11-1-2008

Development of nonlinear reconfigurable control of reconfigurable plants using the FPGA technology

Yi Han
Cape Peninsula University of Technology

Recommended Citation

Han, Yi, "Development of nonlinear reconfigurable control of reconfigurable plants using the FPGA technology" (2008). *CPUT Theses & Dissertations.* Paper 13.
http://dk.cput.ac.za/td_cput/13

Cape Peninsula
University of Technology

DEVELOPMENT OF NONLINEAR RECONFIGURABLE CONTROL OF RECONFIGURABLE PLANTS USING THE FPGA TECHNOLOGY

by

YI HAN

Thesis submitted in fulfilment of the requirements for the degree

Magister Technologiae: Discipline Electrical Engineering

in the Faculty of Engineering

at the Cape Peninsula University of Technology

Supervisor: Professor R. Tzoneva

Bellville
November 2008

DECLARATION

I, YI Han, declare that the contents of this dissertation/thesis represent my own unaided work, and that the dissertation/thesis has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

Signed	**Date**

ABSTRACT

As one of the biggest developing country in the world, South Africa is developing very fast resent years. The country's industrialization process is rapidly evolved. The manufacturing industry as one of the most important sections of the industrialization is playing a very heavy role in South Africa's economic growth. Big percentage of population is involved in the manufacturing industry. It is necessary to keep and enhance the competitiveness of the South Africa's manufacturing industry in the world wide. But the manufacturing companies are facing with unpredictable market demands and global competitions. To overcome these challenges, the manufacturing companies need to produce new products which can cater to the market demand as soon as possible.

Reconfigurable Manufacturing System (RMS) is one of the possible solutions for the manufacturing companies to produce the suitable product for the market in a short period of time with low cost and flexibility. That is because the RMS can be reconfigured easily according to the required specifications for manufacturing the appropriate product for the market and with above mentioned characteristics. Now, RMS is considered as one of the promising concepts for mass production. As one of the very latest research fields, many companies, universities and institutions have been involved to design and develop RMSs. South Africa as one of the most important manufacturing country in the world, her own universities and researchers has the obligation to study this field and follow the newest development steps.

In this project, a lab-scaled reconfigurable plant and a Field Programmable Gate Array (FPGA) technology based reconfigurable controller are used to realize and verify the concepts of the RMS in order to find the methodology of developing RMSs. The lab-scaled reconfigurable plant can be reconfigured into the inverted pendulum and the overhead crane. Although it is not used for manufacturing purpose, it can be used to verify the RMS concepts and the control strategies applied to it. Furthermore, control of the inverted pendulum and the overhead crane are both typical problems in the control field. It is meaningful to develop the controllers for them. As the reconfigurable plant is configured, the reconfigurable controller is reconfigured synchronously in order to produce the proper control signal for the reconfigured plant. In this project, both linear and nonlinear control strategies are deployed. Good results are received.

The outcomes of the project are mainly for the education and fundamental research purposes, but the developed control strategies have significant sense towards the military missile guidance and the overhead crane operation in industry.

ACKNOWLEDGEMENTS

I wish to express my heartfelt thanks to:

- My parents Shiwei Han and Yindi Liu, who always love, support and encourage me.
- My supervisor, Professor Raynitchka Tzoneva for her supervision, guidance, dedication, help, suggesting, understanding and support. She let me know what a real educator is. Thank you again, Prof.
- Mr. Tom van Breda for his support and understanding.
- Mr. Dennis for his help toward the project Real-Time implementation.
- All the stuff in Electrical Engineering Department at Cape Peninsula University of Technology for their support.
- Aunt SuLan Wang, Aunt SuPing Liu, Aunt Sufeng Liu, Uncle Xiangling Liu and other family members for their support and help.
- Miss. Caragh Barwise for her help toward my writing.

TABLE OF CONTENTS

CHAPTER ONE: INTRODUCTION

CHAPTER TWO: LITERATURE REVIEW

CHAPTER THREE: LINEAR AND NONLINEAR CONTROL THEORY

CHAPTER FOUR: MATHEMATICAL MODEL OF THE PENDULUM SYSTEM

CHAPTER FIVE: LINEAR CONTROL

CHAPTER SIX: LYAPUNOV STABILITY THEORY BASED MODEL REFERENCE CONTROL DESIGN

CHAPTER SEVEN: FEEDBACK LINEARIZATION METHOD BASED CONTROL SYSTEM DESIGN

CHAPTER EIGHT: REAL-TIME IMPLEMENTATION CONFIGURATION AND RESULTS

CHAPTER NINE: CONCLUSION AND FUTURE DIRECTION OF RESEARCH

REFERENCE

LIST OF FIGURES

LIST OF TABLES

APPENDIX/APPENDICES

GLOSSARY

Terms/Acronyms/Abbreviations	**Definition/Explanation**
AI	Analog Input
AO	Analog Output
CLB	Configurable Logic Block
CSIR	Council for Scientific and Industrial Research
DAQ	Data Acquisition
DMS	Dedicated Manufacturing System
DSAFC	Direct Self-Structuring Adaptive Fuzzy Control
DSP	Digital Signal Processor
e.m.f	Electromagnetic force
ERC/RMS	Engineering Research Centre of Reconfigurable Machining System
EU	European Union
FLC	Fuzzy Logic Control
FMC	Flexible Manufacturing system Complex
FMS	Flexible Manufacturing System
FPGA	Filed Programmable Gate Array
GDP	Gross Domestic Product
GPCTR	General Purpose Counter
I/O	Input/Output
IOB	Input/Output Block
IOCC	I/O Communication and Control
IP	Inverted Pendulum

IR	Interconnect Resource
LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench
LDM	Lyapunov Direct Method
LQG	Linear-Quadratic-Gaussian
LQR	Linear Quadratic Regulator
LTI	Linear Time-Invariant
LTV	Linear Time-Varying
MOSYN	Modular Synthesis of Advanced Machine Tooles
MIMO	Multi-Input-Multi-Output
MITI	Ministry of International Trade and Industry
MRAC	Model Reference Adaptive Control
MRC	Model Reference Control
NI	National Instruments
NLIP	Nonlinear Inverted Pendulum
NLOC	Nonlinear Overhead Crane
OC	Overhead Crane
OP	Optimal Control method
Op-Amp	Operational Amplifier
PAC	Programmable Automation Controller
PC	Personal Computer
PD	Proportional Derivative
PDC	Parallel Distributed Compensation
PLC	Programmable Logic Controller
PP	Pole-Placement method
PPP	Predict Pole-Placement
RC	Resistor-Capacitor
RIO	Reconfigurable I/O
RMS	Reconfigurable Manufacturing System

RTWT	Real-Time Windows Target
SBC	Single Board Computer
SIMO	Single-Input-Multi-Output
SISO	Single-Input-Single-Output
SMC	Sliding Mode Control
T-S	Takagi-Sugeno

MATHEMATICAL NOTATION

Symbols/Letters	Definition/Explanation
α	Angular acceleration
$\eta_j \in \mathbf{R}^{n-\rho}$	The normal coordinates
λ	The flux linking the coil
μ	A function
θ	The angular displacement between the rod and the vertical direction
$\dot{\theta}$	The angular velocity of the pendulum
$\ddot{\theta}$	The angular acceleration of the pendulum
ρ	The relative degree
σ	The denominator of the pendulum system nonlinear mathematical model in state space
σ'	The denominator of the pendulum system linear mathematical model in state space
τ	Torque
ω	Angular velocity
ζ	The output of the integrator
$\zeta' \in \mathbf{R}^l$	The output of the integrator
$\zeta \in \mathbf{R}^n$	The new states of the system after the state transformation
$\tilde{\zeta}(\mathbf{t})$	The tracking error
ϖ	The linear system for controllability investigation

Γ=[B AB…A^{n-1}B]	The controllability matrix
Φ	The system transition matrix
Ω	The Whole region in space
$[\chi_1, \chi_2 \cdots \chi_l]^T$	The relative degree vector
Ξ	Generalized force
Ψ	Matrix $\begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & 0 \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ a_1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$
∞	Infinite
$[z_1 \quad z_2 \quad z_3 \quad z_4]^T = [x \quad \dot{x} \quad \theta \quad \dot{\theta}]^T$	The states used in the pendulum system nonlinear state space representation
$[z_{1L} \quad z_{1L} \quad z_{1L} \quad z_{1L}]^T$	The states used in the pendulum system linear state space representation
$(\Delta x, \Delta y, \Delta z)$	The object translational movements in 3-Dimension
$\forall$	For any
∇V	A special form of gradient
∇T	The Jacobian matrix
$(\hat{x}, \hat{y}, \hat{z})$	The object origin position coordinates in 3-Dimension
(x’, y’, z’)	The object final position coordinates in 3-Dimension
a	Acceleration of a object when force is applied
$\hat{a}$	The estimate parameters in MRAC
A	The state matrix in LTI system
A_d	The state matrix of the reference mathematical model
A_Φ	The closed-loop constant matrix
$A \in R^{n \times n}$	The constant state matrix for LTI system

$A(t)$	The state matrix in LTV system
A_L	The state matrix of the linearized pendulum mathematical model
A_{lin}	By using feedback linearization method linearized system state matrix
b	The friction coefficient between the cart wheel and the track of the pendulum system
b_m	The viscous friction coefficient of the armature controlled d.c. motor
b_t	The friction coefficient between the rod and the pivot of the pendulum system
$b(x)$	The nonlinear function of states
B	The input matrix in LTI system
B_{Φ}	The closed-loop control matrix
$B \in R^{n \times m}$	The constant control matrix for LTI system
$B(t)$	The input matrix in LTV system
B_L	The input matrix of the linearized pendulum mathematical model
B_{lin}	By using feedback linearization method linearized system input matrix
B'	The magnetic filed flux density
B_{R0}	Sphere of values of x
c	The friction coefficient between the rod and the pivot
C	The output matrix in LTI system
C_{Φ}	The closed-loop output matrix
$C \in R^{1 \times n}$	The constant output matrix for LTI system
$C(t)$	The output matrix in LTV system
d.c.	Direct Current
D	The direct transmission matrix in LTI system
$D(t)$	The direct transmission matrix in LTV system
$e \in R^n$	The error between the desired trajectory and the plant output
e'	The back e.m.f constant

E	A vector of the poles of the closed-loop matrix $[A_{\Phi}-B_{\Phi}K_{\Phi}]$
$E(x)$	The second term of the nonlinear controller in feedback linearization
E'	The total energy
f	Smooth vector fields on R^n
$f'(\zeta)$	New state matrix of a nonlinear after the state transformation
$f(x)$	The nonlinear function of states
f_c	The friction force between the cart wheel and the track of the pendulum system
F	The horizontal control force applied on the cart
g	The gravitational acceleration
g	Smooth vector fields on R^n
$g'(\zeta)$	New input matrix of a nonlinear after the state transformation
G	The adjustment gain matrix in LTI system
$h(t)$	Nonlinear vector valued function in the nonlinear system mathematical model
h.o.t	High order terms
$h'(\zeta)$	New output matrix of a nonlinear model after the state transformation
H	The horizontal reaction force on the rod
H_p	Force applied to the rod perpendicularly caused by horizontal force H of the pendulum system
i	The current flowing the motor wire
i_a	The current flowing the armature controlled d.c motor
I	The moment of inertia of the rod about its center of gravity
l	The length of wire within the field
J	The rod and the pendulum moment of inertia of the pendulum system
J_m	The motor inertia

J_P	The performance index
k	The friction coefficient between the cart and the surface
K	The state feedback gain matrix in LTI system
K_b	The voltage constant
K_D	Potentiometer Gain (derivative control for servo position)
K_L	The feedback gain of the desired mathematical model
K_m	The torque motor constant
K_x	Potentiometer Gain for measuring cart position
K_ω	Tacho Gain for measuring cart velocity
K_Φ	The feedback control gains
$\hat{K}$	The feedback control gains of the pendulum system
L	Distance between the pivot and the center of gravity of the rod
L_a	The inductance of the armature controlled d.c. motor
L’	Lagrangian
m	Mass of the rod
m_P	The payload mass of the crane
m_t	The mass of the cart in the crane
M	Mass of the cart
M_P	Mass of the pendulum
M’	The second term of the Lyapunov function first derivative
n	System order
N	The number of the coil turns
O	The observability matrix of LTI system
$P \in R^{n \times n}$	Positive-definite real symmetrical matrix
P_1; P_2; P_3; P_4; P_5	The desired poles of the closed-loop system

P_g	Potentiometer Gain (proportional control for servo position)
P_s	Servo amplifier Gain
P_Φ	The matrix of the Riccati equation
q	The generalized coordinate
q_i^*	Smooth vector fields on R^n, $i=1,2,\ldots,n$
$Q; R; Q_\Phi; R_\Phi$	The weighting matrix in the performance index. Positive-definite (or positive-semi-definite) Hermitian or real symmetric matrix
r	The reference signal
r	The armature controlled d.c. motor radius to belt
r_c	The distance between the centre line of the shaft and the conductor
R'	The controllability matrix of LTI system
R_a	The resistance of the armature controlled d.c. motor
t_0	The initial time
t_1	The ending Time
$[t_0, t_1]$	Time interval
T	The torque generated by the d.c. motor
$T(x)$	The transformation matrix
T_ω	The torque generated by the servo motor
T'	The kinetic energy
u	The control signal
u_{zd}	The zero-dynamic system input signal
$u(t)$	The input vector in LTV system
v	New input of the linearized nonlinear system
$v \in R^m$	The control vector for the model reference system
V	The vertical reaction force on the rod
V_a	The voltage applied to the armature circuit
$V(x)$	Lyapunov function

V'	The potential energy
V_p	Force applied to the rod perpendicularly caused by vertical force V of the pendulum system
$\dot{V}(x)$	The first derivative of the Lyapunov function respect to time
W	The angular momentum
x	The displacement of the cart from origin
$x(t)$	The state vector in LTV system
X_0	The initial state of the general system
$\dot{x}$	The velocity of the cart
$\ddot{x}$	The acceleration of the cart
$x_d \in R^n$	The desired states vector of the model
x_e	The equilibrium point
x_G	The horizontal distance between the origin and the center of gravity of the pendulum and rod when the pendulum system is configured as the inverted pendulum and the overhead crane
y	The displacement of the pendulum from origin
y	The plant output
$y(t)$	The output vector in LTV system
y_d	The desired output
y_d^{sp}	The desired output set point value
y_G	The vertical distance between the origin and the center of gravity of the pendulum and rod when the pendulum system is configured as the inverted pendulum and the overhead crane
y_m	The desired trajectory
y^{sp}	The set point
z	The states of the linearized nonlinear system
z0	The initial conditions

CHAPTER ONE
INTRODUCTION

1.1 Awareness of the problem

Manufacturing industry remains an important foundation of the global economy which accounts large percentages of the Gross Domestic Product (GDP) (Moyne, Korsakas and Tilbury, 2004:1). South Africa's economy has grown rapidly in recent years and has had significant impact on the world economy (Nayyar, D, 2008:1). As one of the biggest developing countries in the world, the manufacturing industry plays a very heavy role in contribution to determining the country's economic growth. The latest released data show that 24% of South Africa GDP are contributed by the manufacturing industry in 2007 (http://learn.eastday.com/view.jsp?code=1119) and manufacturing industry growth rate is 14.8% over the same period of the previous year from January to July 2007 (South Africa Reserve Bank, 2008). This data reveals how import the manufacturing industry is for South Africa's economic. Additionally, the manufacturing industry also provides a locus for stimulating the growth of other activities, such as services, and achieving specific outcomes, such as value addition, employment creation and economic empowerment (http://www.info.gov.za/aboutsa/economy.htm#manufacturing). In order to maintain the rapid economic growth rate and reduce the unemployment rate, it is very important to keep and enhance the competitiveness of the South Africa's manufacturing industry in the world.

In South Africa, automotive manufacturing industry, information technology products manufacturing, and white goods manufacturing are the pillar manufacturing industries. Especially the automotive industry, it is extremely important for the South Africa's economy. This sector accounts for about 10% of South Africa's manufacturing exports, making it a crucial cog in the economy. Locally, the automotive sector is a giant, contributing about 7.5% to the country's GDP and employing around 36000 people (http://www.southafrica.info/business/economy/sectors/automotive-overview.htm). These key manufacturing industries share several common characteristics such as the large amount of production, labour-intensive production, huge investment and high added values. At the same time, the manufacturing industries also face some common challenges and problems such as unpredictable market demands, global competition and raw material prices and labour prices rising. These idiosyncrasies of the manufacturing industry can be easily understood by using the automotive industry as a typical example.

Now days, new model cars go to the market much more frequently than ever before. This requires the automotive manufacturers to change their production line more often, especially when the demands are fluctuating or the specifications are changing, in order to make it capable to produce the suitable product to the market.

Traditionally, in order to produce the new product, a big portion of the old production line must be rebuilt. This means the manufacturing company must invest a huge amount of money, old equipment will be left unused and during the production line rebuilt period, all the production activities are stopped. Furthermore, the longer the manufacturing company takes to rebuild their new manufacturing line, the more its new product will lose competiveness in the market. Several ineradicable disadvantages of rebuilding the production line can be easily seen, such as: huge amount of investment, long period of production downtime and wasting of resources and labours. As a result, the manufacturing companies desire novel concepts to solve these problems in order to reduce the additional investment, reduce the production downtime, optimize the resources usage and most importantly, the manufacturing companies could produce new products which can cater for the market demand as soon as possible. The Reconfigurable Manufacturing System (RMS) is introduced as one of the prospective conceptual solutions (Lee and Tilbury, 2003:3402).

1.2 Reconfigurable manufacturing system (RMS)

1.2.1 Introduction of the reconfigurable manufacturing system

The RMS provides the manufacturing industry with possibilities to produce the suitable product for the market in short periods of time with low cost and high flexibility. RMSs are currently considered as one of the promising concepts for mass production. In this system, quick start for new products, the rapid alteration of the manufacturing system capacity, and the fast integration of new process technologies into an existing system can be realised by rearranging or changing modular components (Lee and Tilbury, 2003:3402). The RMS is defined as: one designed at the outset for rapid change in its structure, as well as its hardware and software components, in order to quickly adjust its production capacity and functionality within a part family in response to sudden market changes or intrinsic system change (Koren et al., 1999:528).

The RMS takes advantage of the mass production of the dedicated manufacturing systems (DMS) and the flexibility of the flexible manufacturing system (FMS). So it provides a cost-effective solution under changing the manufacturing line because it reduces the downtime, provides high speed capability and functional reliability solution.

A complete RMS includes following main subsystems:

- Reconfigurable machine (plant)
- Reconfigurable controller
- Methodology for the system design, diagnostics and reconfiguration

In these three main subsystems: The reconfigurable machine (plant) subsystem represents the system hardware. It is the system that is necessary to be controlled. Normally it is the actual plant that can be reconfigured according to the specified requirements. The reconfigurable controller subsystem represents the reconfigurable controller which consists of the control hardware and control software. The reconfigurable controller is one of the most important subsystems of the modern manufacturing control system. It manages and optimizes the entire RMS. After the reconfigurable machine is reconfigured, the reconfigurable controller needs to be reconfigured according to the requirement of the reconfigurable system in the form of hardware and/or software, in order to realize the desired functions. The third subsystem consists of design software and reconfigurable logic, algorithms and methods which are realized in the RMS software to coordinate and adjust the process of reconfiguration. The third subsystems control design functions are object of investigation and development in the project (Koren et al., 1999:535).

1.2.2 Characteristics of the reconfigurable manufacturing system

RMS must be designed at the outset to be reconfigurable, and must be created by using hardware and software modules which can be integrated quickly and reliably. Otherwise, the reconfiguration process will be both lengthy and impractical. In order to achieve these goals, all of its subsystems must possess the following key characteristics:

- **Modularity** – All the major components in the reconfigurable manufacturing system are modular.
- **Integrability** – There are designed interfaces between different modules for easy integration. The integrated system performance is predicted based on a given performance of its components and the interfaces of both software and machine hardware modules.
- **Customization** – This characteristic includes the customized flexibility and customized control. The customized flexibility means that machines are built around parts of the family that are being manufactured and provide only the flexibility needed for those specific parts, thereby reducing cost. Customized control is achieved by integrating control modules with the aid of open-architecture technology, providing the exact control functions needed. Open-architecture technology allows adding, upgrading and swapping components to the system. It provides the ability of easy reconfiguration to the reconfigurable manufacturing system.

- **Convertibility** – In a reconfigurable system the optimal operating mode is configured in batches that should be completed during one day, with short conversion times between batches.
- **Scalability** – The ability to easily change production capacity by rearranging an existing manufacturing system and/or changing the production capacity of reconfigurable stations. Scalability is the counterpart characteristic of convertibility.
- **Diagnosibility** – Detection unacceptable part quality is critical in reducing ramp-up time in RMS. As production systems are reconfigurable and are modified more frequently, it becomes essential to rapidly tune the newly reconfigured system so that it produces quality parts. (Koren et al., 2004:530)

These key characteristics distinguish the RMS from the traditional manufacturing production line. They also bring the key research issues of RMS out.

1.2.3 Research issues of the reconfigurable manufacturing system

In order to posses above mentioned key characteristics of the RMS, the following key research issues in the process of designing and operating RMS must be studied. These key research issues can be classified into system-level issues, component-level issues and ramp-up time reduction issues. The project investigations are part of the components-level research issues. The detailed key issues on different levels are presented in Table 1.1 below.

Table 1.1: RMS key research issues (Mehrabi, Ulsoy & Koren, 1998:17-19)

System-level design issues	Component-level issues	Ramp-up time reduction issues
Analysis and design of the full process from recognizing customer needs (or anticipated needs) through operation selection and system specification	Development of fundamental principles and techniques for the design and analysis of reconfigurable machines along with their controllers	Design of robust components that can operate reliably and safely under different operating conditions
Analysis of the impact of system configuration on reliability, quality, and cost	Design and development of a set of simple reconfigurable machines and controller to quickly produce two different parts for the proof of concept	Development of systematic approaches and fundamental principles to indentify root-causes of components failure, and quality and process variations
Economic analysis of various system configurations and their selection		
Development of a systematic approach for design of RMS at the system level		

1.2.4 Historic and current development of the reconfigurable manufacturing system

Reconfigurable manufacturing is one of the latest developments in the general field of computer-integrated manufacturing system. But the concept of RMS is proposed since 1960s. The research history of the RMS can also ascended since that time when the firm called “Moilns Company Ltd” developed the first integrated manufacturing system. Until now, RMSs have been developed extensively since 1970s in both the theoretical level and practical level. In Japan, Europe and USA the related projects have been started since 1980s and 1990s. Several landmark achievements are realized, including: The Flexible Manufacturing system Complex (FMC) project provided by Ministry of International Trade and Industry (MITI) in Tsukuba, Japan. In early 1990s, the European Union (EU) launched the European Modular Synthesis of Advanced Machine Tools (MOSYN) project, lead by the Hannover University. This project looks at customer-specific configurations of modular machine tools. In 1996 an Engineering Research Centre of Reconfigurable Machining System (ERC/RMS) was founded at the University of Michigan, Michigan, USA. The researches focused on three main research main areas, reduction of design lead-time for reconfigurable system, design of reconfigurable machines and their reconfigurable controllers, and reduction of ramp-up time. Regarding issues of control, research efforts have been focused on open control architectures. In the world wide, there are three most important initiatives in open architecture control system: The EU project, OSACA and its German successor, HÜMONS; the Japanese initiative OSEC; and the North American OMAC-TEAM project (Koren et al., 2004:531).

Till now, Japan, EU and USA are the three major contributors to the research of RMS in the world. The world biggest car manufacturer TOYOTA has successfully implemented the RMS concept in its production line. Several practical experiments have been successfully done in the laboratories as well. South Africa is one of the important manufacturing countries in the world, but the RMS research is relatively lagged. There is no university or company involved in this area. Three years ago the Council for Scientific and Industrial Research (CSIR) started a program for develop of RMS. The project under consideration develops methods, algorithms and software for development of the reconfigurable control methodology by studying a lab-scale reconfigurable plant.

1.3 Reconfigurable system in this project

This project provides investigation and development of the methodology for control development of the reconfigurable system. In this project, the hardware part includes a lab-scaled reconfigurable pendulum plant and an embedded logic controller. The reconfigurable plant can be reconfigured into either the inverted pendulum mode or the overhead crane mode by simply turn it upside down. The inverted pendulum and overhead crane are introduced in Chapter 2. The existing mathematical models of the inverted pendulum and overhead crane are also studied in the same chapter. In Chapter

3, this lab-scaled reconfigurable plant is described in detail before its mathematical models are derived. This lab-scaled reconfigurable plant consists of the plant module and the controller module. They are integrated together through a 26 way IDC microcomputer header. The input and output signals can be monitored by using the incidental Data Acquisition (DAQ) system. The described properties reveal that this lab-scaled reconfigurable plant has such characteristics as modularity, integrability, convertibility and diagnosibility. It fulfils the requirements of a module of RMSs. An embedded logic controller is developed as the reconfigurable controller in this project. It is based on the Field Programmable Gate Array (FPGA) technology in order to realize the key characteristics of the RMS control. FPGA technology is studied in Chapter 8 in detail and the Real-Time implementation results are presented in the same chapter as well. The software package that is used in this project includes the Mathworks Matlab/Simulink platform and the National Instruments (NI) Laboratory Virtual Instrumentation Engineering Workbench (LabVIEW) platform.

Several different linear and nonlinear control strategies including the pole-placement method, Linear Quadratic Regulator (LQR) method, Model Reference Control (MRC) method, Lyapunov Direct Method (LDM), the feedback linearization method, are used for the controller design. These control strategies and theories are introduced in Chapter 3 and applied in Chapter 4, 6 and 7 respectively, Matlab/Simulink and LabVIEW.

The project is done by following the sequences of background study, mathematical model derivation, controller design, system simulation and Real-Time implementation. Firstly, RMS, the inverted pendulum, the overhead crane and the control strategies are studied. Then, the nonlinear mathematical models of the reconfigurable plant are derived. By linearizing the nonlinear models, the linear models are obtained. Thirdly, different controllers are designed based on different control methods. Then, the closed-loop systems are simulated and analyzed on the simulation platform. Once the proper results are achieved, the project shifts to the Real-Time implementations platforms. The simulation and Real-Time implementation are done in different software environments and Real-Time implementation platforms respectively.

The outcomes of the project have noteworthy sense for the South Africa’s manufacturing industry. The results also have significant meaning for the military industry, in fundamental research in terms of missile and rocket guidance study and industry applications in terms of overhead crane control in industry. It can also be used for fundamental research and education purposes.

1.4 Research aim and objectives

The aim of the project is to develop methodology for design and implementation of the reconfigurable nonlinear control systems in Real-Time based on embedded FPGA technology.

The methodology is developed based on the following objectives:

- Development of the linear and nonlinear mathematical models for the reconfigurable plant
- Development of modern control methods for the reconfigurable plant
- Evaluation of the model derivation and modern controller design methods by simulation
- Real-Time implementation of the closed-loop systems using embedded platforms
- Building of the methodology for design of reconfigurable embedded control systems

1.5 Statement of the problem

The research problem is stated as follows:

To develop models, methods and algorithms for design of the nonlinear control those are capable of being implemented in Real-Time in the frameworks of embedded software and hardware platform. The FPGA based controllers with LabVIEW programming are used for controller implementation.

The above stated research problem consists of the following sub-problems:

1.5.1 System modeling sub-problems

1.5.1.1 Sub-problem 1: Nonlinear mathematical models development

The complete nonlinear mathematical models for different configuration of the lab-scale RMS are derived. The nonlinear mathematical model derivations are one of the most important tasks of the project.

1.5.1.2 Sub-problem 2: Linear mathematical models development

The complete linear mathematical models for different configuration of the lab-scaled RMS are obtained by linearizing the RMS nonlinear mathematical models.

The process of the system modelling is an important approach of the system familiarization and understanding. The reconfigurable controllers are designed based on the system mathematical models and the modern control strategies.

1.5.2 Sub-problem 3: Reconfigurable controllers design and development

The linear controllers and nonlinear controllers design methods are applied for designing and developing the reconfigurable controllers. Different modern control methods are applied, such as: pole-placement method; linear quadratic method; model reference control method; Lyapunov direct method and the feedback linearization method, to achieve the desired system responses.

1.5.3 System simulation and emulation sub-problems

1.5.3.1 Sub-problem 4: Mathematical models simulation and emulation

In order to verify, confirm and finalize the system mathematical models, the models which are mentioned in section 1.5.1.1 and 1.5.1.2 are simulated in Matlab/Simulink software environment.

1.5.3.2 Sub-problem 5: Closed-loop system simulation and emulation

The closed-loop systems, which consist of the reconfigurable plant models and their corresponding controllers, are simulated in order to verify that the controllers are properly designed. Once the good results of the closed-loop simulation are obtained, the project shifts to the Real-Time implementation platform.

1.5.4 Real-Time implementation

The Matlab/Simulink with DAQ system platform is used first, in order to confirm that the systems are implementable on the Matlab Real-Time Target platform. If the Real-Time implementation is achieved on this platform, the corresponding controllers are designed in LabVIEW environment. This is to reduce the programming development time. Then the Real-Time implementation is shifted to the LabVIEW and CompactRIO FPGA platform. The above mentioned software and hardware are introduced in Chapter 8.

1.5.4.1 Sub-problem 6: Real-Time implementation on DAQ system platforms

The Real-Time implementations on DAQ system based platforms:

- Matlab/Simulink with DAQ card Real-Time implementation platform

1.5.4.2 Sub-problem 7: Real-Time implementation on embedded system platforms

- LabVIEW with NI CompactRIO FPGA Real-Time implementation platform is used

1.5.5 Results comparison

1.5.5.1 Sub-problem 8: Simulation results comparison

The systems performances are studied by analysing and comparing the simulation results. The analysis focuses on the system performances. The comparisons concentrate on the simulation results that are obtained by using different control strategies. Through the comparison, the effects of each control method on the pendulum system are studied, revealed and summarised.

1.5.5.2 Sub-problem 9: Real-Time implementation results comparison

The Real-Time implementation platforms are compared by referring the requirement of the RMS. The obtained Real-Time implementation results from the different software and hardware platforms and the different control strategies are compared. The comparison shows which platform is better in terms of the system performance, such as: the smoothness of the transition behaviour, the stability of the system and the reliability of the system.

1.5.6 Sub-problem 10: Metrology for control reconfiguration development

The methodology of developing and implementing RMS control is concluded from the process of this lab-scaled RMS plant control development. This lab-scaled RMS has the key characteristics of RMS, so the developed methodology can be used for developing the industry RMS control.

1.6 Hypothesis

Certain hypothesises are the preconditions of the research. The hypothesises are based on the existing control theories and technologies. They can be formulated as:

1.6.1 From the theoretical point of view:

- The mathematical models of the nonlinear reconfigurable plants can be derived based on the existing mathematical and physical principles
- The modern control methods can be applied for the linear and nonlinear controller design
- The nonlinear controllers will result in closed-loop systems with better dynamic behaviours than the dynamics obtained from the linear control system

1.6.2 From the implementation point of view:

- The pendulum system has the key characteristics of the RMS. It is considered as a RMS in lab-scale variant.
- All the parameters needed for the calculation, simulation and implementation can be obtained from the pendulum plant or from measurements
- The embedded software can implement the nonlinear functions of the nonlinear control laws

- The embedded hardware can be reconfigured according to the required applications
- The integration between the software and hardware can be done on different implementation platforms
- The reconfiguration of the controller can be achieved in Real-Time

1.7 Delimitation of research

The research project is delimited in the following sections:

- **Modelling:** The modelling of the reconfigurable plant focuses on the inverted pendulum and the overhead crane plant models derivation only.
- **Control strategies:** MRC theory is used to organise the structure of the entire system, where the used linear reference models are designed based on the linear mathematical models and the LQR method. Two theories for design of nonlinear controllers are used – the Lyapunov second method and the geometrical theory for linearization of the closed-loop system. Firstly, Lyapunov direct method is used to analyse the closed-loop systems stability. Secondly, it is also used to design a nonlinear controller. The feedback linearization method is the other method that is used for nonlinear controller design. Other control strategies are not applied in the reconfigurable system
- **Simulation platforms:** All the simulations are done in the MATLAB/Simulink software environment only
- **Real-Time implementation platforms:** The Real-Time implementations are deployed on the following platforms:
  a) NI DAQ card with MATLAB/Simulink platform
  b) NI DAQ card with LabVIEW platform
  c) NI CompactRIO FPGA with LabVIEW platform

1.8 Assumptions

The assumptions are made according to the different parts of the research work. These assumptions include:

- The reconfigurable plant is working according to the prescribed instructions
- The mathematical models can be derived according to the existing physical laws. The linearization of the nonlinear models can lead to linear models capable of being used for design of linear controllers in neighbourhood of the selected equilibrium points
- The linear and nonlinear control methods and theories are powerful enough to support the design of a reconfigurable controller
- The simulation environment results provide good approximation to the Real-Time implementation results
- The hardware can provide high performance, based on the high speed I/O port and accurate signal processing technology associated with noise filtering

1.9 Motivations for the research

There are several motivations for doing this research project. First of all, from the social contribution point of view: Manufacturing industry is one of the major industries contributing to the economic growth in South Africa. It is necessary to keep South Africa’s manufacturing industry to be comparable world wide. RMS provides the possibilities that the manufacturing companies can produce the suitable product to the market in a short period of time with low cost and flexibility. Now, RMS is regarded as the trend and the future of the manufacturing industry and the existing RMSs have proven their great advantages. In other word, RMS gives manufacturing companies the flexibility and the possibility of joining the intense market competition in the global economic process. On the other hand, there are demands for RMS, because the manufacturing companies need more advanced RMSs for greater efficiency, lower cost, easier configuration and so on.

Secondly, from the country scientific competiveness point of view, research of RMS is one of the latest, most complex and across research area in the world. Many companies, universities, institutes and even governments have taken part in the RMS study. Huge amount of funds have been invested in this area. But, still there has not been big involvement from South African. As the ponderance of RMS, South Africa’s universities have the responsibility to join the research field of RMS and the research outcomes can serve the country’s manufacturing industry and economic development.

Thirdly, from the control theory point of view, there are several technique challenges to motivate the research, such as:

- The research is based on the nonlinear dynamic model of the system. This model is difficult to be used for the Real-Time control and several difficulties with the nonlinear models are involved:
  a) They do not follow the principle of superposition (linearity and homogeneity).
  b) They may have multiple isolated equilibrium points (linear systems can have only one).
  c) They may exhibit properties such as limit-cycle, bifurcation, chaos.
  d) Finite escape time: The state of an unstable nonlinear system can go to infinity in finite time.
- Difficulties with the control of the inverted pendulum and the overhead crane:
  a) In the inverted pendulum configuration, the task is to control an inherently unstable system. In order to balance the pendulum in the inverted position the pivot must continuously be moved.
  b) In the overhead crane configuration, the task is to control the position of the load which possesses great oscillatory dynamics.
- Difficulties in applying the nonlinear control strategies
  a) The Lyapunov direct method is applied to design one of the nonlinear controllers. It is well known that the Lyapunov function is difficult to be found.
  b) The feedback linearization method is the other nonlinear control method used to design the nonlinear controller in this project. The difficulty of using the feedback linearization method is how to select the reference model dynamics and how to make the nonlinear controller robust.
- Difficulties in the Real-Time implementations:
  a) Many variables are difficult to be measured.
  b) Many parameters are difficult to be determined, especially in the nonlinear processes

c) Compatibility between different software and hardware platforms for embedded control implementation

d) New Real-Time implementation platforms as FPGAs are used

1.10 Research methods

The field of RMS is a complex discipline which requires integrated approach for providing research activities. That is why different research methods are used during the different phases of the research work on the project.

- Descriptive method: To describe the plants to be controlled as objects of control – The pendulum system. It is necessary to obtain information about the pendulum system parts: the hardware structure, the components specifications, the physical setup and the involved physical phenomenon. The information is gathered by observation, technology study, measurements, questionnaires and interviews with the operating personnel.
- Experimental method: The cause and effect dependencies in the processes are studied in order to take data needed for mathematical model parameter estimation. The experiments are provided on the basis of information obtained from the descriptive method application, after planning, preparation and implementation of the planned actions.
- Design method: Development of methods for design of control based on the existing control techniques. The information from the process description and model development is used. Novel technique or application of the existing knowledge is realized. Research on the control strategies, development of mathematical models and closed-loop systems simulations are done.
- Design method: Combine and integrate different hardware and software for Real-Time implementation of the closed-loop system. Work with different hardware and software platforms, software for Real-Time implementation is written.
- Experimental method: To study the behaviour of the process under the influence of the designed control action. The control methods and Real-Time control implementations are finalized.
- Descriptive method: To describe the designed system with all specific characteristics and functions. To produce the methodology for the operator and guideline for the system control. Information is gathered from all above steps.

1.11 Literature survey

Judging by the number of publications in journals and at conferences, there are many researchers working in the field of nonlinear control, RMS, inverted pendulum, overhead crane and DAQ. A few of them are involved with FPGA. This information was gathered from consulting books, journal publications, the Internet, and conference proceedings. Different authors have different views on many issues and different solutions are proposed and implemented. The relative references are studied in Chapter 2 including the existing mathematical models, control strategies and the implementation platforms. In the other chapters, the other existing work are referenced or used as a basis for further developments.

1.12 Chapter Breakdown

This thesis consists of nine chapters. These chapters include the background information, model development, controller design, system simulation, system implementation, results of the project and conclusion.

Chapter 1 presents the background study of the project. The RMS is introduced by associating the challenges and problems in manufacturing industry in South Africa, especially the automotive manufacturing industry. The pendulum system as a lab-scale reconfigurable system is briefly introduced. The project statements include: the research aim and objectives, statement of the problem, hypothesis, research delimitation, research motivations, assumptions and research methods.

Chapter 2 presents the existing work that is related to this project. It includes the study of the lab-scaled RMS, the existing mathematical models of the inverted pendulum and overhead crane. The existing Real-Time implementation platforms are studied as well.

Chapter 3 provides the background study of both linear and nonlinear control theory by referring to the control of the inverted pendulum and overhead crane. The typical linear control strategies include the root locus method, the pole-placement method, the quadratic optimal control method and the typical nonlinear control strategies include the model reference control method, Lyapunov stability theory and feedback linearization method. The existing works of controller design for the inverted pendulum and the overhead crane are studied as well.

Chapter 4 presents the detailed mathematical models derivation. The pendulum system is described in detail first. Based on the understanding of the pendulum system physical structure, the nonlinear and linear mathematical models are derived. The models are also simulated in Matlab.

Chapter 5 shows the linear controller design based on the pole-placement method and the LQR method and the corresponding closed-loop simulation results. More importantly, by using the LQR method, the designed linear closed-loop system is used as the reference model in the nonlinear control system design. The closed-loop systems based on the pole-placement and LQR method are simulated and the simulation results are given.

Chapter 6 describes the nonlinear controller design based on LDM and the closed-loop simulation results. The common procedure of nonlinear controller design based on LDM is introduced firstly. By using this common procedure the LDM based nonlinear controllers for controlling the inverted pendulum and the overhead crane are designed. The closed-loop systems are simulated and the simulation results are presented.

Chapter 7 presents the nonlinear controller design based on the feedback linearization method and the closed-loop simulation results. The new states are selected. The nonlinear models are converted to linear representation. By using the linear control technique, linear controllers are designed for the linearized closed-loop system. The closed-loop systems are simulated and the simulation results are shown.

Chapter 8 describes the Real-Time implementation platforms configuration structure and the Real-Time implementation results. The used different hardware and software are introduced in detail. The Real-Time implementation results are shown.

Chapter 9 provides the detailed conclusions and the future research direction in the field of RMS.

1.13 Publications

1. Han. Y., Tzoneva. R. and Behardien. S. 2007, “MATLAB, LabVIEW and FPGA Linear Control of an Inverted Pendulum”. Proceedings of Africon 2007; CD-ROM; Ref. No. 1-4244-0987-X/07/$25.00 ©2007 IEEE. IEEE Africon 2007 international conference, Windhoek, Namibia.
2. Han. Y., Tzoneva. R. 2008. “Reconfigurable Plant Control by Using Lyapunov Stability Theory based Model Reference Control Design”, submitted to SAIEE Research Journal.
3. Han. Y., Tzoneva. R. 2008. “Reconfigurable Plant Control by using Feedback Linearization Method with Optimal Control Method”, submitted to SAIEE Research Journal.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

The research work of this project focuses on development of the control methodology for RMS. The RMS used in this project can be reconfigured into the inverted pendulum mode and overhead crane mode. Control of the inverted pendulum and the overhead crane are typical problems in automation control field, which have been studied extensively. The existing works include the mathematical models derivation, and control strategies design. The Real-Time implementation platforms for controlling of the inverted pendulum and the overhead crane are studied and summarized in this chapter. These related latest research works in interrelated fields are studied in the sequence of:

- Reconfigurable system as inverted pendulum and overhead crane
- Linear and nonlinear control strategies
- FPGA technology

Amount them, the central elements involved in the project are the nonlinear controllers design and Real-Time control implementation on the embedded FPGA platform.

In this chapter, section 2.1 recapitulates the content of the chapter. Section 2.2 studied the reconfigurable system used in this project, as well as the existing mathematical models of the inverted pendulum and the overhead crane. In section 2.3, the control theories, techniques and methods are studied associated with the controlling of the inverted pendulum and the overhead crane. Some of the existing Real-Time implementation platforms are presented in section 2.4. At the end, the conclusion is given in section 2.5.

2.2 Reconfigurable system used in this project

The RMS is introduced in Chapter 1. In this project, a lab-scale RMS is used. This RMS is a pendulum system manufactured by Bytronic Limited, Staffordshire, England. Originally, the pendulum system is not used for manufacturing purposes, but mainly for educational and research purposes. In this project, the pendulum system is not used for manufacturing purpose either, but because it can be reconfigured into the inverted pendulum mode and the overhead crane model, it is an excellent example for demonstrating the basic idea of the reconfigurable plant.

Both the inverted pendulum and the overhead crane are nonlinear mechanical systems and both of them have corresponding meaningful applications. The inverted pendulum can be used as a simplified representation of the rocket or missiles guidance system (Tan and Habib, 2007:371) or it can be used to represent the human stances, such as stationary human standing (Barauskas, Krušinskienė, 2007:612) or human walking dynamic (Kuo, 2007:617). The problem of balancing an inverted pendulum by moving the pivot cart in horizontal direction is also commonly used in control education to provide a dramatic experiment and to illustrate the difficulties in controlling a plant which has unstable open-loop poles. The overhead crane is used widely in industry, in transportation systems and rehabilitation equipment (Chang et al, 2007). It works as a robot in many ways. It transports all kinds of massive goods. It is desired for the overhead crane to transport its payloads to the desired location as fast and as accurately as possible without collision with other equipment. Furthermore, the swing of the payload should be suppressed as much as possible (Liu et al, 2005). Furthermore, both the inverted pendulum and overhead crane are used as typical test benches for new control strategies. Because of these exclusive characteristics, both systems have been studied extensively in terms of their physical dynamics, and different control strategies have been applied.

The development of the nonlinear and linear mathematical models of the inverted pendulum and the overhead crane in this project, based on the other researchers' and engineers' previous works is described in Chapter 4. In this section, some of the existing typical mathematical models of both the inverted pendulum and the overhead crane are studied. Basically, there are two approaches for the system model derivation: Newtonian's approach and Lagrangian approach. Both approaches are studied in this section. The system linear mathematical models can be obtained from the corresponding nonlinear mathematical models through model linearization. For the dynamics of the inverted pendulum and overhead crane, four states are considered. These four states are the cart position state x, the cart velocity state $\dot{x}$, the rod angular position state θ and the rod angular velocity state $\dot{\theta}$. In the following sections, the inverted pendulum and overhead crane are studied.

2.2.1 Inverted pendulum

There are several different kinds of the inverted pendulum systems, in terms of the structures setup, providing a variety of interesting control challenges (Bugeja, 2003), such as: the single rod with the pendulum on a cart which is studied in this project; the double inverted pendulum on a cart setup (Niemann, Poulsen, 2005:145), the triple inverted pendulum on a cart configuration (Su and Woodham, 2003:425), the parallel-type double inverted pendulum on a cart (Yi et al, 2000:105) and the rotational inverted pendulum (Yavin, 1999:131). Also, some inverted pendulum can move in 3-dimensions (Chaturvedi et al, 2008). The different inverted pendulum systems have different behaviours which correspond to different mathematical models. For this project, the studies focus on the inverted pendulum that has a single rod with pendulum weight on a cart. In the following sections, the Newton's approach and the Lagrangian approach are studied respectively for the mathematical model derivations.

2.2.1.1 Inverted pendulum models derivation based on Newton's approach

Newton's approach is the most common way for deriving the inverted pendulum mathematical model. The classical Newton's approach derives the system mathematical models by using Newton's motion laws and Euler equations. If the inverted pendulum models are derived based on the Newton's approach, firstly, the forces applied on the rod, the pendulum and the cart are analysed, and secondly the torque balance on the rod is studied. The force analysis is based on the Newton's second law. The torque balance analysis is based on the forces applied on the rod and their corresponding moments. In this section, some existing typical inverted pendulum mathematical model derivations are investigated.

One kind of the typical inverted pendulum only consists of the rod, the pendulum is not involved. The mathematical models of this kind of the inverted pendulum have been developed by different researches in different projects, such as M. Bugeja (2003), D. Angleli (2001), A. Irfan and J. Ashraf et al (2000). The mathematical model can be expressed as:

$$I\ddot{\theta} + c\dot{\theta} = VL\sin\theta - HL\cos\theta \qquad (2.1)$$

$$F - H = M\ddot{x} + k\dot{x} \qquad (2.2)$$

where $V = m\frac{d^2}{dt^2}(L\cos\theta) + mg$; $H = m\frac{d^2}{dt^2}(x + L\sin\theta)$; m is the mass of the rod; M is the mass of the cart; L is the distance between the pivot and the center of gravity of the rod; x is the displacement of the cart from origin; g is the gravitational acceleration; θ is the angular displacement between the rod and the vertical direction; c is friction coefficient between the rod and the pivot; k is the friction coefficient between the cart and the surface; I is the moment of inertia of the rod about its center of gravity; V and H are the vertical and horizontal reaction forces on the rod and F is the horizontal control force applied on the cart (Bugeja, 2003). If no additional specification, these symbols are used with the same meaning in this chapter.

The above mentioned mathematical model of the inverted pendulum has relatively closed formation to the model of the pendulum system when it is set into the inverted pendulum mode. But there is one important element of the system, the mass of the pendulum, which is not considered in this model, so the mathematical model does not contain the force analysis of the pendulum. The moment of inertia of the pendulum is not included either. Another incomplete fact is that the moment of inertia of the rod is not specified scientifically, no analysis and calculation procedures are presented. On the basis of the consideration above, this kind of mathematical model is not appropriate to be used to represent the inverted pendulum that is used in this project.

In M. Henders and A. Soudack's (1996) work, the system mass is concentrated at the top of the rod, the center of gravity is the center of the pendulum (Ogata, 2002:88). In this case, the rod mass and the moment of inertia of the pendulum about its center of gravity are ignored. This mathematical model is expressed as:

$$(M+m)\ddot{x}+mL\ddot{\theta}=F \tag{2.3}$$

$$mL^2\ddot{\theta}+mL\ddot{x}=mgL\theta \tag{2.4}$$

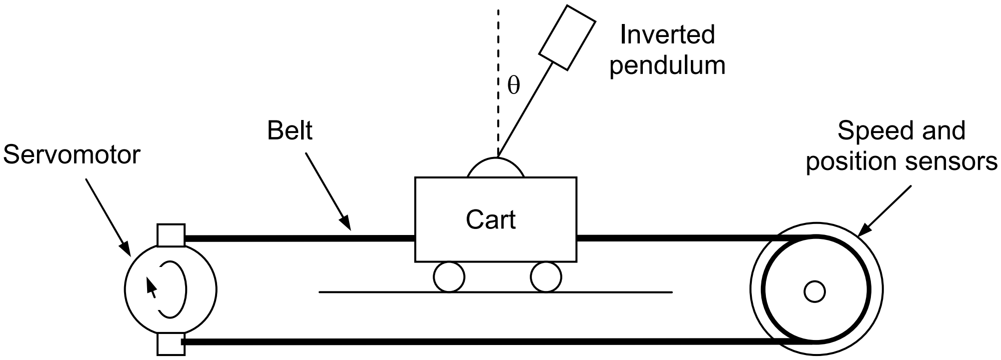


Figure 2.1: Inverted pendulum physical mechanism setup schematic

The assumption has been made in their work that the pendulum is capable of unlimited rotation, and the lateral excursion of the cart is also unlimited (Henders and Soudack, 1996). These assumptions are different from the physical pendulum system which is shown in Figure 2.1 and considered in the project. The pendulum system does not have the capability to rotate unlimitedly. The angular position θ is limited in the range of $-60° < \theta(t) < 60°$ from the vertical direction by a limitation mechanism. Also, the track length of the pendulum is limited to 50cm, and the initial position of the cart is in the middle of the track. Under these circumstances, the existing mathematical models of the inverted pendulum are incomplete. They are not sufficient to describe the pendulum system that is configured into the inverted pendulum mode.

The other difference between the considered pendulum system and the described in the papers inverted pendulums is the output signal. According to the structure of the pendulum system, the only output signal of the pendulum system is the position signal of the pendulum y. It is expressed as: $y = x + L\sin\theta$. However, in most of the existing works, there are two output signals which are the cart position signal x and the pendulum angular signal θ, in the symbolic form: $y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x \\ \theta \end{bmatrix}$, (J. Lam's, P. Gawthrop and L. Wang, 2006). Usually only one of these two outputs is used for control purposes.

Because of the above mentioned reasons, these mathematical models are not suitable for describing the inverted pendulum that is being used in this particular project. But the essential idea of how to derive the mathematical model of the inverted pendulum based on the Newton's laws is the same. Based on it, accurate and integrated nonlinear and linear mathematical models of the inverted pendulum are derived in Chapter 4.

2.2.1.2 Inverted pendulum model derivation based on Lagrangian approach

The Lagrangian approach provides an alternative method which is relatively simpler for developing of the mathematical models. For a conservative system, Lagrangian (L') is defined as kinetic energy (T') of the system minus its potential energy (V'). It can be expressed as: L'=T-V' (Blair, 2002:4). The Lagrangian's equation is defined as:

$$\frac{d}{dt}\left(\frac{\partial L'}{\partial \dot{q}}\right) - \frac{\partial L'}{\partial q} = 0 \qquad (2.5)$$

where q is the generalized coordinate.

Substitute the Lagrangian expression directly into the Lagrangian's equation; the obtained differential equations describe the motion of the system. For the cart, inverted pendulum system, it has two degrees of freedom. Two generalized coordinates are needed to represent the system fully. These two coordinates are chosen as the horizontal displacement of the cart from the origin x and the angular displacement of the pendulum from vertical direction θ.

Lagrangian approach is used a lot for development of the inverted pendulum mathematical models. For example, in the work of Ö. Altinöz (2007:277), the inverted pendulum motion equations have been derived (Altinöz, 2007:278):

$$F = (M + m)\ddot{x} + mL\ddot{\theta}\cos\theta - mL\dot{\theta}^2 \tag{2.6}$$

$$(-I + mL^2)\ddot{\theta} + mgL\sin\theta = mL\ddot{x}\cos\theta \tag{2.7}$$

It can be seen from this model that there is no friction considered.

A well organized mathematical model of the inverted pendulum by using Lagrangian approach is done by A. Stimac (1999). In his work, the generalized forces are analysed first:

$$\Xi_x = f(t) - c\dot{x} \tag{2.8}$$

$$\Xi_\theta = -k\dot{\theta} \tag{2.9}$$

Then the Lagrange's equations for x and θ can be expressed respectively:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = \Xi_x \tag{2.10}$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = \Xi_\theta \text{ (Stimac, 1999:9)} \tag{2.11}$$

Simplifying and rearranging the Lagrange's equations, the obtained system equations are:

$$(M + m)\ddot{x} + k\dot{x} - mL\cos\theta\ddot{\theta} + mL\sin\theta\dot{\theta}^2 = F(t) \tag{2.12}$$

$$-mL\dot{x}\cos\theta + (mL^2 + I)\ddot{\theta} + c\dot{\theta} + mgL\sin\theta = 0 \text{ (Stimac, 1999:11)} \tag{2.13}$$

But the inverted pendulum system that he used does not have pendulum mass, only rod is involved. This is different from the pendulum system that is used in this project. Similar work has been done by different researches, such as: A. Formal'skii (2006:56-64), A. Brizard, G. González and so on.

2.2.2 Overhead crane

There are several different kinds of overhead cranes. The cart can move in one-dimension or two-dimensions, in other words, the cart moves along a straight line or on a surface. As a common fact, the systems have the same dynamics on their own directions (D. Liu et al, 2005). The payload is connected to the cart through a flexible cable (d'Andréa-Novel and Coron, 2000) or a rigid rod. The payload is fixed or it can move along the cable or rod. All these factors make the mathematical models of the overhead crane to be different. The reconfigurable plant used here can be configured into an overhead crane in which the cart can only move in one dimension. The pendulum corresponds to the payload in this system and it can move along the rod. The overhead crane is an inherently stable system. Its mathematical model can be derived by using the similar method that is used for the inverted pendulum mathematical model derivation. In this section, some existing overhead crane models are studied.

2.2.2.1 Overhead crane model derivation based on Newton's laws

The overhead crane studied by Fang et al (2001) has very similar structure with the overhead crane that is used in this project. But there are some assumptions made to simplify the model of the crane system, which include:

- the cart can move only in one direction, i.e. only a planar swinging of the load is considered;
- the load can be regarded as a point mass;
- the cable has negligible mass and it is rigid;
- nonlinearities like backlash or coulomb friction are not considered (Fang et al, 2001:3767)

In the considered pendulum system, the configuration is that the cart can only move in one dimension. The payload is a pendulum and it is homogeneous medium which can be considered as a point mass. Based on the frame of the pendulum system, the above first two assumptions are taken in this project. The third assumption ignores the mass of the rod and the forth assumption ignores the nonlinearities of the system. These two assumptions are not considered in the project. In the developed overhead crane mathematical model, the rod mass and the nonlinearities are considered for developing the complete models.

In the work of Fang, another assumption is made:

- The angular position and velocity of the payload and the planar position and velocity of the gantry are measurable (Fang et al, 2001:3766)

In the pendulum system, there is a servo potentiometer to measure the angle signal $\theta(t)$ and a multi-turn potentiometer to measure the position signal $x(t)$. But there is no sensor for $\dot{\theta}(t)$ measurement. This assumption is not adopted.

The above mentioned assumptions are included in the work of T. Burg et al (1996) as well. In addition, the other two assumptions are made:

- The angular position $\theta(t)$ of the payload mass is assumed to always satisfy the inequality $-90° < \theta(t) < 90°$ as measured from the vertical position
- The control input to the system is the mechanical force applied directly to the cart (Burg et al, 1996:3155)

The angular position of the payload mass of the considered pendulum system is limited by an angle limitation mechanism. The pendulum system consists of a servo system. It generates the force which is applied on the cart to control the payload. So these two assumptions accord with the structure of the physical pendulum system and they are adopted. Based on the above mentioned assumptions, the equations of motion for the overhead crane system are (Burg et al, 1996:3156):

$$m_p L^2 \ddot{\theta} + m_p L \ddot{x} \cos\theta + m_p g L \sin\theta = 0 \qquad (2.14)$$

$$m_p L \ddot{\theta} \cos\theta + (m_p + m_t)\ddot{x} - m_p L \dot{\theta}^2 \sin\theta + b\dot{x} = F \qquad (2.15)$$

where m_p is the payload mass, which equivalent to the mass of the pendulum; m_t is the mass of the cart (Burg et al, 1996:3156). Other coefficients share the same meaning as predefined symbols.

But this model does not contain a friction between the rod and the pivot and the mass of the rod. It is not sufficient to represent the overhead crane that is used in the project. Based on the selected assumptions and Newton's laws, the nonlinear and linear overhead crane mathematical models are derived in Chapter 4.

2.2.2.2 Overhead crane models derivation based on the Lagrangian approach

The Lagrangian approach can also be used for development of the overhead crane model. In the work of D. Liu et al (2005), this approach is adopted and they derived the mathematical model for a two dimension overhead crane and one dimension overhead crane respectively. The similar work of the one dimension overhead crane model development is done by R. Huilgol (1995), and the obtained mathematical model equations are:

$$(M+m)\ddot{x}+mL\ddot{\theta}\cos\theta-mL\dot{\theta}^2\sin\theta=F \tag{2.16}$$

$$mL^2\ddot{\theta}+mL\ddot{x}\cos\theta+mgL\sin\theta=0 \text{ (Liu, 2005:508)} \tag{2.17}$$

In this model all frictions are ignored. The mathematical model is not integrated.

Till now, it can be seen that there is not a complete nonlinear mathematical model for the inverted pendulum and overhead crane which can be used fully to describe the pendulum system that is used in this project. In addition, it can be noticed that the Newtonian's approach of the inverted pendulum and overhead crane mathematical models derivation is clearer and easier for one to understand, as well. This is because one goes through the analysis of the forces and the torques applied to the rod, the pendulum and the cart during the model derivation process. This derivation procedure gives one a better understanding of the systems internal dynamics. The Lagrangian approach is based on the energy analysis, it is simpler in terms of the derivation once the procedure is mastered, but it is not intuitionistic for one to understand how the system behaves. So, in this project, the Newtonian's approach is adopted.

Both linear and nonlinear models of the inverted pendulum and overhead crane mathematical models are derived in this project. The linear models are relative simpler and they cannot represent the physical system accurately. The nonlinear models can describe the physical system exactly, but with more complicated expressions. The derived nonlinear mathematical models in this project have the features of accuracy and integrity. All the factors that affect the system behaviour are considered, such as the friction between the rod and the pivot, friction between the cart and the surface, the moment of inertia of the rod and pendulum and so on. The linear mathematical models of the system are obtained by linearizing the nonlinear mathematical models. The mathematical model derivations are done in Chapter 4.

2.3 Control strategies

As the pendulum system is accepted as a test bench of the new control strategies, different control theories and methods have been used to control the inverted pendulum and overhead crane. These control theories include from classical control to modern control; from frequency domain to time domain and from linear control to nonlinear control. The applied control methods include the phase-plane method, the root locus method (Lundberg and Roberge, 2003:4399-4404), the well-known pole-placement method (Ogata, 2002:850-855), the linear state feedback control method (Z. Lin et al, 1996:933-937), the LQR control method, (http://www.engin.umich.edu/group/ctm/examples/pend/digINVSS.html), the famous PID control method (Irfan et al, 2000), the fuzzy logic control (FLC) method, model reference control method, feedback linearization method (Bugeja, 2003), neural networks and so on. The above mentioned control methods have been studied extensively and some of these methods are standard examples which can be found in the control system text book. As part of the previous work, the applications of the two most common methods – the pole-placement method and the LQR control method, on the pendulum system for the case of linearized models are presented in Chapter 5.

With the development of the control theories, control techniques, modern computation technology and because of the disadvantages of the linear control strategies, the control methods that have been used to control the inverted pendulum and the overhead crane have been transferred from linear to nonlinear control theories and methods. During this transformation, many researchers have gone into the nonlinear control theories and methods. Many novel theories, techniques and approaches have been introduced including feedback linearization, variable structure control (sliding model control), backstepping, regulation control, nonlinear H_∞ control, internal model principle, and H_∞ adaptive fuzzy control. Some of these theories and approaches have been well adopted to control the inverted pendulum, the overhead crane and other systems.

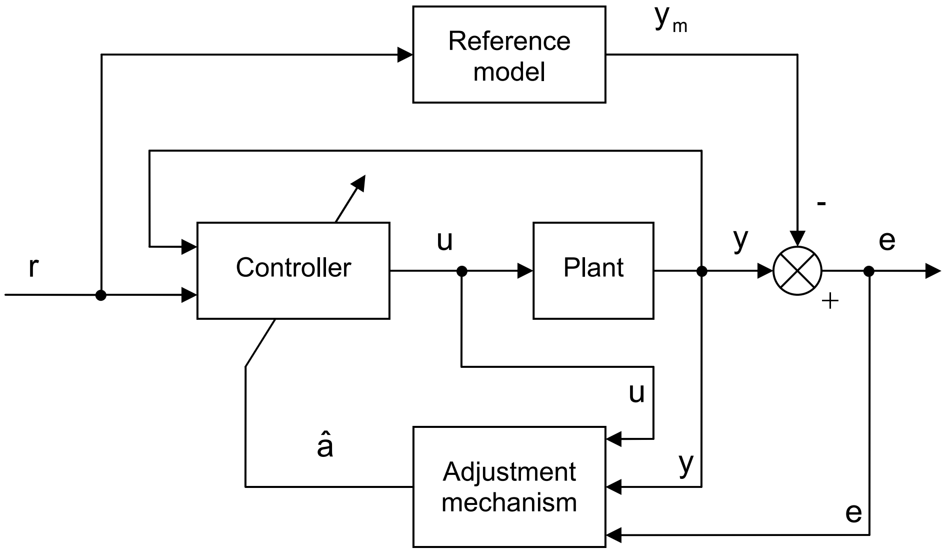
In recent years, the further work of controlling the inverted pendulum and overhead crane is focused on the combined linear-nonlinear, nonlinear-nonlinear control strategies. The combined control strategies demonstrate an incredible advantage over the individual control strategies in terms of the system responses and performance indices. Now, one may realise that this is also one of the development directions of the control theory. Different kinds of hybrid combinations of different control methods have been developed and applied onto controlling the inverted pendulum and the overhead crane. In the work of P. Phan and T. Gale (2008:871-899), a direct self-structuring adaptive fuzzy control (DSAFC) scheme is presented, which combines the self-structuring with adaptive control and FLC. This proposed control scheme is applied to control an inverted pendulum and a magnetic levitation system (Phan, Gale, 2007:871). An LQR-based linear control strategy, valid in the “linear region” of operation of the pendulum, is generalized to a nonlinear direct fuzzy control design and used to control an inverted pendulum in a large range (Yurkovich, Widjaja, 1996:455). The predictive control and pole-placement are brought together to form the predictive-pole-placement (PPP) algorithm in continuous-time by

Gawthrop and Ronco (2002). The method is used to control an inverted pendulum system in P. Gawthrop and L. Wang's work (Gawthrop, Wang, 2005:1347). Q. Wu et al (2002) did work on combining the feedback control and neural network for controlling of a base-excited inverted pendulum (Wu, 2002:267). An on-line adaptive control scheme based on feedback-error-learning is proposed and applied to inverted pendulum by X. Ruan et al (2006). An adaptive sliding model fuzzy control approach is proposed for a two-dimensional overhead crane by D. Liu et al (2004). This method combines the robustness of the sliding-model-control (SMC) and independence of the FLC. In B. d'Andréa-Nove and F. Boustany's work (1991), the linearization technique and the Lyapunov approach were combined with adaptive control to deal with the problem of parametric uncertainties and the methods are applied to control an overhead crane (1998). In the work of Ö. Altinöz (2007), the adaptive backstepping controller for inverted pendulum is designed. Adaptive nonlinear control is well known for solving the parameter uncertainty problems. The method of backstepping is a novel nonlinear control technique based on the Lyapunov design approach and its key idea is to drive the error equation to zero by designing unique control and parameter adjustment laws.

On the basis of the said above it can be concluded that a lot of work has been done for controlling the inverted pendulum and the overhead crane by using different individual or combined control strategies. Among them, the adaptive control, feedback linearization control and Lyapunov stability theory attract more attention from researchers. This is because of the robustness of the adaptive control; the simplicity of the feedback linearization control and the universality of the Lyapunov stability theory. Because of the limitation of pages, not all of the above mentioned work can be presented in this thesis. However some of the representative control methods and theories, which have been used to control the inverted pendulum and the overhead crane, either in the individual forms or in the combined forms are studied in detail in the following sections.

2.3.1 Adaptive control involved in the control of inverted pendulum and overhead crane

Adaptive control of nonlinear systems has drawn much attention during last decade and many significant developments have been achieved (Yang and Wang, 2006:932). Adaptive control is used to control the system with uncertainty parameters. For instance, it can be used for ship steering. For a long distance cruise, the ship is controlled by the automatic steering. However, the dynamic of the ship strongly depends on many uncertain parameters, such as water depth, ship loading, wind speed and other factors. Adaptive control can be used to achieve good control performance under varying operating conditions, and at the same time avoid the energy loss caused by excessive rudder motion (Slotine and Li, 1991:313). Adaptive control has been used in other fields, for example: robot manipulation, aircraft control, process control and so on. Though the adaptive control system can be used for linear and nonlinear system, the essence of the adaptive control is nonlinear and it is intimately connected with Lyapunov theory which is reviewed in a later section. There are two main methods for adaptive controller construction. One is called model-reference adaptive control method and the other is called self-tuning method (Slotine and Li, 1991:315).



â is the estimated parameters

Figure 2.2: General structure of MRAC scheme

(Adapted from Ioannou and Sun, 2003:314)

Model reference adaptive control (MRAC) is one of the model-based approaches, which combines the model reference control (MRC) and adaptive control, has been widely used in control of nonlinear systems. The quantity of literature that has been published in this field proves this point. A model reference adaptive control system can be schematically represented by Figure 2.2. It consists of four basic parts: a plant containing unknown parameters, a reference model for compactly specifying the desired output of the control system, a feedback control law containing adjustable parameters, and an adjustment mechanism for updating the adjustable parameters (Slotine and Li, 1991:315).

The plant is assumed to have a known structure, although the parameters are unknown. The reference model is chosen to generate the desired trajectory, y_d, which the plant output y must follow. The tracking error $e=y-y_d$ represents the deviation of the plant output from the desired trajectory (Ioannou and Sun, 2003:313). The plant parameters, such as the tracking error signal, control signal and output signal are sent to the controller via the adjustment mechanism. Often, only the tracking error is sent to the controller via the adjustment mechanism. When the plant parameters are exactly known, the controller should make the plant output identical to the reference model. When the plant parameters are not known, the adaptation mechanism will adjust the controller parameters to ensure a perfect tracking is asymptotically achieved. The adjustment mechanism is used to adjust the parameters in the control law. The adaptation law searches for parameters such that the response of the plant under adaptive control becomes the same as the reference model. The essential idea is to make the tracking error converge to zero (Slotine and Li, 1991:316). A very important issue in adaptation design is to combine an adaptation mechanism that always ensures the control system is stable and the tracking error converges to zero while the plant parameters are varied. Frequently, Lyapunov theory is used to study the system stability by mean of Lyapunov functions.

It can be seen that the desired trajectory, in the MRAC system, is no longer specified by a constant value, but by the output signal of a reference model. The control signal is generated based on the plant parameters. During this process, the tracking error converges to zero. It is easy to distinguish the MRAC from others, because it consists of a reference model and an adjustment mechanism which is associated with the controller. The reference model is usually linear, stable, and of the same of order as the nonlinear plant. The reference model states are expected to be tracked by the plant states (Jain et al, 2004:939) as it is in the MRC. A model reference control structure has been successfully combined with fuzzy neural network (Chen, Teng, 1994:291).

As was mentioned previously, Lyapunov theory is associated with adaptive control frequently for stability analysis. In fact, Lyapunov theory is one of the most important approaches used for the system stability analysis. Once a controller is designed, it is necessary to study the closed-loop system stability. Only when the closed-loop system is proven stable, then the controller design is successful. In the next section, through a detailed example, the Lyapunov direct method which is one of the most important parts of the Lyapunov theory is presented. A close look at what Lyapunov stability contributed for the controlling of the inverted pendulum and the overhead crane is presented as well.

2.3.2 Lyapunov stability theory involved in the control of inverted pendulum and overhead crane

Lyapunov stability theory has been widely used in the control engineering and it is also involved in the control of an inverted pendulum quite frequently. Particularly the Lyapunov direct method is extensively applied for analysis of the system stability and design the controller based on the Lyapunov functions. The Lyapunov function made a remarkable impact on stabilization theory. It transformed the stability descriptions into tools for solving the stabilization tasks (Cai et al, 2006: 598). The Lyapunov stability theory is studied in Chapter 3 in detail. Here, referring to the control of the inverted pendulum, some exclusive Lyapunov functions are presented. First of all, the definition of the Lyapunov function is given:

Definition 2.1: A smooth, proper and positive definite function V: $R^n \rightarrow R^+$ is a Lyapunov function of the nonlinear system:

$$\dot{x} = f(x) + g(x)u \qquad (2.18)$$

where $x \in R^n$ and $u \in R^n$ are the state and input of the system respectively. f(x) and g(x) are the nonlinear functions.

if:

$$\inf\{L_f V(x) + L_g V(x)u\} < 0 \qquad (2.19)$$

for $x \neq 0$ (Cai et al, 2006:598,599).

where $L_f V(x)$ is the Lie derivative of V(x) with respect to f and $L_g V(x)$ is the Lie derivative of V(x) respect to g.

The Lyapunov function can be used to analyse the system stability. In the work of I. Abdelmalek et al (2007), a quadratic Lyapunov function and a non-quadratic Lyapunov function are used to analyse the closed-loop system stability based on the derived parallel distributed compensation (PDC) scheme of a Takagi-Sugeno (T-S) fuzzy system. How to examine the stability of a system based on Lyapunov functions is presented. A general procedure of stability study is done. In this work, the inverted pendulum is represented by T-S fuzzy model whose construction is based on the identification (fuzzy modelling) using input-output data or derivation from given non-linear system equations. (Abdelmalek et al., 2007:598).

A quadratic Lyapunov function is defined as follows:

Definition 2.2: The system $\dot{x}(t) = f(x(t), u(t))$ is said to be quadratically stable if there exists a quadratic function: $V(x(t)) = x^T(t)Px(t)$, $V(0) = 0$, satisfying the following conditions:

$$V(x(t)) > 0,\ \forall x(t) \neq 0 \Leftrightarrow P > 0 \text{ and } \dot{V}(x(t)) < 0,\ \forall x(t) \neq 0 \tag{2.20}$$

If V exists, it is called a Lyapunov function.

The stability conditions referring to a quadratic Lyapunov function for the above closed-loop system were derived as follows by Tanaka and Sugeno in 1992.

Theorem 2.1: The fuzzy system can be stabilized via the PDC controller if there exists a common positive definite matrix X and M_i, i=1,…,r, such that:

$$-XA_i^T - A_iX + M_i^TB_i^T + B_iM_i > 0 \tag{2.21}$$

$$-XA_i^T - A_iX - XA_j^T - A_jX + M_j^TB_i^T + B_iM_j + M_i^TB_j^T + B_jM_i > 0 \tag{2.22}$$

for all i<j such that $h_i \cap h_j \neq \varnothing$, where

$$X = P^{-1},\ M_i = F_iX \tag{2.23}$$

whereas the single quadratic Lyapunov function is given by

$$V(x(t)) = x(t)X^{-1}x(t) \tag{2.24}$$

The problem of finding a Lyapunov function is transferred to finding the appropriate positive definite matrices M_i and X. If positive definite matrices M_i and X exist, then, the feedback gain F_i can be calculated. Here, the PDC controller is borrowed to demonstrate the procedures. Other control methods can be used in a similar manner. In the work, it is mentioned that because of the various fuzzy rules, it can constrain a range of positive definite matrices selection. But if the control strategy is not like FLC which is based on the different control rules, the quadratic Lyapunov function is sufficient for system stability analyse. Furthermore, the quadratic Lyapunov function can be used to analyse and design a suitable controller for the nonlinear system, as proven by this work.

The controller design based on Lyapunov function is based on the definition of Lyapunov function. Referring to Definition 2.1: Lyapunov function itself must be positive definite V>0 and its first derivative referring time must be negative definite $\dot{V} < 0$. If a control signal u is included in the Lyapunov function, by using the constraining conditions of the definition of Lyapunov function, the controller can be found.

In this work, a non-quadratic Lyapunov function is shown as well which is defined as:

Definition 2.3: (Tanaka and Sugeno, 1992) Equation:

$$V(x(t)) = \sum_{i=1}^{r} h_i(z(t))\, x^T P_i x(t) \tag{2.25}$$

is a Lyapunov function, where P_i is a positive definite matrix. (Khalil, 1996; Tanaka et al., 2003)

By using this non-quadratic Lyapunov function and following the above mentioned procedures, a less conservative positive definite matrix solution can be found (Abdelmalek, 2007:41). Applying this approach, the objective of balancing and stabilizing the pendulum is realized with success for different initial conditions (Abdelmalek, 2007:46).

Other Lyapunov functions have been proposed as well. R. Lozano et al (2000:197) presented a control strategy and a Lyapunov function for the inverted pendulum based on the total energy of the system. In order to present the used Lyapunov function, the system model and energy equation are presented first. In this work, the model of the inverted pendulum is given by:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{2.26}$$

Where:

$$q = \begin{bmatrix} x \\ \theta \end{bmatrix};\ M(q) = \begin{bmatrix} M+m & mL\cos\theta \\ mL\cos\theta & mL^2 \end{bmatrix};\ C(q,\dot{q}) = \begin{bmatrix} 0 & -mL\sin\theta\dot{\theta} \\ 0 & 0 \end{bmatrix};$$

$$G(q) = \begin{bmatrix} 0 \\ -mgL\sin\theta \end{bmatrix} \text{ and } \tau = \begin{bmatrix} F \\ 0 \end{bmatrix}.$$

The potential energy of the pendulum can be defined through $G(q)$ as:

$$G(q) = \frac{\partial P}{\partial q} = \begin{bmatrix} 0 \\ -mgL\sin\theta \end{bmatrix} \tag{2.27}$$

The total energy of the cart and pole system is given as:

$$E(q,\dot{q}) = K(q,\dot{q}) + P(q) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + mgL(\cos\theta - 1); \tag{2.28}$$

Based on the total energy equation, the proposed Lyapunov function is expressed as:

$$V(q,\dot{q}) = \frac{k_E}{2}E(q,\dot{q})^2 + \frac{k_v}{2}\dot{x}^2 + \frac{k_x}{2}x^2 \tag{2.29}$$

where k_E, k_v and k_x are strictly positive constants. It can be seen that $V(q,\dot{q})$ is a positive semi-definite function. Its first derivative according to time is expressed as:

$$\dot{V} = k_E E\dot{E} + k_v \dot{x}\ddot{x} + k_x x\dot{x} \qquad (2.30)$$

In the work of C. van Kats (2004), a Proportional Derivative (PD) controller is designed and a proposed Lyapunov function is used to analyse the closed-loop system stability. The Lyapunov function is expressed as:

$$V = \frac{1}{2}\theta^2 + \frac{1}{2}\left(\dot{\theta} + a\theta^2\right)^2 \qquad (2.31)$$

where a is a constant coefficient. The time derivative of V is:

$$\dot{V} = -a\theta - b\left(\dot{\theta} + a\theta\right)^2 \qquad (2.32)$$

It can be seen, this Lyapunov function has the quadratic form as well. Its first derivative $\dot{V}$ can be easily proven as negative definite function.

There are other Lyapunov functions that have been successfully applied to analyse the stability of the inverted pendulum or design a controller for the inverted pendulum. From above listed Lyapunov functions, one can easily see that the quadratic Lyapunov function has the inherent predominances in comparison with others, for instances: It is a well studied Lyapunov function which has been applied extensively for different linear and nonlinear systems and good results have been obtained; It has a much simpler expression, the simpler expression is very meaningful and helpful once the closed-loop system is complicated, because there are huge amount of derivations and calculations involved, a relative simpler Lyapunov function will reduce the mathematical work dramatically. Finally, though Abdelmalek et al (2007) did not used quadratic Lyapunov function to analyse the stability of the closed-loop inverted pendulum system, they have proved that the quadratic Lyapunov function is sufficient to analyse the stability of the inverted pendulum indirectly. Furthermore, the use of the quadratic Lyapunov function can be extended for controlling of the overhead crane easily. As it is known, the overhead crane is inherently stable. The other typical and lately used nonlinear control technique of feedback linearization is studied in next section.

2.3.3 Feedback linearization involved in the control of inverted pendulum and overhead crane

Feedback linearization is a nonlinear control design method. It has attracted a great deal of research interest in recent years. The idea of this approach is to algebraically transform a nonlinear system dynamic into a (full or partly) linear one by choosing a different state representation (Hedrick and Girard, 2005:134). Then the linear control techniques can be applied (Slotine and Li, 1991:207). The feedback linearization approach has been applied successfully to address many real control problems. These include the control of an electromagnetic suspension system (Arreola, 2004:2153-2160), pendulum system (Hassanzadeh et al, 2008:1322-1328), spacecraft (Bajodah, 2006:35-53), electrohydraulic actuated mechanism (Engleder, 2007:448-456), car-pole system (Bedrossian, 1992:1987-1992), bank-to-turn missile system (Huang and Lin, 1993:569-573), and so on. Here, some of the successful applications of the feedback linearization method are presented.

In the work of C. Chen et al (2007), proposed a feedback linearization for tracking disturbance decoupling control. The closed-loop system with the proposed controller is valid for any initial condition. The controller is successfully applied on the nonlinear AMIRA ball and beam system which has similar dynamic to the inverted pendulum.

A summarized proposed controller derivation is described as follows:
A nonlinear control system with disturbances is considered:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, ..., x_n) \\ f_2(x_1, x_2, ..., x_n) \\ \cdot \\ \cdot \\ f_n(x_1, x_2, ..., x_n) \end{bmatrix} + \begin{bmatrix} g_1(x_1, x_2, ..., x_n) \\ g_2(x_1, x_2, ..., x_n) \\ \cdot \\ \cdot \\ g_n(x_1, x_2, ..., x_n) \end{bmatrix} u + \sum_{i-1}^{p} q_i^* \theta_i \tag{2.33}$$

$$y(t) = h(x_1, x_2, ..., x_n) \tag{2.34}$$

This system can be presented in a compact for as:

$$\dot{x}(t) = f(x(t)) + g(x(t))u + \sum_{i=1}^{p} q_i^* \theta_i \tag{2.35}$$

$$y(t) = h(x(t)), \tag{2.36}$$

where $x(t) := [x_1(t), x_2(t), ..., x_n(t)]^T \in R^n$ is the state vector, $u \in R^1$ is the input, $y \in R^1$ is the output, $\theta := [\theta_1(t), \theta_2(t), ..., \theta_p(t)]^T$ is a bounded time-varying disturbance vector, f, g,

$q_1^*,\ldots,q_p^*$ are smooth vector fields on R^n, and $h(x(t))\in R^1$ is a smooth function. The nominal system is then defined as follows:

$$\dot{x}(t) = f(x(t)) + g(x(t))u \tag{2.37}$$

$$y(t) = h(x(t)), \tag{2.38}$$

with relative degree r.

$$L_g L_f^k h(x(t)) = 0,\ k < r-1 \tag{2.39}$$

$$L_g L_f^{r-1} h(x(t)) \neq 0 \tag{2.40}$$

for all $x\in R^n$ and $t\in[0, \infty)$.

The desired output trajectory $y_d(t)$ and its first r derivative are all uniformly bounded and:

$$\left\|\left[y_d(t), y_d^{(1)}(t), \ldots, y_d^{(r)}(t)\right]\right\| \leq B_d \tag{2.41}$$

where B_d is some positive constant.

Under the assumption of well-defined relative degree, it has been shown that the mapping:

$$\phi_i(x(t)) := \xi_i(t) = L_f^{i-1} h(x(t)),\ i = 1, 2, \ldots, r \tag{2.42}$$

$$\phi_k(x(t)) := \eta_k(t),\ k=r+1, r+2, \ldots, n \tag{2.43}$$

and satisfying

$$L_g \phi_k(x(t)) = 0,\ k=r+1, r+2, \ldots, n \tag{2.44}$$

is a diffeomorphism onto image. For the sake of convenience, define that trajectory error to be:

$$e_i(t) := \xi_i(t) - y_d^{(i-1)}(t),\ i=1, 2, \ldots, r \tag{2.45}$$

$$e(t) := [e_1(t), e_2(t), \ldots, e_r(t)]^T \in R^r \tag{2.46}$$

and $\xi(t) := [\xi_1(t), \xi_2(t), \ldots, \xi_r(t)]^T \in R^r$ (2.47)

$$\eta(t) := [\eta_1(t), \eta_2(t), \ldots, \eta_r(t)]^T \in R^r \tag{2.48}$$

$$\begin{aligned} q(\xi(t), \eta(t)) &:= [L_f\phi_{r+1}(t), L_f\phi_{r+2}(t), \ldots, L_f\phi_n(t)]^T \\ &:= [q_{r+1} \quad q_{r+2} \quad \cdots \quad q_n]^T \end{aligned} \tag{2.49}$$

Theorem 2.2: Suppose that there exists a continuously differentiable function which has V: $R^{n-r}\rightarrow R^+$ such that the following three inequalities hold for all $\eta\in R^{n-r}$:

(a) $\omega_1 \| \eta \|^2 \le V(\eta) \le \omega_2 \| \eta \|^2$, $\omega_1, \omega_2 > 0$, (2.56)

(b) $\nabla_t V + (\nabla_\eta V)^T q_{22}(t, \eta, 0) \le -2\alpha_x \| \eta \|^2$, $\alpha_x > 0$, where $q_{22}(t, \eta, 0) = q(\xi, \eta)$, (2.57)

(c) $\| \nabla_\eta V \|^2 \le \varpi_3 \| \eta \|$, $\varpi_3 > 0$ (2.58)

Then, the tracking problem with almost disturbance decoupling is globally solvable by the controller defined by:

$$u = \left[L_g L_f^{r-1} h(x(t))\right]^{-1} \left\{ \begin{array}{l} -L_f^r h(x) + y_d^{(r)} - \varepsilon^{-r} \alpha_1 \left[L_f^0 h(x) - y_d\right] - \varepsilon^{1-r} \alpha_2 \left[L_f^1 h(x) - y_d^{(1)}\right] \\ - \cdots - \varepsilon^1 \alpha_r \left[L_f^{r-1} h(x) - y_d^{(r-1)}\right] + K_x^T x \end{array} \right\} \quad (2.59)$$

where $K := [k_1, k_2, \ldots k_n]$ is some adjustable real matrix. ε is adjustable positive constant.

In the work of Sharma and Kar (2007), the feedback linearization method is combined with adaptive control to control a chaos system. The adaptive control is used to estimate the uncertain parameters in the model. The selected reference model has the same order as the plant and is a stable linear system. The system unknown parameters are evolved using an adaptation law. The adaptation law aims to drive the system towards the ideal condition to achieve the perfect matching with the reference model.

A recurrent high-order neural network for both identifying and controlling unknown chaotic systems is proposed by Z. Lu et al (2006:2337-2354). In this work, the feedback linearization technique is used in an adaptive manner. The Lyapunov theory is used to prove the boundedness of the asymptotic stability of the tracing error (Lu et al, 2005:2337-2354).

An exact feedback linearization of the mobile robots is proposed in the work of K. Park et al (2000). Based on newly proposed coordinates, it is possible to transform the point stabilization problem into a problem of controlling a linear time-invariant system. By using the well-established linear control theory, the problem of the robot point stabilization is solved (Park et al, 2000). In this work, the linear feedback control is used. The developed kinematic system of the mobile robot satisfies the necessary and sufficient conditions for the state-space exact feedback linearization. The results can be extended to other kinematic system of the mobile robots with some modification (Park et al, 2000).

From review of the above literature, it can be seen that the feedback linearization approach has been well adopted in different fields which are facing nonlinear control problems. The inverted pendulum is a typical nonlinear system with the property of being under-actuated. In the work of C. Chen et al (2007), a good controller, which is almost universal, is proposed. However in that work, the desired trajectories are not specified.

In section 2.3, different control theories, techniques and methods are briefly studied associated with controlling of the inverted pendulum and overhead crane. Based on the study of the existing research work, two methods are proposed in this project for the controller design. One is Lyapunov stability theory based model reference control design, the other is the feedback linearization based model reference control design. In next section, some existing Real-Time implementation platforms are introduced.

2.4 Hardware implementation platform

Most of the existing work and relative publications only focus on the simulation work and most of the simulations are done in the Matlab environment. Very limited studies have been done on the Real-Time implementations. In the work of J. Lam a general purpose data acquisition and control board MultiQ-3 associated with Matlab/Simulink is used to implement the controller and analyze data for the inverted pendulum (2004). Similarly, in N. Muškinja and B. Tovornik’s work (2006) control algorithm and data acquisition and visualization are realized with Matlab/Simulink, and Real-Time xPC target products on a personal computer (PC) with 12-bit AD/DA converter. The TMS320C6711 Digital Signal Processor (DSP) is used as the controller by M. Bugeja (2003) for the swing-up and stabilizing an inverted pendulum. The DSP (MS320C50) combined with a FPGA is used to implement the control of a biped robot by S. Oh et al (2003). In the work of A. Irfan et al (2000), a Single Board Computer (SBC) and a PI and PD op-amp based analog electronics circuit are used to construct a PID controller for controlling of an inverted pendulum. In the work of D. Liu et al (2004), the control algorithm for controlling of a two-dimensional swing-free transporting overhead crane is implemented on a Pentium III 800MHz PC running under the Windows operation system.

From the existing implementation works, it can be seen that basically there are four ways to realize the Real-Time implementations. They are DAQ, DSP, SBC and analog circuits. In this project, the controller is designed as a reconfigurable controller which is used to implement complex nonlinear control algorithms in the industrial environment. So, it is necessary for the complicated nonlinear control algorithm to be deployed. It must be reprogrammed quickly in order to synchronize with the plant reconfiguration. According to these requirements, the implementation platform must have characteristics such as:

sufficient high speed Input/Output (I/O) ports, field programmable ability, and capacity to deal with complicated mathematical calculation. The FPGA can fulfil the above mentioned requirement. Furthermore, each I/O port of the FPGA can be defined by the user which makes it much more flexible, this is meaningful when it comes to the hardware design in terms of reducing the wiring and simplifies the design. Considering the application of the RMS in the industry, the reconfigurable controller must have the properties of:

- Big capacity to accommodate huge control algorithms
- Flexibility in terms of reconfiguration friendly
- Customization in terms of reconfigurablility according to the requirements
- Reliability for extended running periods without problem
- Safety in terms of it not being harmful to the works and other equipment around
- Easy maintenance in terms of easily de-bugging and fixing
- Diagnosibility in terms of self-diagnosis
- Cost efficiency, no additional cost

In the previously mentioned papers above Real-Time implementation platforms cannot fulfil the above described properties. The DAQ must work with PC which limits the flexibility and reliability. The DSP is a good alternative candidate besides its cost efficiency. SBP has limited I/O ports and limited capability and speed. Analog circuit is not suitable for digital control and it is difficult to implement complex nonlinear control algorithms. As a contrast, the FPGA based controller, for example, NI CompactRIO FPGA is manufactured according to the industry standards. It can fulfil the above mentioned requirements. It is suitable for use in industry. Additionally, LabVIEW as the support software for the NI CompactRIO is well developed software. This provides easy compatibility for the software and hardware integration.

In this project, the NI CompactRIO is adopted as the final Real-Time implementation platform as it is exclusively advantageous. The DAQ system based Real-Time implementation platform is used as well, but it is only used as the transition to prove that the designed controllers are capable to control the inverted pendulum and the overhead crane in Real-Time.

2.5 Conclusion

In this chapter, some of the existing mathematical models of the inverted pendulum and the overhead crane are studied. Some control strategies for controlling the inverted pendulum and the overhead crane have gone through and some existing Real-Time implementation platforms are shown as well. By studying these existing works, it proves that the complete mathematical models for the pendulum system are necessary to be derived. These mathematical models include the inverted pendulum nonlinear and linear models, the overhead crane nonlinear and linear models. The proposed control strategies include a Lyapunov stability theory based model reference control design and a feedback linearization based model reference control design. In the next chapter, some typical control strategies either linear or nonlinear are studied, especially, the Lyapunov direct method, the feedback linearization method and the model reference method. From the implementation point of view, FPGA is the best choice to fulfil the requirement of the reconfigurable controller. The detailed descriptions of the FPGA are presented in Chapter 8.

CHAPTER THREE
LINEAR AND NONLINEAR CONTROL THEORY

3.1 Introduction

The world consists of all kinds of different systems, big or small, simple or complicated. The human practice actions aim at knowing and changing the world. The control theory is the subject that studies different systems and finds the universal common control law. Nowadays the application of the control theory extends all over every aspect of the modern life. It becomes one essential artifice that human beings can use to understand and remodel the world.

Control theory is the base of the automation. Automation releases the human from burdensome, even dangerous physical force labour, and it is also the key technique that boosts the work productivity and the product quality. For a given system, characterized with nonlinear behaviour and outside environmental uncertainty, control based only on the human sense or experience can not reach the high precise control requirements. Better solutions are given by the control theory. The control theory must give the best or satisfied control strategies in order to get the quantification precision or ideal targets. It is based on application of the mathematical methods. For every system to be controlled, the control theory must solve the following three basic questions: (Guo, 1997)

- Is the system controllable
- How to overcome the uncertainty of the system structure and the influence of the disturbance
- How to find the detailed and satisfied implementable control strategy

In order to answer the above mentioned questions, different control theories, techniques and methods are developed. Here are some examples. The primary thing of studying a system is to find the system mathematical model. Because of the complexity of the system, it is difficult to derive the accurate model only based on the physical, chemical and biological laws directly. As an alternative, by using the system input and output data, the parameters of the model can be found using the theory of identification. Because of many systems, states are not measurable and the disturbances include random noise, which initiated the development of the signal filtering theory. Since many system parameters are not a pre-determined and they are time varying, the robust control theory and adaptive control theory are carried out. For the more uncertain and more complex systems, the artificial neural networks and artificial intelligence are necessary to be deployed. With the aim of finding the idiographic control method, it is required to apply the system optimal mathematical theories and methods. For implementing these control strategies, the advanced computation machine, different instruments and implementation components are used. Finally, the main concern is whether the system is stable under the closed feedback loop, and whether the system achieves the control target. To answer these questions on the theory level, the high-order nonlinear random dynamic equations must be solved.

Since the end of the 1950s, the control theory went through a transformation from the classical control theory to modern control theory. The classical control theory focuses on the single-input and single-output (SISO) linear system, the typical theory includes: Routh-Hurwitz stability criterion, Nyquist analysis, Bode chart, Ziegler-Nichols tuning, Wiener filtering and so on. Euler equation and stationary random processes are the main mathematical tools used during the classical control period. The modern control theory, contacted almost all the sub-branches of the applied mathematics with the wider and deeper research spectrum. The modern control theory includes the following algebraic theories, such as: Pontryagin maximum theory, Bellman's dynamic programming, Kalman filtering theory, state space model's controllability, observability and feedback stabilization. During the past forty years, modern control theory developed rapidly driven by the engineering techniques need and computer technology development. Important progress has been made especially in the area of: nonlinear control, distributed parameter control, stochastic control, robust control, adaptive control and so on.

With the rapid development of new computer and high technologies and with the growth of economical competitiveness in recent years, people face more and more complex dynamic systems where the requirements towards the system accuracy, efficiency and reliability are very high. New control problems in space technology, life science, industrial processes, society and environmental fields are bringing new challenges towards the control theories. The complex systems have the following common properties, such as: randomicity of the system dynamics and disturbances; high nonlinearity of the system dynamic; roughness of the measured information; the mixture of the continuous and discrete; high order and distribution of the state variables; strong coupling between subsystems. The complex system control has broken the limitation of the traditional control concepts and method in scale, complexity and agility. It requires the controller to have the ability to "study" and "recognize" the system, to adapt and to be robust towards the disturbances. It can be seen that the direction of development of control theory is to investigate more and more advanced intelligent control methods in order to process more and more complicated practical systems. Control theory is a highly integrated cross subject, it has interactions with mathematics, computer science and engineering technique, it also has strong links with physical science and management science. The general concepts involved in the control theory have important influence over the technology progress and society development.

As mentioned previously, the control theory is a powerful tool to guide the practice. In this chapter, the linear control theory and nonlinear control theory are presented in section 3.2 and 3.3 respectively. Referring to this project, the main issue of the project is to deal with the nonlinear controller design which is discussed in detail. The linear control theory as the relative knowledge and fundamental control theory will be introduced briefly first.

3.2 Linear control theory

Linear control theory focuses on the linear control system analyses and design. When the linear control theory is used to analyse a control system, the system or plant model and the corresponding controller are given. Linear control theory deals with linear systems, but in the real world most systems are nonlinear. The linear systems have always being considered as a special case of the nonlinear systems. However, in many cases the nonlinearity is very small and can be neglected, or the limits of operation are small enough to allow a linear analysis to be applied (D’Azzo and Houpis, 1988:18). If the given model is nonlinear, it is necessary to find the equivalent linearized model first. Then, based on the system model and pre-designed controller, the performance of the control system is studied. The linearization can be done through the Taylor series expansion. The detailed procedures are revealed in a later section of the chapter. When the techniques are used to design the controller for a given process, the procedures are reversed. At this time, the control specifications are given. By applying the control theories and design techniques, a proper controller must be designed. Then, the designed controller should control the system to meet all specifications such as the required rising time, overshoot, steady state error and others.

3.2.1 Linear system

A system satisfying the superposition principle can be classified as linear system. A linear system can be described in the complex domain by a transfer function, or in time domain it can be described by a set of first order differential or difference equations for continuous time and discrete time respectively. To simplify the complexity, the systems considered in this chapter are all continuous ones and expressed in time domain. As an example, the continuous Linear Time-Varying (LTV) system equation is presented in state space form as follow:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \qquad (3.1)$$

where $x(t) \in R^n$ is the state vector. $y(t) \in R^l$ is the output vector and $u(t) \in R^m$ is the input vector. $A(t) \in R^{n \times n}$ is the state matrix, $B(t) \in R^{n \times m}$ is the input matrix, $C(t) \in R^{l \times n}$ is the output matrix and $D(t) \in R^{l \times m}$ is the direct transmission matrix.

As a very conventional control strategy, the linear control theory has been well developed and applied. The applications have covered such a broad range. Here the linear control theories are applied for linear system analyses and controller design.

3.2.2 Linear system performance analyses

Linear system performance analyses can be deployed in the complex domain and the time domain. In complex domain, the transfer function is used to represent the system, either open loop system or closed-loop system. As a well known mathematical tool, the Laplace transform is used to transform the system from time domain differential equations to the complex domain. The system performance analyses in complex domain take the study of the system steady state response to various inputs signals, for instance, impulse input, step input and ramp input. One of the very important benefits of this method to analyse linear systems is that the linearity and the algebraic feature of the Laplace transform allows for the modularizing of the total system into subsystems and for the combining of subsystems into large super systems in an easy and simple way. On the other hand, the transfer function based control strategies have the limitation to be applied on the system with non-constant parameters and multiple input multiple output (MIMO) systems. At this stage, the time domain state space analyses are introduced (Batzel, 2001:21). The state space theory has the advantages and includes features which are capable to handle MIMO systems, incorporates optimal design and provides numerical methods for solutions.

At the system performance analysis stage, stability, controllability and observability are the most important properties of a system. These properties must be investigated to determine the level and usefulness of control that can be exerted on the system (Mohammad et al, 1992:56). For a Linear Time-Invariant (LTI) system, these properties can be determined by the characteristic equation in complex domain. In time domain, they can be determined by the state space representation matrices (A, B, C and D) directly as shown in system equation 3.1. These properties of the LTI system and LTV system are studied briefly below because the linear control is not the main issue of the project.

3.2.2.1 Stability analyses

Stability is one of the most import properties of the system. Generally speaking, a system is useful only when it is stable. Stability is considered as a fundamental problem. Let it is assumed that the system is at some equilibrium state and is perturbed initially. If the trajectory of the perturbed system eventually returns to the equilibrium state, the system is stable. If the system trajectory moves further away from the equilibrium state, the system is unstable (Mohammad et al, 1992:56). There are stability definitions for a linear system:

a) A linear system is stable if all poles of the closed-loop transfer function are in the left half complex plane.
b) A linear system is stable if a bounded input produces a bounded output.
c) A linear system is stable if the system impulse response integral is bounded.
d) A linear system is stable if all poles of the matrix equation $(sI-A)^{-1}$ are in the left half of the s plane. (Sage, 1981:153)

There are several methods that can be used to determine the system stability, such as: Routh Hurwits criterion, Nyquist criterion. Both of them are well known and easily applied. There is also gain margin and phase margin method which can be used for stability assessment.

3.2.2.2 Controllability

A system controllability is investigated in the following way: a given system is at its initial state $x(t_0)$, if there exist such control input $u(t)$ that can transfer the system from its initial state to the desired state, the system is controllable. Only when the system is controllable, its dynamics can be influenced by the input. This definition implies that $u(t)$ is able to affect each state variable $x(t)$ (D'Azzo and Houpis, 1988: 467). A LTI system is controllable if the controllability matrix has full rank. The controllability matrix can be expressed as: $R=[A \quad AB \quad A^2B \ldots. A^{n-1}B]$, where n is the number of the states. A LTV system is controllable at time t_0, if and only if, for a moment $t_1>t_0$ the rows of $\varpi(t)=\Phi(t_0,t)B(t)$ are linearly independent in the period $[t_0,t_1]$, where $\Phi(t_0,t)$ is the system transition matrix. (Mohammad et al, 1992:59)

3.2.2.3 Observability

A system is said to be completely observable if every initial state $x(t_0)$ can be exactly determined from the measurements of the output $y(t)$ over the finite interval of time $t_0 \le t \le t_f$. This definition implies that every state of $x(t)$ affects the output $y(t)$. (D'Azzo and Houpis, 1988: 467) For LTI systems, if the rank of the observability matrix $O=[C \quad CA \quad CA^2 \ldots\ldots CA^{n-1}]$ has same value as the number of the states, then the system is observable. A LTV system is observable if and only if the columns of the matrix $C(t)\ \Phi(t_0,t)$ are linearly independent in the interval $[t_0,t_1]$.

3.2.3 Linear system controller design

As the counter part of the linear system performance analysis, in the linear system controller design phase, the design specifications are given. Based on the study of the plant, a controller is necessary to be designed. The controller should be capable of letting the system follow the desired trajectory. The linear system controller design techniques are well studied in both classical and modern control era. Several different techniques have been developed for both complex domain and time domain. Here some of them are briefly described, for example, the root locus method, the pole placement method and the quadratic optimal control method.

3.2.3.1 The root locus method

The root locus method is a graphic method applied for the SISO LTV control systems. The fundamental idea of the root locus method is to add poles and zeros to the system's open loop transfer function and to force the root loci to pass through the desired closed-loop poles in the s-plane (Ogata, 2002:417). In the sense of controller design, the effect of the poles and zeros on the open-loop transfer function is studied first. Based on the understanding of the effect, the proper poles and zeros are selected to be put to the open-loop transfer function in order to get the desired root loci.

In essence, the root loci in the root locus design are reshaped through the adding poles and zeros so that a pair of dominant closed-loop poles can be placed at the desired location. The extra poles added to the open-loop transfer function can pull the root locus to the right which can increase the system instability and slow down the settling of the response. On the other hand, the extra zeros added to the transfer function can pull the root locus to the left which can make the system more stable and speed up the settling of the response.

The procedure of adding poles and zeros to reshape the root locus in order to improve the system performance is called compensation or stabilization. When the system is compensated, it shows better performance, such as higher stability and better transient response. The steady state error can be reduced by using a bigger system gain (D'Azzo and Houpis, 1988: 343). The compensation either *lead* or *lag* can be deployed by using all kinds of different electrical or mechanical means, for example, Operational Amplifiers (Op-Amp), Resistor-Capacitor (RC) network and spring-dashpot systems.

3.2.3.2 The pole placement method

The root locus method is based on the use of transfer function in the complex domain. It is used to deal with the SISO systems. For the MIMO systems, the pole placement method, which is also known as the pole assignment method, is used. It is similar to the root locus method excepting that in the pole placement method all the closed-loop poles are placed to the desired locations rather than only the dominant closed-loop poles.

In many design problems, a system is stabilized through state feedback. Here, the basic problem of pole placement by state feedback is to find a feedback control law: u(t)=Gr(t)+Kx(t) for a LTI system:

$$\begin{aligned}\dot{x}(t) &= Ax(t)+Bu(t)\\ y(t) &= Cx(t)+Du(t)\end{aligned}, \qquad x(t_0)=x_0 \tag{3.2}$$

The closed-loop system can be expressed as:

$$\begin{aligned}\dot{x}(t) &= (A-BK)x(t)+BGr(t)\\ y(t) &= (C-DK)x(t)+DGr(t)\end{aligned} \tag{3.3}$$

where $x(t)\in R^n$ is the state vector; $y\in R^l$ is the outputs vector; $u\in R^m$ is the inputs vector; $r\in R^m$ is the reference signal. $A\in R^{n\times n}$, $B\in R^{n\times m}$, $C\in R^{l\times n}$ and $D\in R^{l\times m}$ are constant matrices with respected dimensions. Matrix $G\in R^{m\times m}$ is an adjustment gain. It is used when a nonsingular transformation on the input is applied. Matrix $K\in R^{m\times n}$ is called the state feedback gain matrix. One must be aware that only when the LTI system is completely controllable, then the eigenvalues of the closed-loop dynamic system can be arbitrarily chosen. To find the feedback gain matrix K, the standard procedures in control text books can be followed. The detailed procedures are described in Chapter 5. There commands which can be used for computation of the feedback gain matrix in Matlab are also introduced in Chapter 5. By using these commands, the problem can be solved easily.

3.2.3.3 The quadratic optimal control method

An advantage of the quadratic optimal control method over the pole-placement method is that the former provides a systematic way of computing the state feedback control gain matrix (K. Ogata, 2002:897). In modern optimal control theory of linear deterministic dynamic system, represented in continuous time by Equations (3.2), the linear state feedback control signal is expressed as: $u(t)=-Kx(t)$. By optimizing the value for the feedback gain $K(t)$, the following performance criterion which is also called performance index is minimized:

$$J=\int_0^{\infty}\left(x^TQx+u^TRu\right)dt \qquad (3.4)$$

where $Q\in R^{n\times n}$ is a positive definite (or positive-semidefinite) Hermitian or real symmetric matrix and $R\in R^{m\times m}$ is a positive-definite Hermitian or real symmetric matrix.

The criterion is selected wisely. It requires minimization of “square” of input, which is the control signal. In other words, it minimizes the input energy required to control a given system. On the other hand, it requires minimization of the “square” of the state variables. That means the differences between the desired trajectories and actual system states trajectories are minimized. The minimization can be interpreted as the goal of bringing the system as close as possible to the desired trajectories while optimizing the energy usage (http://www.ece.rutgers.edu/~gajic/psfiles/lqr.pdf). For the regulation problem, the origin is zero. If the problem is to regulate the state variables to some constant values, the problem becomes set point control problem. This transformation can be done easily by shifting the origin.

In order to find the linear feedback law, there are standard steps to derive the method equations. The obtained controller equation is:

$$u(t)=-Kx(t)=-R^{-1}B^TPx(t) \qquad (3.5)$$

The matrix P must satisfy the following reduced Riccati equation:

$$A^TP+PA-PBR^{-1}B^TP+Q=0 \qquad (3.6)$$

Now the procedure of designing a linear quadratic feedback controller can be summarized:

- Solve the reduced-matrix Riccati equation for the matrix P. If a positive-definite symmetrical matrix P exists, the system is stable.
- Substitute this matrix P into the Riccati equation, the matrix K is the optimal matrix.

There are other methods to analyse the linear system's performances as well as other linear controller and observer design methods, such as the Linear-Quadratic-Gaussian control (LQG), H_∞ control method and so on. As they are not the research emphases, they are not studied here.

The linear system control theory has its drawbacks. Because the physical world is nonlinear essentially, linear systems cannot fully accomplish the desired goals. These limits and drawbacks force the engineers to look for more capable, more efficient control strategies. The nonlinear control theories are more advantageous in control strategies compared to the linear control ones.

3.3 Nonlinear control theory

Nonlinear control theory is the theory used to analyse and design the nonlinear control systems. Nonlinear control system contains nonlinear components; they can be either in the nonlinear plant or in the nonlinear controller, even exist in both of them at the same time. From a mathematical point of view, the nonlinear systems can be defined relatively easier by excluding linear parts from the systems. If the system satisfies both additively and homogeneity properties, the system is considered as a linear system, otherwise it is a nonlinear system. Commonly, a nonlinear dynamic system can be expressed by a set of nonlinear differential equations in the form of:

$$\dot{x} = f(x,t) \tag{3.7}$$

In which $f \in R^{n\times 1}$ is a nonlinear vector function, and $x \in R^n$ is the state vector. The number of states n is called the order of the system.

The nonlinear methods for analysis and design of control system are developed in order to achieve the following goals:

Improvement of existing control systems: Linear control methods rely on the key assumption of small range operation for the linear model to be valid (Slotine and Li, 1991:2). In nature, most physical systems are inherently nonlinear, which means the linear model cannot completely describe the physical systems and its dynamics, because the nonlinearities of the system are not included in the model. Because of the above mentioned reason, the linear model is not sufficiently valid when the operation range is large. On the other hand, the nonlinear system can fully describe the system model and based on the more accurate model the controller can be designed to be more efficient.

The operation range of the nonlinear control system is also wider compared to the linear control system.

Analysis of hard nonlinearities: In control engineering systems, there is much nonlinearity whose discontinuous nature does not allow linear approximation. These so-called "hard nonlinearities" include Coulomb friction, saturation, dead-zones, backlash, and hysteresis (Slotine and Li, 1991:2). The linear control methods are not applicable for these systems. Because the nonlinearities frequently cause chaos in the control system, the nonlinear control theory must be developed to compensate for the inherent nonlinear performance of these systems.

Dealing with model uncertainties: Many model parameters in the control problems are not determinable due to the circumstances changing. An application of a linear controller based on incorrect values of the model parameters can cause significant performance degradation or even instability of the closed-loop system. Nonlinearities can be intentionally introduced into the controller part of the control system so that model uncertainties can be tolerated (Slotine and Li, 1991:2-3).

Design simplicity: Nonlinear controller designs are often deeply related with the essential physical phenomenon of the plants. This makes the nonlinear control designs possibly simpler and more efficient than their linear corresponded designs.

Above mentioned reasons give an answer to the question: why must nonlinear control theory is developed? Thus, as a subject, nonlinear control is an important field of control system analysis and design. In the analysis, a nonlinear closed-loop system is assumed to have been designed, and the behaviour characteristics are necessary to be determined (Slotine and Li, 1991:1). There are several well developed techniques to analyse nonlinear control system, including: Describing function method; Phase plane method; Lyapunov stability analysis method; Singular perturbation method; Small-gain theorem. For the nonlinear controller design, the nonlinear plant to be controlled and some specifications of the closed-loop system behaviour are given. The task is to construct a controller so that the closed-loop system meets desired characteristics (Slotine and Li, 1991:1). The relative nonlinear controller design techniques include: Trial-and-error method; Feedback linearization method; Lyapunov redesign method; Nonlinear damping method; Sliding mode control method; Gain scheduling method; Robust control method; Adaptive control method and so on.

From the application point of view, the nonlinear control design and implementation requires big amount of computation. This was the limitation in the past for the application and development of the nonlinear control techniques. With the popularization of the computers and their price going down, the nonlinear control design and application for large range operations has attracted particular attention. The considered problem, control of an inverted pendulum in a large range, is a typical application of the nonlinear control. The task in this project is to design the suitable nonlinear controllers for the reconfigurable plant. Two typical nonlinear strategies are chosen: the Lyapunov direct method and the feedback linearization method. In this project, both of them are used to design the nonlinear controllers by associating with the model reference control (MRC) method. In the following sections, the MRC method, the Lyapunov stability theory and the feedback linearization are studied.

3.3.1 Model reference control (MRC)

One useful method for specifying system performance for nonlinear and uncertain systems is by means of a model that will produce the desired output (Datta, Loannou, 1995:2370-2387). The objective of the MRC is to develop a control strategy that forces the plant dynamics to follow the dynamics of an ideal model. The ideal model is not the actual hardware. It can be only a mathematical model simulated on a computer. In the MRC system, the output of the model and that of the controlled system are compared and the error vector – e(t) is used to generate the control signals (Uçar, 2005:713). The MRC has been used to obtain acceptable performance in some very difficult control problems involving the systems containing nonlinearity and/or time varying parameters (Datta, Loannou, 1995:2370-2387).

In the complex domain, the task is to design a controller whose closed-loop transfer function matches the reference model. In the time domain, the closed-loop response is necessary to follow the reference model response. The use of a reference model to follow is worthwhile in that the model is a system that has such good performance that it is worth imitating. (Paz, 2002:6). The reference model is set as the example that the closed-loop will follow. It must contain certain characteristics to ensure it has the distinct advantages, such as:

- Internal stability
- Steady-State Tracking capability
- Transient performance according to the designer’s requirements

3.3.1.1 Internal stability

Internal stability implies that the closed-loop system is a stable transfer function and that the controller cancels neither unstable poles nor zeros (Paz, 2002:7), because the unstable poles and zeros cancelation cannot make sure that all poles are stable. Internal stability also implies that the controller used is a proper controller (i.e., the relative degree of the controller be greater than or equal to zero) (Paz, 2002:7). For a proper reference model, it must have a relative degree that is greater than equal to that of the plant. In general, the relative degree of the model should be set to equal to the plant and requires that the relative degree is zero. Generally speaking, it is preferable to choose a model with the minimal relative degree and minimal order. A minimal order model is a model with a constant numerator that has the appropriate relative degree.

3.3.1.2 Steady-state tracking

Normally, for any problem, there are an infinite number of models to choose from. However, a good model makes all the difference. It provides a basis for conduct among infinitely many choices. Generally in the complex domain, if the numerators of the closed-loop model transfer function is chosen carefully for a minimum order and minimum relative degree, and if the model is stable, then the model will asymptotically track the reference signal (Paz, 2002:9). But there is still infinite number of choices for a model.

3.3.1.3 Transient performance

It is difficult to define a good transient performance in order to find a good model. For different applications, the same transient performance can be perfect or disastrous disaster because the application determines what transient performance is good. But a good transient performance model shares some common properties. For example: It can be simpler model – the first and second order model. If the plant is of relative degree one or two, these may be the right ones. And it is an optimal model. These are models in which the very best model for a given criteria are provided. Since there are a lot of potential criteria, quite a few models are provided.

These characteristics ensure the reference model can be chosen from infinite choices and they also ensure that the reference model has as close as possible behaviours to the ideal model. The reference model provides the desired trajectory. The different control strategies can be applied to design the controllers, which can guarantee the closed-loop system has an identical response to the reference model. Here, the Lyapunov stability theory and the feedback linearization method are introduced on the basis of a reference model.

3.3.2 Lyapunov stability theory

3.3.2.1 Introduction of Lyapunov stability theory

Unstable system is typically useless and dangerous. For a control system, one of the most important properties is whether the system is stable. The stability of the control system should be carefully studied no matter whether the system is linear or nonlinear. Lyapunov theory is the most useful and common method for studying the stability of the nonlinear control system. The theory is introduced in the late 19th century by the Russian mathematician Alexander Mikhailovich Lyapunov. There are two methods for stability analysis, linearization method (indirect method) and direct (Lyapunov second) method. Today, Lyapunov's linearization method has come to represent the theoretical justification of linear control, while Lyapunov's direct method has become the most important tool for nonlinear system analysis and design. Together, the linearization method and the direct method constitute the so-called Lyapunov stability theory (Slotine and Li, 1991:41). In the following content, the linearization method is mentioned briefly as the relative knowledge. As a focus of the stability study and Lyapunov's direct method application for the nonlinear system controller design in this project, the Lyapunov direct method is studied carefully in detail.

3.3.2.2 Lyapunov's linearization method

The local stability of a nonlinear system is considered by the Lyapunov's linearization method. It is a formalization of the intuition that a nonlinear system should behave similarly to its linearized approximation for a small range motions (Slotine and Li, 1991:53). As the linearized approximation is studied, the linear control techniques are used. If the linear control is stable, it guarantees the stability of the original physical system locally.

For a particular system, its linearization can be easily found by ignoring all terms that have order higher than 1 in the corresponding Taylor series expansion of the nonlinear system derived for the equilibrium point x_e, $f(x_e)=0$. It can be expressed as follows:

Consider the autonomous system described in Equation (3.1). If $f(x)$ is continuously differentiable, the system can be written as:

$$\dot{x}(t)=\left(\frac{\partial f}{\partial x}\right)_{x_e=0}+f_{h.o.t.}(x) \tag{3.8}$$

where $f_{h.o.t}$ stands for higher-order-terms in x. Let $A=\left(\frac{\partial f}{\partial x}\right)_{x_e=0}$, then the system:

$$\dot{x}(t)=Ax(t) \tag{3.9}$$

is called linearization of the original nonlinear system at equilibrium point 0.

Investigation of the solution of the linearized system in Equation (3.9) shows the capabilities of the system to go and stay at the equilibrium point. The following results make the accurate relationship between the stability of the linear system and its original nonlinear system. They are known as the Lyapunov's linearization theorem.

Theorem 3.1: Lyapunov's linearization theorem:

- If the linear system is strictly stable (i.e., if all eigenvalues of A are strictly in the left-half complex plane), then the equilibrium point is asymptotically stable (for the actual nonlinear system)
- If the linearized system is unstable (i.e., if at least one of the eigenvalues of A is strictly in the right-half complex plane), then the equilibrium point is unstable (for the nonlinear system).
- If the linearized system is marginally stable (i.e., all eigenvalues of A are in the left-half complex plane, but at least one of them is on the $j\omega$ axis), then one cannot conclude anything from the linear approximation (the equilibrium point may be stable, asymptotically stable, or unstable for the nonlinear system). (Slotine and Li, 1991:55)

Lyapunov's linearization theorem is based on the approximation of the original system. Because of the linearization approximation, the stability of the original system can be studied in a very small range which is even difficult to be defined. These drawbacks promote the development of the Lyapunov's direct method.

3.3.2.3 Lyapunov's direct method

The Lyapunov's direct method has been well studied and applied for both linear and nonlinear systems. The basic philosophy of Lyapunov's direct method is the mathematical extension of a fundamental physical observation: If the total energy of a mechanical (or electrical) system is continuously dissipated, then the system, whether linear or nonlinear, must eventually settle down to an equilibrium point (Slotine and Li, 1991:55). Then, a system's stability can be concluded by evaluating the variation of a single scalar function. Because of the importance of these functions in the Lyapunov's direct method, they are denoted as Lyapunov functions. The control Lyapunov function

made a marvellous impact on stabilization theory. It has converted stability descriptions into tools for solving stabilization tasks (Cai et al, 2007: 598-602). Lyapunov functions allow the determination of stability for non-linear systems without the need to find exact solutions as by the linearization method.
(http://library.wolfram.com/infocenter/MathSource/547/nl_stab.txt, 18 April 2008).

Before Lyapunov function is studied, some definitions and theorems are needed to be clarified, as: positive definite function, Lyapunov function, Lyapunov theorem for local stability, Lyapunov theorem for global stability, local invariant set theorem and global invariant set theorem. All these definitions and theorems are used to study the stability of linear or nonlinear systems. Furthermore they are also used to design the controller for different systems. Each of the above mentioned definitions and theorems are given as background knowledge below.

Definition 3.1: (Global positive definite): A scalar continuous function V(x) is said to be locally positive definite if:

Let $x \in B_{R_0}$ for x=0, V(0)=0 and for $x \neq 0 \rightarrow V(x) > 0$, where B_{R_0} is a sphere of values of x.

If the above property holds over the whole state space, then V(x) is said to be globally positive definite.

Definition 3.2: (Lyapunov function): If, in a sphere B_{R_0}, the function V(x) is positive definite and has continuous partial derivatives, and if its time derivative along any state trajectory of the system (3.2) is negative semi-definite, i.e., $\dot{V}(x) \leq 0$.
then V(x) is said to be a Lyapunov function for the system.

Theorem 3.2: (Local stability): If, in the sphere B_{R_0}, there exists a scalar function V(x) with continuous first partial derivatives such that:

- V(x) is positive definite (locally in B_{R_0})
- $\dot{V}(x)$ is negative semi-definite (locally in $\dot{V}(x) \leq 0$)

then the equilibrium point 0 is stable. If, actually, the derivative $\dot{V}(x)$ is locally negative definite in B_{R_0}, $\dot{V}(x) < 0$, then the stability is the asymptotic one.

Theorem 3.3: (Global stability): Assume that there exists a scalar function V of the state x, with continuous first derivatives such that:

- $V(x)$is positive definite
- $\dot{V}(x)$ is negative definite
- $V(x)\rightarrow\infty$ as $\|x\|\rightarrow\infty$

then the equilibrium at the origin is globally asymptotically stable.

Theorem 3.4: (Local invariant set theorem): Consider an autonomous system of the form (3.7), with f continuous, and let V(x) be a scalar function with continuous first partial derivatives. Assume that:

- For some l>0, the region Ω_l defined by V(x)<l is bounded
- $\dot{V}(x) \le 0$ for all x in Ω_l

Let R be the set of all points within Ω_l where $\dot{V}(x)$=0, and M be the largest invariant set in R. Then, every solution x(t) originating in Ω_l tends to M as t→∞.

Theorem 3.5: (Global invariant set theorem): Consider an autonomous system of the form (3.1), with f continuous and let V(x) be a scalar function with continuous first partial derivatives. Assume that

- V(x) $\rightarrow\infty$ as $\|x\|\rightarrow\infty$
- $\dot{V}(x) \le 0$ over the whole state space

Let R be the set of all points within Ω_l where $\dot{V}(x)$=0, and M be the largest invariant set in R. Then all solutions globally asymptotically converge to M as t→∞.

Applying the above theorems, it is not difficult to study the system's stability or to design a particular controller for linear or nonlinear system based on the Lyapunov's direct method. Even appears that a common method of studying the systems' stability can be found. Unfortunately, the other problem which is how to find the Lyapunov functions for a specific nonlinear problem arises. In fact, there is no general approach to construct Lyapunov functions for nonlinear systems. This is a fundamental drawback of the Lyapunov's direct method. Therefore, the construction of Lyapunov function becomes the bottleneck of the design technique developed by using Lyapunov function. Faced with specific systems, one has to use experience, intuition, and physical insights to find and construct a suitable Lyapunov function (Slotine and Li, 1991:55). Two useful methods, one is called Krasovskii's method, the other is called variable gradient method can provide systematic ways to determine a suitable Lyapunov function, and are introduced briefly in the following sections. (http://netlib.org/linalg/html_templates/node20.html, February 20th, 2007)

3.3.2.3.1 Lyapunov analysis of LTI systems

For a stable LTI system: $\dot{x}(t) = Ax(t)$, its Lyapunov function satisfies:

$$V(x)>0,\ V(0)=0 \tag{3.10}$$

$$\dot{V}(x) = \frac{dV}{dt} = \frac{\partial V}{\partial x}\frac{dx}{dt} \leq 0 \tag{3.11}$$

A LTI system is stable, if such a scalar function V(x) associated with the system and satisfying the both conditions exists. The system is asymptotically stable if $\dot{V}$ is negative definite. A quadratic Lyapunov function can be chosen as:

$$V(x) = x^T(t)Px(t) \tag{3.12}$$

where $P \in R^{n\times n}$ is a given symmetric positive definite matrix ($P=P^T>0$). Then:

$$\dot{V}(x) = \dot{x}^T(t)Px(t) + x^T(t)P\dot{x}(t) = x^T(t)\left(A^TP + PA\right)x(t) = -x^T(t)Qx(t) \tag{3.13}$$

where $Q \in R^{n\times n}$ is a positive definite matrix ($Q=Q^T>0$). Equation $A^TP+PA=-Q$ is called the Lyapunov algebraic equation. Now the Lyapunov stability theory for the LTI System can be formulated as:

Theorem 3.6: A necessary and sufficient condition for a LTI system $\dot{x}(t) = Ax(t)$ to be strictly stable is that, for any symmetric positive definite matrix Q, a unique matrix P, the solution of the Lyapunov algebraic equation is symmetric and positive definite.

3.3.2.3.2 Krasovskii's method

Krasovskii's method provides a Lyapunov function candidate for autonomous nonlinear system in the form of $V(x)=f^Tf$. The basic idea of the method is to find out whether this particular choice indeed leads to a Lyapunov function (Slotine and Li, 1991:84). The method is well concluded in the Generalized Krasovskii theorem, stated as follows:

Theorem 3.7: (Generalized Krasovskii theorem) Consider the autonomous system defined by Equation (3.1), with the equilibrium point of interest being the origin, and let A(x) denote the Jacobian matrix of the system. Then, a sufficient condition for the origin to be asymptotically stable is that there exist two symmetric positive definite matrices P and Q, such that, ∀A≠0 the matrix

$$F(x) = A^TP + PA + Q \tag{3.14}$$

is negative semi-definite in some neighbourhood Ω of the origin. The function $V(x)=f^TPf$ is then a Lyapunov function for the system. If the region Ω is the whole state space, and if in addition, $V(x)\to\infty$ as $\|x\|\to\infty$, then the system is globally asymptotically stable. (Slotine and Li, 1991:85)

3.3.2.3.3 The variable gradient method

The variable gradient method is another useful method for finding suitable Lyapunov functions. The theory of the method is to assume a certain form for the gradient of an unknown Lyapunov function first, then find the Lyapunov function itself by integrating the assumed gradient (Slotine and Li, 1991:86). In this method, a special form of gradient ∇V is assumed instead of assuming a special form for the Lyapunov function V. The detailed mathematical expression and explanation can be found in relative nonlinear control text books.

Lyapunov stability theory has been studied and applied extensively. It is one of the fundamental tools to study the system stability. Additionally, based on the Lyapunov direct method, nonlinear control law can also be derived. In this project, a Lyapunov stability theory based MRC by using the Lyapunov direct method design is applied to control the inverted pendulum and the overhead crane. This part of work is presented in chapter 6.

3.3.3 Feedback linearization

3.3.3.1 Introduction of the feedback linearization

Feedback linearization as an approach for the nonlinear control design has drawn a lot of attention in past decade. It has been studied extensively. Many related papers have been published and many ramifications have resulted. “The central idea of the feedback linearization approach is to algebraically transform a nonlinear system dynamics into a (fully or partly) linear one, so that linear control techniques can be applied.” (Slotine and Li, 1991:207) Although this approach is linearizing the nonlinear system, it is completely different from the conventional linearization. The feedback linearization is done through changing of variables and a suitable control input is derived that simply makes a linear approximation of the system dynamics.

Feedback linearization has been widely used in different fields. The successful applications include the control of helicopters, high performance aircraft, industrial robots and biomedical devices. The advantages of feedback linearization are that the design of control can be used generally, in the sense that the same principle can be used on all systems of the right type. Moreover, extensions have been developed to take into account possible model inaccuracies.

As the control tasks can be divided into two categories: stabilization (regulation) and tracking (servo) which is more difficult that the other generally, the study of feedback linearization approach is also considered for these two kinds of problems. The typical examples of stabilization tasks are temperature control of refrigerator, altitude control of aircraft and position control of robot arms. In the tracking task, the controller aims to drive the system output to follow the reference time-varying trajectories. Typical examples include the ship cruise liner following the specified path and a robot arm moving in a desired manner. In the following sections, feedback linearization approach is studied in detail.

Here, the feedback linearization theory is studied generally. The overall idea of the theory is presented. In Chapter 7, the feedback linearization theory is studied in a deeper manner and applied to the pendulum system.

3.3.3.2 Important terminologies

Before the input-output linearization is introduced, there is some terminology that needs to be clarified and understood. This terminology includes: the relative degrees, the internal dynamics and the zero dynamics.

Relative degree: The number of differentiation needed to be taken on the output y in order to generate an explicit relationship between the output y and the input u. For example, if the output y is necessary to be differentiated ρ times, in order a clear relationship between output y and input u to be obtained, the system is said to have relative degree ρ. This terminology is consistent with the notation of relative degree in linear system which is the excess of poles over zeros. It can also be shown that for any controllable system of order n, it will take at most n differentiations of any output for the control input to appear. This can be understood naturally: if it took more than n differentiations, the order of the system would be higher than n; if the control input never appeared, the system would not be controllable (Slotine and Li, 1991:218).

Internal dynamics: A part of the system dynamics that has been rendered "unobservable" in the input-output linearization. This part of the dynamics is called the internal dynamics, because it cannot be seen from the external input-output relationship. If the internal dynamics are stable – the states remain bounded during tracking or stabilizing, the problem is solved. If not, the designed controller is worthless, because the instability of the internal dynamics would imply undesired phenomena such as the burning-up of fuses or the violent vibration of mechanical members.

Zero-dynamics: A relatively simpler way of studying the system internal dynamics is by studying the zero-dynamics. The zero-dynamics are defined as the internal dynamics of the system when the system output is kept at zero by the input. For linear systems the stability of the internal dynamics can be determined by the locations of zeros. If all zeros are in the left-half complex plane the global asymptotic stability of the zero-dynamics is guaranteed, and the stability of the zero-dynamics implies the global stability of the internal dynamics. In nonlinear systems, the local asymptotic stability of zero-dynamics is enough to guarantee the local asymptotic stability of the internal dynamics for both stabilizing and tracking problems. But, no results can be drawn on the global stability or even large range stability for internal dynamics of nonlinear systems, for instance, only local stability is guaranteed for the internal dynamics even if the zero-dynamics is globally exponentially stable (Slotine and Li, 1991:227).

Two useful remarks can be made about the zero-dynamics of nonlinear systems. First, the zero-dynamics is an intrinsic feature of a nonlinear system, which does not depend on the choice of control law or the desired trajectories. Secondly, examining the stability of zero-dynamics is much easier than examining the stability of internal dynamics, because the zero-dynamics only involve the internal states. (Slotine and Li, 1991:228)

Diffeomorphisms: A function μ: $R^n \rightarrow R^n$, defined in a region Ω, is called a diffeomorphism if it is smooth, and if its inverse μ^{-1} exists and is smooth. If the region Ω is the whole space R^n, then $\mu(x)$ is called global diffeomorphism. Global diffeomorphisms are rare, and therefore one often looks for local diffeomorphisms.

After clarify the terminologies that are used for study of the feedback linearization theory. The feedback linearization theory is studied generally in the following sections.

3.3.3.3 Nonlinear system in companion form

Feedback linearization can be directly applied if a nonlinear system is in the companion form which is also known as the controllability canonical form. The companion form of a nonlinear system can be expressed as:

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ \cdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_2 \\ \cdots \\ x_n \\ f(x)+b(x)u \end{bmatrix} \equiv x^n = f(x)+b(x)u \qquad (3.15)$$

where u is the scalar control input, x is the scalar output of interest, $\mathbf{x} = [x, \dot{x}, \ldots, x^{(n-1)}]^T$ is the state vector, and f(x) and b(x) are nonlinear functions of the states. This form is unique in the fact that, although derivatives of x appear in this equation, no derivative of the input u is present.

If a system can be expressed in the companion form, by using the control input (assume b(x) to be non-zero) can be expressed as:

$$u = \frac{1}{b(x)}[v - f] \qquad (3.16)$$

Then the system nonlinearities are cancelled and the system is expressed in the linear form as:

$$x^{(n)} = v \qquad (3.17)$$

Thus, the control law for the stabilization task is:

$$v = -k_0 x - k_1 \dot{x} - \cdots - k_{n-1} x^{(n-1)} \qquad (3.18)$$

It is necessary carefully to choose the k_i, (i=0, 1,…,n-1) so that the polynomial $s^n + k_{n-1}s^{n-1} + \cdots + k_0$ has all it roots strictly in the left-half complex plane which means the system turns to exponentially stable.

If the task is tracking a desired output $y_d(t)$, the control law is:

$$v = x_d^{(n)} - k_0 e - k_2 \dot{e} - \cdots - k_{n-1} e^{(n-1)} \qquad (3.19)$$

where $e(t)=x(t)-x_d(t)$ is the tracking error. This control law leads the system to exponentially convergent tracking.

It can be seen that the control derivation is straightforward if the nonlinear system is in the companion form. But if the nonlinear system is not in the companion form, it either must be transformed into the companion form or partially be linearized instead of be fully linearized. The following subsections include the input-state linearization and input-output linearization which deal with these kinds of problems.

3.3.3.4 Input-state linearization of a SISO system

3.3.3.4.1 Input-state linearization definition

The input-state linearization definition is given as:

Definition 3.3: A single-input nonlinear system in the form of:

$$\dot{x} = f(x) + g(x)u \tag{3.20}$$

with f(x) and g(x) being smooth vector fields on R^n, is said to be input-state linearizable if there exists a region Ω in R^n, a diffeomorphism ϕ: $\Omega \rightarrow R^n$, and a nonlinear feedback control law

$$u = \alpha(x) + \beta(x)v \tag{3.21}$$

such that the new state variable $z=\phi(x)$ and the new input v satisfy a linear time-invariant relation

$$\dot{z} = Az + bv \tag{3.22}$$

where $A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}$; $b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$; z is the linearizing state; u is the linearizing control law.

3.3.3.4.2 Condition for input-state linearization

The following theorem defines the condition for a nonlinear state equation in the form of $\dot{x} = f(x) + g(x)u$ to be input-state linearized. This is one of the most fundamental results of feedback linearization theory.

Theorem 3.8: The nonlinear system in the form of $\dot{x} = f(x) + g(x)u$, with f(x) and g(x) being smooth vector fields, is input-state linearizable if, and only if, there exists a region Ω such that the following conditions hold:

- The vector fields $\{g, ad_f g, ..., ad_f^{n-1} g\}$ are linearly independent in Ω
- The set $\{g, ad_f g, ..., ad_f^{n-2} g\}$ is involutive in Ω

where $ad_f g$ is Lie derivative.

3.3.3.4.3 Procedure of forming the input-state linearization

The procedure of input-state linearization can be performed through the following steps:

- Construct the vector field $g, ad_f g, \ldots, ad_f^{n-1} g$ for the given system
- Check whether the controllability and involutivity conditions are satisfied
- If both are satisfied, find the first state z_1 (the output function leading to input-output linearization of relative degree ρ) from the equation:

$$L_g z_1 = L_{ad_f g} z_1 = \ldots = L_{ad_f^{n-2} g} z_1 = 0 \text{, i.e.,} \tag{3.23}$$

$$\nabla z_1 ad_f^i g = 0 \text{ , i=0, …, n-2} \tag{3.24}$$

$$\nabla z_1 ad_f^{n-1} g \neq 0 \tag{3.25}$$

- Compute the state transformation $z = \begin{bmatrix} z_1 & L_f z_1 & \cdots & L_f^{n-1} z_1 \end{bmatrix}^T$ and the input transformation $u = \alpha(x) + \beta(x) v$, with

$$\alpha = -\frac{L_f^n z_1}{L_g L_f^{n-1} z_1} \tag{3.26}$$

$$\beta = -\frac{1}{L_g L_f^{n-1} z_1} \tag{3.27}$$

3.3.3.4.4 Controller design based on the input-state linearization

After the system is successfully transferred into the linear form, the controller can be constructed by using the controllers described in section 3.3.3.3 for both stabilization and tracking problems.

3.3.3.5 Input-output linearization of SISO system

A single-input-single-output nonlinear system expressed as:

$$\dot{x} = f(x) + g(x) u \tag{3.28}$$

$$y = h(x) \tag{3.29}$$

where y is the output of the system is considered.

3.3.3.5.1 Generating a linear input-output relation

From Equations (3.28) and (3.29), it can be seen that the output y is indirectly related to u through the state variable x and the nonlinear state equations. The relationship between the output y and the input u can be created by differentiating the output function y repeatedly until the input u appears. Then design u to cancel the nonlinearity.

3.3.3.5.2 Control design procedures

Procedures of control design based on input-output linearization can be summarized in three steps:

- Differentiate the output y until the input u appears
- Choose u to cancel the nonlinearities and guarantee tracking convergence
- Study the stability of the internal dynamics

If the system order equals to the relative degree associated with the input-output linearization, the nonlinear system is fully linearized and the above procedure indeed leads to a satisfactory controller based on an accurate model. If the system order is higher than the relative degree, the nonlinear system is only partly linearized and whether the controller can indeed be applied depends on the stability of the internal dynamics. The internal dynamics stability can be studied through the zero-dynamics study (Slotine and Li, 1991:228).

In the Chapter 7, the feedback linearization theory is studied in a deeper manner for Single-Input-Single-Output (SISO) systems, Multi-Inputs-Multi-Outputs (MIMO) systems and Single-Input-Multi-Outputs (SIMO) systems and the feedback linearization is applied to the pendulum system.

3.6 Conclusion

Control theory has been under development for thousands of years. It becomes more and more useful and powerful than ever before because of its positive impacts. As a cross subject in the modern world, it plays a very important role in different aspects, such as: industry process, national defence, biomedical engineering, social science, economy and even daily life. In this chapter, linear control theory and nonlinear control theory are studied, especially nonlinear control theory. The control theory provides the theoretical basis which is a powerful tool for the engineers and researchers to solve the control problems in different fields. In the project, the MRC associated with the Lyapunov stability theory and the MRC associated with the feedback linearization approach are applied to design control for the inverted pendulum and the overhead crane. Before the control theory is applied, the mathematical models of the plant are developed. In next chapter, the mathematical nonlinear and linear models are developed for the inverted pendulum and the overhead crane.

CHAPTER FOUR
MATHEMATICAL MODEL OF THE PENDULUM SYSTEM

4.1 Introduction

Mathematical model represents certain system or phenomenon by using symbolic mathematical expressions, involving different elements, such as: letters, numbers, mathematical notations, equations and so on. A mathematical model of a dynamic system is defined as a set of equations that represents the dynamics of the systems accurately (Ogata, 2002:53). Normally, they are a set of differential equations. It is one of the most useful tools that humans can utilize to describe the physical world precisely and unambiguously. In this way of representation, the quantified values or letters are used instead of the general describing words. This enables the internal relationships and the system characteristics to be demonstrated. One of the remarkable benefits of using the mathematical model, is that well known mathematical manipulation can be utilized. Additionally, this enables the modern computation technology to be applied through varying computation or simulation software. Mathematical models are broadly used in physical science and engineering fields. Furthermore, they are also used in the social science field – especially in Economics. The process of finding the mathematical model of a system or phenomenon is called “mathematical modelling”. The mathematical modelling processes enable a thorough understanding of the modelled system. It is a process whereby one can also apply the mathematical theories into an application-oriented mathematical implementation.

A given system can be represented by different mathematical models, because when a model is built, different modellers may emphasise different factors in the model, however the essence of these mathematical models is the same or very similar. “An exact plant model should produce the same output response as the plant, provided the input to the model and initial conditions are exactly the same as those of the plant” (Ioannou, Sun, 1995:2). The mathematical model can be represented in transfer function form or state space form. The SISO-LTI systems are represented in a differential equation form or a transfer function form. In these forms, the frequency response and system transient response are easily analysed. On the other hand, state space representations with time domain analyses are used for MIMO system. For a particular control problem, once the mathematical model is obtained, it is analysed and studied in detail in order to find and understand its characteristics and properties.

How good a model is can be measured by the degree of the truth the model presents. Under different circumstances, for the same system, the mathematical model can be highly precise or generally accurate, simple or complicated, linear or nonlinear. A highly precise model is normally more complicated, as it represents the system more completely, although the related expression may be fuzzy. A generally accurate model represents the system adequately with simplified expressions, but probably misses some tiny detail factors. The most famous scientist Einstein said that: "A good theory" (or model) "should be as simple as possible, but not simpler."

The relationship between the linear and nonlinear models is discussed in the previous Chapter 3. This chapter is organized in the following way that: the reconfigurable plant is introduced in the section 4.2. The complete mathematical models of the reconfigurable plant including the inverted pendulum and the overhead crane are developed according to Newton's laws in the section 4.3. Incorporation of the actuator model into the pendulum model is developed in the section 4.4. In the section 4.5, the developed mathematical models are constructed and simulated in the Matlab/Simulink environment based on their parameters. The mathematical models which are built in Matlab/Simulink environment are shown and the simulation results are presented in the section 4.6. The simulation results are discussed in the section 4.7.

4.2 Description of the reconfigurable plant

The reconfigurable plant that is studied in the project is a pendulum system. The pendulum system consists of two modules: the carriage module and the control module.

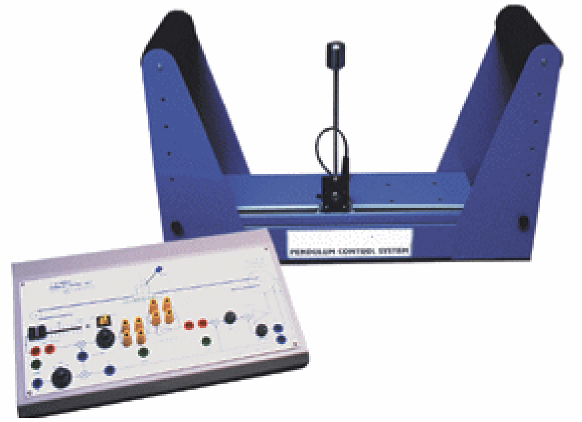


Figure 4.1: **The pendulum system**

(Adopted from http://www.bytronic.net/html/pcs.htmlT, 5th May 2007)

They are linked through a connecting cable. The carriage module consists of a cart that carries a pivoted rod and a pendulum. The cart is mounted on a 50cm long track and driven by a dc servo motor through a toothed belt. The pendulum system is shown in Figure 4.1 and Figure 4.2.

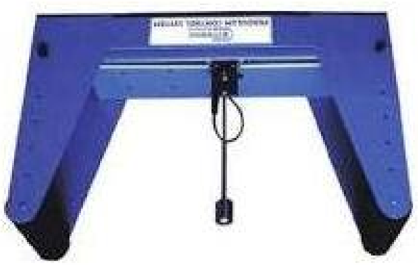


Figure 4.2: The carriage model is configured into overhead crane mode

(Adapted from http://www.bytronic.net/html/pcs.html, 5th May 2007)

The pendulum system can be configured into two modes. When the carriage module stands upright, it behaves as an inverted pendulum, shown in Figure 4.1. When the carriage is simply turned upside down, the rod and the pendulum represent the lifting block of an overhead crane shown in Figure 4.2 (Bytronic international LTD, 2001: 1.1). A mimic diagram is shown on the control module. This mimic diagram is given in Appendix C. 1. The complete circuit diagram of the pendulum system is given in Appendix C. 2. It can be used as an analog linear controller for the pendulum system. But, for this project, the control module is only working as a connector which takes in and sends out signals through its connection slots. The control signal is produced by the controllers which are developed in this project. The developed controllers are deployed in different hardware and software platforms. The detailed technique is discussed in the later chapters.

4.3 Mathematical models derivation for the inverted pendulum system and the overhead crane system

The inverted pendulum is a nonlinear inherently unstable system. The overhead crane is a stable nonlinear system with oscillation. Because of the special characteristics of the inverted pendulum and overhead crane behaviour, controlling the both systems is a typical control problem. Both of them are often used to demonstrate concepts of the control theory such as the stabilization of unstable and oscillating systems. They are also the ideal test bench for control theory verification and implementation. Based on the physical structure of the pendulum system, the inverted pendulum model has been developed first.

4.3.1 Inverted pendulum system

4.3.1.1 Introduction of the inverted pendulum

The specified inverted pendulum consists of a pendulum and a thin rod, the pendulum is attached on the rod, and the pendulum can move up and down along the rod. The pendulum and the rod are mounted on a moving cart through a pivot. This mechanism also ensures that the pendulum system is a two-dimensional mechanism in which the pendulum and the rod move only in one plane. The inverted pendulum is inherently unstable. In order to keep the pendulum standing upright in the inverted position, the cart must continuously move to correct the position of the pendulum. Control of an inverted pendulum is a typical problem in dynamic and control theory; it is also considered as a typical physics problem. It is broadly used as a test bench for control strategies, such as: PID control method (http://www.engin.umich.edu/group/ctm/examples/pend/invPID.html), fuzzy logic control method (Yi et al, 2002: 105) and pole placement control method (http://www.engin.umich.edu/group/ctm/examples/pend/digINVSS.html) and so on. The pendulum system shares the same type of behaviour as the rocket and the missile guidance (Ali, et al., 2000: 3-4), where thrust is actuated at the bottom of a tall vehicle. The other application related with the inverted pendulum is the moving of cargos in the port by cranes. When the goods are transporting, the cranes move carefully ensuring there are no swings or sways. The goods are always kept at the desired position even under certain challenging conditions, for example, when the cranes stop moving or increase their speed.

4.3.1.2 Inverted pendulum mathematical model derivation

The inverted pendulum mathematical model can be derived based on the physical rules and mathematical derivations, such as: Newton’s motion laws and trigonometric functions, for example. The other way to find the inverted pendulum mathematical model is by using the Lagrange’s equations (Stimac, 1999: 8). In this project, the first method is used. In this section, the mathematical model of the inverted pendulum is derived, linearized and presented into state space form. As the inverted pendulum is a MIMO system, the state space mathematical model is created. There exist many mathematical models for the inverted pendulum. It is not difficult for one to find one, but most time these mathematical models are simplified, linearized and most of them do not specify the moment of inertia for the pendulum and the rod. In this section, a full scale mathematical model for the inverted pendulum is derived with a detailed explanation for each step.

The motion of the inverted pendulum system consists of translational movement and rotational movement. In physics, translation is the movement that changes the position of an object in a straight direction. A relative simple mathematical expression can represent the translation as:

$$(\hat{x},\hat{y},\hat{z}) \rightarrow (x+\Delta x, y+\Delta y, z+\Delta z) = (x', y', z') \tag{4.1}$$

where $(\hat{x},\ \hat{y},\ \hat{z})$ are the object original position coordinates, $(\Delta x,\ \Delta y,\ \Delta z)$ are constant vectors which represent the translational movements, (x' ,y' ,z') represent the object final position.

A rotational movement is a movement of an object in a circular motion. A two-dimensional object rotates around a center (or point) of rotation. The point can be within the object, then the object is said to rotate upon itself, or spin. Alternatively, an object's circular motion about an external point is called an orbit or more properly an orbital revolution. For this project, the second condition is applied, as the pendulum moves about the pivot which is at the end of the rod. After the motion of the plant is clarified, the model of the inverted pendulum can be derived according to its movement characteristics based on the physical laws.

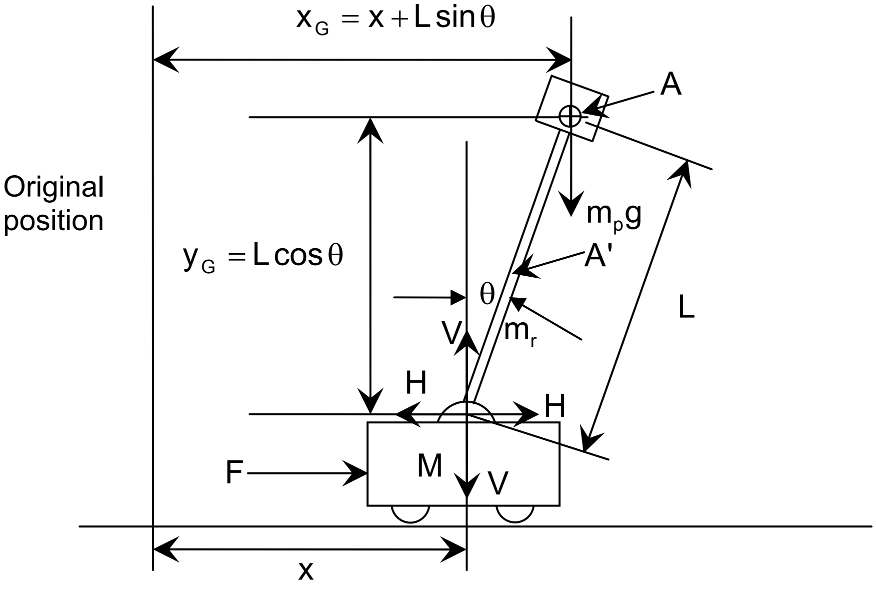


Figure 4.3: **The inverted pendulum system simplified diagram**

Refer to the inverted pendulum the system diagrammatic drawing shown in Figure 4.3. Assume the center of gravity of the pendulum and the rod is at point A; M [kg] is the mass of the cart; m [kg] is the mass of the pendulum and of the rod; L [m] is the length from the pivot to the center of gravity of the pendulum and the rod; θ [rad] is the angle between the rod and the vertical direction. F [N] is the force applied to the cart. x [m] is the displacement of the cart from the original position; H [N] and V [N] indicate the horizontal and vertical reaction forces the rod and cart. The vertical line on the left hand side of the diagram indicates the original position of the cart. This line is also considered as the reference position for the cart. Because the pendulum and the rod have similar motion characteristics, the analysis about the pendulum and the rod are taken as a whole.

Define the x, y coordinates of the center of gravity of the pendulum and the rod at point A as x_G and y_G, then:

$$x_G = x + L\sin\theta \qquad (4.2)$$

$$y_G = L\cos\theta \qquad (4.3)$$

where x_G and y_G represent the displacement of the pendulum and the rod away from the origin referring to their centre of gravity in x and y direction respectively.

The cart, the pendulum and the rod movements can fall into two categories:

- Translational movements for the cart, the pendulum and the rod refer to the original position
- Rotational movements for the pendulum and the rod according to its pivot

4.3.1.2.1 Inverted pendulum mathematical model derivation on the basis of the translational movement

For the inverted pendulum system, the cart, the pendulum and the rod move together along the track horizontally. This movement is translational movement caused by the force horizontally applied on the cart. According to the Newton’s first law, also known as the law of inertia - A physical body will remain at rest, or continue to move at a constant velocity, unless an external net force acts upon it. According to the Newton’s second law – The net force on a body is equal to its mass multiplied by its acceleration; it can be expressed by the equation:

$$F = m \cdot a \qquad (4.4)$$

where F [N] is the net force applied to the body; m [kg] the mass of the body and a [m/s^2] is the acceleration when the force is applied to the body.

Additionally according to the physics law the second derivative of the displacement (respect to time) is the acceleration, which is:

$$\frac{d^2x}{dt^2} = \ddot{x} = a \tag{4.5}$$

where x [m] is the displacement.

Then the mathematical model of the inverted pendulum describing its translational movement can be derived.

By applying Newton's second law, based on the force applied to the cart and the displacement of the pendulum and the rod, the first equation of the inverted pendulum mathematical model is derived. The pendulum and the rod are considered as a whole which shares one center of gravity, so they are analyzed together. H is the horizontal force applied to the cart.

4.3.1.2.1.1 Horizontal force applied to the pendulum and rod

Horizontal force H is expressed. It is the force applied to the pendulum and the rod in the horizontal direction. $x+L\sin\theta$ is the displacement of the pendulum and the rod away from the original position referring to their center of gravity. Then, H can be expressed as follows:

$$H = m \cdot \frac{d^2}{dt^2}\left(x + L\sin\theta\right) \tag{4.6}$$

where $m=m_p+m_r$, m_p is the mass of the pendulum; m_r is the mass of the rod.

Simplify the Equation (4.6):

$$H = m\frac{d}{dt}\left(\dot{x} + \dot{\theta}L\cos\theta\right) \tag{4.7}$$

$$H = m\left[\ddot{x} + \ddot{\theta}L\cos\theta + \dot{\theta}L\dot{\theta}(-\sin\theta)\right] \tag{4.8}$$

$$H = m\ddot{x} + m\ddot{\theta}L\cos\theta - m\left(\dot{\theta}\right)^2 L\sin\theta \tag{4.9}$$

4.3.1.2.1.2 Force analysis on the cart

According to the Newton's second law, the net force applied on the object is equal to its mass multiplied by its acceleration. Here the "body" is referring to the cart which has mass of M. The displacement of the cart is x, so its acceleration is the second derivative of x. Consider the forces applied to the cart shown in Figure 4.4, there are three forces. F is the force applied on the cart to keep the pendulum and the rod in balance. H is the force from the motion of the pendulum and the rod, which has opposite direction of F.

The third force is a small friction force which is caused by the friction between the cart wheel and the surface. It can be expressed as: $f_c=b \cdot v$, where b is the friction coefficient between the cart wheel and the track; v is the velocity of the cart. It is well known that the first derivative of the displacement is the velocity, then $f_c = b \cdot \dot{x}$. This force has opposite direction of F as well.

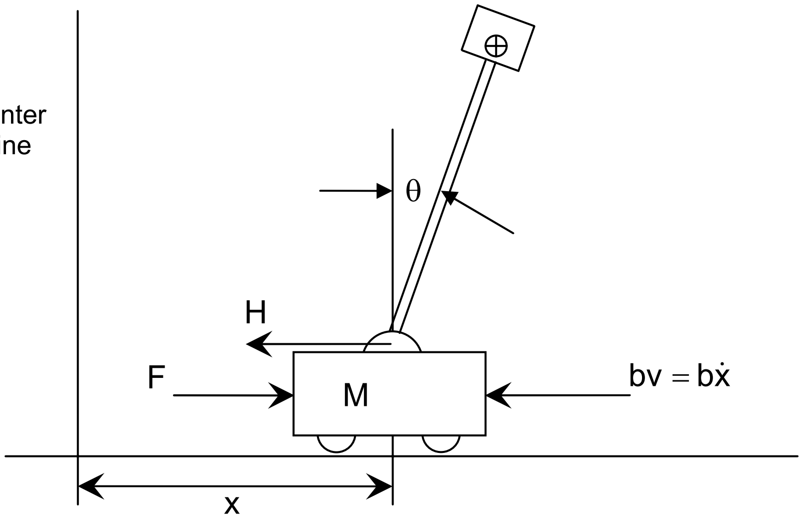


Figure 4.4: **Force analysis on the cart**

Then, according to the Newton's second law, the equation can be expressed as:

$$Ma = F - bv - H \tag{4.10}$$

Rearrange the equation and substitute H into the equation, the first equation to describe the dynamic of the inverted pendulum is obtained as follows:

$$F = Ma + bv + H \tag{4.11}$$

$$F = Ma + bv + m\ddot{x} + m\ddot{\theta}L\cos\theta - m\left(\dot{\theta}\right)^2 L\sin\theta \tag{4.12}$$

$$F = (M+m)\ddot{x} + b\dot{x} + mL\ddot{\theta}\cos\theta - mL\left(\dot{\theta}\right)^2 \sin\theta \tag{4.13}$$

4.3.1.2.2 Inverted pendulum mathematical model derivation on the basis of the rotational movement

4.3.1.2.2.1 Introduction of the torque

Based on the torque balance on the pendulum and the rod, the second inverted pendulum mathematical model equation is derived. In physics, torque τ is defined as: the moment of a force; the measure of a force's tendency to produce torsion and rotation about an axis, equal to the vector product of the radius vector r[m] from the axis of rotation to the point of application of the force F[N]. It can be stated in the equation: $\tau = r \times F$.

If the torque is presented as a function of time, it is the time-derivative of angular momentum:

$$\tau = \frac{dW}{dt} \qquad (4.14)$$

where W is the angular momentum.

Angular momentum on a rigid body can be defined by its moment of inertia J and its angular velocity ω as:

$$W = J \cdot \omega \qquad (4.15)$$

It is known that the first derivative of the angular displacement is the angular velocity and the second derivative of the angular displacement is the angular acceleration. If J is constant, then:

$$\tau = J\frac{d\omega}{dt} = J\frac{d^2\theta}{dt^2} = J\alpha \qquad (4.16)$$

where α is the angular acceleration; θ is the angular displacement of the rigid body.

4.3.1.2.2.2 Torque analysis on the rod and the pendulum

Refer to the torque balancing diagram shown in Figure 4.5. Based on the torque balance on the pendulum and the rod, the second mathematical model equation for the inverted pendulum is derived. The horizontal and vertical forces are split into two directions: the direction perpendicular to the rod and the direction parallel to the rod. These forces perpendicular to the rod are used to form the torque balance equation. The torque applied to the rod causes the rod and the attached pendulum to rotate around the pivot.

Referring to the Figure 4.4, the angle between the vertical direction and the rod is θ. Analyse the vertical force V, the component of force which is perpendicular to the rod can be expressed as:

$$V_P = V \cdot \sin\theta + mg \cdot \sin\theta \qquad (4.17)$$

and the horizontal force H's component that is perpendicular to the rod can be expressed as:

$$H_p = H \cdot \cos\theta \qquad (4.18)$$

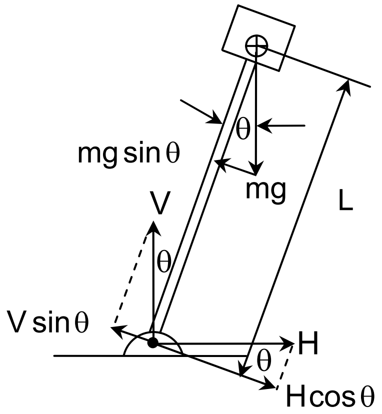


Figure 4.5: **Pendulum and rod torque balancing diagram**

Both forces V_p and H_p are acting at the pivot, so the length of the lever is L. Then, the torque balancing equation for the pendulum and the rod can be determined. The equation is stated as follows:

$$-H \cdot \cos\theta \cdot L + (V + mg) \cdot \sin\theta \cdot L = J\ddot{\theta} + b_t\dot{\theta} \tag{4.19}$$

where J is the moment of inertia for the pendulum and the rod; $\dot{\theta}$ is the angular velocity; $\ddot{\theta}$ is the angular acceleration; b_t is the viscous friction coefficient at the pivot.

The horizontal direction force H is derived in previous section in Equation (4.8). Now, in order to complete the Equation (4.19), the moment of inertia J and the vertical force V need to be determined.

4.3.1.2.2.3 The moment of inertia

According to the definition: the scalar moment of inertia of a point mass rotating about a known axis is defined by:

$$J = m \cdot r^2 \tag{4.20}$$

where m [kg] is the mass and r [m] is the distance of the point mass to the axis of rotation. The pendulum and the rod can be considered together as a point at their center of gravity with the mass of m. The pendulum and the rod are rotating about the pivot. The distance between the pivot and the center of gravity of the pendulum and the rod is L. So the above mentioned definition is applicable. Referring to Equation (4.19) the moment of inertia of the pendulum and the rod J [kg m^2] then can be similarly given as:

$$J = mL^2 \tag{4.21}$$

A characteristic of the considered pendulum system is that the pendulum can be moved along the rod, then the length L will be changed. If the centre of gravity is in the middle of the rod the formula is:

$$J = \frac{1}{3} m L^2 \tag{4.22}$$

where m [kg] is the total mass of the object. L [m] is the distance between the pivot and the centre of the gravity.

4.3.1.2.2.4 Vertical force applied to the pendulum and the rod

The derivation is based on the Newton’s second law. Consider the vertical forces applied to the pendulum and the rod. V is the force applied on the pendulum in the vertical direction. It is caused by the cart movement and the gravitational force. mg is the gravitational force applied on the pendulum. Lcosθ is the displacement of the pendulum from the pivot to its center of gravity in the vertical direction. By using Newton’s second law, V can be expressed as:

$$V = m \frac{d^2}{dt^2} L \cos\theta \tag{4.23}$$

Simplifying the above equation:

$$V = m \frac{d}{dt}\left[\dot{\theta} \cdot L \cdot (-\sin\theta)\right] \tag{4.24}$$

$$V = m\left[\ddot{\theta} \cdot L \cdot (-\sin\theta) - \dot{\theta} \cdot L \cdot \dot{\theta} \cdot (\cos\theta)\right] \tag{4.25}$$

The vertical force applied on the pendulum is stated as:

$$V = -m \cdot \ddot{\theta} \cdot L \cdot \sin\theta - m \cdot (\dot{\theta})^2 \cdot L \cdot \cos\theta \tag{4.26}$$

4.3.1.2.2.5 Formation of the torque balancing equation

Both the horizontal and the vertical forces H and V are derived. Substitute both force H which is expressed in Equation (4.9) and force V which is expressed in Equation (4.26) into Equation (4.19), then:

$$-H \cdot \cos\theta \cdot L + (V + mg) \cdot \sin\theta \cdot L = J\ddot{\theta} + b_t \dot{\theta} \tag{4.19}$$

$$\begin{aligned} &-\left(m\ddot{x} + m\ddot{\theta}L\cos\theta - m(\dot{\theta})^2 L\sin\theta\right) \cdot \cos\theta \cdot L + \\ &+\left(-m \cdot \ddot{\theta} \cdot L \cdot \sin\theta - m \cdot (\dot{\theta})^2 \cdot L \cdot \cos\theta + mg\right) \cdot \sin\theta \cdot L = \\ &= J\ddot{\theta} + b_t \dot{\theta} \end{aligned} \tag{4.27}$$

Simplify the Equation (4.27):

$$\begin{aligned}&-m\cdot\ddot{x}\cdot\cos\theta\cdot L-m\cdot\ddot{\theta}\cdot L^2\cdot\cos^2\theta+m\cdot\left(\dot{\theta}\right)^2\cdot L^2\cdot\sin\theta\cdot\cos\theta+m\cdot g\cdot L\cdot\sin\theta-\\&-m\cdot\ddot{\theta}\cdot L^2\cdot\sin^2\theta-m\cdot\left(\dot{\theta}\right)^2\cdot L^2\cdot\cos\theta\cdot\sin\theta=J\ddot{\theta}+b_t\dot{\theta}\end{aligned}\tag{4.28}$$

Since $\sin^2\theta+\cos^2\theta=1$, then:

$$-m\cdot\ddot{\theta}\cdot L^2-m\cdot\ddot{x}\cdot\cos\theta\cdot L=J\ddot{\theta}+b_t\dot{\theta}-m\cdot g\cdot L\cdot\sin\theta\tag{4.29}$$

$$\left(J+m\cdot L^2\right)\cdot\ddot{\theta}=-m\cdot\ddot{x}\cdot\cos\theta\cdot L-b_t\dot{\theta}+m\cdot g\cdot L\cdot\sin\theta\tag{4.30}$$

Then, the second equation of the inverted pendulum mathematical model is obtained as follows:

$$\left(J+m\cdot L^2\right)\cdot\ddot{\theta}=-m\cdot\ddot{x}\cdot\cos\theta\cdot L-b_t\dot{\theta}+m\cdot g\cdot L\cdot\sin\theta\tag{4.31}$$

Till now, two important equations are obtained – Equation (4.13) and (4.31). These two equations describe the translational and rotational movement of the inverted pendulum respectively. Because this particular project deals with the inverted pendulum that has two-dimensional movements, the pendulum and the rod move only in one plane. These obtained Equations (4.13), (4.31) can fully describe the motion of the inverted pendulum as follows:

$$F=\left(M+m\right)\ddot{x}+b\dot{x}+mL\ddot{\theta}\cos\theta-mL\left(\dot{\theta}\right)^2\sin\theta\tag{4.13}$$

$$\left(J+m\cdot L^2\right)\cdot\ddot{\theta}=-m\cdot\ddot{x}\cdot\cos\theta\cdot L-b_t\dot{\theta}+m\cdot g\cdot L\cdot\sin\theta\tag{4.31}$$

4.3.1.2.3 Conversion of the inverted pendulum model into a state space representation

From Equations (4.13), using the algebra elimination method, $\ddot{x}$ can be expressed as in Equations (4.32):

$$\ddot{x}=\frac{F-b\dot{x}-m\ddot{\theta}L\cos\theta+m\left(\dot{\theta}\right)^2L\sin\theta}{M+m}\tag{4.32}$$

Substitute Equation (4.32) into Equation (4.31) to eliminate $\ddot{x}$ from the equation, then:

$$\left(J+m\cdot L^2\right)\cdot\ddot{\theta}=-m\cdot\frac{F-b\dot{x}-m\ddot{\theta}L\cos\theta+m\left(\dot{\theta}\right)^2L\sin\theta}{M+m}\cdot\cos\theta\cdot L-b_t\dot{\theta}+m\cdot g\cdot L\cdot\sin\theta\tag{4.33}$$

$$\left(J+m\cdot L^2\right)\cdot\ddot{\theta}-m\cdot g\cdot L\cdot\sin\theta+m\cdot\frac{F-b\dot{x}-m\ddot{\theta}L\cos\theta+m\left(\dot{\theta}\right)^2L\sin\theta}{M+m}\cdot\cos\theta\cdot L=-b_t\dot{\theta}\tag{4.34}$$

Rearrange Equation (4.34), $\ddot{\theta}$ can be expressed as follow:

$$\ddot{\theta} = \left[J + m \cdot L^2 - m \cdot \cos\theta \cdot L \cdot \frac{mL\cos\theta}{M+m} \right]^{-1} \times$$

$$\times \left[\begin{array}{l} m \cdot g \cdot \sin\theta \cdot L - \frac{m}{M+m} \cdot \cos\theta \cdot L \cdot F + \frac{m}{M+m} \cdot \cos\theta \cdot L \cdot b\dot{x} - \\ - \frac{m}{M+m} \cdot \cos\theta \cdot L \cdot m\left(\dot{\theta}\right)^2 L\sin\theta - b_t\dot{\theta} \end{array} \right] \tag{4.35}$$

Substitute Equation (4.35) into Equation (4.32) to eliminate $\ddot{\theta}$ from the equation, then $\ddot{x}$ can be stated as:

$$\ddot{x} = \frac{F - b\dot{x} + m\left(\dot{\theta}\right)^2 L\sin\theta}{M+m} - \frac{mL\cos\theta}{M+m} \cdot \left[J + m \cdot L^2 - m \cdot \cos\theta \cdot L \cdot \frac{mL\cos\theta}{M+m} \right]^{-1} \times$$

$$\times \left[\begin{array}{l} m \cdot g \cdot \sin\theta \cdot L - \frac{m}{M+m} \cdot \cos\theta \cdot L \cdot F + \frac{m}{M+m} \cdot \cos\theta \cdot L \cdot b\dot{x} - \\ - \frac{m}{M+m} \cdot \cos\theta \cdot L \cdot m\left(\dot{\theta}\right)^2 L\sin\theta - b_t\dot{\theta} \end{array} \right] \tag{4.36}$$

Simplify Equation (4.36), $\ddot{x}$ can be expressed as:

$$\ddot{x} = \frac{\begin{array}{l} -\left(J + mL^2\right) \cdot b \cdot \dot{x} - m^2 \cdot g \cdot L^2 \sin\theta \cdot \cos\theta + m \cdot L \cdot b_t \cdot \dot{\theta} \cdot \cos\theta + \\ + \left(J + mL^2\right) \cdot mL\dot{\theta}^2 \sin\theta + \left(J + mL^2\right) \cdot F \end{array}}{\left(J + mL^2\right) \cdot \left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2\theta} \tag{4.37}$$

Simplify Equation (4.35), $\ddot{\theta}$ can be expressed as:

$$\ddot{\theta} = \frac{\begin{array}{l} \left(M + m\right) \cdot mg \cdot L \cdot \sin\theta - m^2 \cdot L^2 \cdot \left(\dot{\theta}\right)^2 \cdot \sin\theta \cdot \cos\theta - \left(M + m\right) \cdot b_t \cdot \dot{\theta} + \\ + m \cdot L \cdot \cos\theta \cdot b \cdot \dot{x} - F \cdot m \cdot L \cdot \cos\theta \end{array}}{\left(J + mL^2\right) \cdot \left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2\theta} \tag{4.38}$$

From Equation (4.37) and Equation (4.38), the inverted pendulum nonlinear mathematical model can be put into its first order differential equation form. In order to avoid the confusion of the letter usage, from now one, the states are expressed by the letter z. x is used as the displacement state only.

Let $\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$, then using the Equation (4.37) and Equation (4.38) the state space model can be expressed as follows:

$$\dot{z}_1 = z_2 \tag{4.39}$$

$$\dot{z}_2 = \frac{\begin{bmatrix} -\left(J+mL^2\right)\cdot b\cdot z_2 - m^2\cdot L^2\cdot g\cdot \sin z_3\cdot \cos z_3 + m\cdot L\cdot b_t\cdot z_4\cdot \cos z_3 + \\ +\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot \sin z_3 \end{bmatrix}}{\sigma} + \\ + \frac{\left(J+m\cdot L^2\right)}{\sigma}\cdot F \tag{4.40}$$

$$\dot{z}_3 = z_4 \tag{4.41}$$

$$\dot{z}_4 = \frac{\begin{bmatrix} (M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 - m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 - (M+m)\cdot b_t\cdot z_4 + \\ + m\cdot L\cdot \cos z_3\cdot b\cdot z_2 \end{bmatrix}}{\sigma} + \\ + \left[\frac{-m\cdot L\cdot \cos z_3}{\sigma}\right]\cdot F \tag{4.42}$$

where $\sigma = \left(J+mL^2\right)\cdot(M+m) - m^2\cdot L^2\cdot \cos^2 z_3$ (4.43)

The nonlinear state space model equations are:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} z_2 \\ \dfrac{\begin{bmatrix} -\left(J+mL^2\right)\cdot b\cdot z_2 - m^2\cdot L^2\cdot g\cdot \sin z_3\cdot \cos z_3 + \\ + m\cdot L\cdot b_t\cdot z_4\cdot \cos z_3 + \left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot \sin z_3 \end{bmatrix}}{\sigma} \\ z_4 \\ \dfrac{\begin{bmatrix} (M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 - m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 \\ -(M+m)\cdot b_t\cdot z_4 + m\cdot L\cdot \cos z_3\cdot b\cdot z_2 \end{bmatrix}}{\sigma} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{\left(J+m\cdot L^2\right)}{\sigma} \\ 0 \\ \left[\dfrac{-m\cdot L\cdot \cos z_3}{\sigma}\right] \end{bmatrix}\cdot F \tag{4.44}$$

4.3.1.2.4 Inverted pendulum model linearization

For linear state control of the inverted pendulum, it is necessary to convert the state equations (4.39)-(4.42) to state space representation in the form: $\dot{z} = Az + BF$.

For the trigonometric functions, if θ is a very small angle (smaller than 5°), $\sin\theta\approx\theta$, $\cos\theta\approx 1$. From the Taylor series expansion, the higher order terms are neglected, so $\left(\dot{\theta}\right)^2 \approx 0$, then:

The denominator can be linearized and simplified as follow:

$$
\begin{aligned}
\sigma' &= \left(J + mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2 = \\
&= MJ + mJ + MmL^2 + m^2L^2 - m^2L^2 = \\
&= J\left(M+m\right) + MmL^2
\end{aligned}
\tag{4.45}
$$

Then, the inverted pendulum linear mathematical model in the form of a first order differential equation is obtained as:

$$\dot{z}_1 = z_2 \tag{4.46}$$

$$\dot{z}_2 = \frac{\left[-\left(J+mL^2\right)\cdot b\cdot z_2 - m^2\cdot L^2\cdot g\cdot z_3 + m\cdot L\cdot b_t\cdot z_4 +\right]}{\sigma'} + \frac{\left(J+m\cdot L^2\right)}{\sigma'}\cdot F \tag{4.47}$$

$$\dot{z}_3 = z_4 \tag{4.48}$$

$$\dot{z}_4 = \frac{\left[m\cdot L\cdot b\cdot z_2 + \left(M+m\right)\cdot m\cdot g\cdot L\cdot z_3 - \left(M+m\right)\cdot b_t\cdot z_4\right]}{\sigma'} + \left[\frac{-m\cdot L}{\sigma'}\right]\cdot F \tag{4.49}$$

The linearized mathematical model into state-space form is expressed as follows:

$$
\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix}
=
\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix}
=
\begin{bmatrix}
0 & 1 & 0 & 0 \\
0 & -\frac{\left(J+mL^2\right)\cdot b}{\sigma'} & \frac{-m^2L^2g}{\sigma'} & \frac{mLb_t}{\sigma'} \\
0 & 0 & 0 & 1 \\
0 & \frac{mLb}{\sigma'} & \frac{\left(M+m\right)\cdot mgL}{\sigma'} & -\frac{\left(M+m\right)\cdot b_t}{\sigma'}
\end{bmatrix}
\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix}
+
\begin{bmatrix} 0 \\ \frac{\left(J+mL^2\right)}{\sigma'} \\ 0 \\ -\frac{mL}{\sigma'} \end{bmatrix} F
\tag{4.50}
$$

The linearized model in Equations (4.45) to (4.48) is used in Chapter 5 to derive the linear control of the inverted pendulum. The nonlinear model in Equations (4.39) to (4.42) is used in Chapters 6 and 7 to design nonlinear control based on the Lyapunov second method and based on the feedback linearization approaches respectively.

According to the structure of the pendulum system, the inverted pendulum is controlled through the position of the pendulum. Its displacement is defined as: y=x+Lsinθ, then the linearized output of the model can be expressed as:

$$y = \begin{bmatrix} 1 & 0 & L & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} \tag{4.51}$$

4.3.2 Overhead crane system

4.3.2.1 Introduction of the overhead crane

The overhead crane is widely used in Industry. It is especially used in the heavy assembly industry for mass carrying and assembling. The problem the overhead crane operator faces is how to carry mass from point A to point B safely without oscillations and using the least amount of time to complete the task. Assume one ball is connected with a string that is much longer as compare to the ball's diameter. One holds the end of the string and moves it in a horizontal direction rapidly. If one stops the movement suddenly, because of the inertia, the ball will continue to move in the same direction along a circle which has the radius of the length of the string and one's hand is the circle's origin. Then, because of the gravitational force, the ball falls back. The oscillation starts. When the energy that consists in the system dissipates, the ball stops swing and stay below one's hand. Similarly, if the operator moves the mass fast, he or she cannot stop it at the desired position. Otherwise, the mass can only be moved slowly. A controller is necessary to be developed. The controller can help the operator to move mass rapidly and without oscillations.

In the considered pendulum system, when the carriage module is turned upside down the pendulum acts as an overhead crane. It has exactly the same behaviour as mentioned above. For example, if the position of the cart is changed, the pendulum starts swinging naturally. With the energy dissipating, it eventually stops at a stable equilibrium position where the centre of the mass of the pendulum is below the pivot. The problem is now to control the position of the load, which possesses oscillatory dynamics.

4.3.2.2 Development of the model of the crane

The model of the crane is developed following the same steps as for the model of the inverted pendulum. Because of the similarities, the model derivation for the overhead crane is described in a relatively simpler way as compared to the inverted pendulum model.

The movement of the crane is characterised within the considered above two types:

- Translation movement for the cart, the pendulum and the rod
- Rotational movement for the pendulum and the rod on the pivot

The direction of the movement of the pendulum is the clock one. All assumptions made for the inverted pendulum are considered in this case also.

The coordinates of the crane are (in the IV-th quadrant) of the coordinate plane

$$x_G = x + L\sin\theta \tag{4.52}$$

$$y_G = -L\cos\theta \tag{4.53}$$

4.3.2.2.1 Overhead Crane mathematical model derivation from the translational movement

The balance of the forces acting on the cart is:

$$M\frac{d^2x}{dt^2} + b\frac{dx}{dt} = F - H = F - M\frac{d^2}{dt^2}x_G \tag{4.54}$$

Substitute Equation (4.52) in Equation (4.54), then the first equation for describing the overhead crane is obtained as follow:

$$F = (M+m)\ddot{x} + b\dot{x} + mL\ddot{\theta}\cos\theta + mL\dot{\theta}^2\sin\theta \tag{4.55}$$

This equation is the same as the equation for the inverted pendulum in the quadrant I of the plane.

4.3.2.2.2 Overhead crane mathematical model derivation on the basis of the rotational movement

There are three forces applied on the pendulum and the rod centre of the gravity. H is in the horizontal direction; V and mg are in the vertical direction.

The equation for the balance of the vertical forces for the pendulum and the rod is:

$$-V - mg = m\frac{d^2}{dt^2}y = m\frac{d^2}{dt^2}(-L\cos\theta) \tag{4.56}$$

The signs of V and mg are negative because they are in the IV-th quadrant. Simplify Equation (4.56),

$$V = -mg - m\frac{d^2(-L\cos\theta)}{dt^2} = -mg - mL\ddot{\theta}\sin\theta - mL\dot{\theta}^2\cos\theta \tag{4.57}$$

where $\frac{d^2(-\cos\theta)}{dt^2} = \frac{d(\dot{\theta}\sin\theta)}{dt} = \ddot{\theta}\sin\theta + \dot{\theta}^2\cos\theta$

The equation for the horizontal force H is:

$$H = m\frac{d^2 x_G}{dt^2} = m\ddot{x} + mL\ddot{\theta}\cos\theta - mL\dot{\theta}^2\sin\theta \tag{4.58}$$

The torque balance equation for the pendulum and the rod in the overhead crane mode can be expressed as:

$$J\ddot{\theta} + b_T\dot{\theta} = VL\sin\theta - HL\cos\theta \tag{4.59}$$

Equation (4.57) and (4.58) are substituted in equation (4.59). The second equation for describing the overhead crane is obtained:

$$\left(J + mL^2\right)\cdot\ddot{\theta} + b_T\dot{\theta} + mgL\sin\theta + mL\ddot{x}\cos\theta = 0 \tag{4.60}$$

The model of the crane is given by the Equations (4.55) and (4.60). These equations are used to derive the state space model of the crane.

4.3.3 State space model

From Equation (4.54) and (4.59), by using the algebra elimination method, $\ddot{x}$ and $\ddot{\theta}$ can be expressed as:

$$\ddot{x} = \frac{\begin{bmatrix} -\left(J + mL^2\right)\cdot b\cdot\dot{x} + m^2\cdot L^2\cdot g\cdot\sin\theta\cdot\cos\theta + m\cdot L\cdot b_T\cdot\dot{\theta}\cdot\cos\theta + \\ + \left(J + mL^2\right)\cdot m\cdot L\cdot\left(\dot{\theta}\right)^2\cdot\sin\theta + \left(J + mL^2\right)\cdot F \end{bmatrix}}{\left(M + m\right)\left(J + mL^2\right) - m^2\cdot L^2\cdot\cos^2\theta} \tag{4.61}$$

$$\ddot{\theta} = \frac{\begin{bmatrix} -\left(M + m\right)\cdot m\cdot g\cdot L\cdot\sin\theta - m^2\cdot L^2\cdot\left(\dot{\theta}\right)^2\cdot\sin\theta\cdot\cos\theta - \left(M + m\right)\cdot b_T\cdot\dot{\theta} + \\ + m\cdot L\cdot b\cdot\dot{x}\cdot\cos\theta - F\cdot m\cdot L\cdot\cos\theta \end{bmatrix}}{\left(M + m\right)\left(J + mL^2\right) - m^2 L^2\cos^2\theta} \tag{4.62}$$

The same states as for the inverted pendulum are used:

Let $\begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$ then the state space nonlinear model can be expressed as follows:

$$\dot{z}_1 = z_2 \tag{4.63}$$

$$\dot{z}_2 = \frac{\begin{bmatrix} -\left(J + mL^2\right)\cdot b\cdot z_2 + m^2\cdot L^2\cdot g\cdot\sin z_3\cdot\cos z_3 + \\ + \left(J + mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot\sin z_3 + m\cdot L\cdot b_t\cdot z_4\cdot\cos z_3 \end{bmatrix}}{\sigma} + \frac{\left(J + m\cdot L^2\right)}{\sigma}\cdot F \tag{4.64}$$

$$\dot{z}_3 = z_4 \tag{4.65}$$

$$\dot{z}_4 = \frac{\begin{bmatrix} +m\cdot L\cdot\cos z_3\cdot b\cdot z_2-(M+m)\cdot m\cdot g\cdot L\cdot\sin z_3- \\ -m^2\cdot L^2\cdot z_4^2\cdot\sin z_3\cdot\cos z_3-(M+m)\cdot b_t\cdot z_4 \end{bmatrix}}{\sigma}+\frac{(-m\cdot L\cdot\cos z_3)}{\sigma}\cdot F \qquad (4.66)$$

where $\sigma=(J+mL^2)\cdot(M+m)-m^2\cdot L^2\cdot\cos^2 z_3$ (4.67)

The nonlinear state space equations are:

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\\ \dot{z}_3\\ \dot{z}_4\end{bmatrix}=\begin{bmatrix} z_2 \\ \dfrac{\begin{bmatrix} -(J+mL^2)\cdot b\cdot z_2+m^2\cdot L^2\cdot g\cdot\sin z_3\cdot\cos z_3+ \\ +(J+mL^2)\cdot m\cdot L\cdot z_4^2\cdot\sin z_3+m\cdot L\cdot b_t\cdot z_4\cdot\cos z_3 \end{bmatrix}}{\sigma} \\ z_4 \\ \dfrac{\begin{bmatrix} +m\cdot L\cdot\cos z_3\cdot b\cdot z_2-(M+m)\cdot m\cdot g\cdot L\cdot\sin z_3- \\ -m^2\cdot L^2\cdot z_4^2\cdot\sin z_3\cdot\cos z_3-(M+m)\cdot b_t\cdot z_4 \end{bmatrix}}{\sigma}\end{bmatrix}+\begin{bmatrix} 0 \\ \dfrac{J+m\cdot L^2}{\sigma} \\ 0 \\ \dfrac{-m\cdot L\cdot\cos z_3}{\sigma}\end{bmatrix}\cdot F$$

(4.68)

Linearize the Equation (4.63) to (4.66), and then put them into the state space form:

$$\dot{z}_1=z_2 \qquad (4.69)$$

$$\dot{z}_2=\frac{\left[-(J+mL^2)\cdot b\cdot z_2+m^2\cdot L^2\cdot g\cdot z_3+m\cdot L\cdot b_t\cdot z_4\right]}{\sigma'}+\frac{(J+m\cdot L^2)}{\sigma'}\cdot F \qquad (4.70)$$

$$\dot{z}_3=z_4 \qquad (4.71)$$

$$\dot{z}_4=\frac{\left[m\cdot L\cdot b\cdot z_2-(M+m)\cdot m\cdot g\cdot L\cdot z_3-(M+m)\cdot b_t\cdot z_4\right]}{\sigma'}+\left[\frac{-m\cdot L}{\sigma'}\right]\cdot F \qquad (4.72)$$

where $\sigma'=J(M+m)+MmL^2$ (4.73)

The linearized mathematical model into state-space form is expressed as follows:

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\\ \dot{z}_3\\ \dot{z}_4\end{bmatrix}=\begin{bmatrix}\dot{x}\\ \ddot{x}\\ \dot{\theta}\\ \ddot{\theta}\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & -\frac{\left(J+mL^2\right)\cdot b}{\sigma'} & \frac{m^2L^2g}{\sigma'} & \frac{mLb_t}{\sigma'}\\ 0 & 0 & 0 & 1\\ 0 & \frac{mLb}{\sigma'} & \frac{-(M+m)\cdot mgL}{\sigma'} & -\frac{(M+m)\cdot b_t}{\sigma'}\end{bmatrix}\begin{bmatrix}z_1\\ z_2\\ z_3\\ z_4\end{bmatrix}+\begin{bmatrix}0\\ \frac{\left(J+mL^2\right)}{\sigma'}\\ 0\\ -\frac{mL}{\sigma'}\end{bmatrix}F \tag{4.74}$$

The output signal is expressed as: y=x+Lθ, into state space form, its representation is:

$$y=\begin{bmatrix}1 & 0 & L & 0\end{bmatrix}\begin{bmatrix}z_1\\ z_2\\ z_3\\ z_4\end{bmatrix}=\begin{bmatrix}1 & 0 & L & 0\end{bmatrix}\begin{bmatrix}x\\ \dot{x}\\ \theta\\ \dot{\theta}\end{bmatrix} \tag{4.75}$$

4.3.4 Comparison between the models of the inverted pendulum and the overhead crane

The nonlinear and the linear models of the inverted pendulum in Equations (4.44) and (4.50) and the overhead crane in Equations (4.68) and (4.74) have the same structures and elements. The difference is in the signs in front of $\sin z_3$ and $z_3=\theta$. These expressions give the positions of the angle θ in the coordinate plane.

4.3.5 Parameters of the inverted pendulum and the overhead crane

The parameters used for the inverted pendulum and the overhead crane mathematical model derivation are shown in Table 4.1. These parameters are specified in the developed Matlab software m-files and are used for the simulations and Real-Time implementation.

Table 4.1: **Table of the parameter for the pendulum system**

Symbol	Description	Value	Unit
M	Mass of the cart	0.050	kg
m	Mass of the pendulum and rod ($m=m_p+m_r$)	0.280	kg
m_p	Mass of the pendulum	0.220	kg
m_r	Mass of the rod	0.060	kg
L	Distance between the pivot and the pendulum and rod's centre of gravity	0.22	m
J	Moment of inertia of the pendulum and rod	0.0013	Kgm^2
b	Friction between the cart wheel and the surface	0.1	m/s^3
b_t	Friction at the pivot	0.01	m/s^3

4.4 Mathematical model derivation for the servo system

In the section 4.3, the mathematical models of the inverted pendulum and the overhead crane are derived. From the model, it can be seen that the control signal applied to the pendulum system is force F. In this reconfigurable plant, according to its physical structure the driven force applied to the cart is generated by a servo system. This servo system consists of servomotor, speed amplifier, feedback amplified and a toothed belt. In this section, the d.c. motor mathematical model is derived from basic electromagnetic relationships; the servo system mathematical model is derived based on the study of the servomotor and the servo system formation. Through the study of the servo system, the force applied to the cart can be represented by the electrical signal.

A servomotor which is also called servo is an electro-mechanical device in which an electrical input determines the position of the armature of a motor. Servos are used extensively in robotics and radio-controlled puppets, cars, airplanes and boats (http://www.ee.duke.edu/~cec/final/node59.html). The servomotor used in this pendulum system also has an integral tachometer (Bytronic international LTD, 2001: 1.1). A tachometer is an instrument designed to measure the speed of an object or substance. The d.c. motor is the actuator at the heart of a position control servomechanism such as servomotor. It is the means by which electrical energy is converted to mechanical energy. In this section, the d.c. motor mathematical model derivation is presented firstly.

4.4.1 Direct current servomotor mathematical model derivation

There are armature controlled d.c. motors and field controlled d.c motors. In the first mode the field current is held constant, and an adjustable voltage is applied to the armature. In the second mode the armature current is held constant, and an adjustable voltage is applied to the field. In this pendulum system, an armature controlled d.c. motor is used. In this section, the armature controlled d.c. motor is studied in detail.

4.4.1.1 Basic operation principle of the direct current motor

Consider a piece of wire carrying a constant current i suspended in a magnetic field of uniform flux density B’, where the direction of the current is at right angles to the flux, as shown in Figure 4.6 below. In a d.c. servomotor the magnetic field B’ is usually produced by a strong permanent magnet and this part of the motor is known as the stator.

According to Ampere’s law, force F is produced whose direction is orthogonal to both the current and flux. Its magnitude is given as:

$F=B'li$ (4.76)

where l is the length of wire within the field and i is the current flowing the wire as shown in Figure 4.6.

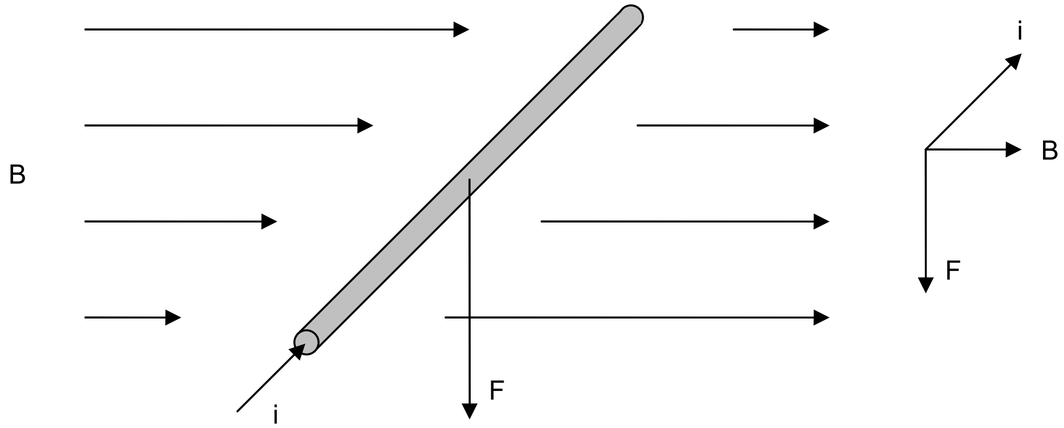


Figure 4.6: **Force analysis of a piece of wire with carrying a constant current suspended in a magnetic field**

Instead of having a single current carrying wire, the coil in the magnetic filed is supported in the middle on a shaft which rotates in bearings at either end of the motor. This part of the motor is known as the rotor or armature. The net effect of the two forces, one upward and one downward, is to exert a turning moment or torque of value:

$T=2Fr_c$ (4.77)

Where r_c is the distance between the centre line of the shaft and the conductor. This causes the coil/shaft assembly to turn with the direction of anticlockwise shown in Figure 4.7. The force direction or the shaft turning direction can be reversed by simply changing the direction of the current in the coil.

In a particular d.c. servomotor, supposing the coil has N turns instead of just one, then the motor torque can be calculated as:

$T_m=2NB'lr_ci$ (4.78)

where N, B', l and r_c are constant for any particular motor. It can also be written as:

$T_m=K_mi$ (4.79)

where $K_m=2NB'lr_c$ is known as the motor's torque constant with units of NmA^{-1}.

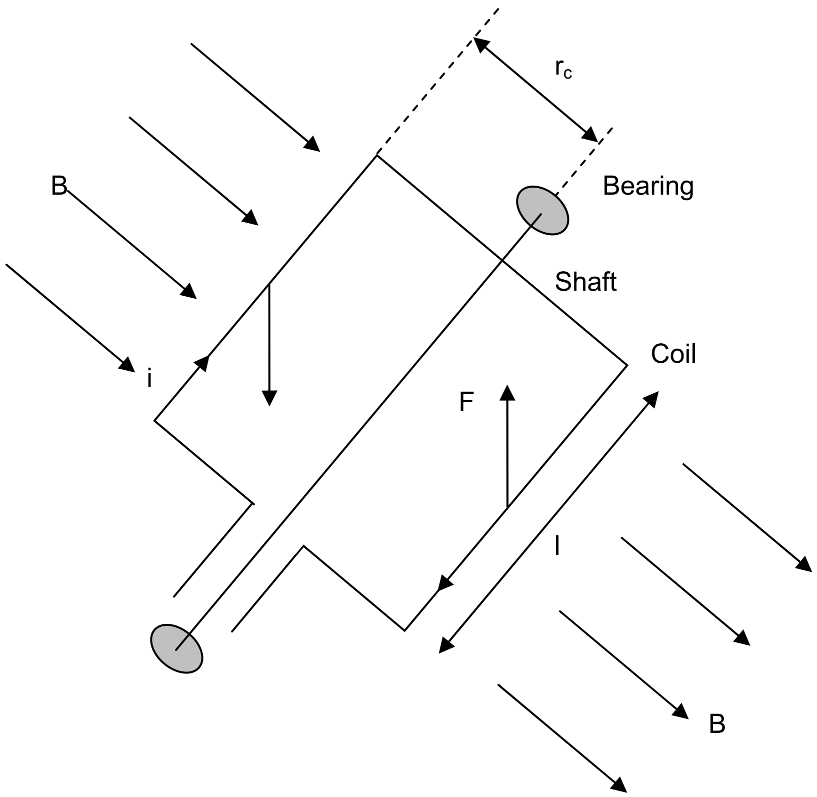


Figure 4.7: Circuit analysis of the d.c. motor coil

According to Faraday's Law, a coil of length l rotating in a uniform magnetic field of flux density B' will generate an electromagnetic force (e.m.f.). In other words, when the coil is driven by electrical energy, the coil generates electrical energy while it is turning at the same time. This e.m.f. is given by:

$$e = -\frac{d\lambda}{dt} \tag{4.80}$$

where λ is the flux linking the coil.

Consider the coil of length l and radius r_c to be at an angle θ_m to the magnetic field. The magnetic flux λ linking the coil is given by:

$$\lambda = B'l2r_c\cos\theta_m \tag{4.81}$$

And the magnetic flux linkage is:

$$\lambda = N\phi = 2NB'lr_c\cos\theta_m \tag{4.82}$$

If the coil is rotating at a constant angular speed ω_m then its angle at any time t is $\omega_m t$, then:

$$e' = -\frac{d\lambda}{dt} = -2NB'Ir_c \frac{d(\cos\omega_m t)}{dt} \qquad (4.83)$$

$$\therefore\ e' = -2NB'Ir_c(-\omega_m \sin(\omega_m t)) = 2NBIr\omega_m \sin(\omega_m\ ) \qquad (4.84)$$

when the coil is at θ_m=90°, the term $\sin(\omega_m t)=1$ and:

$$e' = 2NBIr_c\omega_m = K_b\omega_m \qquad (4.85)$$

where e is called back e.m.f constant, with units of $Vrad^{-1}s^{-1}$. K_b is the voltage constant.

There is a basic relationship between the torque constant and voltage constant.

$$K_m = K_b \times 1.345 \qquad (4.86)$$

But mostly, K_m and K_b are simply considered to be having the same value.

4.4.1.2 Armature controlled direct current motor model derivation

Based on the study of the d.c. motor operation principle, the armature controlled d.c. motor model is derived. In a particular d.c. motor, if V_a is the voltage applied to the armature circuit, it causes a current i_a to flow and produces a proportional torque. This in turn accelerates the armature. As the armature gathers speed, a proportional e.m.f. is generated, according to the back e.m.f., which tends to oppose the current. Eventually, the rotor gathers sufficient speed such that the back e.m.f. just equals the applied voltage. In the steady state, when V_a=e’, no current flows in the coil so there is no further acceleration and the rotor turns at a constant speed.

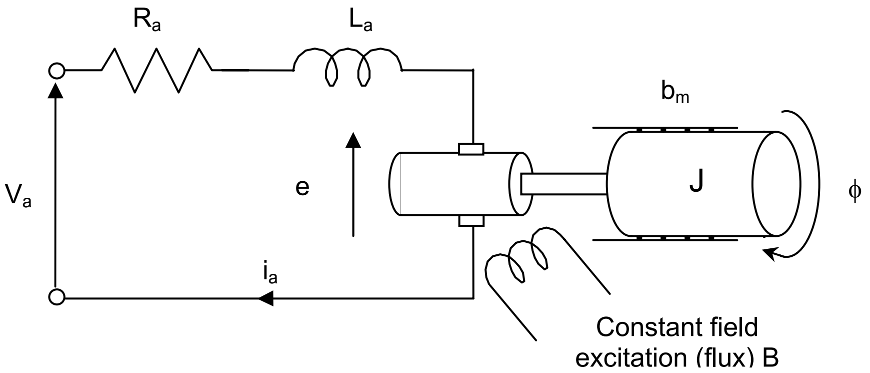


Figure 4.8: The d.c. motor equivalent circuit

However, assumes a perfect motor with no friction at all in the bearings and no electrical power losses. In practices, running a motor at steady speed requires a small amount of current to flow, sufficient to just overcome the friction at that speed. The rotor coil will also have some resistance R_a, so any current will cause ohmic losses and a small voltage drop. The coil also has self-inductance L_a. If the current in the coil changes then there will be an induced back e.m.f which opposes that change. In Figure 4.8, the equivalent circuit of the d.c. motor is shown. From Figure 4.8, the voltage equation can be derived as:

$$V_a = i_a R_a + L_a \frac{di_a}{dt} + e \tag{4.87}$$

or $V_a - e = i_a R_a + L_a \frac{di_a}{dt}$ (4.88)

Take the Laplace transforms of Equation (4.87):

$$V_a(s)\text{-}E(s)=(R_a+L_a s)I_a(s) \tag{4.89}$$

Applying Newton’s Law (by summing moments) for the rotational motion of the motor gives:

$$\sum M = J_m \frac{d\omega}{dt} = T_e - b_m \omega_m \tag{4.90}$$

or $J_m \dot{\omega} + b_m \omega_m = T_e$

where T_e is the input torque to the load system; b_m is the viscous friction coefficient; J_m is the motor inertia.

Thus, the transfer function from the input motor torque to rotational speed changes is:

$$\frac{\omega}{T_e} = \frac{(1/J_m)}{s + b_m/J_m} = \frac{1}{J_m s + b_m} \tag{4.91}$$

From Equation (4.88), if the coil self-inductance is small enough to be neglected, the current draw of the motor can be calculated in the following way:

$$i=[v_a\text{-}e]/R_a \tag{4.92}$$

Substitute Equation (4.85) into Equation (4.92):

$$i=[V_a\text{-}K_b\cdot\omega_m]/R_a \tag{4.93}$$

The torque output of the motor generated by electrical energy is:

$$T_e=K_m i \tag{4.94}$$

where K_m is the motor torque constant.

Substitute Equation (4.93) into Equation (4.94), then:

$$T_e = K_m \cdot [V_a - K_b \cdot \omega_m]/R_a = [V_a - K_b \cdot \omega_m] \cdot \frac{K_m}{R_a} \tag{4.95}$$

T_ω is the load torque which can be expressed as:

$$T_\omega = F \times r \tag{4.96}$$

where r is the motor radius to belt.

From the schematic of the servo system, the following equation is obtained:

$$T_e - T_\omega = J_m \dot{\omega} + b_m \omega_m$$

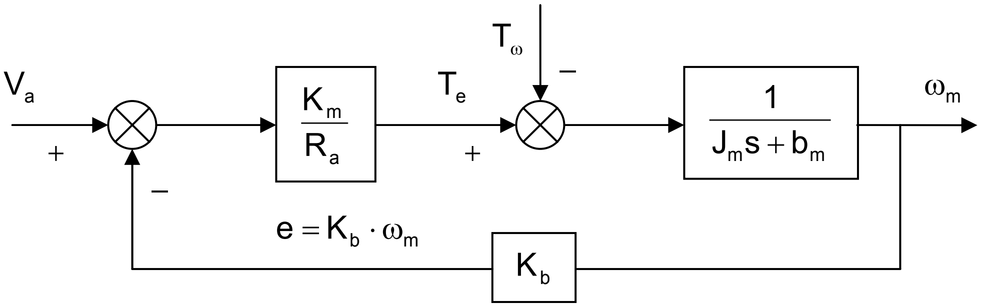


Figure 4.9: **The d.c. motor closed-loop block diagram**

Equation (4.85), (4.87) and (4.94) can describe the characteristics of a d.c. armature controlled motor very well. Combine these three equations, the closed-loop block diagram of a d.c. motor is shown in Figure 4.9.

4.4.2 Servo system mathematical model derivation

The pendulum system in the trainer is controlled by electrical signal. The servo system in the pendulum system converts the electrical control signal u into the mechanical energy in the form of the force F and the torque T. The servo system consists of the servomotor, the servo amplifier and the servo controller. Based on the studying of the d.c. motor and according to the configuration of the pendulum system, the mathematical model of the servo system is derived in this section. In Figure 4.10, the schematic diagram of the servo system is shown. Based on its study, the equivalent block diagram of the servo system can be drawn. It is shown in Figure 4.11.

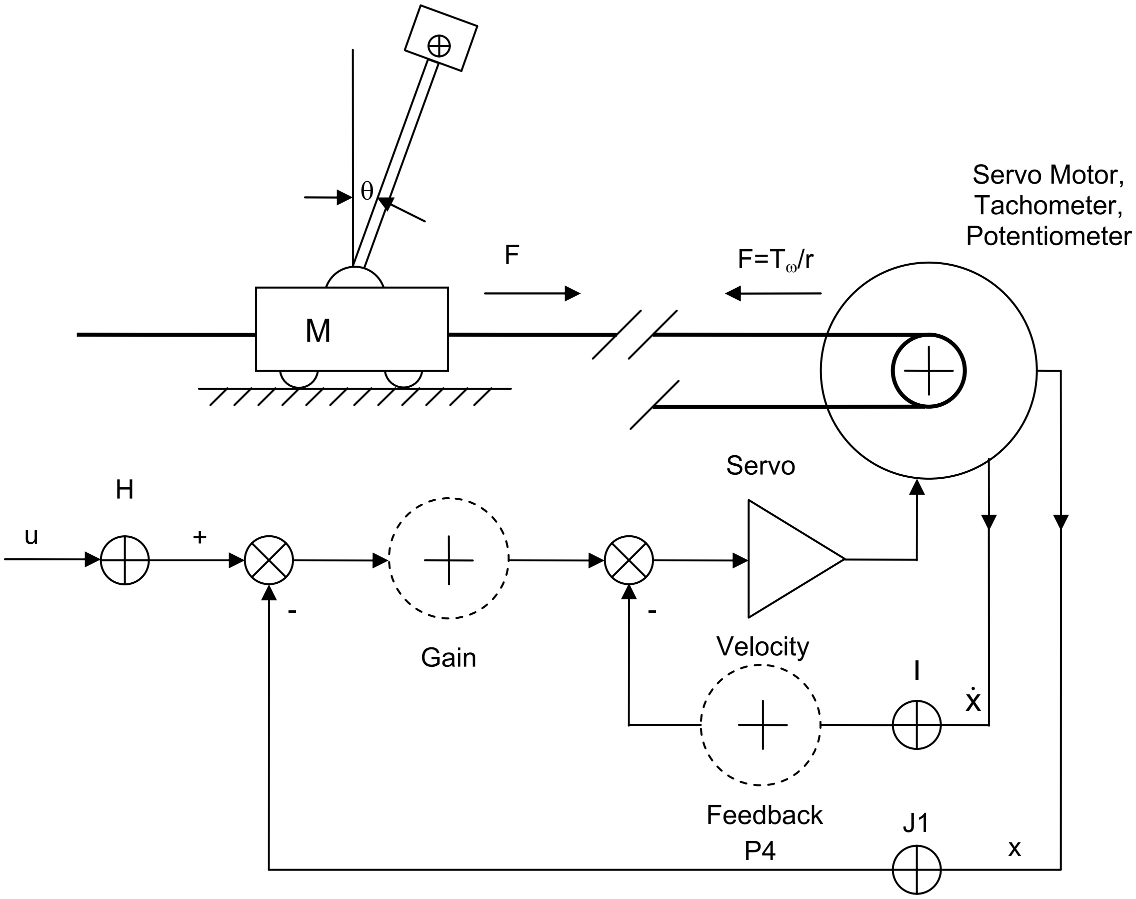


Figure 4.10: **Servo system hardware configuration**

The parameters which are used for the servo system mathematical model derivation are listed in Table 4.2.

Table 4.2: **Table of the servo system parameters** $_g$

Symbol	Name	Value	Unit
P_g	Potentiometer Gain (proportional control for servo position)	270/47	-
P_s	Servo amplifier Gain	750/47	-
K_D	Potentiometer Gain (derivative control for servo position)	0.5(47/56)	-
K_x	Potentiometer Gain for measuring cart position	5/0.125	-
K_ω	Tacho Gain for measuring cart velocity	10/0.25	V/m/s
K_m	Motor torque constant	9	Ncm/A
K_b	Motor voltage constant	10.3	V/1000rpm
r	Motor radius to belt	0.02	m
J_m	Motor inertia	0.00002245	kg·m^2
b_m	Viscous friction coefficient	0.0011225	kg·m^2/s
R_a	Armature resistance	5	Ohm

The block diagram of the servo system simplifies the relationships between different parameters and subsystems in the servo system. The mathematical model of the servo system is derived according to the block diagram.

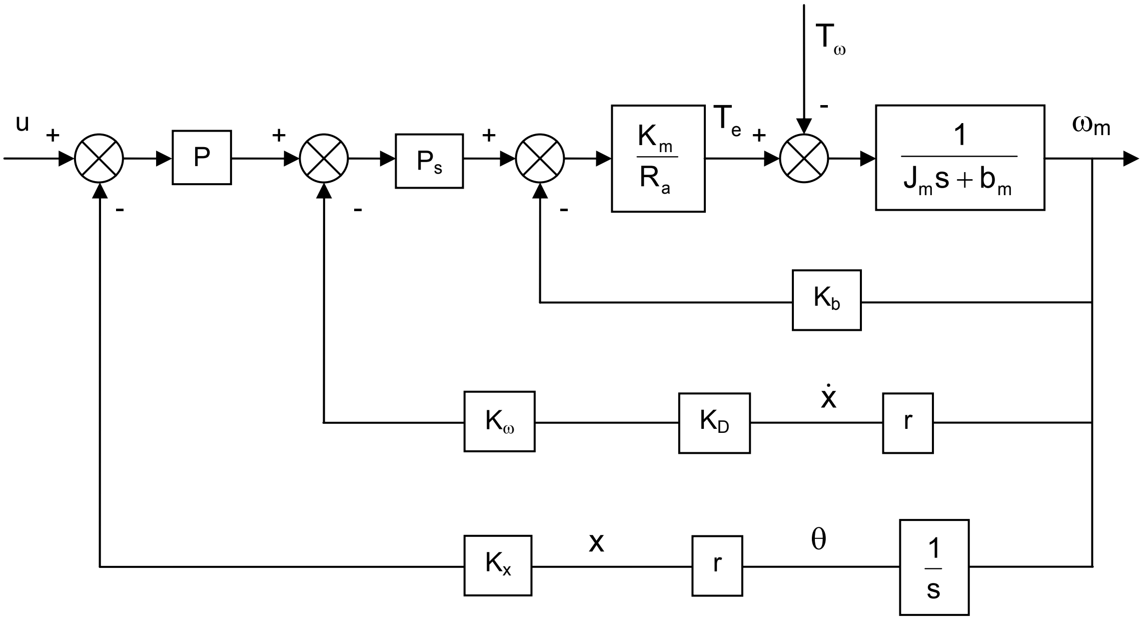


Figure 4.11: Servo system block diagram

Following the servo system block diagram, the mathematical model of the servo system is derived. Equation (4.97) represents the relationship between the torque applied to the motor shaft and the speed of the motor as shown below.

$$T_e - T_\omega = J_m \dot{\omega}_m + b_m \omega_m \tag{4.97}$$

Substitute Equation (4.96) into Equation (4.97), then:

$$T_e - F \cdot r = J_m \dot{\omega}_m + b_m \omega_m \tag{4.98}$$

$$F \cdot r = T_e - J_m \dot{\omega}_m - b_m \omega_m \tag{4.99}$$

$$F = \frac{T_e - J_m \dot{\omega} - b_m \omega}{r} = \frac{T_e}{r} - \frac{J_m \dot{\omega}}{r} - \frac{b_m \omega}{r} \tag{4.100}$$

From the servo system block diagram, T_e can be expressed as:

$$T_e = \left\{ \left[\left[\left(u - \omega_m \cdot r \cdot \frac{1}{s} \cdot K_x \right) \cdot P - \left(\omega_m \cdot r \cdot K_D \cdot K_\omega \right) \right] \cdot P_s \right] - \omega_m \cdot K_b \right\} \cdot \frac{K_m}{R_a} \tag{4.101}$$

Simplify the expression:

$$T_e = \left\{ \left[\left[\left(u \cdot P - \omega_m \cdot r \cdot \frac{1}{s} \cdot K_x \cdot P \right) - \left(\omega_m \cdot r \cdot K_D \cdot K_\omega \right) \right] \cdot P_s \right] - \omega_m \cdot K_b \right\} \cdot \frac{K_m}{R_a}$$

$$T_e = \left\{ \left(u \cdot P \cdot P_s - \omega_m \cdot r \cdot \frac{1}{s} \cdot K_x \cdot P \cdot P_s \right) - \left(\omega_m \cdot r \cdot K_D \cdot K_\omega \cdot P_s \right) - \omega_m \cdot K_b \right\} \cdot \frac{K_m}{R_a}$$

$$T_e = u \cdot P \cdot P_s \cdot \frac{K_m}{R_a} - \omega_m \cdot r \cdot \frac{1}{s} \cdot K_x \cdot P \cdot P_s \cdot \frac{K_m}{R_a} - \omega_m \cdot r \cdot K_D \cdot K_\omega \cdot P_s \cdot \frac{K_m}{R_a} - \omega_m \cdot K_b \cdot \frac{K_m}{R_a}$$

$$T_e = \frac{u \cdot P \cdot P_s \cdot K_m}{R_a} - \frac{\omega_m \cdot r \cdot K_x \cdot P \cdot P_s \cdot K_m}{s \cdot R_a} - \frac{\omega_m \cdot r \cdot K_D \cdot K_\omega \cdot P_s \cdot K_m}{R_a} - \frac{\omega_m \cdot K_b \cdot K_m}{R_a} \tag{4.102}$$

Substitute Equation (4.102) into Equation (4.100), then:

$$F = \frac{T_e - J_m \dot{\omega}_m - b_m \omega_m}{r} = \frac{T_e}{r} - \frac{J_m \dot{\omega}}{r} - \frac{b_m \omega}{r}$$

$$\begin{aligned} &= \left[\frac{u \cdot P \cdot P_s \cdot K_m}{R_a \cdot r} - \frac{\omega_m \cdot r \cdot K_x \cdot P \cdot P_s \cdot K_m}{s \cdot R_a \cdot r} - \frac{\omega_m \cdot r \cdot K_D \cdot K_\omega \cdot P_s \cdot K_m}{R_a \cdot r} - \frac{\omega_m \cdot K_b \cdot K_m}{R_a \cdot r} \right] - \\ &- \frac{J_m \dot{\omega}}{r} - \frac{b_m \omega_m}{r} \end{aligned} \tag{4.103}$$

If φ is the angle of the motor, then the following equations can be formulated: $\varphi r = x$; $\omega = \dot{\varphi}$; $\omega r = \dot{x}$; $\frac{r\omega}{s} = x$ and $r\dot{\omega} = \ddot{x}$. By using these equations, the further simplification is taken, then:

$$\begin{aligned} F &= \frac{P \cdot P_s \cdot K_m}{R_a \cdot r} \cdot u - \frac{K_x \cdot K_m \cdot P \cdot P_s}{R_a \cdot r} \cdot \frac{\dot{x}}{s} - \frac{K_D \cdot K_\omega \cdot P_s \cdot K_m}{R_a \cdot r} \cdot \dot{x} - \\ &- \frac{K_b \cdot K_m}{R_a \cdot r} \cdot \frac{\omega_m \cdot r}{r} - \frac{J_m \dot{\omega}_m}{r} \cdot \frac{r}{r} - \frac{b_m \omega_m}{r} \cdot \frac{r}{r} \end{aligned} \tag{4.104}$$

$$\begin{aligned} F &= \frac{P \cdot P_s \cdot K_m}{R_a \cdot r} \cdot u - \frac{K_x \cdot K_m \cdot P \cdot P_s}{R_a \cdot r} \cdot x + \\ &+ \left[-\frac{K_D \cdot K_\omega \cdot P_s \cdot K_m}{R_a \cdot r} - \frac{K_b \cdot K_m}{R_a \cdot r^2} - \frac{b_m}{r^2} \right] \cdot \dot{x} - \frac{J_m}{r^2} \cdot \ddot{x} \end{aligned} \tag{4.105}$$

Let $F = N_1 u + N_2 x + N_3 \dot{x} + N_4 \ddot{x}$ (4.106)

where $N_1 = \frac{P \cdot P_s \cdot K_m}{R_a \cdot r}$ (4.107)

$$N_2 = -\frac{K_x \cdot K_m \cdot P \cdot P_s}{R_a \cdot r} \tag{4.108}$$

$$N_3 = -\frac{K_D \cdot K_\omega \cdot P_s \cdot K_m}{R_a \cdot r} - \frac{K_b \cdot K_m}{R_a \cdot r^2} - \frac{b_m}{r^2} \tag{4.109}$$

$$N_4 = -\frac{J_m}{r^2} \tag{4.110}$$

Expressed in the state space notations:

$$F = N_1 u + N_2 z_1 + N_3 z_2 + N_4 \dot{z}_2 \tag{4.111}$$

The control u, which represents the reference position input to the servo system P, K_D is:

$$u = \frac{F - N_2 z_1 - N_3 z_2 - N_4 \dot{z}_2}{N_1} \tag{4.112}$$

Equation (4.111) represents the servo system. It can be seen from Equation (4.111) that the input signals to the servo system is the control signal u and the output is the force F applied to the cart. In the next subsection, the complete mathematical model of the pendulum system is derived by combining the inverted pendulum and overhead crane system mathematical models with the servo system mathematical model respectively.

4.5 Complete pendulum system mathematical model derivation

In the Section 4.3, the inverted pendulum and the overhead crane mathematical models are derived. In the Section 4.4, the mathematical model of the servo system is derived. The combined pendulum system mathematical models are derived in this subsection by integrating the inverted pendulum and the overhead crane mathematical models with the servo system mathematical model. The mathematical models derived in the Section 4.3 have for inputs the force F. By combining the servo system with the pendulum system, the system inputs become control signal u.

4.5.1 Complete inverted pendulum mathematical model

The complete inverted pendulum mathematical model is derived by substituting Equation (4.111) into Equation (4.40) and Equation (4.42). Then, the complete inverted pendulum mathematical model is expressed as:

$$\dot{z}_1 = z_2 \tag{4.113}$$

$$\dot{z}_2 = \frac{\begin{bmatrix} -\left(J + mL^2\right)\cdot b \cdot z_2 - m^2 \cdot L^2 \cdot g \cdot \sin z_3 \cdot \cos z_3 + m \cdot L \cdot b_t \cdot z_4 \cdot \cos z_3 + \\ + \left(J + mL^2\right)\cdot m \cdot L \cdot z_4^2 \cdot \sin z_3 \end{bmatrix}}{\sigma} + \\ + \frac{\left(J + m \cdot L^2\right)}{\sigma} \cdot \left[N_1 u + N_2 z_1 + N_3 z_2 + N_4 \dot{z}_2\right] \tag{4.114}$$

$$\dot{z}_3 = z_4 \tag{4.115}$$

$$\dot{z}_4 = \frac{\left[\begin{array}{l}(M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 - m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 - (M+m)\cdot b_t\cdot z_4 + \\ + m\cdot L\cdot \cos z_3\cdot b\cdot z_2\end{array}\right]}{\sigma} + \\ + \left[\frac{-m\cdot L\cdot \cos z_3}{\sigma}\right]\cdot [N_1u + N_2z_1 + N_3z_2 + N_4\dot{z}_2] \tag{4.116}$$

where $\sigma = (J + mL^2)\cdot(M+m) - m^2\cdot L^2\cdot \cos^2\theta$

Simplifying Equation (4.114) and Equation (4.116), then:

$$\dot{z}_1 = z_2 \tag{4.117}$$

$$\dot{z}_2 = \frac{\left[\begin{array}{l}-(J+mL^2)\cdot b\cdot z_2 - m^2\cdot L^2\cdot g\cdot \sin z_3\cdot \cos z_3 + m\cdot L\cdot b_t\cdot z_4\cdot \cos z_3 + \\ + (J+mL^2)\cdot m\cdot L\cdot z_4^2\cdot \sin z_3 + \\ + (J+m\cdot L^2)\cdot N_2\cdot z_1 + (J+m\cdot L^2)\cdot N_3\cdot z_2 + (J+m\cdot L^2)\cdot N_4\cdot \dot{z}_2\end{array}\right]}{\sigma} + \\ + \frac{(J+m\cdot L^2)}{\sigma}\cdot N_1\cdot u \tag{4.118}$$

$$\dot{z}_3 = z_4 \tag{4.119}$$

$$\dot{z}_4 = \frac{\left[\begin{array}{l}(M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 - m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 - (M+m)\cdot b_t\cdot z_4 + \\ + m\cdot L\cdot \cos z_3\cdot b\cdot z_2 + \\ + (-m\cdot L\cdot \cos z_3)\cdot N_2\cdot z_1 + (-m\cdot L\cdot \cos z_3)\cdot N_3\cdot z_2 + (-m\cdot L\cdot \cos z_3)\cdot N_4\cdot \dot{z}_2\end{array}\right]}{\sigma} + \\ + \left[\frac{-m\cdot L\cdot \cos z_3}{\sigma}\right]\cdot N_1\cdot u \tag{4.120}$$

These equations are very complex because of the $\dot{z}_2$ appearing in the expression for $\dot{z}_4$. If the motor inertia J_m is very small and can be neglected which means that $N_4 = -\frac{J_m}{r^2} = 0$, then:

$$F = N_1u + N_2z_1 + N_3z_2 \tag{4.121}$$

The Equations (4.117) to (4.120) can be simplified significantly as:

$$\dot{z}_1 = z_2 \tag{4.122}$$

$$\dot{z}_2=\frac{\left[\begin{array}{l}-\left(J+mL^2\right)\cdot\left(b-N_3\right)\cdot z_2-m^2\cdot L^2\cdot g\cdot\sin z_3\cdot\cos z_3+m\cdot L\cdot b_t\cdot z_4\cdot\cos z_3+\\+\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot\sin z_3+\left(J+m\cdot L^2\right)\cdot N_2\cdot z_1\end{array}\right]}{\sigma}+$$

$$+\frac{\left(J+m\cdot L^2\right)}{\sigma}\cdot N_1\cdot u \tag{4.123}$$

$$\dot{z}_3=z_4 \tag{4.124}$$

$$\dot{z}_4=\frac{\left[\begin{array}{l}\left(M+m\right)\cdot m\cdot g\cdot L\cdot\sin z_3-m^2\cdot L^2\cdot z_4^2\cdot\sin z_3\cdot\cos z_3-\left(M+m\right)\cdot b_t\cdot z_4+\\+m\cdot L\cdot\cos z_3\cdot b\cdot z_2-m\cdot L\cdot\cos z_3\cdot N_2\cdot z_1-m\cdot L\cdot\cos z_3\cdot N_3\cdot z_2\end{array}\right]}{\sigma}+$$

$$+\left[\frac{-m\cdot L\cdot\cos z_3}{\sigma}\right]\cdot N_1\cdot u \tag{4.125}$$

The nonlinear complete inverted pendulum mathematical model can be represented in the following way:

$$\dot{z}_1=z_2 \tag{4.126}$$

$$\dot{z}_2=\overline{f_1}(z)+\overline{g_1}(z)u \tag{4.127}$$

$$\dot{z}_3=z_4 \tag{4.128}$$

$$\dot{z}_4=\overline{f_2}(z)+\overline{g_2}(z)u \tag{4.129}$$

where:

$$\overline{f_1}(z)=\frac{\begin{array}{l}-\left(J+mL^2\right)\cdot\left(b-N_3\right)\cdot z_2-m^2\cdot L^2\cdot g\cdot\sin z_3\cdot\cos z_3+m\cdot L\cdot b_t\cdot z_4\cdot\cos z_3+\\+\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot\sin z_3+\left(J+m\cdot L^2\right)\cdot N_2\cdot z_1\end{array}}{\sigma}$$

$$\overline{f_2}(z)=\frac{\begin{array}{l}\left(M+m\right)\cdot m\cdot g\cdot L\cdot\sin z_3-m^2\cdot L^2\cdot z_4^2\cdot\sin z_3\cdot\cos z_3-\left(M+m\right)\cdot b_t\cdot z_4+\\+m\cdot L\cdot\cos z_3\cdot b\cdot z_2-m\cdot L\cdot\cos z_3\cdot N_2\cdot z_1-m\cdot L\cdot\cos z_3\cdot N_3\cdot z_2\end{array}}{\sigma}$$

$$\overline{g_1}(z)=\frac{\left(J+m\cdot L^2\right)}{\sigma}\cdot N_1$$

$$\overline{g_2}(z)=\frac{-m\cdot L\cdot\cos z_3}{\sigma}\cdot N_1$$

The inverted pendulum mathematical model in Equation (4.126) to (4.129) can be linearized when θ is a very small angle. When <5°θ<5°, sinθ≈0 and cosθ≈1, then:

$$\dot{z}_1=z_2 \tag{4.130}$$

$$\dot{z}_2 = \frac{\left[\begin{array}{l} -\left(J+mL^2\right)\cdot\left(b-N_3\right)\cdot z_2 - m^2\cdot L^2\cdot g\cdot z_3 + m\cdot L\cdot b_t\cdot z_4 + \\ +\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot z_3 + \left(J+m\cdot L^2\right)\cdot N_2\cdot z_1 \end{array}\right]}{\sigma'} + \frac{\left(J+m\cdot L^2\right)}{\sigma'}\cdot N_1\cdot u \qquad (4.131)$$

$$\dot{z}_3 = z_4 \qquad (4.132)$$

$$\dot{z}_4 = \frac{\left[\begin{array}{l} (M+m)\cdot m\cdot g\cdot L\cdot z_3 - m^2\cdot L^2\cdot z_4^2\cdot z_3 - (M+m)\cdot b_t\cdot z_4 + \\ +m\cdot L\cdot b\cdot z_2 - m\cdot L\cdot N_2\cdot z_1 - m\cdot L\cdot N_3\cdot z_2 \end{array}\right]}{\sigma'} + \left[\frac{-m\cdot L}{\sigma'}\right]\cdot N_1\cdot u \qquad (4.133)$$

where $\sigma' = J(M+m)+MmL^2$ (4.45)

If $\left(\dot{\theta}\right)^2 \approx 0$, then the complete inverted pendulum state space linear mathematical model can be expressed as:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\left(J+mL^2\right)N_2}{\sigma'} & -\frac{\left(J+mL^2\right)\cdot\left(b-N_3\right)}{\sigma'} & \frac{-m^2L^2g}{\sigma'} & \frac{mLb_t}{\sigma'} \\ 0 & 0 & 0 & 1 \\ \frac{-mLN_2}{\sigma'} & \frac{mL\left(b-N_3\right)}{\sigma'} & \frac{(M+m)\cdot mgL}{\sigma'} & -\frac{(M+m)\cdot b_t}{\sigma'} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{J+mL^2}{\sigma'}N_1 \\ 0 \\ -\frac{mL}{\sigma'}N_1 \end{bmatrix} u$$

(4.134)

4.5.2 Complete overhead crane mathematical model

Follow the same procedure of the complete inverted pendulum mathematical model derivation, the complete overhead crane mathematical model is derived by substituting Equation (4.121) into Equation (4.64) and Equation (4.66). Then, the complete overhead crane mathematical model is expressed as:

$$\dot{z}_1 = z_2 \qquad (4.135)$$

$$\dot{z}_2 = \frac{\left[\begin{array}{l} -\left(J+mL^2\right)\cdot b\cdot z_2 + m^2\cdot L^2\cdot g\cdot \sin z_3\cdot \cos z_3 + \\ +\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot \sin z_3 + m\cdot L\cdot b_t\cdot z_4\cdot \cos z_3 \end{array}\right]}{\sigma} + \qquad (4.136)$$
$$+\frac{\left(J+m\cdot L^2\right)}{\sigma}\cdot\left[N_1u+N_2z_1+N_3z_2\right]$$

$$\dot{z}_3 = z_4 \qquad (4.137)$$

$$\begin{aligned}\dot{z}_4 = {} & \frac{\begin{bmatrix} +m\cdot L\cdot\cos z_3\cdot b\cdot z_2-(M+m)\cdot m\cdot g\cdot L\cdot\sin z_3- \\ -m^2\cdot L^2\cdot z_4^2\cdot\sin z_3\cdot\cos z_3-(M+m)\cdot b_t\cdot z_4 \end{bmatrix}}{\sigma}+ \\ & +\frac{(-m\cdot L\cdot\cos z_3)}{\sigma}\cdot\left[N_1u+N_2z_1+N_3z_2\right]\end{aligned}\tag{4.138}$$

Simplifying the Equation (4.137) and (4.138), the overhead crane mathematical model can be expressed as:

$$\dot{z}_1 = z_2 \tag{4.139}$$

$$\dot{z}_2 = \frac{\begin{bmatrix} -\left(J+mL^2\right)\cdot\left(b-N_3\right)\cdot z_2+m^2\cdot L^2\cdot g\cdot\sin z_3\cdot\cos z_3+ \\ +\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot\sin z_3+m\cdot L\cdot b_t\cdot z_4\cdot\cos z_3+ \\ +\left(J+m\cdot L^2\right)\cdot N_2\cdot z_1 \end{bmatrix}}{\sigma}+\frac{\left(J+m\cdot L^2\right)}{\sigma}\cdot N_1\cdot u \tag{4.140}$$

$$\dot{z}_3 = z_4 \tag{4.141}$$

$$\dot{z}_4 = \frac{\begin{bmatrix} +m\cdot L\cdot\cos z_3\cdot b\cdot z_2-(M+m)\cdot m\cdot g\cdot L\cdot\sin z_3- \\ -m^2\cdot L^2\cdot z_4^2\cdot\sin z_3\cdot\cos z_3-(M+m)\cdot b_t\cdot z_4- \\ -m\cdot L\cdot\cos z_3\cdot N_2\cdot z_1-m\cdot L\cdot\cos z_3\cdot N_3\cdot z_2 \end{bmatrix}}{\sigma}+\frac{(-m\cdot L\cdot\cos z_3)}{\sigma}\cdot N_1u \tag{4.142}$$

The nonlinear complete overhead crane mathematical model can be represented in the following way:

$$\dot{z}_1 = z_2 \tag{4.143}$$

$$\dot{z}_2 = \overline{\overline{f_1}}(z)+\overline{\overline{g_1}}(z)u \tag{4.144}$$

$$\dot{z}_3 = z_4 \tag{4.145}$$

$$\dot{z}_4 = \overline{\overline{f_2}}(z)+\overline{\overline{g_2}}(z)u \tag{4.146}$$

where:

$$\overline{\overline{f_1}}(z)=\frac{\begin{gathered}-\left(J+mL^2\right)\cdot\left(b-N_3\right)\cdot z_2+m^2\cdot L^2\cdot g\cdot\sin z_3\cdot\cos z_3+\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot\sin z_3+ \\ +m\cdot L\cdot b_t\cdot z_4\cdot\cos z_3+\left(J+m\cdot L^2\right)\cdot N_2\cdot z_1\end{gathered}}{\sigma}$$

$$\overline{\overline{f_2}}(z)=\frac{\begin{gathered}+m\cdot L\cdot\cos z_3\cdot b\cdot z_2-(M+m)\cdot m\cdot g\cdot L\cdot\sin z_3-m^2\cdot L^2\cdot z_4^2\cdot\sin z_3\cdot\cos z_3- \\ -(M+m)\cdot b_t\cdot z_4-m\cdot L\cdot\cos z_3\cdot N_2\cdot z_1-m\cdot L\cdot\cos z_3\cdot N_3\cdot z_2\end{gathered}}{\sigma}$$

$$\overline{\overline{g}}_1(z) = \frac{\left(J + m \cdot L^2\right)}{\sigma} \cdot N_1$$

$$\overline{\overline{g}}_2(z) = \frac{\left(-m \cdot L \cdot \cos z_3\right)}{\sigma} \cdot N_1$$

After the linearization, the complete overhead crane mathematical model can be expressed as:

$$\dot{z}_1 = z_2 \tag{4.147}$$

$$\dot{z}_2 = \frac{\begin{bmatrix} -\left(J + mL^2\right) \cdot \left(b - N_3\right) \cdot z_2 + m^2 \cdot L^2 \cdot g \cdot z_3 + \\ +\left(J + mL^2\right) \cdot m \cdot L \cdot z_4^2 \cdot z_3 + m \cdot L \cdot b_t \cdot z_4 + \\ +\left(J + m \cdot L^2\right) \cdot N_2 \cdot z_1 \end{bmatrix}}{\sigma'} + \frac{\left(J + m \cdot L^2\right)}{\sigma'} \cdot N_1 \cdot u \tag{4.148}$$

$$\dot{z}_3 = z_4 \tag{4.149}$$

$$\dot{z}_4 = \frac{\begin{bmatrix} +m \cdot L \cdot b \cdot z_2 - (M + m) \cdot m \cdot g \cdot L \cdot z_3 - m^2 \cdot L^2 \cdot z_4^2 \cdot z_3 - \\ -(M + m) \cdot b_t \cdot z_4 - m \cdot L \cdot N_2 \cdot z_1 - m \cdot L \cdot N_3 \cdot z_2 \end{bmatrix}}{\sigma'} + \frac{(-m \cdot L)}{\sigma'} \cdot N_1 u \tag{4.150}$$

Then the complete overhead crane state space linear mathematical model can be expressed as:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\left(J + mL^2\right)N_2}{\sigma'} & -\frac{\left(J + mL^2\right) \cdot \left(b - N_3\right)}{\sigma'} & \frac{m^2L^2g}{\sigma'} & \frac{mLb_t}{\sigma'} \\ 0 & 0 & 0 & 1 \\ \frac{-mLN_2}{\sigma'} & \frac{mL\left(b - N_3\right)}{\sigma'} & -\frac{(M + m) \cdot mgL}{\sigma'} & -\frac{(M + m) \cdot b_t}{\sigma'} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{J + mL^2}{\sigma'} N_1 \\ 0 \\ -\frac{mL}{\sigma'} N_1 \end{bmatrix} u \tag{4.151}$$

Till now, all the mathematical models of the pendulum system are derived. This includes the inverted pendulum and overhead crane linear and nonlinear mathematical models, the servo system mathematical model and the integrated pendulum system mathematical models for the inverted pendulum and the overhead crane respectively. The servo system is used to convert control signal from its electrical form to a mechanical form.

The integrated pendulum system mathematical models are much complicated. This can complicate the controllers' designs. In the following controller design chapters, the inverted pendulum and overhead crane linear and nonlinear mathematical models are used directly without the servo system, in order to simplify the controller design process. The servo system is added after the design of the controllers, in order to convert the control signal from electrical form to mechanical form.

The inverted pendulum and overhead crane linear and nonlinear mathematical models are simulated in the following sections.

4.6 Pendulum system mathematical models simulation

The linear and nonlinear mathematical models of the inverted pendulum and the overhead crane are built and simulated in Matlab/Simulink environment. Equation (4.44) and (4.68) can be written in the following form:

$$\dot{z} = f(z) + g(z)u = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} f_1(z) \\ f_2(z) \\ f_3(z) \\ f_4(z) \end{bmatrix} + \begin{bmatrix} g_1(z) \\ g_2(z) \\ g_3(z) \\ g_4(z) \end{bmatrix} u \tag{4.152}$$

where $f_i(z)$ and $g_i(z)$, $i = \overline{1,4}$ are continuous nonlinear functions.

Then the nonlinear mathematical model can be constructed by using the nonlinear functions f(z) and g(z) whose Simulink block diagrams are shown in Figure 4.12 and 4.13. Here the inverted pendulum's nonlinear Simulink model is shown in Figure 4.14. The linear Simulink model is shown in Figure 4.15. For the overhead crane, the structures are the same; just two values in the function f have opposite signs.

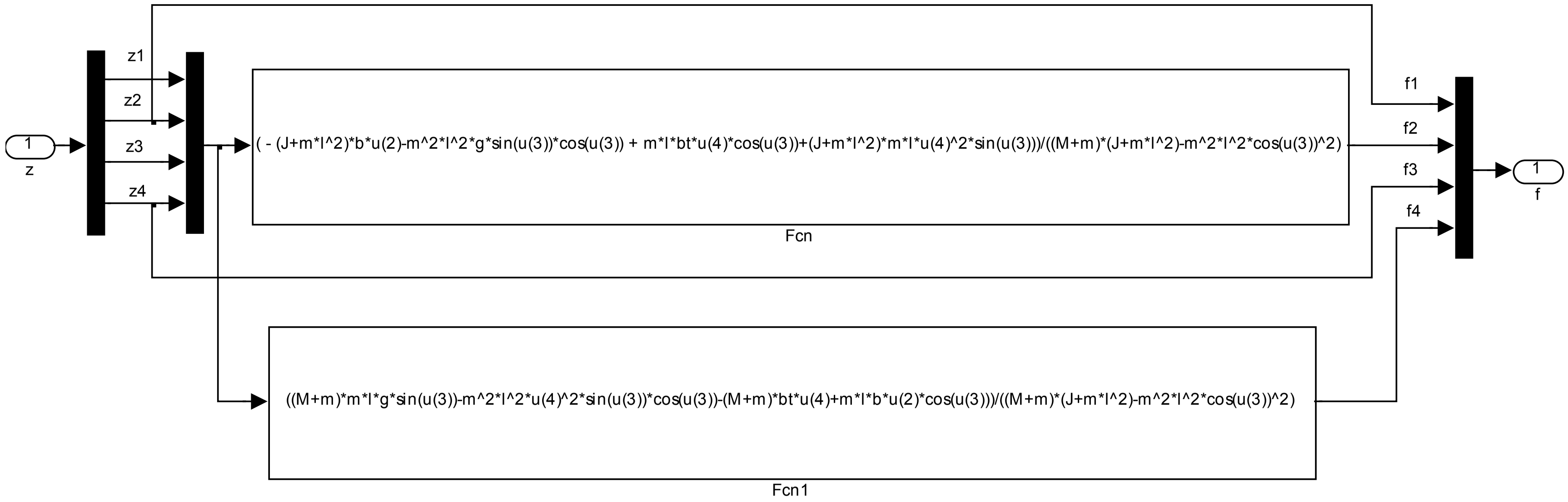

1
z
z1
z2
z3
z4
(- (J+m*l^2)*b*u(2)-m^2*l^2*g*sin(u(3))*cos(u(3)) + m*l*bt*u(4)*cos(u(3))+(J+m*l^2)*m*l*u(4)^2*sin(u(3)))/((M+m)*(J+m*l^2)-m^2*l^2*cos(u(3))^2)
Fcn
((M+m)*m*l*g*sin(u(3))-m^2*l^2*u(4)^2*sin(u(3))*cos(u(3))-(M+m)*bt*u(4)+m*l*b*u(2)*cos(u(3)))/((M+m)*(J+m*l^2)-m^2*l^2*cos(u(3))^2)
Fcn1
f1
f2
f3
f4
1
f

Figure 4.12: The nonlinear function f(z)

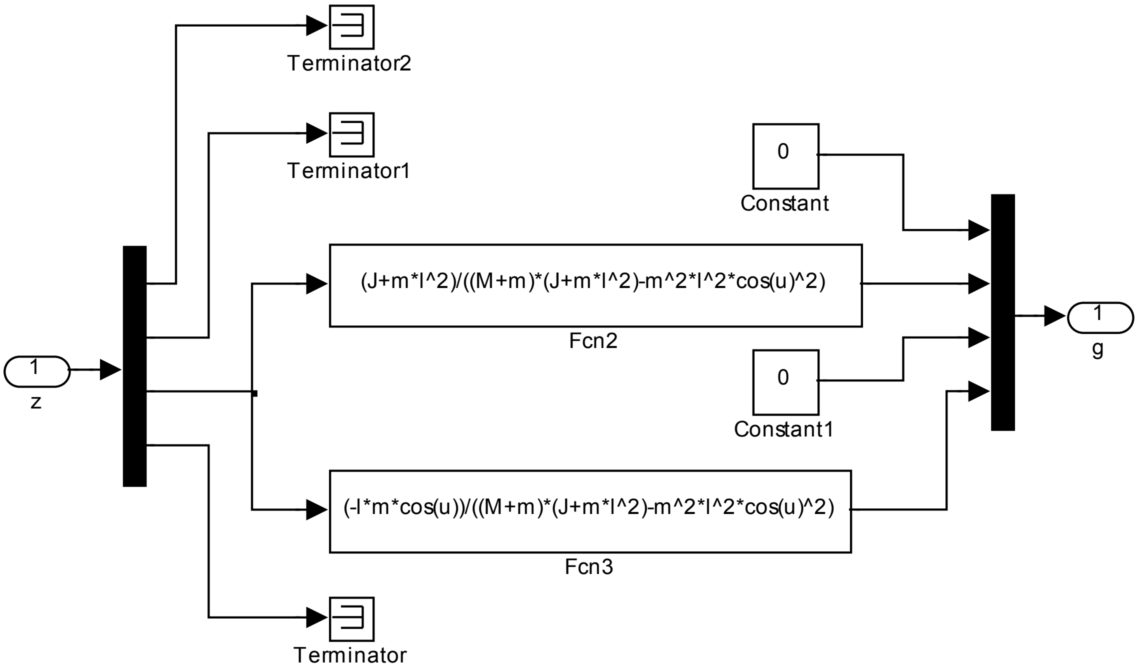

Terminator2
Terminator1
Constant
0
(J+m*l^2)/((M+m)*(J+m*l^2)-m^2*l^2*cos(u)^2)
Fcn2
1
z
1
g
0
Constant1
(-l*m*cos(u))/((M+m)*(J+m*l^2)-m^2*l^2*cos(u)^2)
Fcn3
Terminator

Figure 4.13: The nonlinear function g(z)

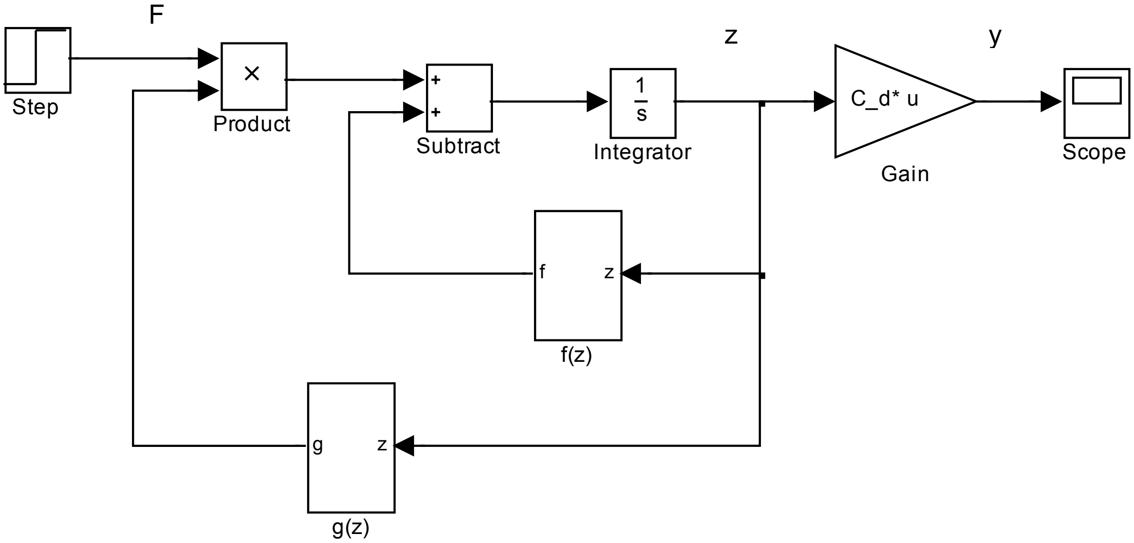


Figure 4.14: Inverted pendulum nonlinear model

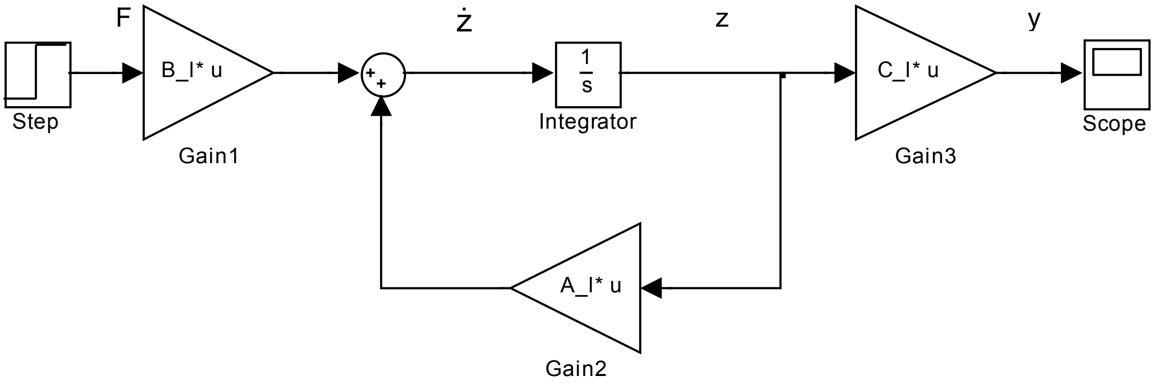


Figure 4.15: **Inverted pendulum linear model**

The inverted pendulum and the overhead crane share the same parameters. The parameters necessary for the calculation are introduced in Matlab workspace by the same program which is called "*model_test.m*". The corresponding Simulink files called "*model_test.mdl*". The m-file is given in the Appendix A. 1.

The simulation is given for a step input F=0.1m. The results are shown in Figure 4.16 to 4.25, where NLIP stands for Non-Linear Inverted Pendulum, NLOC for Non-Linear Overhead Crane. The results signals include the states signals and the output signals. The simulation results are discussed in the follow section.

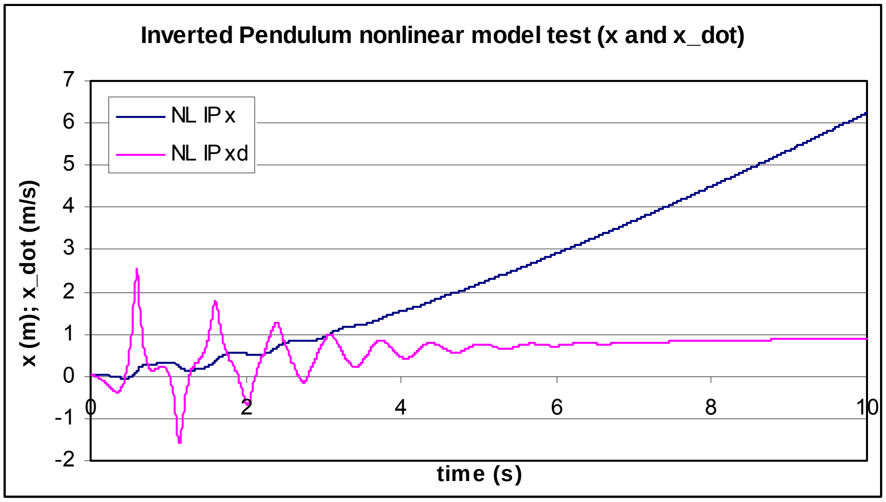


Figure 4.16: **Inverted pendulum nonlinear model states x, x_dot responses when the step input is 0.1m**

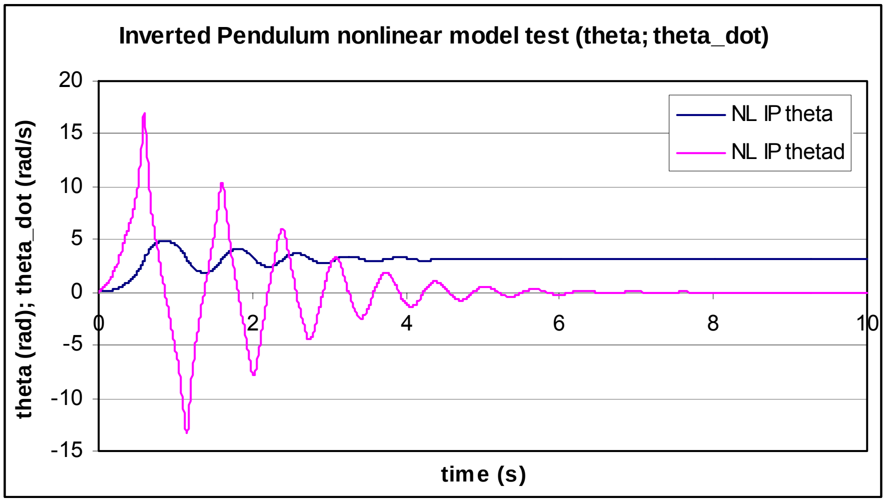


Figure 4.17: Inverted pendulum nonlinear model states theta, theta_dot responses when the step input is 0.1m

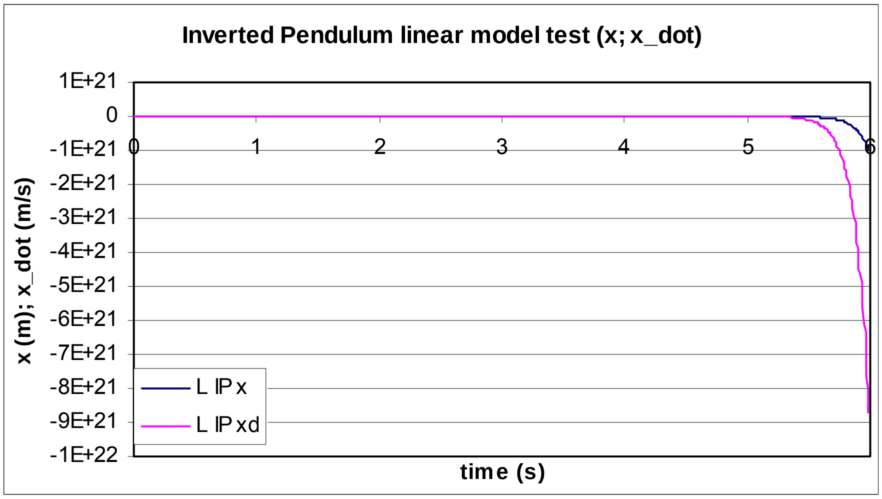


Figure 4.18: Inverted pendulum linear model states x, x_dot responses when the step input is 0.1m

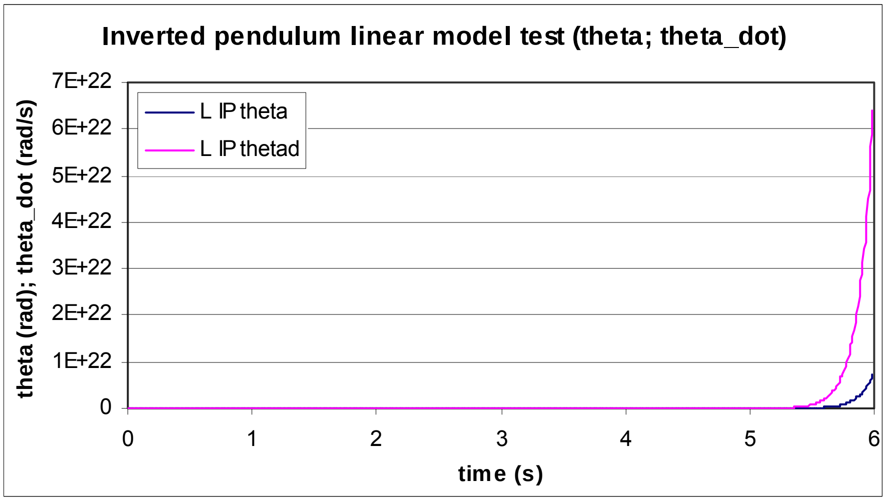


Figure 4.19: Inverted pendulum linear model states theta, theta_dot responses when the step input is 0.1

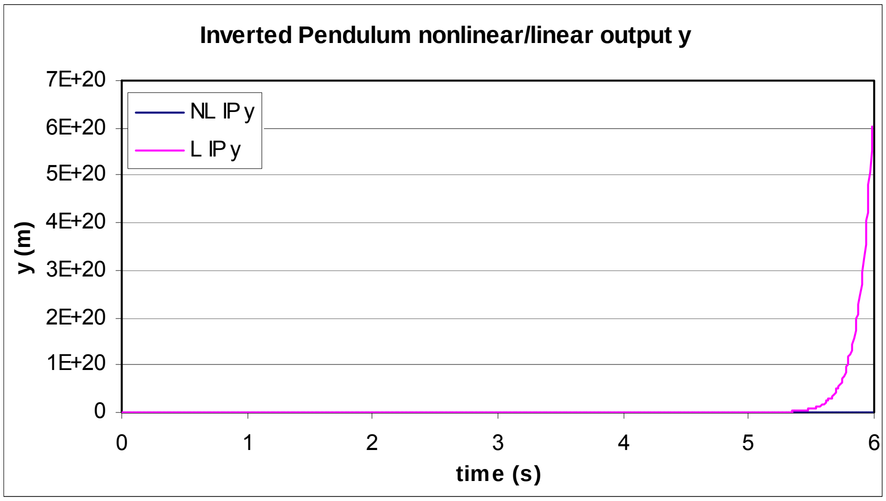


Figure 4.20: Inverted pendulum nonlinear output and linear output y when the step input is 0.1m

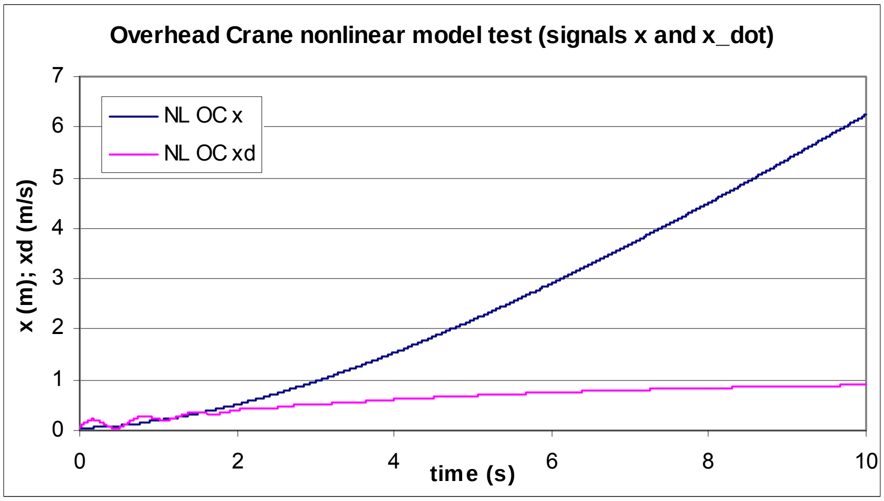


Figure 4.21: Overhead crane nonlinear model states x, x_dot responses when the step input is 0.1m

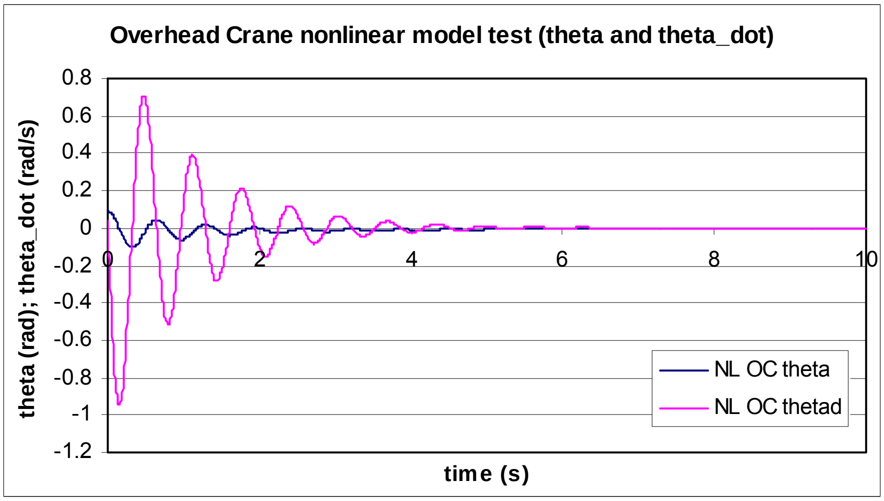


Figure 4.22: Overhead crane nonlinear model states theta, theta_dot responses when the step input is 0.1m

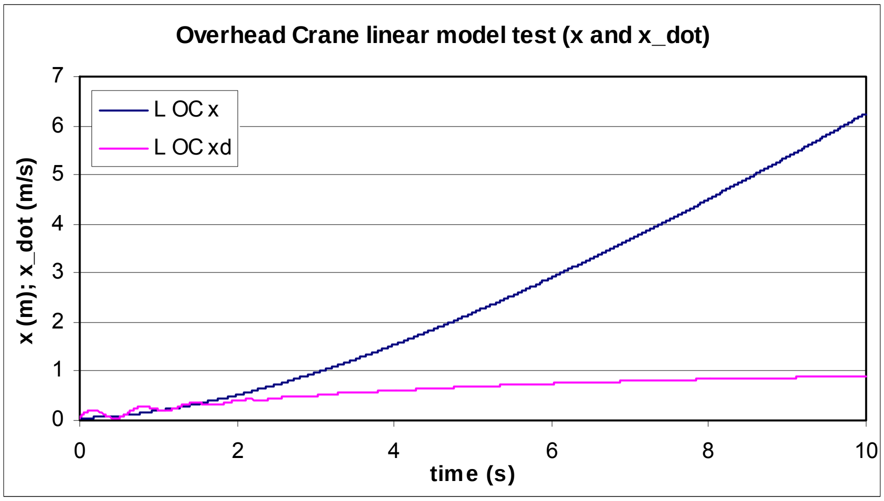


Figure 4.23: Overhead crane linear model states x, x_dot responses When the step input is 0.1m

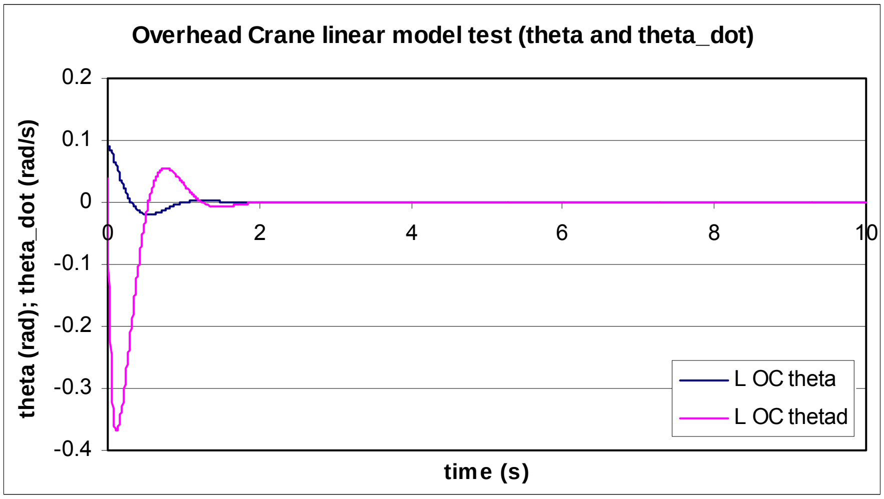


Figure 4.24: Overhead crane linear model states theta, theta_dot responses when the step input is 0.1

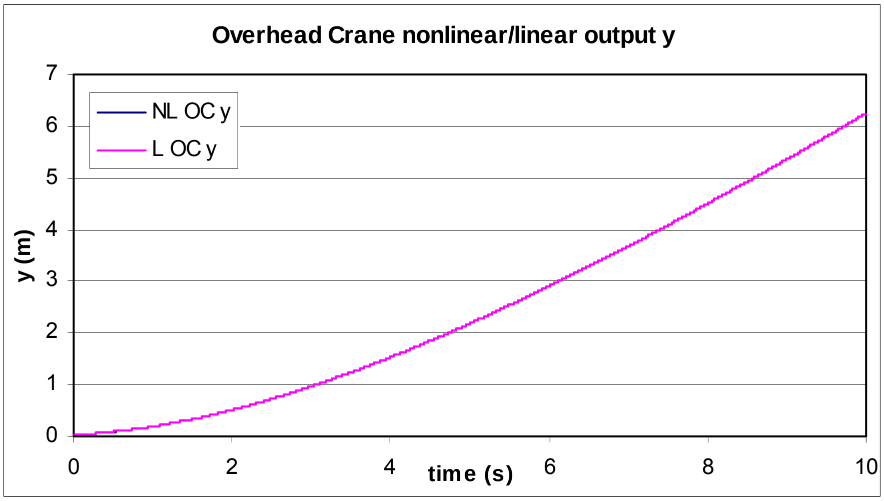


Figure 4.25: Overhead crane nonlinear output and linear output y when the step input is 0.1m

4.7 Discussion of the results

The behaviours of the inverted pendulum and the overhead crane are shown in Figure 4.16 to 4.25. Referring to the physical pendulum system, the above obtained results are discussed.

The linear model is a good approximation of the plant behaviour only around the non-stable equilibrium point $\theta=0$. As the plant is not under control and step continuous force is applied, the pendulum is moving far from the point $\theta=0$. Then the values of the position and the velocity of the position are becoming negative. This can be seen also from the linearized equations.

The position state response and velocity state response of the inverted pendulum and the overhead crane nonlinear models are shown in Figure 4.16 and 4.20. Under the constant force driven, eventually the cart moves with a constant acceleration in other words, the speed of the cart increases by a constant number towards to infinity in one direction.

For the angle state response and the angular velocity state response, the inverted pendulum and the overhead crane exhibit different behaviours. The inverted pendulum is inherently unstable. If there is no proper controller in the system, the pendulum falls, oscillates and stops at the position under the pivot. From Figure 4.17, it can be seen that the angle signal swings from its initial condition and settles at $\pi\approx3.14159$, which means the pendulum is at the down position. This position is the other equilibrium point of the system. In Figure 4.22, the overhead crane angle state swings from its initial condition and stops at 0 with a much small amplitude.

4.8 Conclusion

In this chapter, the nonlinear and linear mathematical models of the inverted pendulum and overhead crane are developed and simulated in Matlab/Simulink environment. From the simulation results, the behaviours of the system are understood deeper and wider.

Based on these mathematical models, the linear controllers and nonlinear controllers for the force F are developed. Different control strategies are used. In Chapter 5, the pole-placement method and quadratic optimal control method is deployed. The Lyapunov direct method based MRC and feedback linearization method based MRC are developed in Chapter 6 and Chapter 7 respectively. The real-time controllers for the armature control of the motor driving the pendulum are derived and implemented in Chapter 8.

CHAPTER FIVE
LINEAR CONTROL

5.1 Introduction

The mathematical models of the inverted pendulum and overhead crane are developed in the previous chapter. The task of the project is to develop nonlinear controllers based on the MRC theory. Two kinds of nonlinear control techniques are used, the Lyapunov direct method and the feedback linearization method. According to the theory of the MRC, it is necessary to create the reference models. For this project, two reference models are needed. One is used for the inverted pendulum control, the other for the overhead crane. These two reference models generate the desired trajectories which the actual system will follow under the nonlinear control signal. To complete the project, the first target now becomes to find the reference models.

For this project, linear models are selected as reference models. In order to find the reference models, the linearized plant is studied and the linear controllers are designed based on the pole-placement method and quadratic optimal control method. By studying the linear system, the reference models are derived and a better understanding of the plant is obtained.

This chapter is organized in the following way, in the section 5.1, the introduction gives the overview of the chapter and explains the purposes of studying the linearized plant. The pole-placement method and quadratic optimal regulator method are reviewed in the section 5.2. As the application of the controller design methods, the linear controllers are designed, based on different control techniques for both the inverted pendulum and the overhead crane, in the section 5.3. In this section, the feedback control matrices are calculated by one of the most used computation tools – Matlab/Simulink. In the section 5.4, the obtained results are discussed. The conclusion is presented in the section 5.5. The discussion of the results and the conclusion are given in the sections 5.4 and 5.5 respectively.

5.2 Review of the state space linear controller design methods

The linear controllers are designed by using the pole-placement method and the quadratic optimal regulator method. These two methods are very useful and powerful for the linear controller design. They have been used extensively as the time-domain methods. The controllers designed by using these two methods are in the form of a state feedback gain matrix K. In this chapter, these two methods are studied on the theory level, in terms of the controller derivation.

5.2.1 Pole-placement method

The root locus method is based on the transfer function in the complex domain. It is used to deal with the SISO systems. In the root locus method, only the dominant poles have the desired damping ratio and undamped natural frequency. The nondominant closed-loop are ignored. The pole-placement method, which is also called pole assignment method, is similar to the root locus method with the exception that in the pole-placement method all the closed-loop poles are placed to the desired locations rather than only the dominant closed-loop poles. The closed-loop poles can be placed at arbitrarily chosen locations with the precondition that the system must be completely state-controllable. (Ogata, 2002:827).

In many design problems, a system is stabilized through state feedback. Consider a control system as follows:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \qquad x(0) = x_0 \\ y(t) &= Cx(t) + Du(t)\end{aligned} \tag{5.1}$$

where $x(t) \in R^{n\times 1}$ is the state vector; y is the output scalar; $u \in R^{m\times 1}$ is the control vector. $A \in R^{n\times n}$ is the state matrix; $B \in R^{n\times 1}$ is the control matrix; $C \in R^{1\times n}$ is the output matrix and D is a constant, $x(0)$ is the initial state condition.

The control signal is chosen as:

$$u=-Kx(t) \tag{5.2}$$

where $K \in R^{1\times n}$ is the feedback gain matrix.

Substitute Equation (5.2) into Equation (5.1), then:

$$\dot{x}(t) = (A - BK)x(t) \tag{5.3}$$

Solve Equation (5.3), the solution is given as:

$$x(t) = e^{(A-BK)t}x(0) \tag{5.4}$$

The eigenvalues of the matrix A-BK determine the stability and transient response characteristics. If matrix K is selected appropriately, the matrix A-BK is asymptotically stable matrix and for $x(0)\neq 0$, it is possible to make $x(t)$ to approach zero as t approaches infinity. The eigenvalues of matrix A-BK are called the regulator poles. If these regulator poles are placed in the left-half s plane, then x(t) approaches zero as t approaches infinity (Ogata, 2002:829).

It can be proven that a necessary and sufficient condition for arbitrary pole placement is that the system must be completely state controllable. If the system is completely controllable, all eigenvalues can be arbitrarily placed by choosing the controller matrix according to the equation:

$$K = \begin{bmatrix} \delta_n & \delta_{n-1} & \cdots & \delta_1 \end{bmatrix} (\Gamma \Psi)^{-1} \tag{5.5}$$

where

a) $\Gamma = \begin{bmatrix} B & AB & \cdots & A^{n-1}B \end{bmatrix}$ is the controllability matrix;

b) $\Psi = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}$. The a_i's in Ψ are coefficients of the characteristic polynomial: $|sI - A| = s^n + a_1 s^{n-1} + \cdots + a_{n-1}s + a_n$.

c) In the preceding equation $\begin{bmatrix} \delta_n & \delta_{n-1} & \cdots & \delta_1 \end{bmatrix}$ can be solved from the system characteristic equation:

$|sI - T^{-1}AT + T^{-1}BKT| = 0$, if the parameters are substituted, then

$$s^n + (a_1 + \delta_1)s^{n-1} + \cdots + (a_{n-1} + \delta_{n-1})s + (a_n + \delta_n) = 0$$

There are other ways to find the feedback gain matrix K. They all have the same approach.

The above described pole-placement method derivation is for the system where the reference input is always zero. If the reference input to the system is always a constant which can be zero or a nonzero constant, the system is called a regulator system. With the help of the modern computation tools, the feedback gain matrix K can be easily calculated.

5.2.2 Quadratic optimal regulator method

There are several good reasons for developing the optimal control method. First of all, an advantage of the optimal control method over the pole-placement method is that the former provides a systematic way of computing the state feedback control gain matrix (Ogata, 2002:897). Secondly, in the pole-placement method the closed-loop pole locations must be determined, but the designer may not really know where they are located. The optimal control method avoids finding the desired pole locations. Thirdly, for the same system, there is not unique control law based on the pole-placement method. The designer may not know how to choose the best one. The control law of the optimal control method optimizes performance of the system in the precise sense, and the above difficulties are avoided (Friedland, 1986: 337).

In modern optimal control theory of linear deterministic dynamic systems, represented in continuous time by:

$$\dot{x}(t) = Ax(t) + Bu(t), \; x(0) = x_0 \tag{5.6}$$

the linear state feedback control signal is expressed as:

$$u(t) = -Kx(t) \tag{5.7}$$

where $u \in R^m$ is the vector of the control signal, $B \in R^{n \times m}$, $K \in R^{m \times n}$.

By optimizing the value for the feedback gain K, the following performance criterion which is also called performance index is minimized:

$$J_P = \int_0^{\infty} \left(x^T(t) Q x(t) + u^T(t) R u(t) \right) dt \tag{5.8}$$

where Q is a positive definite (or positive-semidefinite) Hermitian or real symmetric matrix and R is a positive-definite Hermitian or real symmetric matrix.

Matrices Q and R are the weighting matrices. They are often called the state weighting matrix and control weighting matrix, respectively (Friedland, 1986:339). Weighting matrices Q and R along with matrices A and B define the dynamic of the process. But it is impossible to predict the effect of the weighting matrices on a system. The suitable approach for finding the feedback gain matrix K would be as follows: a range of weighting matrices Q and R are chosen. Simulate the system with these selected weight matrices and examine the corresponding closed-loop response. The gain matrix K that generates the response closest to meeting the desired objective is the ultimate selection (Friedland, 1986:341).

The criterion is selected wisely. It requires minimization of “square” of input, which is the control signal. In other words, it minimizes the input energy required to control a given system. On the other hand, it requires minimization of the “square” of the state variables. That means the differences between the desired trajectories and actual system states trajectories are minimized. The minimization can be interpreted as the goal of bringing the system as close as possible to the desired trajectories while optimizing the energy usage (http://www.ece.rutgers.edu/~gajic/psfiles/lqr.pdf). For the regulation problem, the origin is zero. If the problem is to regulate the state variables to some constant values, the problem becomes set point control problem. This transformation to the regulation problem can be done easily by shifting the origin.

The solution of the optimal problem is determined by one of two dynamic optimization mathematical techniques: dynamic programming and calculus of variations. The solution is given in terms of the solution of the famous Riccati equation. It can be shown (http://www.ece.rutgers.edu/~gajic/psfiles/lqr.pdf) that the required optimal solution for the feedback gain is given by:

$$u(t) = -Kx(t) = -R^{-1}B^{T}Px(t) \qquad (5.9)$$

The symmetrical positive definite matrix $P \in R^{n \times n}$ must satisfy the following algebraic Riccati equation:

$$A^{T}P + PA - PBR^{-1}B^{T}P + Q = 0 \qquad (5.10)$$

Now the procedure of design a linear quadratic feedback controller can be summarized as follow:

- Solve the reduced-matrix Riccatie equation for the matrix P. If a positive-definite matrix P exists, the system is stable.
- Substitute this matrix P into the feedback gain Equation (5.9), the matrix K is the optimal matrix.

5.3 Linear controller design for the inverted pendulum and overhead crane

Based on the linearized mathematical models of the inverted pendulum in Equations (4.46) to (4.49) and the overhead crane in Equation (4.69) to (4.72), the linear controllers are designed and simulated. Here the pole-placement method and the quadratic optimal regulator method are used for the controllers design. The controllers are computed in Matlab and the closed loop system is simulated in Simulink.

5.3.1 State Space Pole-placement Method Controller Design

The control method is based on the pole-placement design techniques in state space. The linearized models of the inverted pendulum and overhead crane are used. This implies that the obtained stabilizing control will only ensure local stability in the vicinity of the upright equilibrium point, the point about which the equations were linearized.

The pole-placement method permits the design of a linear controller which achieves arbitrary desired closed-loop poles. The desired closed-loop poles should be chosen wisely so that the desired closed-loop characteristics are achieved, for example

- The settling time is approximately 2~3 seconds
- The maximum overshoot is 3%~5% in the step response of the cart position

The final control law, from this design, is the result of a matrix multiplication between the state vector z and a gain matrix of compatible dimension K, such that the control signal is

$u=-Kz(t)$

where $z \in R^{4\times1}$ is the state space vector of the inverted pendulum and the overhead crane

For this project, the aim is to:

- Keep the inverted pendulum upright as much as possible which is $z_3=\theta=0$ for $t\to\infty$ and
- Control the position of the cart according to a given set point y^{sp} for the output, for example, move the cart in a step manner.
- The process dynamic behaviour must meet the specification mentioned above

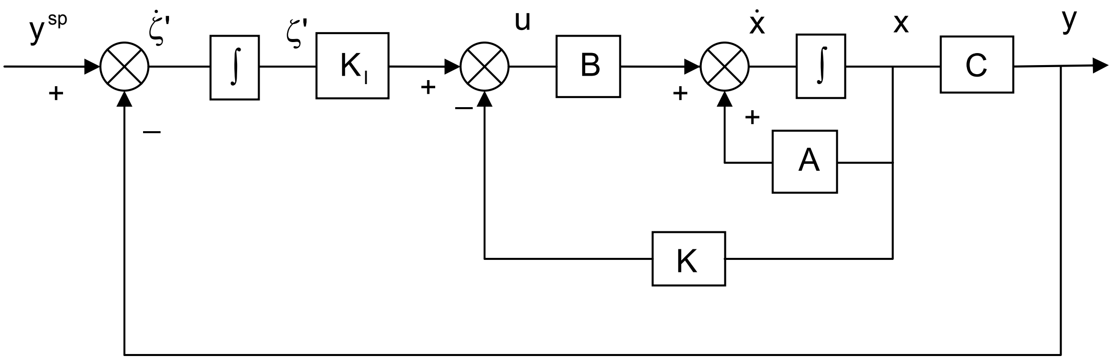


Figure 5.1: Type 1 servo system

To control the position of the cart, with zero steady state error according to the required set point y^{sp}, a type 1 servo system is needed (Ogata, 2002:850). There is no built in integrators in the inverted pendulum and the overhead crane. Both of them are type 0 plants. Therefore, the position signal y is sent back to the input to be compared with the set point and an integrator is used in the feed-forward path as shown in Figure 5.1. The output considered for both the inverted pendulum and the overhead crane is given by Equation (4.51).

The general approach of developing a servo system is presented shortly now. From the above block diagram, the following differential and algebraic equations are obtained:

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{5.11}$$

$$y(t) = Cx(t) \tag{5.12}$$

$$u(t) = -Kx(t) + K_I\zeta \tag{5.13}$$

$$\dot{\zeta}'(t) = y^{sp}(t) - y(t) = y^{sp}(t) - Cx(t),\ \zeta' \in R^l \tag{5.14}$$

where ζ' is the output of the integrator.

It can be seen there are two differential equations which can be grouped into one state space model as follow:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \zeta(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} y^{sp}(t) \tag{5.15}$$

It can be stated, for $y^{sp}(t)$ constant that the requirements for $y^{sp}(t)=y(t)$ are equivalent to the requirements that $x(\infty)$, $\zeta'(\infty)$, $u(\infty)$ approach constant values and in the steady state $\dot{\zeta}'(t) = 0$. Then $r(\infty)$ and $y(\infty)$ are constant.

At steady state the equation of the extended system (5.15) can be written as:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\zeta}'(t) \end{bmatrix} = \begin{bmatrix} A & 0_{n\times l} \\ -C & 0_{l\times l} \end{bmatrix} \begin{bmatrix} x(\infty) \\ \zeta'(\infty) \end{bmatrix} + \begin{bmatrix} B \\ 0_{l\times m} \end{bmatrix} u(\infty) + \begin{bmatrix} 0_{n\times l} \\ 1_{l\times l} \end{bmatrix} y^{sp}(\infty) \tag{5.16}$$

If y^{sp} is a step input applied at $t=0$, then for $t>0$ it is a constant. By subtracting Equation (5.16) from Equation (5.15) it is obtained:

$$\begin{bmatrix} \dot{x}(t) - \dot{x}(\infty) \\ \dot{\zeta}'(t) - \dot{\zeta}'(\infty) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) - x(\infty) \\ \zeta'(t) - \zeta'(\infty) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} [u(t) - u(\infty)] \tag{5.17}$$

After a new definition of variables:

$$x(t) - x(\infty) = x_e(t) \tag{5.18}$$

$$\zeta'(t) - \zeta'(\infty) = \zeta'_e(t) \tag{5.19}$$

$$u(t) - u(\infty) = u_e(t) \tag{5.20}$$

the Equation (5.17) can be written as:

$$\begin{bmatrix} \dot{x}_e(t) \\ \dot{\zeta}'_e(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x_e(t) \\ \zeta'_e(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_e(t)$$

(5.21)

where $u_e(t) = -Kx_e(t) + K_I\zeta_e(t)$ (5.22)

If the vector $\begin{bmatrix} x_e(t) \\ \zeta'_e(t) \end{bmatrix} = e(t)$, then the Equation (5.21) and (5.22) can be written as:

$$\dot{e}(t) = A_\Phi e(t) + B_\Phi u_e(t),\ e \in R^{n+l},\ u_e \in R^m \qquad (5.23)$$

$$u_e(t) = \hat{K}e(t) \qquad (5.24)$$

where $A_\Phi = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}$, $B_\Phi = \begin{bmatrix} B \\ 0 \end{bmatrix}$, $\hat{K} = \begin{bmatrix} K & K_I \end{bmatrix}$, $K \in R^{m\times(n+l)}$ (5.25)

Following the described general approach for design of the type 1 servo system above, a control servo system for the inverted pendulum and the overhead crane can be developed. For the considered case n=4, m=1, L=1. The matrix C=[1 0 L 0]. From the manual of the pendulum system “Document for the Bytronic Pendulum Control System”, it can be seen that inverted pendulum and overhead crane share the same natural frequency 6.16rad/s and damping ratio 0.0033 (Bytronic international LTD, 2001: 5.5).

In the manual a second order mathematical model G_p is given as:

$$G_p(s) = \frac{1}{1 + \frac{2\zeta}{\omega_n}s + \frac{1}{\omega_n^2}s^2} = \frac{1}{1 + 0.00122s + 0.0264s^2} \qquad (5.26)$$

In the denominator, the term 0.00122s is very small, so it is neglected. Then, the inverted pendulum mathematical model can be presented as:

$$G_p = \frac{1}{1 + 0.0264s^2} \qquad (5.27)$$

A suitable compensator G_c is given at the same time, which is presented as:

$$G_c(s) = \frac{1 + 0.1s}{1 + 0.01s} \qquad (5.28)$$

By using the given mathematical model and suitable compensator, the closed-loop system poles can be determined. The closed-loop transfer function G can be expressed as:

$$G = \frac{G_c(s)G_p(s)}{1 + G_c(s)G_p(s)} \qquad (5.29)$$

Substitute $G_p(s)$ and $G_c(s)$ into Equation (5.29), then:

$$G=\frac{\frac{1}{1+0.0264s^2}\cdot\frac{1+0.1s}{1+0.01s}}{1+\frac{1}{1+0.0264s^2}\cdot\frac{1+0.1s}{1+0.01s}}=\frac{1+0.1s}{\left(1+0.0264s^2\right)\left(1+0.01s\right)+1+0.1s} \qquad (5.30)$$

From the characteristic equation, the closed-loop poles can be determined by solving equation in Matlab:

$$(1+0.0264s^2)(1+0.01s)+1+0.1s=0 \qquad (5.31)$$

The results are:

-96.495614428933182941444876409678

-1.7521927855334085292775617951608-8.6855424835188583024103261365093*i

-1.7521927855334085292775617951608+8.6855424835188583024103261365093*i

For the overhead crane the closed-loop poles can be obtained by solving equation:

$$(1-0.0264s^2)(1+0.01s)+1+0.1s=0$$

The results are:

-103.32303041979847555614431132269

-10.383999162801621108324101821326

-7.0609687430031455521797904986407

The same closed-loop poles are placed at $z=p_i$, $(i=1, 2, 3, 4, 5)$, where p_1=-1.752+8.685j, p_2=-1.752-8.685j, p_3=-16, p_4=-17 and p_5=-100. Among these poles, poles p_1, p_2 and p_5 are chosen according to the given closed-loop poles from the manual. Poles p_3 and p_4 are located far left of the dominant pair of closed-loop poles and therefore, their effect over the overall response is minimal. After the poles are chosen, it is necessary to prove that the closed-loop systems are completely state controllable. This can be done in Matlab. By using command '*ctrb*', the controllability matrix can be calculated. If the matrix has full rank, then the system is controllable. The controller design follows the standard procedures. In the project, the calculations are done through Matlab. The function '*place'* is used to compute the feedback gain matrix $\hat{K}$. Please refer to Appendix A. 2 and Appendix A. 3 for the associated m-file "*PP_OP_IP.m*" and "*PP_OP_OC.m*".

The calculation results of the feedback gain for the inverted pendulum and the overhead crane respectively are:

$$\hat{K}=\begin{bmatrix}15.7724 & 3.0233 & -11.1434 & -0.1357 & -28.7641\end{bmatrix} \text{ (Inv. Pend.)} \qquad (5.26)$$

$$\hat{K}=\begin{bmatrix}-16.1362 & -3.9658 & -11.0047 & -1.7931 & 28.7641\end{bmatrix} \text{ (Crane)} \qquad (5.27)$$

These matrices are used for the simulations.

The simulation of the closed loop system is done in the Simulink environment. In the Figure 5.2, the Simulink block diagram "*Linear_PP_OP.mdl*" for the closed-loop system is shown. This Simulink block diagram is associated with the above mentioned two m-files. The first m-file "*PP_OP_IP.m*" is for the inverted pendulum. The second one "*PP_OP_OC.m*" is for the overhead crane. These two m-files are used to compute the feedback gain matrix $\hat{K}$ based on the pole-placement method. Once the calculation is done, all the parameters and the calculation results including the feedback gain matrices values are restored in the workspace of the Matlab. All the values that are necessary for running the Simulink block diagram can be found in the workspace. Once the Simulink block diagram starts running, it takes the necessary parameters from the workspace in order to complete the simulation.

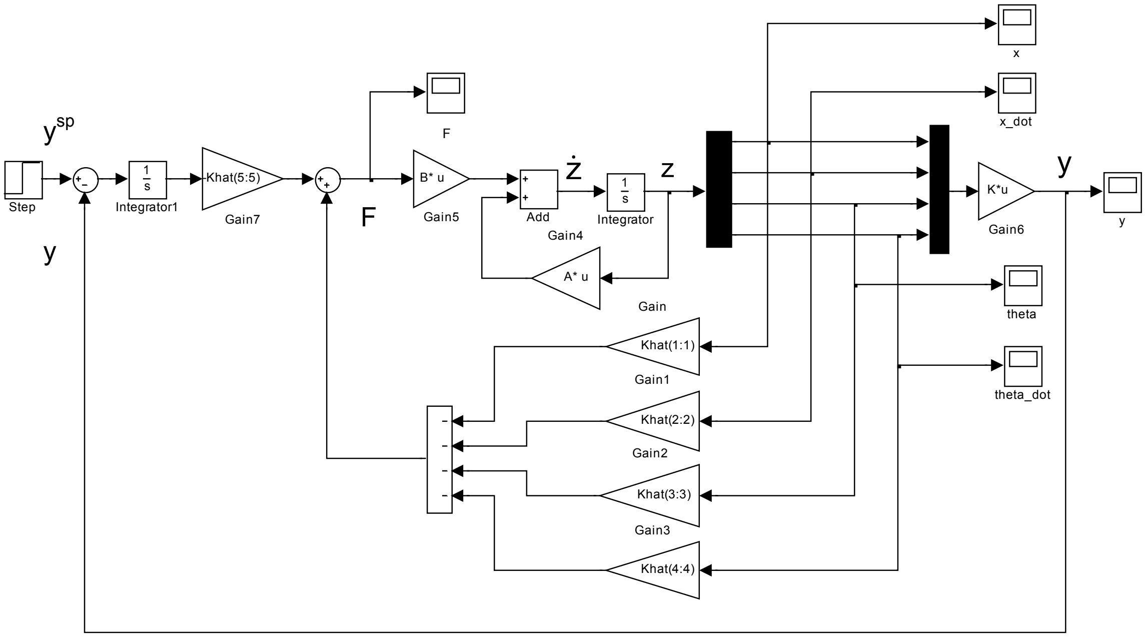


Figure 5.2: **Simulink block diagram for closed-loop simulation with controller design based on the pole-placement method**

The Simulink block diagram is built for one output y. It is constructed as: y=x+Lθ, where x is the cart position signal and θ is the angle signal which shows how much the pendulum is away from the vertical direction. There are five states x [m], $\dot{x}$ [m/s], θ [rad], $\dot{\theta}$ [rad/s] and ζ [m], the feedback gain matrix has dimension of 1 by 5. The first four elements of the matrix are multiplied with the first four states respectively. The fifth element is placed after the added integrator. The error signal is the difference between the set point and the actual output signal.

For the inverted pendulum and the overhead crane, the initial conditions are z(0)=[0.03 0.05 0.09 0.04] and z(0)=[0.03 0.05 0.35 0.04] respectively. These two vectors specify the initial values for the four states x, $\dot{x}$, θ and $\dot{\theta}$. The third states for specifying the initial angle values are different in these two vectors. For the inverted pendulum, the third state is 0.09 radian which means initially the pendulum is around 5 degree away from the vertical direction. For the overhead crane, the third value is 0.35 radian which corresponds the pendulum is 20 degree away from the vertical direction at the beginning. The set point (desired position) is set by three different values: 0.0m, 0.05m and 0.1m. The simulations are done by utilizing all of these three values. The signals shown below include the output signal y; the control signal F; the position signal x, the velocity signal $\dot{x}$, the angle signal θ and the angular velocity signal $\dot{\theta}$. These simulation results are shown in the figures following. They are organized in such a manner that: for each set point the signal y, u, x, $\dot{x}$, θ and $\dot{\theta}$ of the inverted pendulum and the overhead crane are presented.

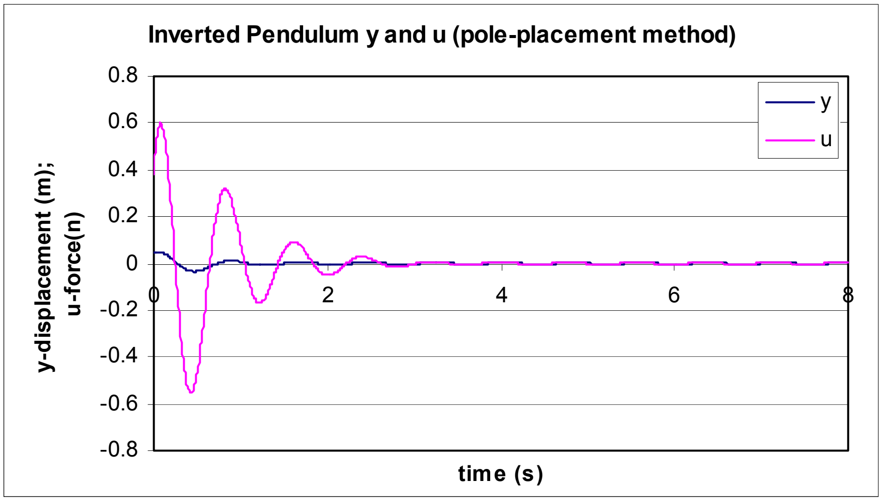


Figure 5.3: Inverted pendulum signal **y** and **F** when the set point is 0

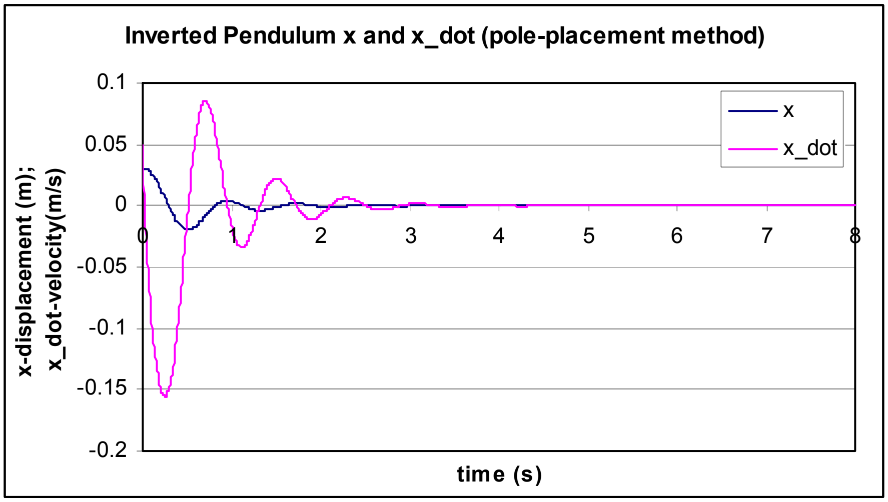


Figure 5.4: Inverted pendulum state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0

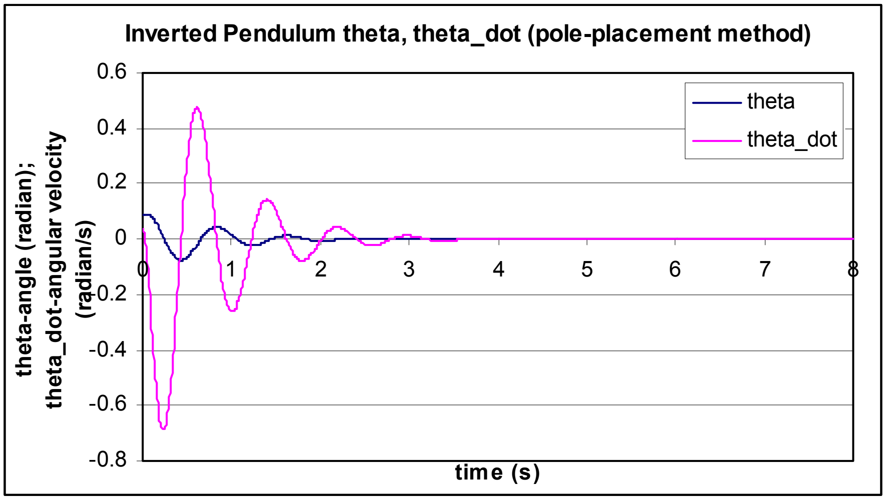


Figure 5.5: Inverted pendulum state signal $\boldsymbol{\theta}$ and $\dot{\theta}$ when the set point is 0

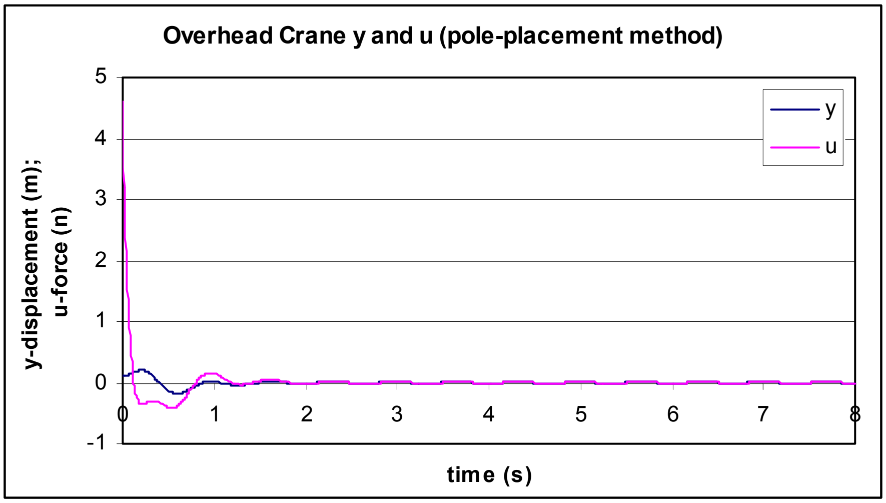


Figure 5.6: Overhead crane signal **y** and **F** when the set point is 0

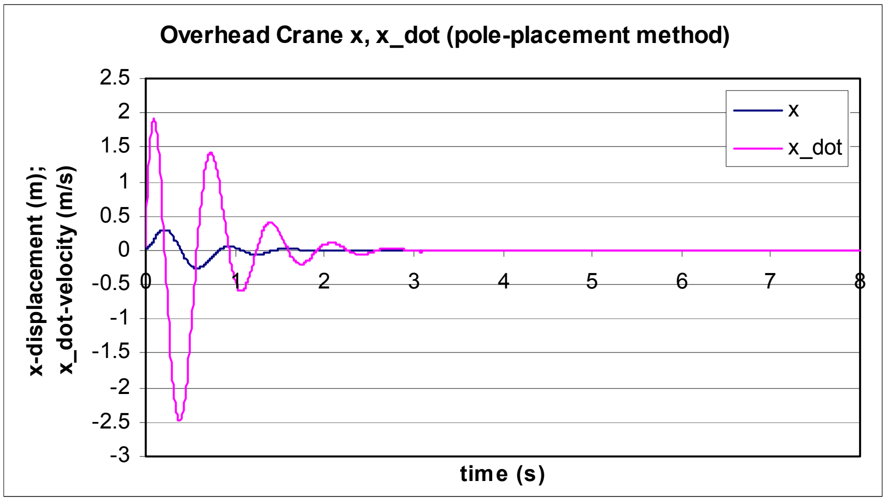


Figure 5.7: Overhead crane state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0

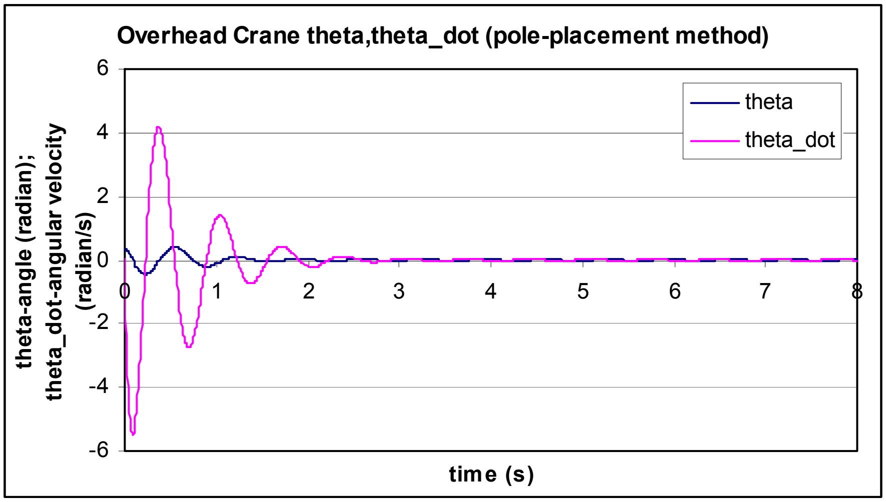


Figure 5.8: Overhead crane state signal $\boldsymbol{\theta}$ and $\dot{\theta}$ when the set point is 0

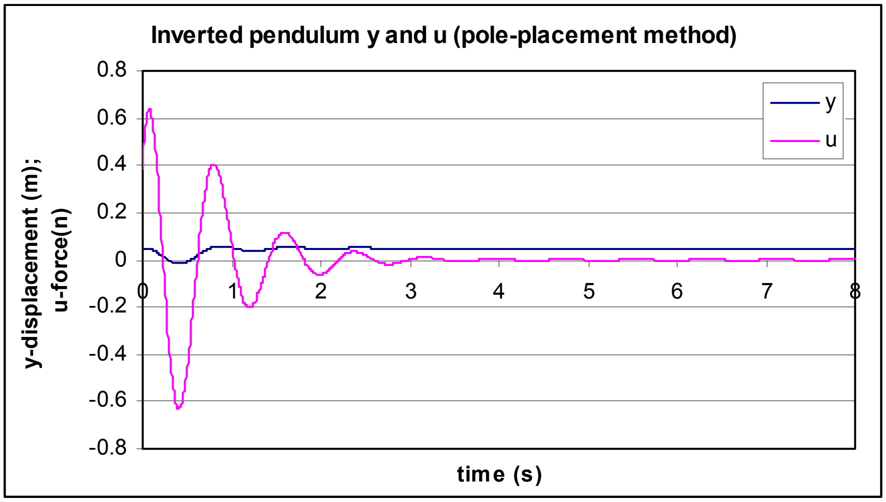


Figure 5.9: Inverted pendulum signal **y** and **F** when the set point is 0.05m

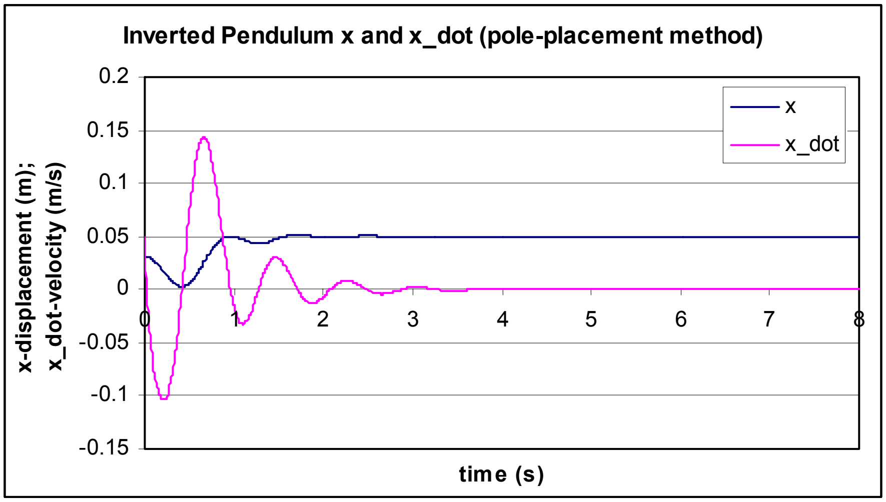


Figure 5.10: Inverted pendulum state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0.05m

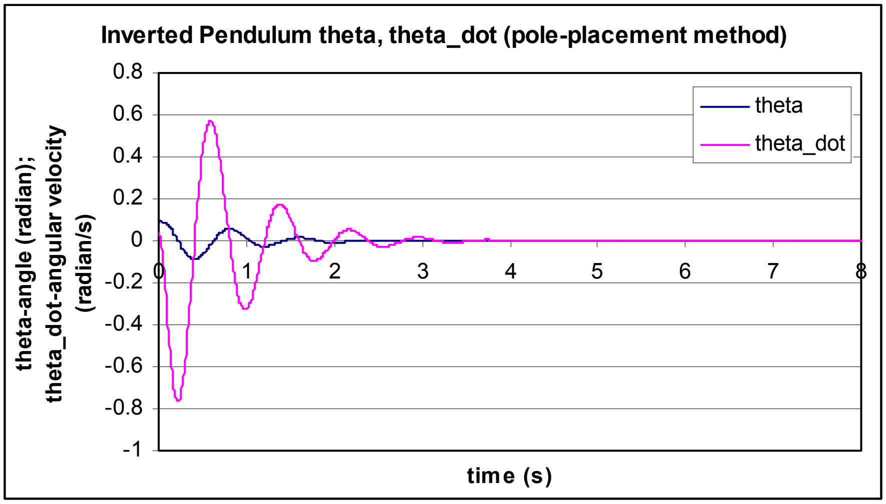


Figure 5.11: Inverted pendulum state signal $\boldsymbol{\theta}$ and $\dot{\theta}$ when the set point is 0.05m

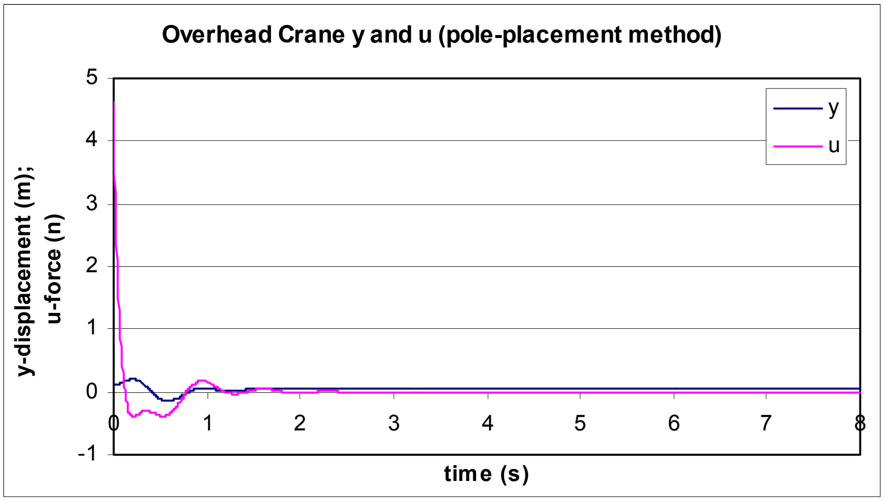


Figure 5.12: Overhead crane signal **y** and **F** when the set point is 0.05m

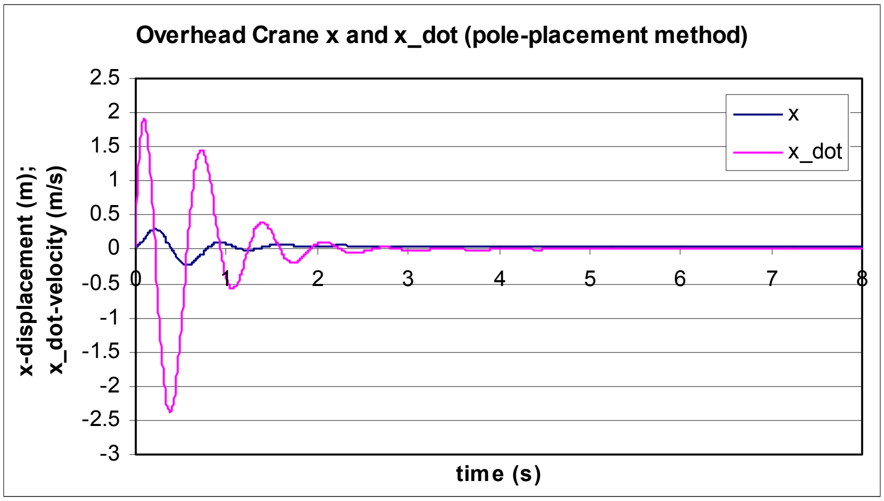


Figure 5.13: Overhead crane state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0.05m

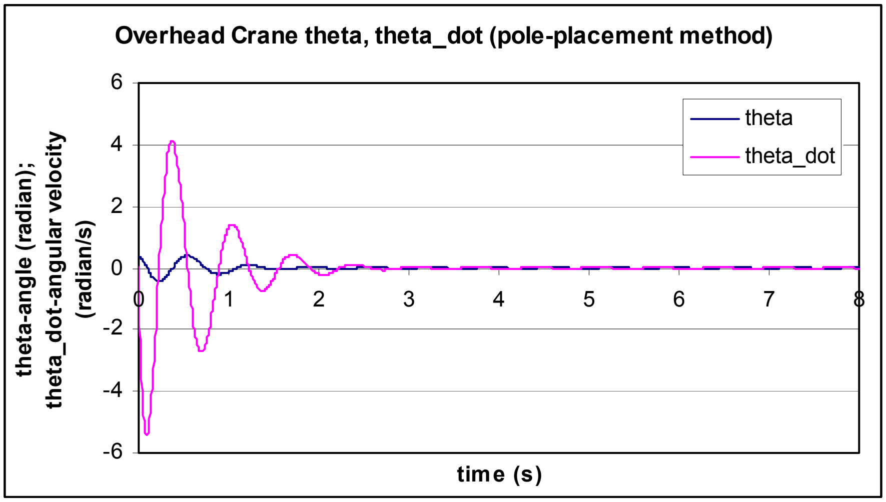


Figure 5.14: Overhead crane state signal $\boldsymbol{\theta}$ and $\dot{\theta}$ when the set point is 0.05m

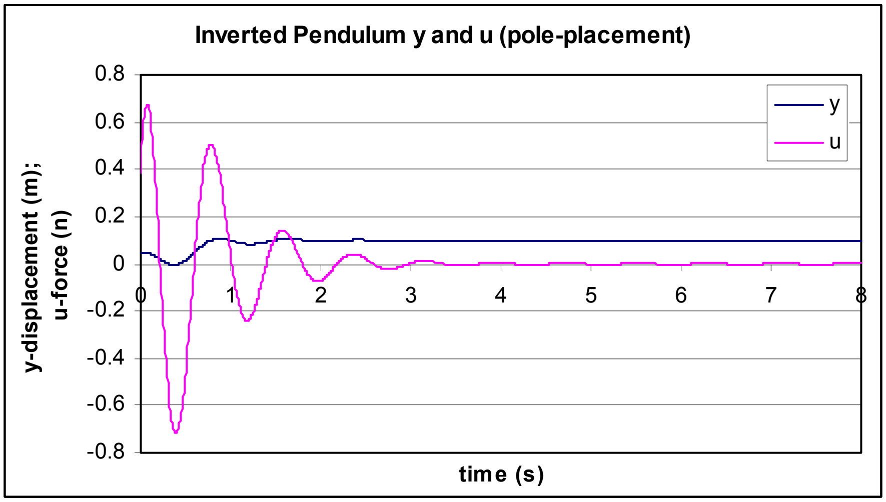


Figure 5.15: Inverted pendulum signal **y** and **F** when the set point is 0.1m

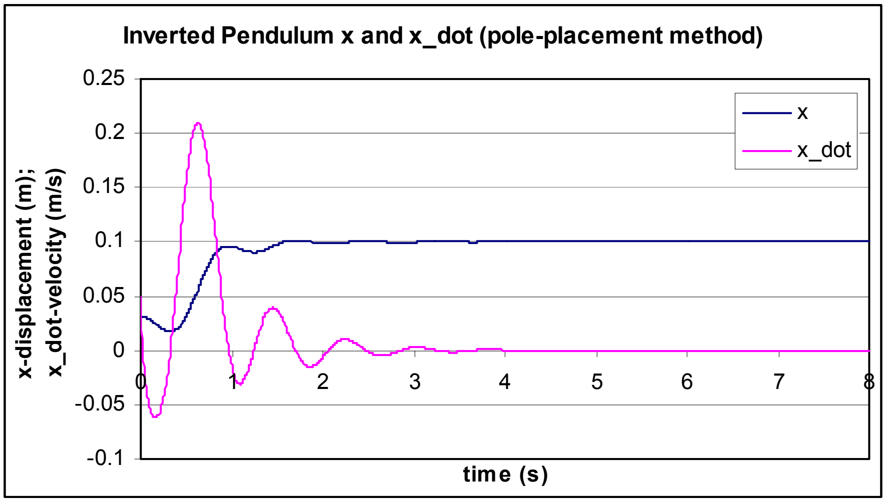


Figure 5.16: Overhead crane state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0.1m

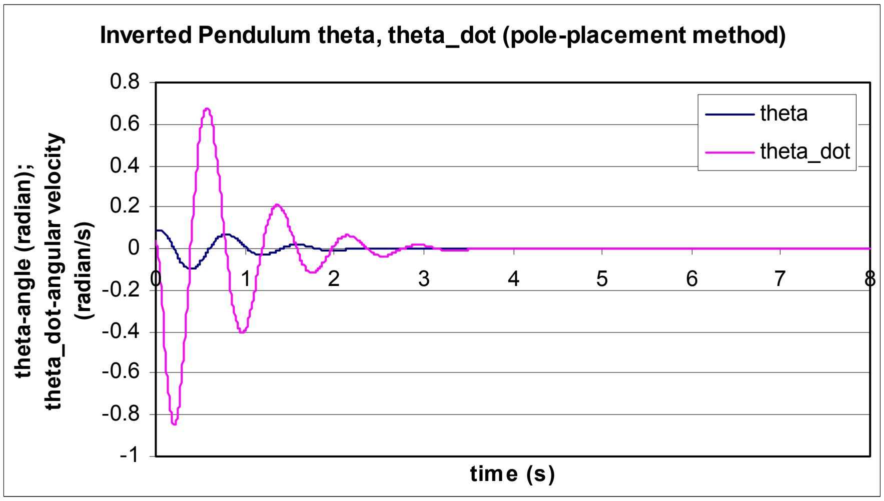


Figure 5.17: Inverted pendulum state signal $\boldsymbol{\theta}$ and $\dot{\theta}$ when the set point is 0.1m

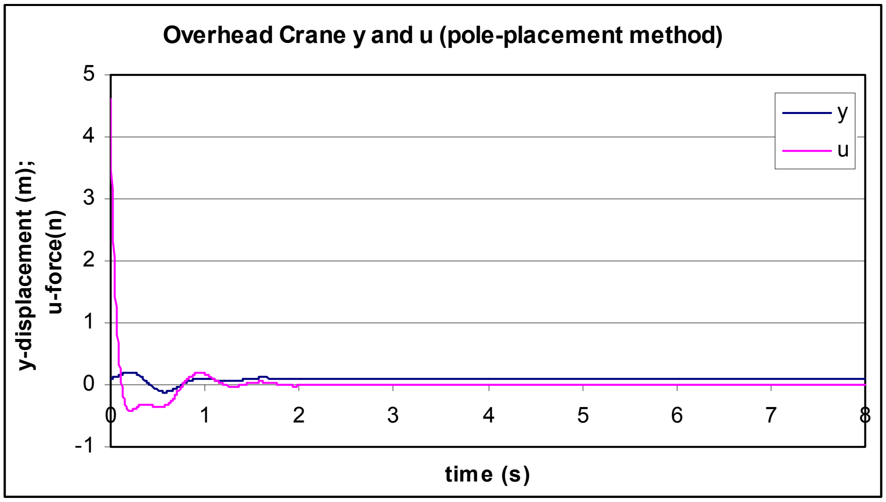


Figure 5.18: Overhead crane signal **y** and **F** when the set point is 0.1m

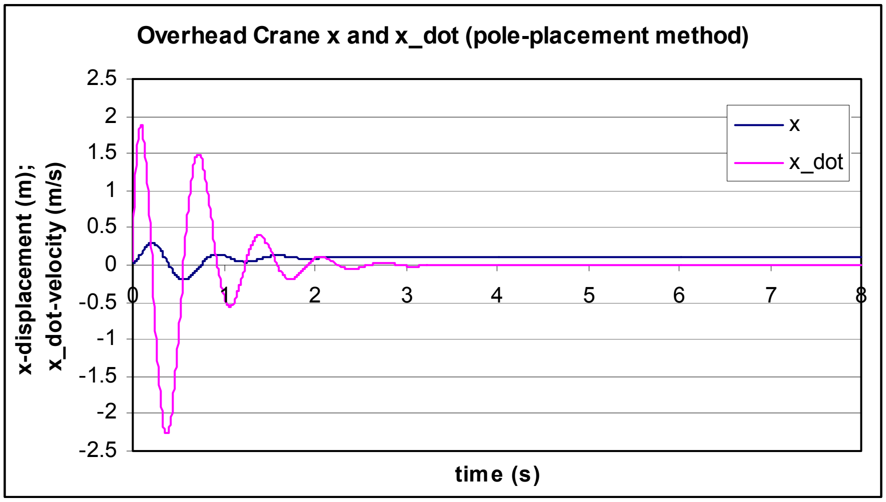


Figure 5.19: Overhead crane state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0.1m

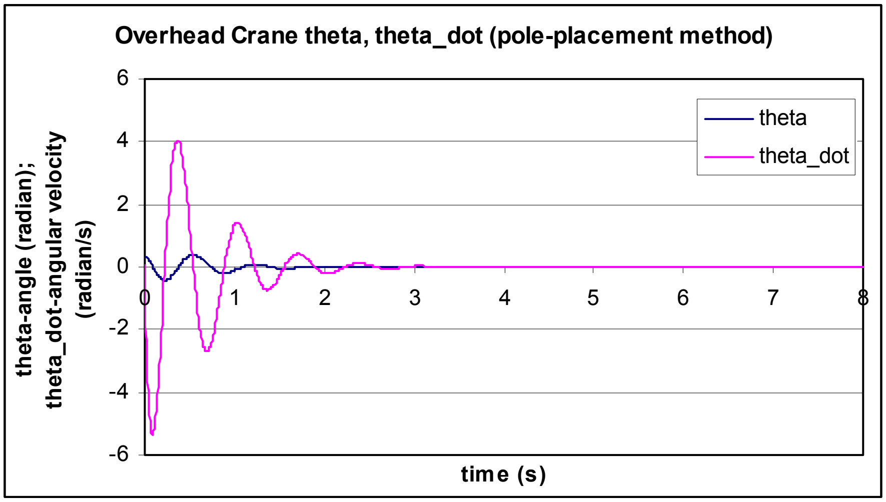


Figure 5.20: Overhead crane state signal **θ** and $\dot{\theta}$ when the set point is 0.1m

The controllers design based on pole-placement method has been done for the inverted pendulum and the overhead crane. The simulations have been done as well. The simulation results are shown above. In the next subsection, the linear quadratic optimal method control is used for the linear controller design.

5.3.2 Linear quadratic optimal controller design

This method is based on a LQR design using the linearized system (5.23) (5.24). In section 5.2.2, the method has been studied. In a LQR design, the gain matrix $\hat{K}$ for a linear state feedback control law $u(t) = -\hat{K}e(t)$ is found by minimizing the performance index in the form of:

$$J = \int_0^\infty e(t)^T Qe(t) + u(t)^T Ru(t)dt \qquad (5.28)$$

where $Q \in R^{5\times 5}$ is a positive definite (or positive semidefinite) Hermitian or real symmetric matrix. $R \in R^1$ is a positive definite Hermitian or real symmetric matrix. The matrices Q and R are weighting parameters that penalize certain states or control inputs and $y^{sp}=0$. In Matlab, the command “lqr” is used to solve the continuous-time, linear, quadratic regulator problem. This command computes the feedback gain matrix $\hat{K}$.

5.3.3 Set point control

For the LQR method, the matrices Q and R determine the relative importance of the error and the expenditure of the energy. They are selected based on the experience gained from the simulations. The following values are chosen for control of the inverted pendulum and overhead crane respectively:

$$Q_{ip}=\text{diag}\,\{8900 \quad 900 \quad 80000 \quad 100 \quad 50000\} \qquad R_{ip}=1 \qquad (5.29)$$

$$Q_{oc}=\text{diag}\,\{50000 \quad 500 \quad 10000 \quad 100 \quad 500000\} \qquad R_{oc}=1 \qquad (5.30)$$

The initial conditions and the set points for the different variables are the same as these used for the pole-placement method simulation. By using the method described above and the selected weighting matrices, the linear controllers are designed and applied to the linear mathematical models of the inverted pendulum and overhead crane. Based on this design, the controller gain matrix $\hat{K}$ for the linearized inverted pendulum and overhead crane are:

$$\hat{K} = [207.0768 \quad 75.8834 \quad -294.3823 \quad 4.6024 \quad -223.6068] \text{ (Inv. Pend.)} \qquad (6.6)$$

$$\hat{K} = [-462.785 \quad -116.1857 \quad -414.2684 \quad -39.8377 \quad 707.1068] \text{ (Crane)} \qquad (6.7)$$

For the simulation, the same Simulink block diagram “*Linear_PP_OP.mdl*” is used, but it is also associated with these two m-files which are called “*PP_OP_IP.m*” and “*PP_OP_OC.m*”. These m-files contain the commands and parameters for the quadratic optimal controllers design. For the detail of the m-files, please refer to Appendix A. 2 and Appendix A. 3.

The simulation results are shown in the figures below. The signals shown below include the output signal y, the position signal x, the velocity signal $\dot{x}$, the angle signal θ and the angular velocity signal $\dot{\theta}$. They are arranged in the same order as mentioned above for the inverted pendulum and overhead crane, respectively.

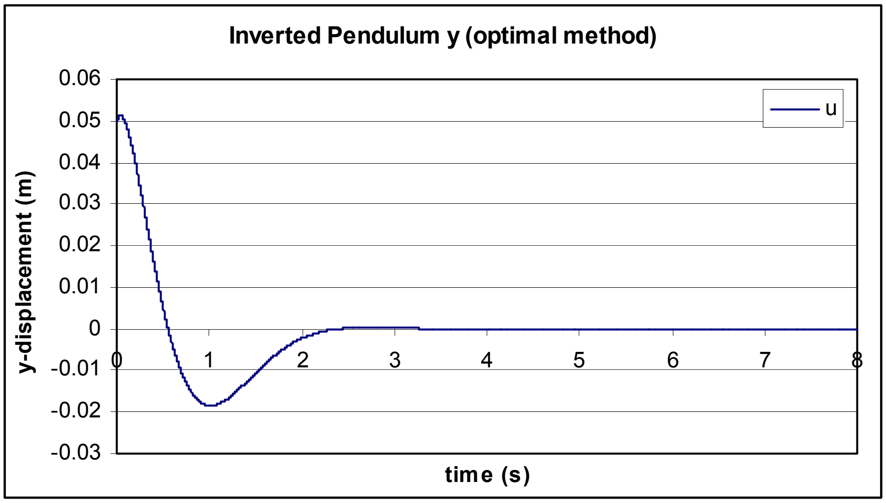


Figure 5.21: Inverted pendulum output signal **y** when the set point is 0

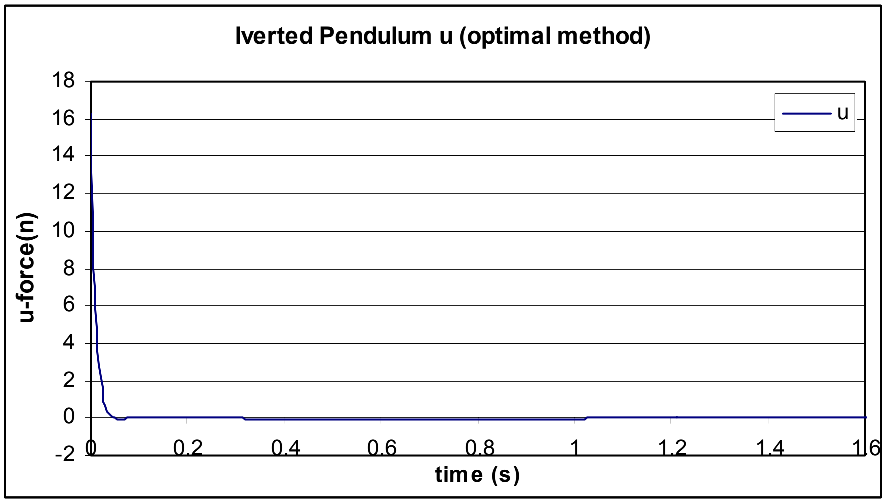


Figure 5.22: Inverted pendulum control signal **F** when the set point is 0

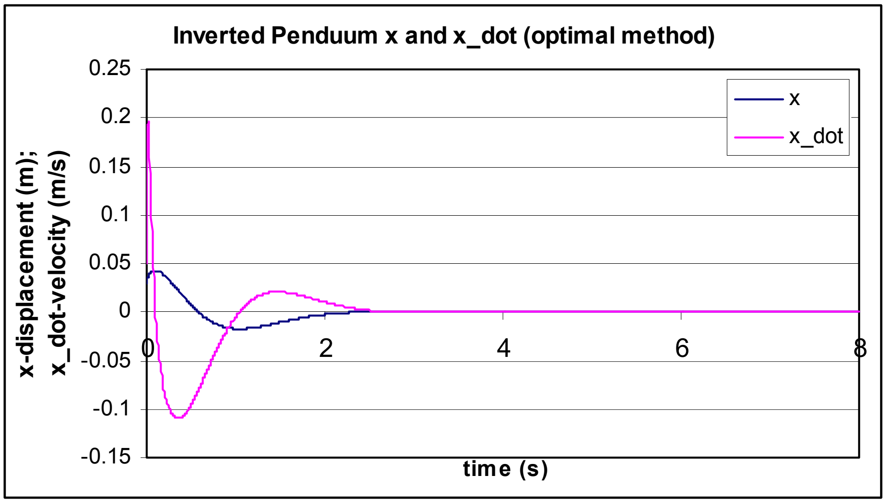


Figure 5.23: Inverted Pendulum state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0

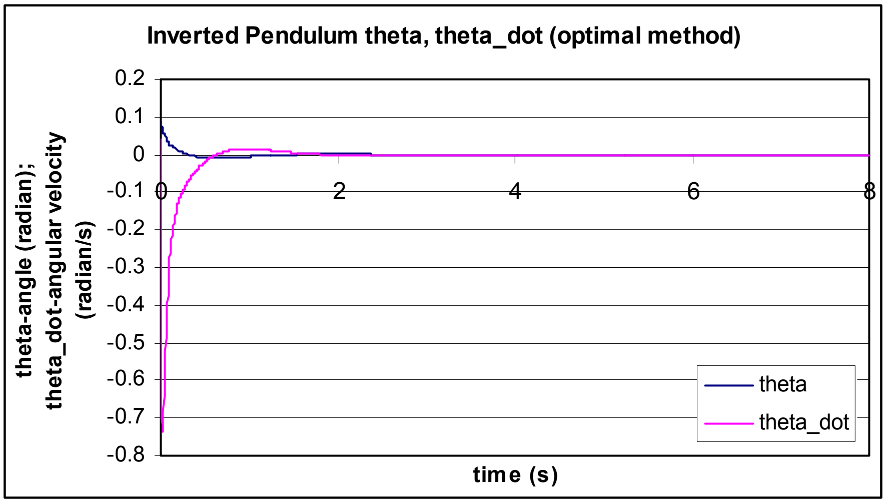


Figure 5.24: Inverted pendulum state signal **θ** and $\dot{\theta}$ when the set point is 0

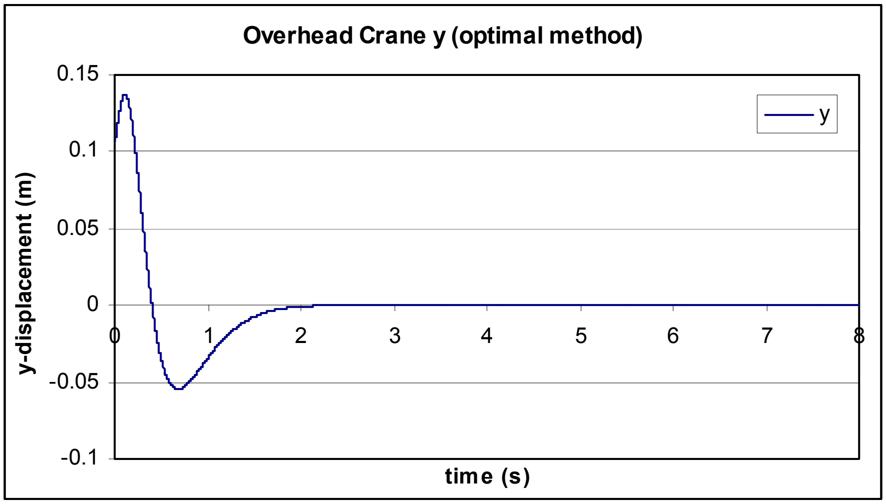


Figure 5.25: Overhead crane output signal **y** when the set point is 0

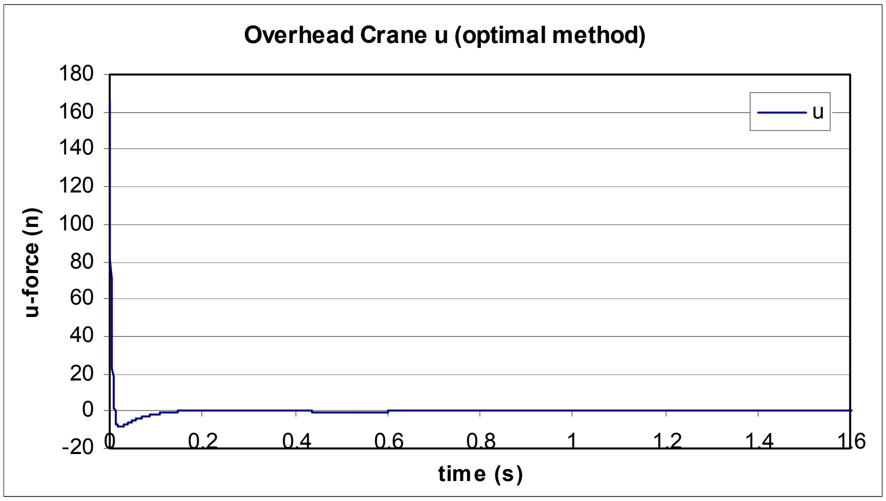


Figure 5.26: Overhead crane control signal **F** when the set point is 0

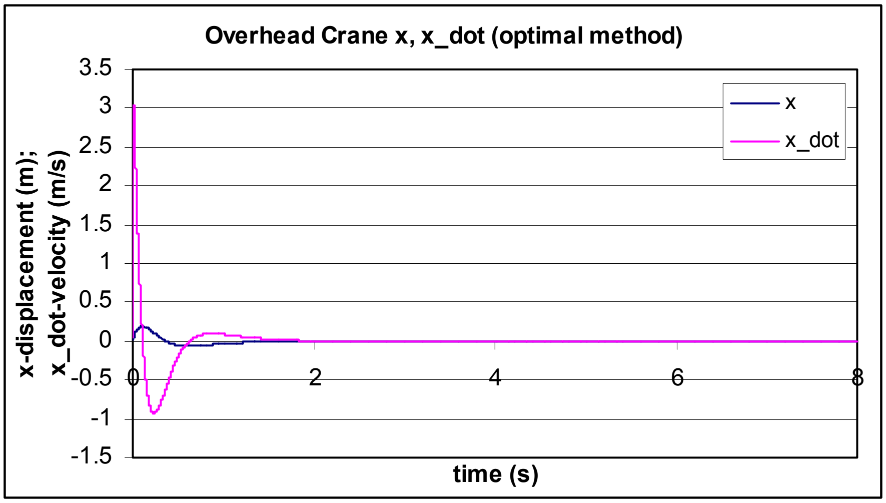


Figure 5.27: Overhead crane state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0

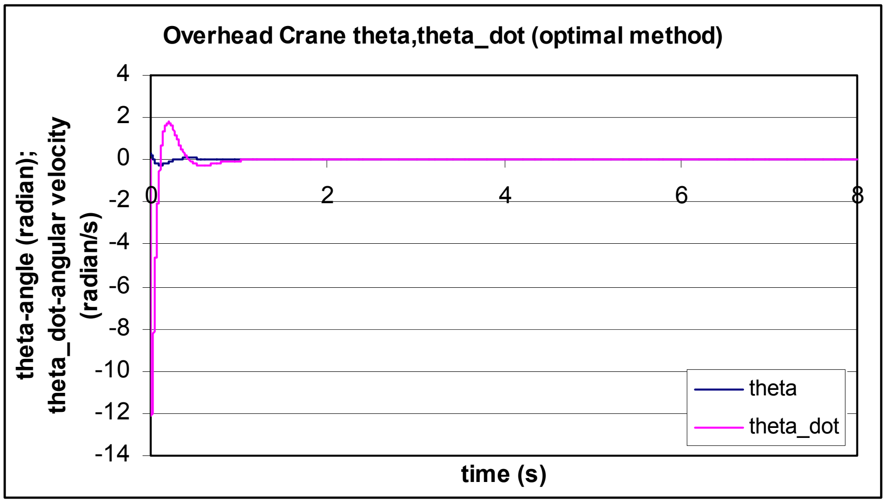


Figure 5.28: Overhead crane state signal $\boldsymbol{\theta}$ and $\dot{\theta}$ when the set point is 0

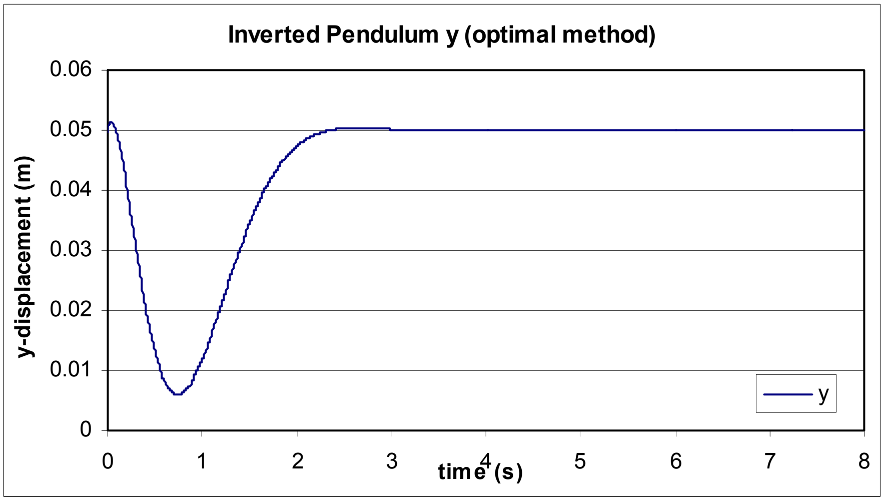


Figure 5.29: Inverted pendulum output signal **y** when the set point is 0.05m

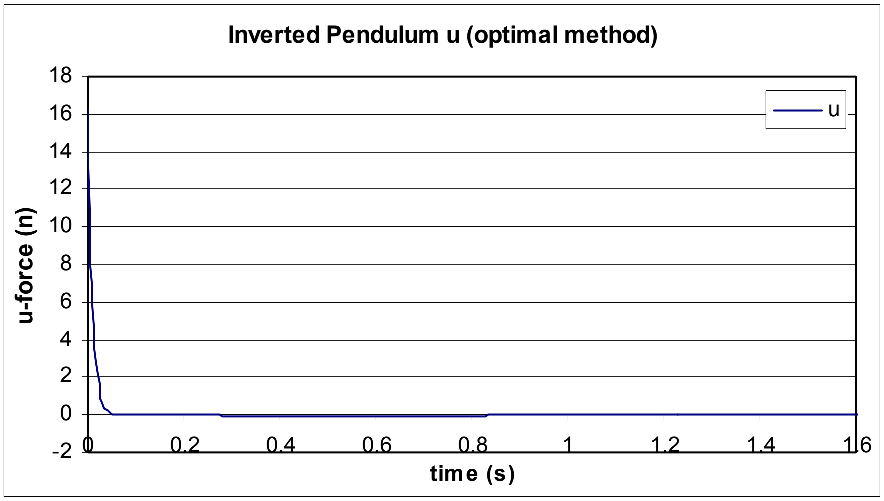


Figure 5.30: Inverted pendulum control signal **F** when the set point is 0.05m

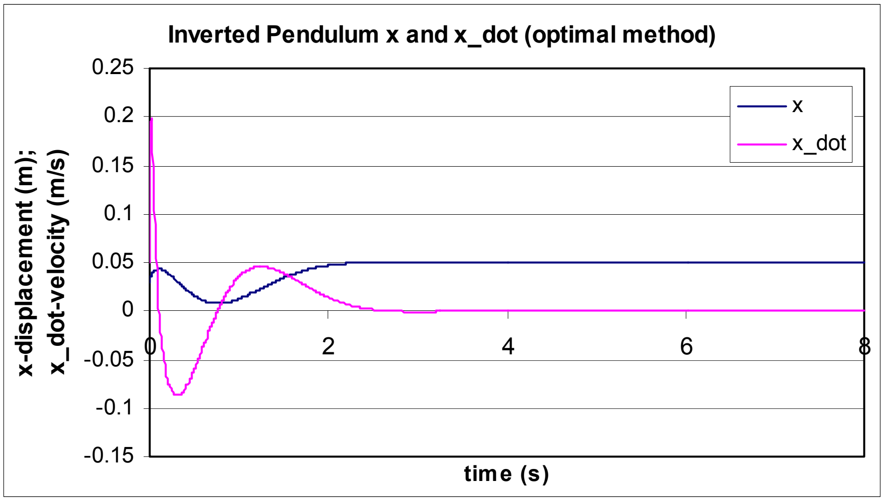


Figure 5.31: Inverted Pendulum state signal **x** and $\dot{x}$ when the set point is 0.05m

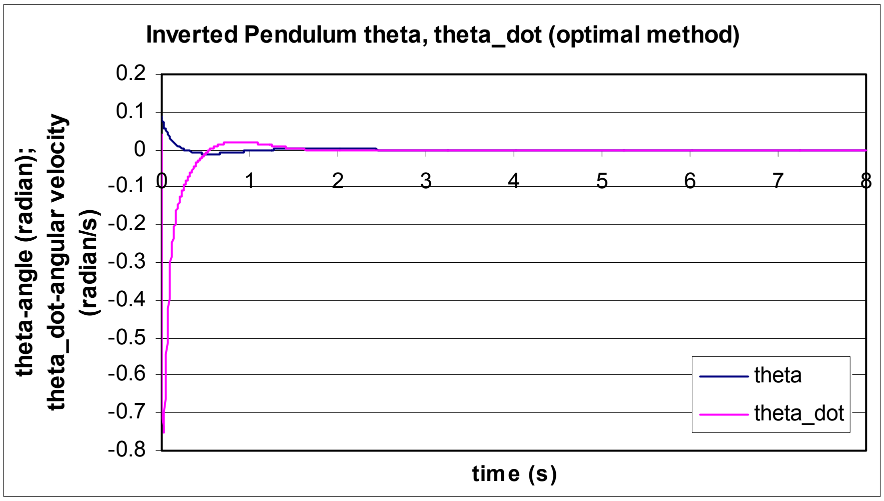


Figure 5.32: Inverted pendulum state signal θ and $\dot{\theta}$ when the set point is 0.05m

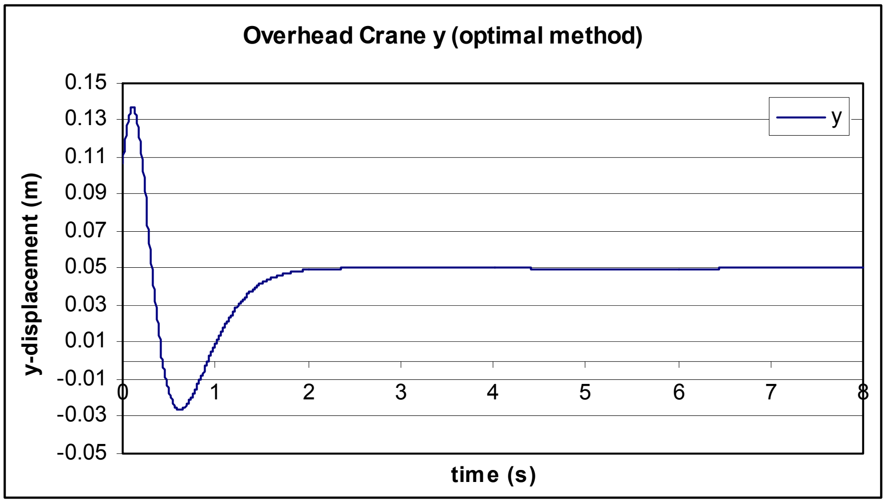


Figure 5.33: Overhead crane output signal **y** when the set point is 0.05m

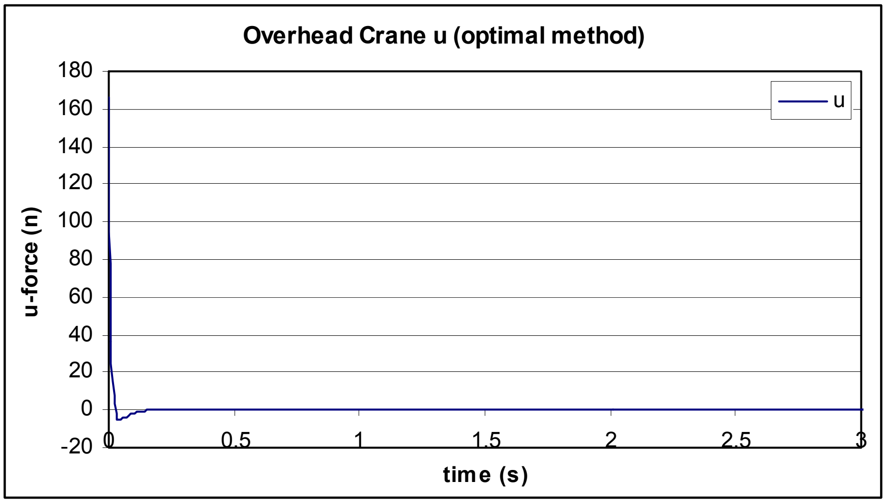


Figure 5.34: Overhead crane control signal **F** when the set point is 0.05m

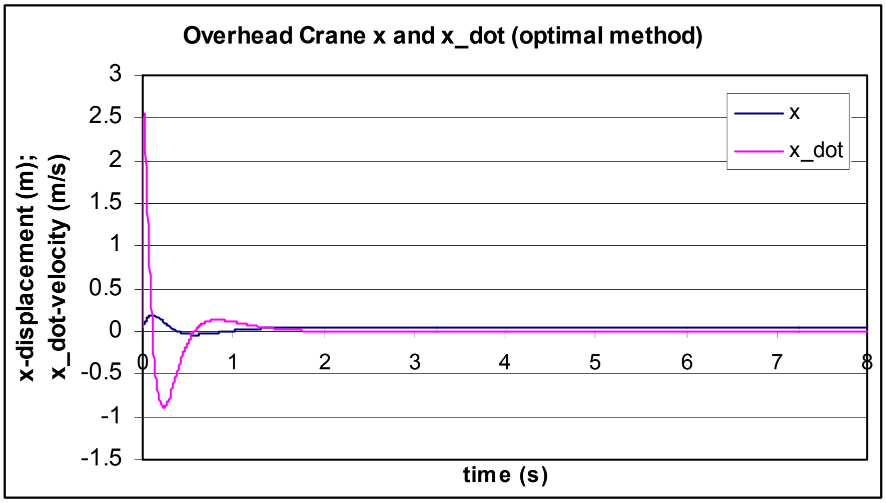


Figure 5.35: Overhead crane state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0.05m

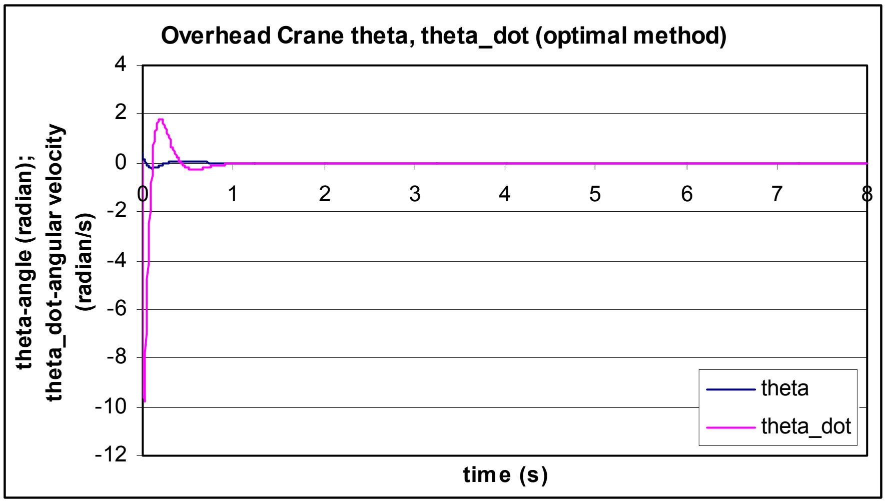


Figure 5.36: Overhead crane state signal **θ** and $\dot{\theta}$ when the set point is 0.05m

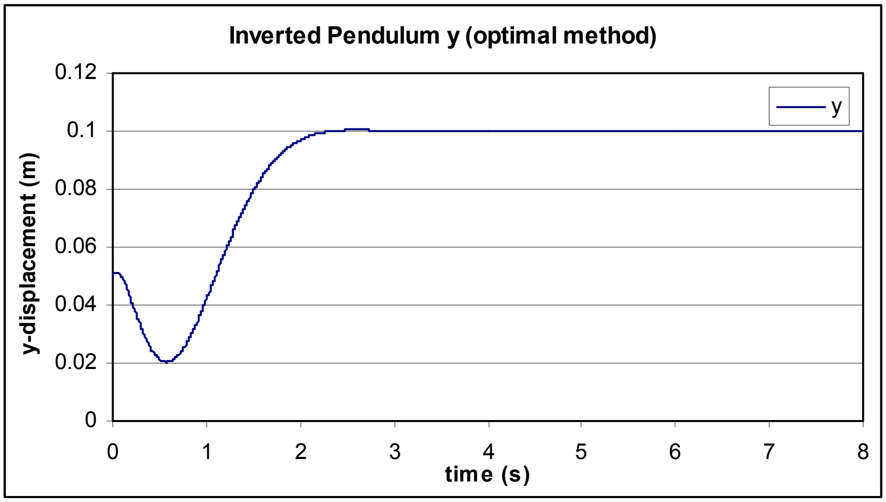


Figure 5.37: Inverted pendulum output signal **y** when the set point is 0.1m

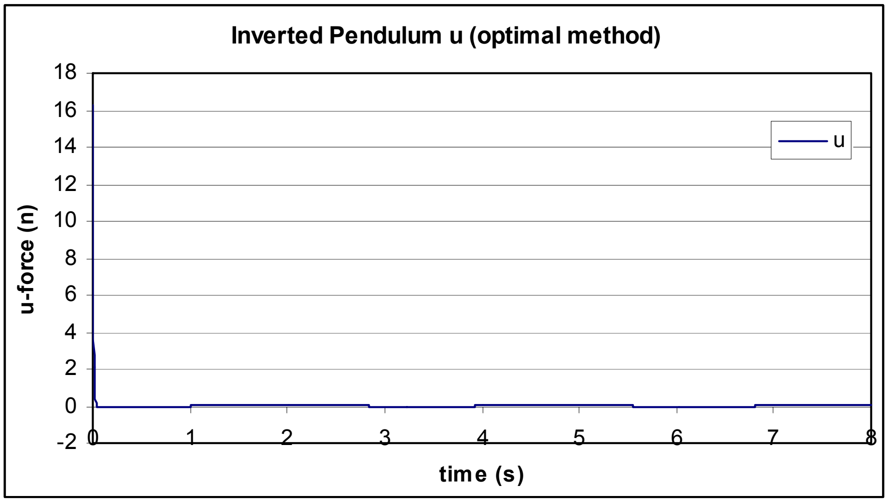


Figure 5.38: Inverted pendulum control signal **F** when the set point is 0.1m

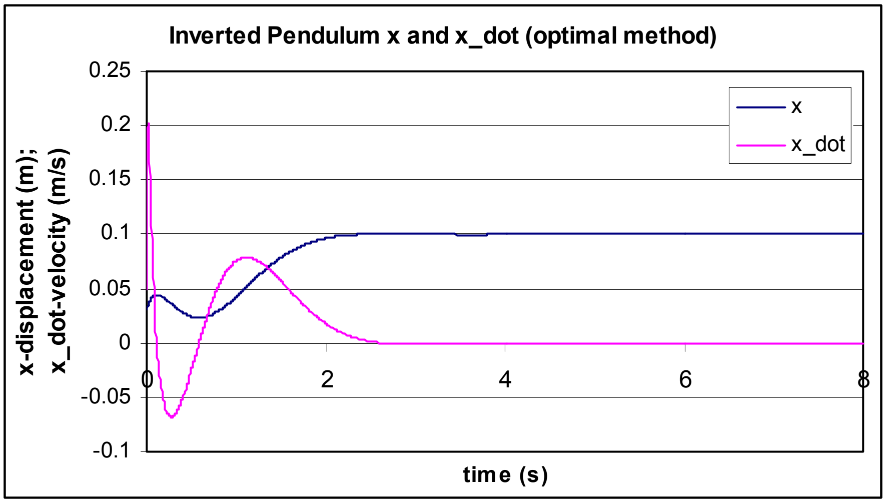


Figure 5.39: Inverted Pendulum state signal **x** and **$\dot{x}$** when the set point is 0.1m

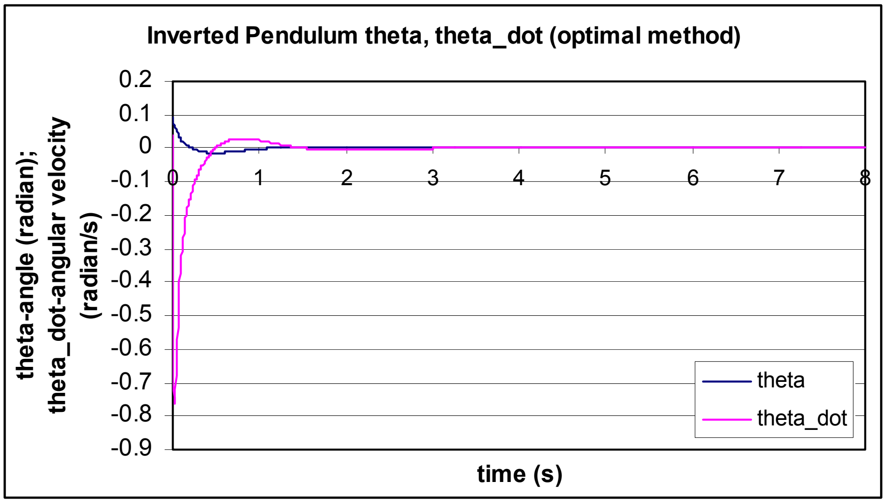


Figure 5.40: Inverted pendulum state signal **θ** and $\dot{\theta}$ when the set point is 0.1m

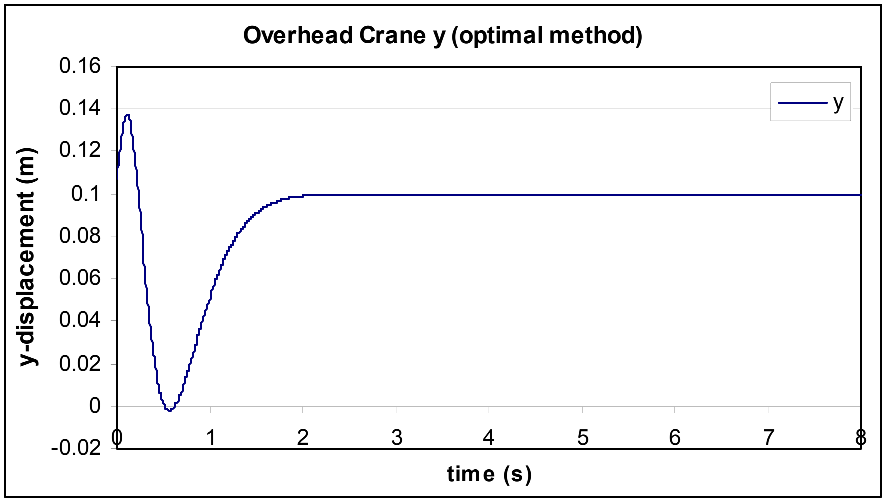


Figure 5.41: Overhead crane output signal **y** when the set point is 0.1m

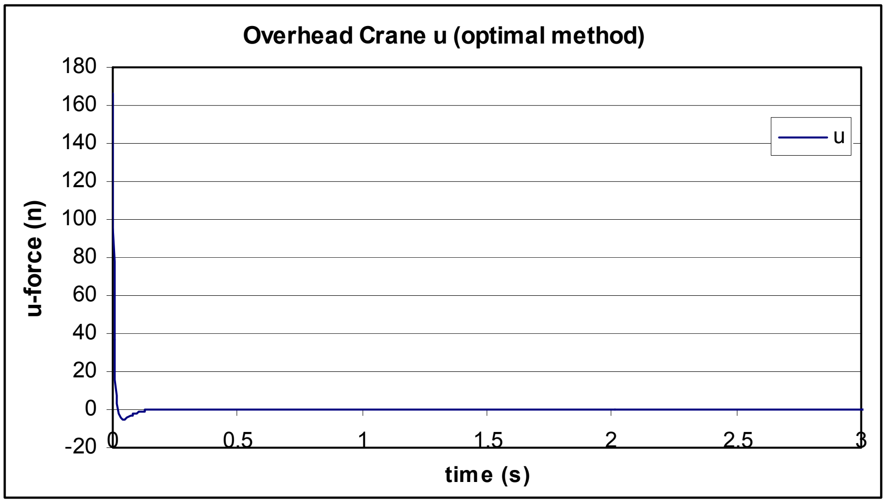


Figure 5.42: Overhead crane control signal **F** when the set point is 0.1m

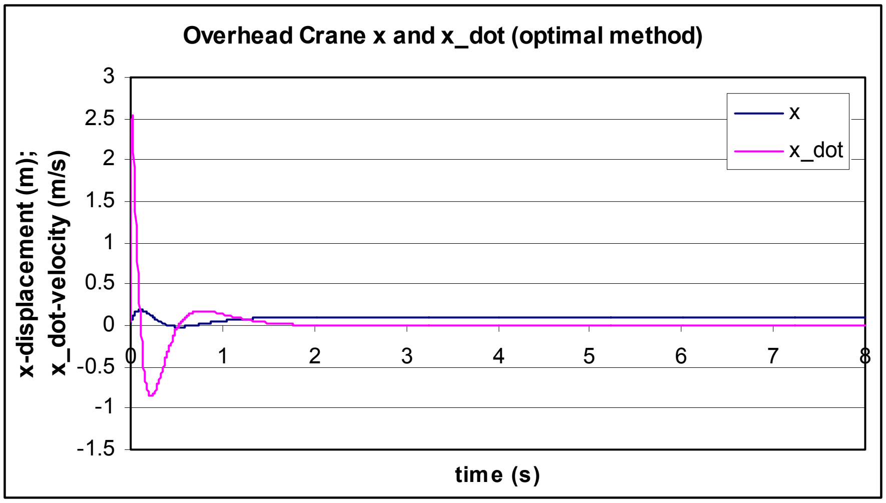


Figure 5.43: Overhead crane state signal **x** and $\dot{\mathbf{x}}$ when the set point is 0.1m

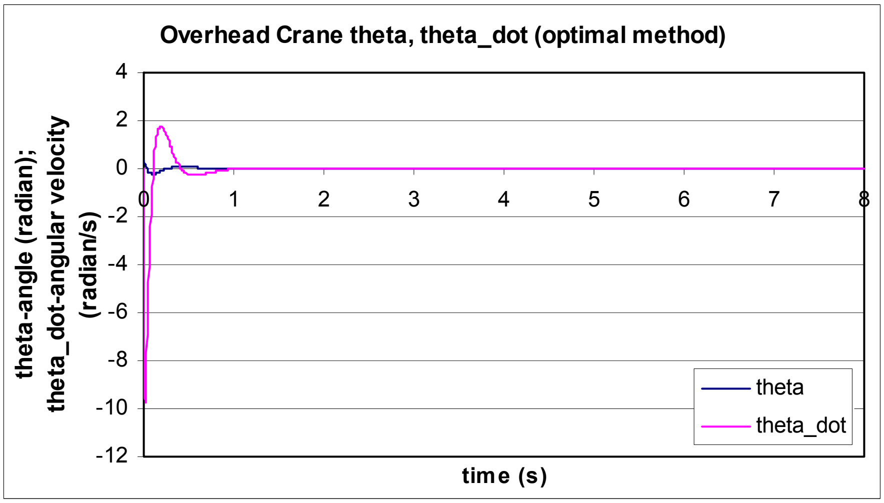


Figure 5.44: Overhead crane state signal **θ** and $\dot{\theta}$ when the set point is 0.1m

5.4 Discussion of the results

According to the simulation results, the closed-loop systems with the quadratic optimal controllers have better response than the one with the pole-placement controllers. The output signal is critically damped. Overshoot is less than 1%. The system responses are faster than those of the corresponding pole-placement method designed closed-loop system. As a result, the linear systems with the quadratic optimal controllers are selected as reference models for the design of the nonlinear controllers. These references are used in the nonlinear controller design for both Lyapunov direct method and feedback linearization method. The specifications of the dynamic output behaviour according to the obtained results are given in Table 5.1. The closed-loop poles of the systems are given in Table 5.1 as well.

Table 5.1: Table of the simulation results comparison

Set point	characteristics	Inverted pendulum (pole-placement)	Overhead crane (pole-placement)	Inverted pendulum (Optimal)	Overhead crane (Optimal)
0	Time Delay	0	0	0	0
	Overshoot	5%	200%	2%	200%
	Rising Time	0.8s	0.04s	0.05s	0.02s
	Steady State error	0.01	0.001	0.01	0.001
	Settling Time	2.5s	1.8s	2.3s	1.8s
0.05	Time Delay	0	0	0	0
	Overshoot	3%	200%	1%	150%
	Rising Time	0.8s	0.03s	0.05s	0.01s
	Steady State error	0.005	0.001	0.001	0.003
	Settling Time	1.5s	1.4s	2.1s	2s
0.1	Time Delay	0	0	0	0
	Overshoot	3%	200%	1%	40%
	Rising Time	0.9s	0.05s	1.8s	0.001s
	Steady State error	0.005	0.001	0.001	0.001
	Settling Time	1.8s	1.7s	2s	1.7s

5.5 Conclusion

In this chapter, the pole-placement method and the linear optimal control method are studied. Based on the developed linearized mathematical models of the inverted pendulum and overhead crane, the linear controllers are designed based on the pole-placement method and linear optimal control method. From the simulation results, it can be seen that the closed-loop systems with linear optimal controller have better responses. In Chapter 6 and Chapter 7, the closed-loop system with linear optimal controller is used as a reference model for the design of the nonlinear controllers.

CHAPTER SIX
LYAPUNOV STABILITY THEORY BASED MODEL REFERENCE CONTROL DESIGN

6.1 Introduction

Lyapunov stability theory allied with the model reference control method has been used to control complicated nonlinear systems. Especially, the Lyapunov Direct Method (LDM) which is also known as the Lyapunov second method has been widely used. It has become one of the most important tools for nonlinear system analysis and nonlinear controller design (Slotine and Li, 1991:41). The application of the Lyapunov direct method requires the Lyapunov function to be constructed first. Then the Lyapunov function is used to obtain a rigorous mathematical technique for designing the control vector (Ucar, 2005). In this chapter, the Lyapunov stability based model reference control method is deployed. Nonlinear, linearizing the closed-loop system according to the linear reference model controller is designed based on the requirements for negative definiteness of the Lyapunov function first derivative. Referring to the previous Chapter 5, a linear quadratic regulator is used to control the reference model and optimize the performance of the entire closed-loop system.

According to the nonlinear controller design procedure, this chapter is organized as follows: in the section 6.2, the common nonlinear controllers' design is based on the Lyapunov direct method and the model reference control method. In the section 6.3, the common controller developed from the section 6.2 is applied to the inverted pendulum and overhead crane models. The Simulink block diagrams, which are used for the simulations and the simulation results, are shown in the section 6.3.

6.2 Design of the nonlinear controller

One useful method for specifying system performance is by means of a model that will produce the desired output for a given input (Landau, 1979). The basic idea of the MRC is to develop a control strategy that can force the plant dynamics to follow the dynamics of an ideal model. The ideal model needs not be actual hardware. It can only be a mathematical model simulated on a computer. In a MRC system, the output or states of the model and that of the plant are compared. The difference is used to generate the nonlinear or linear control signals. The MRC has been used to obtain acceptable performance in some very difficult control problems involving the systems containing nonlinearity and/or time varying parameters (Landau, 1979). In this project, the MRC associated with the Lyapunov second method for stability is used to design control of the inverted pendulum and the overhead crane.

The general procedure developed for design of a nonlinear controller based on the Lyapunov second method consists of the following steps:

6.2.1 Step 1: Find the plant mathematical model

The task in this step is to find the nonlinear model of the system. The plant is described by a set of first order nonlinear differential equations, in the form of:

$$\begin{aligned} \dot{x} &= f(x,u,t) \quad x(t_0) = x_0 \\ y &= h(x) \end{aligned} \tag{6.1}$$

where $x \in R^n$ is the state vector; $u \in R^m$ is the control vector; $f \in R^n$ is a nonlinear vector valued function; $y \in R^l$ is the vector of the output; $h(t) \in R^l$ is a nonlinear vector valued function.

Referring to the project, the task is to find the nonlinear models of the inverted pendulum and the overhead crane. This has been done in the Chapter 4 by Equations (4.40) to (4.43) and (4.63) to (4.67), respectively.

6.2.2 Step 2: Determining the specifications for the behaviour of the closed-loop system

It is desired that the controlled closed-loop system follows closely the reference model system. The problem for design of a controller is to calculate a controller that always generates a signal that forces the plant states towards the reference model states. With the time tending to infinity, the error vector between the system state vector and the desired state vector has to go to zero. The closed-loop system configuration is given in Figure 6.1 shown below.

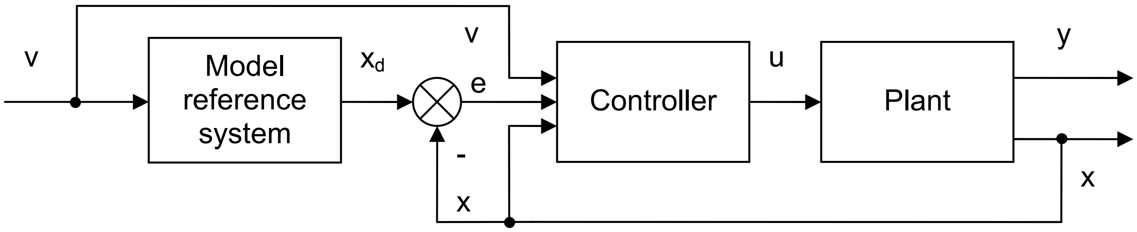


Fig 6.1: **General block diagram for model-reference control**

The model reference system can be different, linear or nonlinear, time-invariant or time variant, and so on. In this particular project, it is assumed that the model reference system is linear time-invariant system, which is given by the equation:

$$\begin{aligned}&\dot{x}_d(t) = Ax_d(t) + Bv(t) \quad x(0) = x_0\\&y = Cx_d(t)\end{aligned} \qquad (6.2)$$

where $x_d \in R^n$ is the desired states vector of the model; $v \in R^m$ is the control vector for the model reference system; $A \in R^{n\times n}$ is the constant state matrix; $B \in R^{n\times m}$ is the constant control matrix and $C \in R^{1\times n}$ is the constant output matrix.

The matrix A of the reference model is selected equal of the matrix A of the closed-loop optimal control system for the linearized models of the inverted pendulum in Equation (4.50) or the overhead crane in Equation (4.74), respectively. The matrix B is selected equal of the matrix B of the linearized models. The control input v is selected in such a way that the state vector x_d follows some desired trajectory, which then will be followed by the actual plant.

6.2.3 Step 3: Determining the error between the reference model states and the plant states

The error e(t) can be expressed as follows:

$$e(t) = x_d(t) - x(t) \qquad (6.3)$$

Where $e \in R^n$, x(t) is the actual state of the plant.

The requirements towards the closed-loop systems are that the error e(t) has to be reduced to zero by a suitable control vector u(t) when the time goes to infinity. In order to integrate both the reference model equation and the plant equation into the error Equation (6.3), it is necessary to differentiate the error Equation (6.3) according to the time. Once done, both model equations can be substituted into the differentiated equation, which is shown as follows:

$$\begin{aligned}&\dot{e}(t) = \dot{x}_d(t) - \dot{x}(t) =\\&= Ax_d(t) + Bv(t) - f(x,u,t) =\\&= Ax_d(t) + Ax(t) - Ax(t) + Bv(t) - f(x,u,t) =\\&= A[x_d(t) - x(t)] + Ax(t) + Bv(t) - f(x,u,t) =\end{aligned}$$

$\because$ $e(t)=x_d(t)-x(t)$, then the equation can be simplified as:

$$\dot{e}(t)=\dot{x}_d(t)-\dot{x}(t)=$$

$$=Ae(t)+Ax(t)-f(x,u,t)+Bv(t) \tag{6.4}$$

Equation (6.4) is a differential equation for the error vector.

The problem now is to design a controller so that at the equilibrium state $x=x_d$, $\dot{x}=\dot{x}_d$ or $e=\dot{e}=0$. Thus the equilibrium e=0 will be the origin of the coordinate system.

6.2.4 Step 4: Design of a controller

Based on the understanding of the Lyapunov direct method, the positive definite Lyapunov function V for the system is constructed and its time derivative $\dot{V}$ is examined. If $\dot{V}$ is negative definite, that means that the energy contained in the system is continuously dissipating. The system is moving towards the stable equilibrium. The following sub-steps present the procedures of how the nonlinear controller design is based on the Lyapunov direct method.

6.2.4.1 Construction of a Lyapunov function for the system

There is no general method for finding and constructing a Lyapunov function for a nonlinear system. In this project, the Lyapunov function is assumed to be done by a quadratic form:

$$V(e)=e^T(t)Pe(t) \tag{6.5}$$

where $P\in R^{n\times n}$ is positive definite real symmetrical matrix. Because the function V(e) is in quadratic form and the matrix P is positive definite, it is true that V(e) is positive definite.

6.2.4.2 Calculation of the first time derivative of the function of Lyapunov according to the trajectory of the error equation

Lyapunov function V(e) is defined in Equation (6.5). Differentiating the positive definite function V(e) along the system trajectory, its time derivative is obtained as follows:

$$\dot{V}(e)=\dot{e}^T(t)Pe(t)+e^T(t)P\dot{e}(t)=$$

$$=[Ae(t)+Ax(t)-f(x,u,t)+Bv(t)]^T\ Pe(t)=$$

$$+e^T(t)P[Ae(t)+Ax(t)-f(x,u,t)+Bv(t)]^T Pe(t)=$$

$$=\left[e^T(t)A^T+x^T(t)A^T-f^T(x,u,t)+v^T(t)B^T\right]Pe(t)+$$
$$+e^T(t)P\left[Ae(t)+Ax(t)-f(x,u,t)+Bv(t)\right]=$$

$$=e^T(t)A^TPe(t)+x^T(t)A^TPe(t)-f^T(x,u,t)Pe(t)+v^T(t)B^TPe(t)+$$
$$+e^T(t)PAe(t)+e^T(t)PAx(t)-e^T(t)Pf(x,u,t)+e^T(t)PBv(t)=$$

$$= e^T(t)\left[A^TP + PA\right]e(t) + 2M \tag{6.6}$$

where $2\,M = x^T(t)A^TPe(t) + e^T(t)PAx(t) - f^T(x,u,t)Pe(t) - e^T(t)Pf(x,u,t) +$

$$+ v^T(t)B^TPe(t) + e^T(t)PBv(t) =$$

$$= e^T(t)PAx(t) + e^T(t)PAx(t) - e^TPf(x,u,t) - e^T(t)Pf(x,u,t) +$$

$$+ e^TPBv(t) + e^T(t)PBv(t) =$$

$$= 2e^T(t)P\left[Ax(t) - f(x,u,t) + Bv(t)\right] \tag{6.7}$$

This derivation is based on the fact that P is a symmetrical matrix with the property of $P^T = P$ or

$$M' = e^T(t)P\left[Ax(t) - f(x,u,t) + Bv(t)\right] \tag{6.8}$$

M' is a scalar quantity

6.2.4.3 Calculation of the control vector

The assumed function V(e) is a Lyapunov function, it is not only positive definite, but also its first derivative must be negative definite. The first derivative of the function V(e) is expressed in Equation (6.9) as follows:

$$\dot{V}(e) = e^T(t)\left[A^TP + PA\right]e(t) + 2M' \tag{6.9}$$

The time derivative of the Lyapunov function consists of two terms. In order $\dot{V}(e)$ to be negative definite, the two terms have to either be negative definite at the same time or one term must be negative definite and the other term can be negative definite or equal to zero. For easier analysis and understanding, these two terms are examined separately in conditions one and two.

6.2.4.3.1 Condition one: The condition for the first term to be negative definite

It can be seen in Equation (6.10), the first term of $\dot{V}(t)$ is the quadratic form:

$$e^T(t)\left[A^TP + PA\right]e(t) = -e^T(t)Q\,e(t) \tag{6.10}$$

where $A^TP + PA = -Q$.

The necessary condition for Equation (6.10) to be negative definite is that the matrix Q must be a positive definite matrix. That means matrix P must be carefully selected in order to ensure Q is positive definite. However, this “natural” approach may lead to an inconclusive result, for example, the system is stable but with non-positive definite matrix Q (Slotine and Li, 1991:81).

A relatively simpler and more useful way is to define a positive definite matrix Q first, then to solve for P from the Lyapunov Equation (6.10), then examine whether P is positive definite. If P is positive definite, then the first term of $\dot{V}(e)$ is negative definite, and the problem with the selection of the Lyapunov function matrix is solved.

6.2.4.3.2 Condition two: The condition for the second term to be negative definite

For the second term, in order to satisfy the condition of the Lyapunov function time derivative negative definiteness, it is straight forward that M'<0 or M'=0. That means, the control signal u(t) must be selected carefully in order to make M' negative definite or equal to zero.

On the basis of condition 1 and 2, it can be concluded that M' can be made negative or equal to zero through convenient and appropriate choice of the plant control vector F(t). Then from noting that $V(e)\rightarrow 0$, can be seen that the equilibrium state e=0 is asymptotically stable in the large.

Condition 1 can always be fulfilled by a proper choice of P since the first term is in quadratic form. The problem now is to choose an appropriate vector F(t) so that M is either zero or negative scalar quantity. The calculations of F(t) can be done with selected values of a matrix P or a matrix Q.

The general procedures for designing a nonlinear controller based on the Lyapunov stability and MRC method are described above. Now the method is applied to the inverted pendulum and the overhead crane. The developed control should always keep the inverted pendulum standing up and the oscillations of the overhead crane have to be reduced fast. The specifications determined for the linear control systems are used for design of the MRC systems in this chapter too.

6.3 Development of MRC system for the inverted pendulum based on the Lyapunov direct method

The same method is applied to the inverted pendulum and overhead crane. In the thesis, the application of the method to the inverted pendulum is described in detail. For the overhead crane, it is described briefly, because only the model of the overhead crane is different whilst the steps of the procedure are the same.

6.3.1 Design of the nonlinear Lyapunov controller for the model of the inverted pendulum

Based on the study and understanding of the MRC theory, the Lyapunov stability theory, the Lyapunov direct method and LQR control method, the following sub-sections cover the general procedures as described above, and applied to the inverted pendulum system. The nonlinear controller for controlling of the inverted pendulum is developed step by step.

6.3.1.1 The nonlinear model of the inverted pendulum

The nonlinear model of the inverted pendulum is developed in previous Chapter 4. In order to avoid the confusion of the letters usage, the states of the inverted pendulum are denoted by z, where $z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$.

Then, the nonlinear mathematical model of the inverted pendulum can be expressed as follows:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} z_2 \\ \dfrac{\begin{bmatrix} -\left(J+mL^2\right)\cdot b\cdot z_2 - m^2\cdot L^2\cdot g\cdot \sin z_3\cdot \cos z_3 + \\ +m\cdot L\cdot b_t\cdot z_4\cdot \cos z_3 + \left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot \sin z_3 \end{bmatrix}}{\sigma} \\ z_4 \\ \dfrac{\begin{bmatrix} (M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 - m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 - \\ -(M+m)\cdot b_t\cdot z_4 + m\cdot L\cdot \cos z_3\cdot b\cdot z_2 \end{bmatrix}}{\sigma} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{\left(J+m\cdot L^2\right)}{\sigma} \\ 0 \\ \dfrac{-m\cdot L\cdot \cos z_3}{\sigma} \end{bmatrix}\cdot F$$

(6.11)

$$y(t) = Cz(t) \qquad (6.12)$$

where $\sigma = (M+m)(J+mL^2) - m^2L^2\cos^2 z_3$; u=F and C=[1 0 L 0].

Or the model can be written in the following common form:

$$\dot{z}(t) = f\left(z(t)\right) + g\left(z(t)\right)F(t) \qquad (6.13)$$

$$y(t) = Cz(t) \qquad (6.14)$$

6.3.1.2 The model of the desired system

The linear reference model for this project can be written in the form of:

$$\begin{aligned}&\dot{x}_d(t) = Ax_d(t) + Bv(t) \quad x_d(0) = x_{d0}\\ &y_d(t) = Cx(t)\end{aligned} \tag{6.15}$$

where $x_d \in R^4$ is the desired state space vector, $v \in R^1$ is the control vector for the reference model, $A \in R^{4\times4}$, $B \in R^{4\times1}$ and $B \in R^{1\times4}$ are the state space, control and output matrices of the reference model, x_{d0} is the initial state.

The reference model is designed in two steps:

- The first step: A LQR system is designed based on the linearized inverted pendulum mathematical model following the Chapter 5 derivations but without considering the integration of the error signal between the set point and the output of the plant
- The second step: The state matrix A_d of the reference model is found in such a way that:

$$A_d = A_L - B_L K_L \tag{6.16}$$

where $A_L = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{-B(J+mL^2)}{\sigma'} & \dfrac{-(ml)^2 g}{\sigma'} & \dfrac{mLB_T}{\sigma'} \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{mLB}{\sigma'} & \dfrac{m(M+m)Lg}{\sigma'} & \dfrac{-B_T(M+m)}{\sigma'} \end{bmatrix}$ is the state matrix of the linearized model; $B_L = \begin{bmatrix} 0 \\ \dfrac{J+mL^2}{\sigma'} \\ 0 \\ \dfrac{-Lm}{\sigma'} \end{bmatrix}$ is the control matrix of the linearized model;

$\sigma' = J(M+m) + MmL^2$ and $K_L \in R^{1\times4}$ is the feedback gain matrix.

Because the set points are different, the corresponding reference models are different from each other. In the Table 6.1, the weighting matrices $Q \in R^{4\times4}$, $R \in R^{1\times1}$, the feedback gain K_L and the desired system state matrices A_d are specified for different set points.

Table 6.1: Parameters table of the reference model

set point	Q	R	K_L	$A_d=A_L-B_L\times K_L$
0.0m	$\begin{bmatrix}540&0&0&0\\0&8&0&0\\0&0&10&0\\0&0&0&1\end{bmatrix}$	1	$\begin{bmatrix}-23.2379\\-12.5704\\-35.5126\\-4.9353\end{bmatrix}^T$	$\begin{bmatrix}0&1&0&0\\122.5976&65.7911&173.8900&26.2568\\0&0&0&1\\-509.2664&-273.2940&-681.6246&-109.7323\end{bmatrix}$
0.05m	$\begin{bmatrix}530&0&0&0\\0&8&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	1	$\begin{bmatrix}-23.0217\\-12.4343\\-35.0857\\-4.8945\end{bmatrix}^T$	$\begin{bmatrix}0&1&0&0\\121.4571&65.0726&171.6377&26.0413\\0&0&0&1\\-504.5290&-270.3096&-672.2686&-108.8374\end{bmatrix}$
0.1m	$\begin{bmatrix}530&0&0&0\\0&8&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	1	$\begin{bmatrix}-23.0217\\-12.4343\\-35.0857\\-4.8945\end{bmatrix}^T$	$\begin{bmatrix}0&1&0&0\\121.4571&65.0726&171.6377&26.0413\\0&0&0&1\\-504.5290&-270.3096&-672.2686&-108.8374\end{bmatrix}$

The matrix B can be selected equal to the matrix B of the linearized model.

$$B=\begin{bmatrix}0\\ \dfrac{J+mL^2}{\sigma'}\\ 0\\ \dfrac{-Lm}{\sigma'}\end{bmatrix}=\begin{bmatrix}0\\ b_1\\ 0\\ b_2\end{bmatrix}=\begin{bmatrix}0\\ 5.2758\\ 0\\ -21.9153\end{bmatrix} \tag{6.17}$$

The matrix C is C=[1 0 L 0]=[1 0 0.22 0]

6.3.1.3 Determination of the error between the reference model and plant states

$$e(t)=x_d(t)-z(t),\ e\in R^{4\times 4} \tag{6.18}$$

The error e(t) has to be reduced to zero by a suitable control vector u. The differential equation of the error is:

$$\begin{aligned}\dot{e}(t)&=\dot{x}_d(t)-\dot{z}(t)=A\,x_d(t)+B\,v(t)-f(z(t))-g(z(t))u(t)=\\&=A\,x_d(t)+A\,z(t)-A\,z(t)+B\,v(t)-f(z(t))-g(z(t))u(t)=\\&=A(x_d(t)-z(t))+A\,z(t)+B\,v(t)-f(z(t))-g(z(t))u(t)=\\&=A\,e(t)+A\,z(t)+Bv(t)-f(z(t))-g(z(t))u(t)\end{aligned} \tag{6.19}$$

The problem is to design a control vector F(t), such that at the equilibrium state $z=x_d,\dot{z}_d=\dot{x}_d,e=\dot{e}=0$ is achieved.

6.3.1.4 Design of the nonlinear controller

6.3.1.4.1 Construction of the Lyapunov function

Construction of the Lyapunov function for the error differential equation shown in Equation (6.18) is:

$$V(e) = e^T(t)Pe(t)\,, \tag{6.20}$$

where P is a symmetrical positive definite matrix , $P \in R^{n\times n}$

6.3.1.4.2 Calculation of the first derivative of the function of Lyapunov

$$\dot{V}(e) = \dot{e}^T(t)Pe + e^T(t)P\dot{e}(t) =$$

$$= \left[Ae(t) + Az(t) + Bv(t) - f(z(t)) - g(z(t))\,F(t)\right]^T Pe(t) +$$
$$+ e^T(t)P\left[Ae(t) + Az(t) + Bv(t) - f(z(t)) - g(z(t))F(t)\right] =$$

$$= \left[e^T(t)A^T + z^T(t)A^T + v^T(t)B^T - f^T(z(t)) - F^T(t)\,g^T(z(t))\right]Pe(t) +$$
$$+ e^T(t)P\left[Ae(t) + Az(t) + Bv(t) - f(z(t)) - g(z(t))F(t)\right] =$$

$$= e^T(t)A^TPe(t) + z^TA^TPe(t) + v^T(t)B^TPe(t) - f^T(z(t))Pe(t) - F^T(t)g^T(z(t))Pe(t) +$$
$$+ e^TPAe(t) + e^T(t)PAz(t) + e^T(t)PBv(t) - e^T(t)Pf(z(t)) - e^T(t)Pg(z(t))F(t) =$$

$$= e^T(t)A^TPe(t) + e^TPAe(t) + e^T(t)PAz(t) + e^T(t)PAz(t) + e^T(t)PBv(t) + e^T(t)PBv(t) -$$
$$- e^T(t)Pf(z(t)) - e^T(t)Pf(z(t)) - e^T(t)Pg(z(t))F(t) - e^T(t)Pg(z(t))F(t) =$$

$$= e^T(t)\left[A^TP + PA\right]e(t) + 2e^T(t)P\left[Az(t) + Bv(t) - f(z(t)) - g(z(t))F(t)\right] =$$

$$= -e^T(t)Qe(t) + 2M'$$

$Q = A^TP + PA$ is a positive definite matrix, Q is symmetrical.

$$M' = e^T(t)P\left[Az(t) + Bv(t) - f(z(t)) - g(z(t))F(t)\right] \tag{6.21}$$

The obtained equation is the equation of the first derivative of the Lyapunov function. In order for the error in the closed-loop system to go to zero as time t→∞, it is necessary for this equation to be negative definite. The first part is a negative definite as Q is selected to be positive definite. Then the second part M’ can be made zero or negative M’≤0 by a proper selection of the control F(t).

6.3.1.4.3 Calculation of F(t)

Calculation of the nonlinear control is done by some mathematical transformations of the expression for M’:

$$M' = e^T(t)P\left[Az(t) + Bv(t) - f(z(t)) - g(z(t))\cdot F(t)\right] \le 0$$

$$M' = e^T(t)P\left[Az(t) + Bv(t) - f(z(t))\right] - e^T(t)Pg(z(t))\cdot F(t) \le 0$$

$$e^T(t)P[Az(t)+Bv(t)-f(z(t))] \le e^T(t)Pg(z(t)) \cdot F(t) \qquad (6.22)$$

The expressions from both sides of the equations are scalars, which depend on time. That is why it is possible to divide both sides by $e^T(t)Pg(z(t))$ and to obtain:

$$[e^T(t)Pg(z(t))]^{-1} e^T(t)P[Az(t)+Bv(t)-f(z(t))] \le F(t)$$

or $F(t) \ge [e^T(t)Pg(z(t))]^{-1} e^T(t)P[Az(t)+Bv(t)-f(z(t))]$ (6.23)

6.3.1.5 Diagram of the closed-loop system

Based on the above equations, the diagram of the closed-loop system can be drawn. The expression for the nonlinear controller is multiplied by a number $n>0$ in order to make the fulfilment in the Equation (6.23) stronger. The nonlinear control given by the Equation (6.23), makes the first derivative of the Lyapunov function negative and linearizes the closed-loop system consisting from the nonlinear controller and the plant and equals to the reference model. The block diagram of the closed-loop system is given in Figure 6.2.

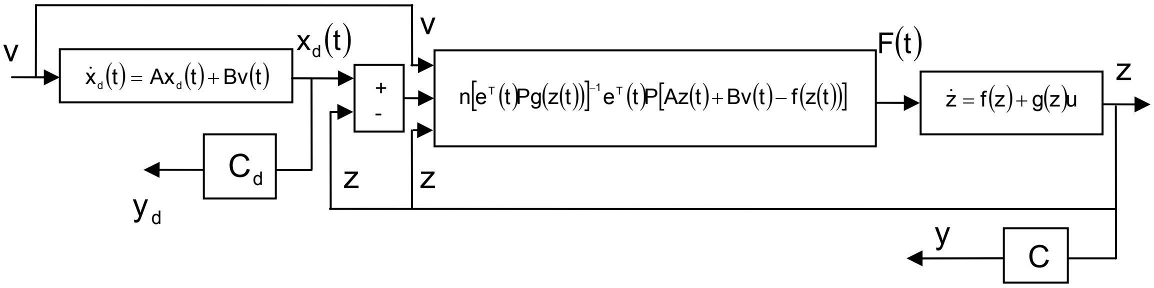

v
$\dot{x}_d(t) = Ax_d(t) + Bv(t)$
$x_d(t)$
v
+
-
$n[e^T(t)Pg(z(t))]^{-1} e^T(t)P[Az(t)+Bv(t)-f(z(t))]$
F(t)
$\dot{z} = f(z) + g(z)u$
z
C_d
y_d
z
z
y
C

Figure 6.2: **Block diagram of the closed-loop system**

6.3.2 Design of the linear optimal control v of the closed-loop system

It can be seen that the desired vector x_d depends on the input control vector $v(t)$ for the reference model. Different values of $v(t)$ will determine different values of $x_d(t)$. It can be seen from the expression for the nonlinear control in Equation (6.23) that the values of the $F(t)$ depend on the parameters of the nonlinear plant model. Real implementation of the nonlinear control then cannot be very successful because of the disturbance influence and parameter variance. The linearizing and stabilizing effects could be lost. This means that additional control has to be designed to make the closed-loop system output exactly equal to the desired behaviour of the closed-loop system. It is possible to apply the theory of optimal linear quadratic control, in order to design optimal controller for the system of the reference model.

6.3.2.1 Specification of the closed-loop system for the reference model and the linearized closed-loop system

It is supposed that the desired output of the entire closed-loop system is a set point value y^{sp}. Based on this assumption, it is necessary to determine the optimal controller in such a way that $y(t)=y^{sp}$.

6.3.2.2 Formulation of the problem for linear optimal control

The problem in the designing of the optimal linear control v(t) is formulated and solved following the development in Chapter 5, sections (5.32) and (5.33). The problem is solved for the output $y_d=x_d+L\theta_d=z_{1d}+Lz_{3d}$, but the real-time implementation is done for the real output $y=x+L\theta$ in order to additionally compensate for all disturbances influencing the plant.

Following the derivations in a Chapter 5, the vector $\begin{bmatrix} x_{de}(t) \\ \zeta_e(t) \end{bmatrix} = e_d(t)$ and the model of the error between the reference model states and their steady state values can be written in the following form:

$$\dot{e}_d(t) = A_\Phi e_d(t) + B_\Phi v_e(t),\ e_d \in R^{n+1},\ v_e \in R^m \qquad (6.38)$$

$$v_e(t) = \hat{K} e_d(t) \qquad (6.39)$$

where $A_\Phi = \begin{bmatrix} A & 0 \\ -C_d & 0 \end{bmatrix},\ B_\Phi = \begin{bmatrix} B \\ 0 \end{bmatrix},\ K = [K \quad K_I],\ K \in R^{m\times(n+1)}$ (6.40)

The basic idea of obtaining Equation (6.38), (6.39) is that the servo-problem is converted to a problem for design of a regulator system in which the set point in the criterion is zero. The problem can then be formulated in such a way that finds the matrix of the regulator K to fulfil the criterion:

$$J_P = \int_0^\infty \left[\| e(t) \|_Q^2 + \| v_e(t) \|_R^2 \right] dt,\ Q \in R^{(n+1)(n+1)},\ R \in R^{m\times m} \qquad (6.41)$$

is minimized under the model Equation (6.38). The solution of this problem is based on the linear-quadratic controller design theory.

6.3.2.3 Solution of the problem for design of the optimal regulator

The solution of the problem in Equation (6.38), (6.39) and (6.41) is given by the equation

$$v_e(t) = -K_\Phi e_d(t) = -Kx_{de}(t) + K_I\zeta(t) = -R^{-1}B_\Phi^T P_\Phi e_d(t) \tag{6.42}$$

where P_Φ is a solution of the equation of Riccati; $K_\Phi=[K \quad K_L]\in R^{1\times 5}$

The solution of the problem in Equation (6.38), (6.39) and (6.41) is done by using Matlab function '*lqr*':

$$[K_\Phi, \quad P_\Phi, \quad E] = lqr(A_\Phi, B_\Phi, Q, R)$$

where K_Φ is the matrix of the regulator, P_Φ is the matrix of the Riccati equation and E is a vector of the poles of the closed-loop matrix $[A_\Phi - B_\Phi K_\Phi]$. If the system is stable all the poles have to be with negative real part.

The control $v_e(t)$ can be obtained in the following way:

$$v_e(t) = K_\Phi e_d(t) = -Kx_{de}(t) + K_I\zeta_e(t) = -K[x_d - x_d(\infty)] + K_I[\zeta(t) - \zeta(\infty)] \tag{6.43}$$

The steady state values are:

$$\dot{x}_d(\infty) = 0 = A_d x(\infty) + B_d v(\infty) \tag{6.44}$$

$$\dot{\zeta}(\infty) = 0 = y^{sp} - Cx_d(\infty) \tag{6.45}$$

In a vector matrix form

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} A & B \\ -C & 0 \end{bmatrix} \begin{bmatrix} x_d(\infty) \\ v(\infty) \end{bmatrix} + \begin{bmatrix} 0 \\ y^{sp} \end{bmatrix} \tag{6.46}$$

The rank of the matrix $\begin{bmatrix} A & B \\ -C & 0 \end{bmatrix}$ has to be checked to be equal to $(n+1)$. Then,

$$\begin{bmatrix} x_d(\infty) \\ v(\infty) \end{bmatrix} = -\begin{bmatrix} A & B \\ -C & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ y^{sp} \end{bmatrix} \tag{6.47}$$

Also, $v(\infty) = -Kx_d(\infty) + K_I\zeta(\infty)$, then $v_e(t) = -Kx_d(t) + K_I\zeta(t) - v(\infty)$

From here $\zeta(\infty) = \dfrac{1}{K_I}[v(\infty) + Kx(\infty)]$ $\tag{6.48}$

The algorithm of calculation is:

1. Determine K_Φ
2. Determine $x_d(\infty)$, $v(\infty)$.
3. Determine $v_e(t)$
4. Determine $\zeta(\infty) = \dfrac{1}{K_I}[v(\infty) + Kx_d(\infty)]$

Then $v(t) = v_e(t) + v(\infty) = -Kx_d(t) + K_I\zeta(t) = -Kx_d(t) + K_I\left[y^{sp}(t) - y(t)\right]$ (6.49)

6.3.2.4 Application of the obtained linear control to the nonlinear control system and the reference model

The diagram obtained for the nonlinear control is extended with the optimal control v shown in Figure 6.3.

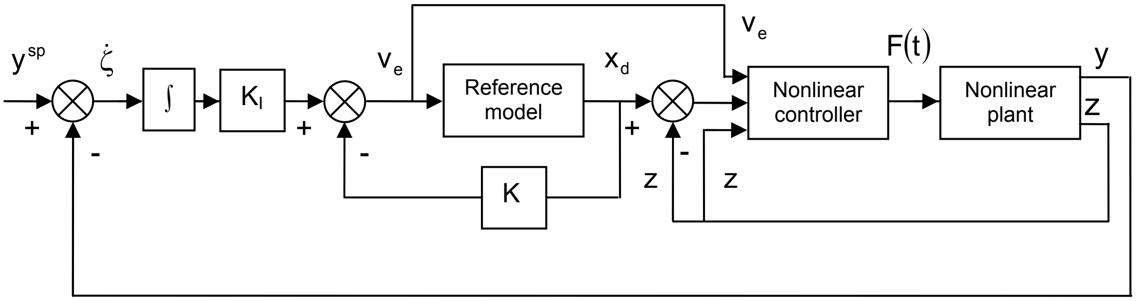


Figure 6.3: **The structure block diagram of the Lyapunov stability based MRC system**

6.3.2.5 Design of the linear controller K_Φ for the inverted pendulum plant

The matrices A, B, C are given from the part for designing of the linear controller. The matrices A_Φ and B_Φ are formed as:

$$A_\Phi = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}, \; B_\Phi = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

where $A_\Phi = \begin{bmatrix} A \in R^{4\times4} & 0 \in R^{4\times1} \\ C \in R^{1\times4} & 0 \in R^{1\times1} \end{bmatrix}$ and $B_\Phi = \begin{bmatrix} B \in R^{4\times1} \\ 0 \in R^{1\times1} \end{bmatrix}$.

The matrix $R_\Phi \in R^{1\times1}$ is $R_\Phi=1$. The matrices $Q_\Phi \in R^{5\times5}$ is selected differently for different set points. In Table 6.2, the matrices Q_Φ are shown.

The feedback control gain K_Φ can be found by using the Matlab command "*lqr*".

$$[K_\Phi, \;\; P_\Phi, \;\; E] = \mathrm{lqr}(A_\Phi, B_\Phi, Q, \; R)$$

For different set points, the values of the feedback control gain K_Φ are shown in Table 6.2.

Table 6.2: **Parameters table for linear control design**

Set point	Q_{Φ}	K_{Φ}
0.0m	$\begin{bmatrix} 9000 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 700000 \end{bmatrix}$	$\begin{bmatrix} -389.6059 & -83.6857 & -181.6120 & -23.2996 & 836.6600 \end{bmatrix}$
0.05m	$\begin{bmatrix} 9000 & 0 & 0 & 0 & 0 \\ 0 & 110 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 600000 \end{bmatrix}$	$\begin{bmatrix} -362.6252 & -77.5148 & -166.2922 & -21.1849 & 774.5967 \end{bmatrix}$
0.1m	$\begin{bmatrix} 9000 & 0 & 0 & 0 & 0 \\ 0 & 130 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 220000 \end{bmatrix}$	$\begin{bmatrix} -261.5103 & -63.8087 & -134.7821 & -17.8526 & 469.0416 \end{bmatrix}$

The obtained values for $K=[K_{\Phi 11}\ \ K_{\Phi 12}\ \ K_{\Phi 13}\ \ K_{\Phi 14}]$ and $K_I=-K_{\Phi 15}$ are used in Simulink diagram to form the closed-loop system.

6.4 Simulation

The simulation is done in the Matlab/Simulink environment. In this section, the associated m-files and the Simulink block diagrams are introduced first, and then the simulation results for both inverted pendulum and overhead crane are shown. The results are discussed at the end of the section.

6.4.1 Associated m-files

There are two m-files used for the simulation in this chapter. These two m-files are: named as “*LDM_IP.m*” and “*LDM_OC.m*”. “*LDM_IP.m*” deals with the inverted pendulum and “*LDM_OC.m*” deals with the overhead crane. These two m-files calculate the linear feedback control gain, form the reference model, calculate the closed-loop feedback control gain and trigger the Simulink files. The complete programs code is in Appendix A. 4 and Appendix A. 5 respectively.

6.4.2 Simulink block diagram

The system is simulated in Simulink. The overall Lyapunov direct method Simulink block diagram “*LDM_IP.mdl*” is shown in Figure 6.4. It consists of several subsystems, including:

- The linear controller subsystem (Figure 6.5)
- The linear model subsystem – from Chapter 4
- The nonlinear controller subsystem (Figure 6.6)
- The nonlinear model subsystem – from Chapter 4
- The derivative of the Lyapunov function subsystem (Figure 6.7)

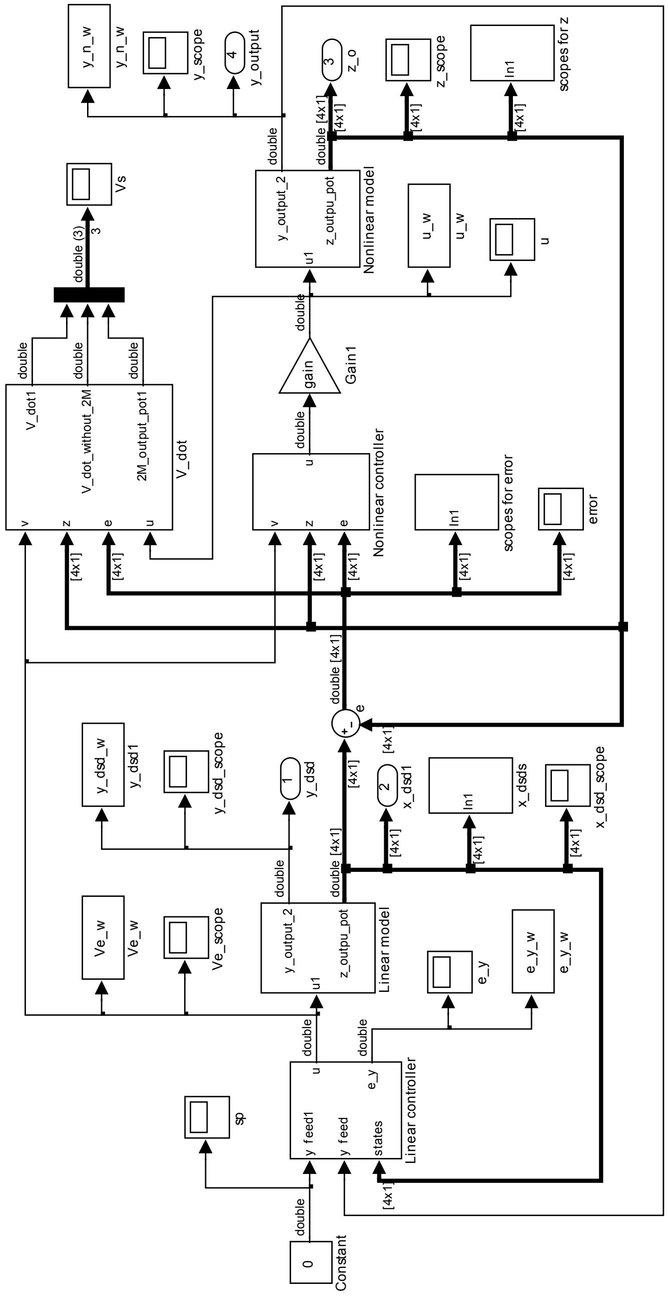


Figure 6.4: Overall Lyapunov Direct method Simulink block diagram

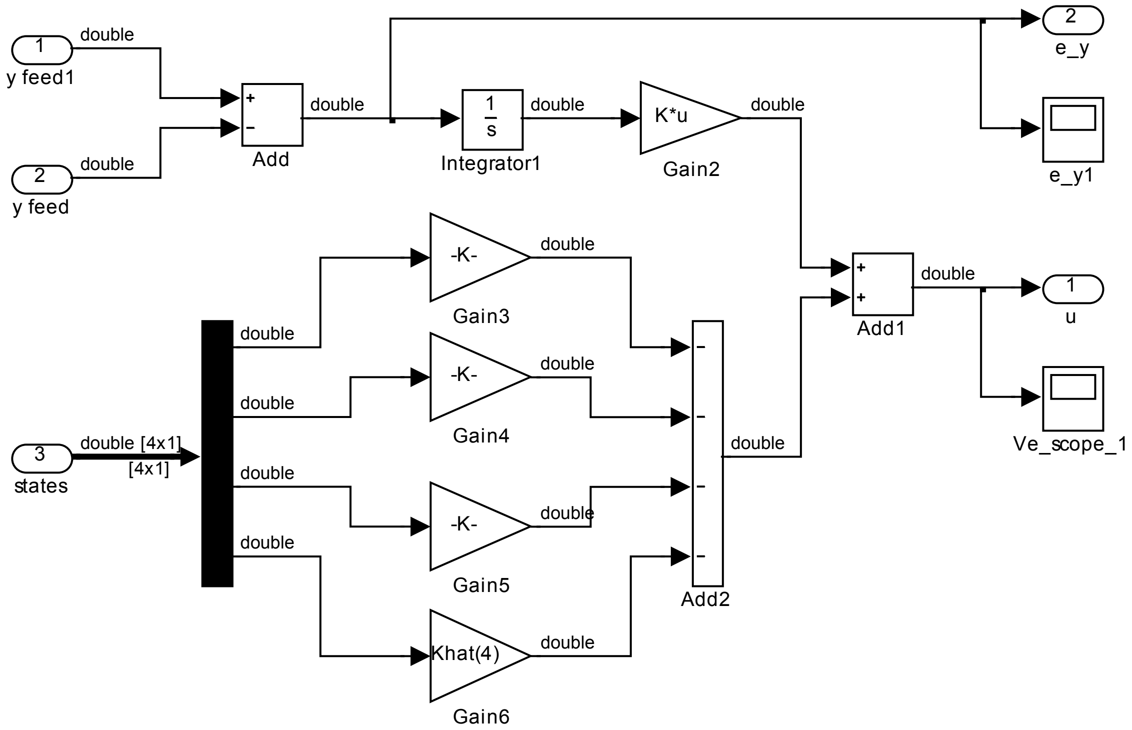


Figure 6.5: Linear optimal controller Simulink block diagram

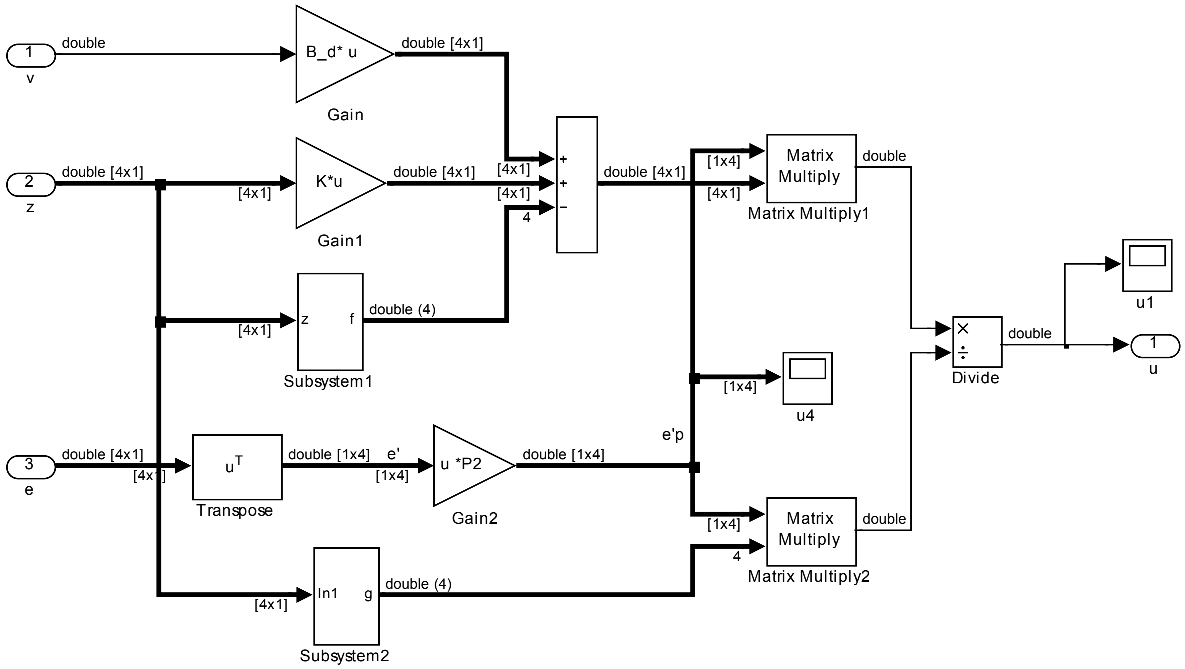


Figure 6.6: Nonlinear controller Simulink block diagram

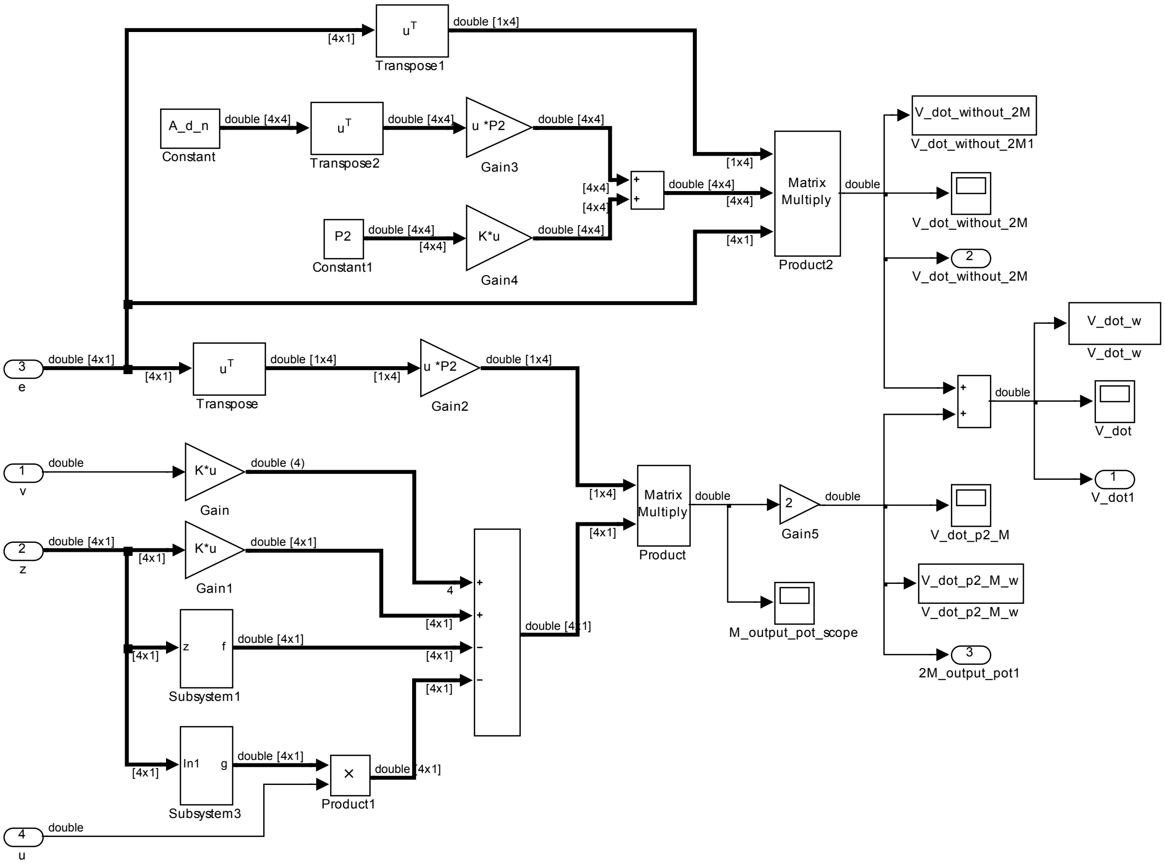


Figure 6.7: Derivative of the Lyapunov Function Simulink block diagram

The linear controller subsystem and the linear model subsystem form the reference model system. The nonlinear controller subsystem produces the nonlinear control signal which corresponds to Equation (6.23). The derivative of the Lyapunov function subsystem is used to monitor the stability of the system. The Simulink block diagrams of the linear model and the nonlinear model are shown in Chapter 4.

As the Lyapunov direct method is applied on the inverted pendulum and the overhead crane, the differences exist in the models. The Lyapunov Simulink block diagram for the inverted pendulum can be converted to an overhead crane block diagram by simply replacing all the inverted pendulum models with the overhead crane models. These changes take place in the “nonlinear model” subsystem, the “nonlinear controller” subsystem and the “*v_dot*” subsystem. The Simulink file for the overhead crane is called “*LDM_OC.mdl*”.

6.4.3 Simulation results

The simulation results are presented. These results include the following signals: the linear/nonlinear states, linear/nonlinear outputs, linear/nonlinear control signals, error signals between four states and the signal of the derivative of the Lyapunov function. The notation linear is used for the reference model variables and nonlinear for the inverted pendulum and overhead crane variables.

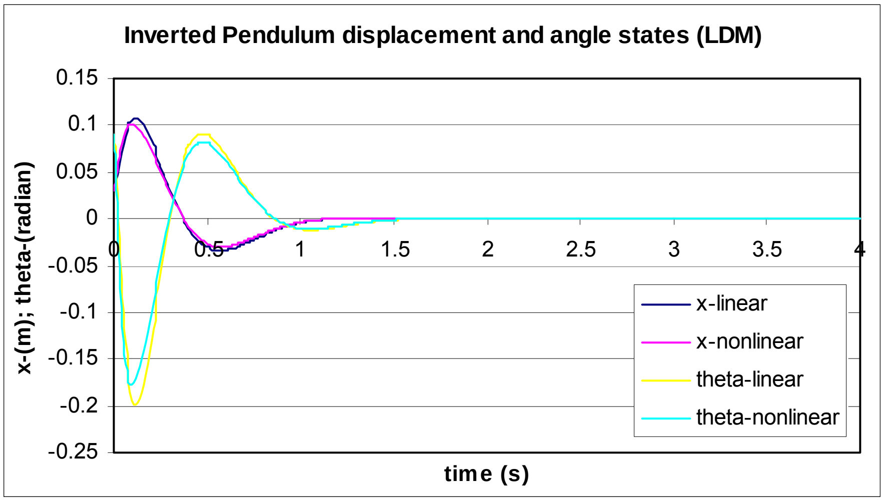


Figure 6.8: Inverted pendulum displacement and angle state signals when set point is 0m

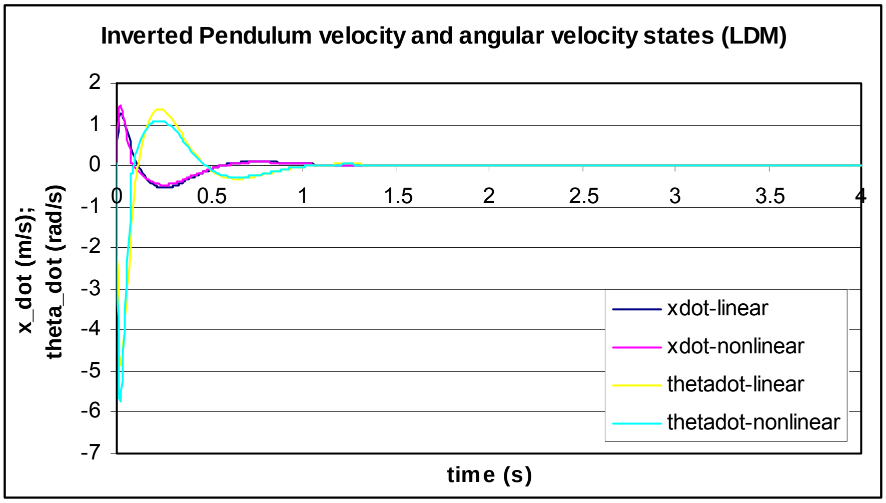


Figure 6.9: Inverted pendulum velocity and angular velocity state signals when the set point is 0m

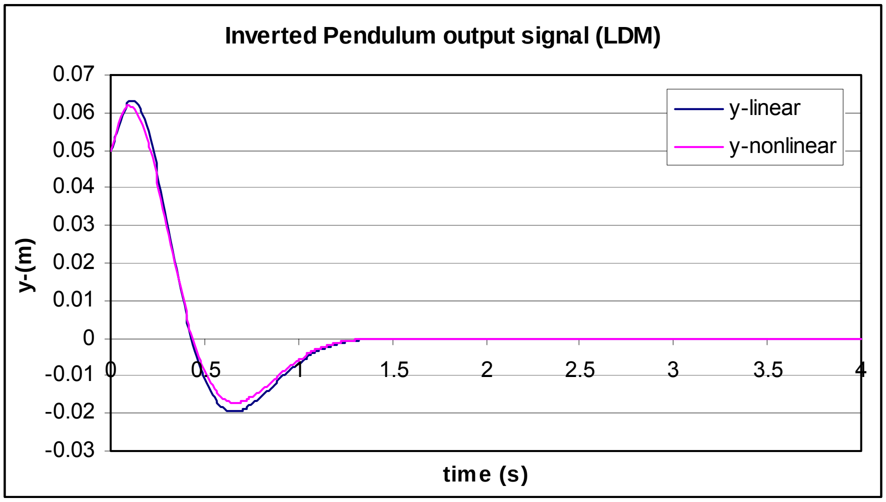


Figure 6.10: Inverted pendulum output signals when the set point is 0m

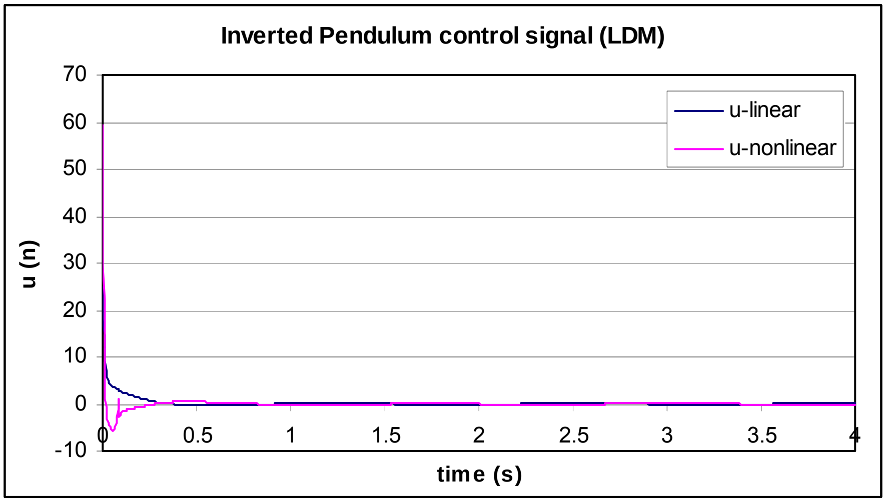


Figure 6.11: Inverted pendulum control signals when the set point is 0m

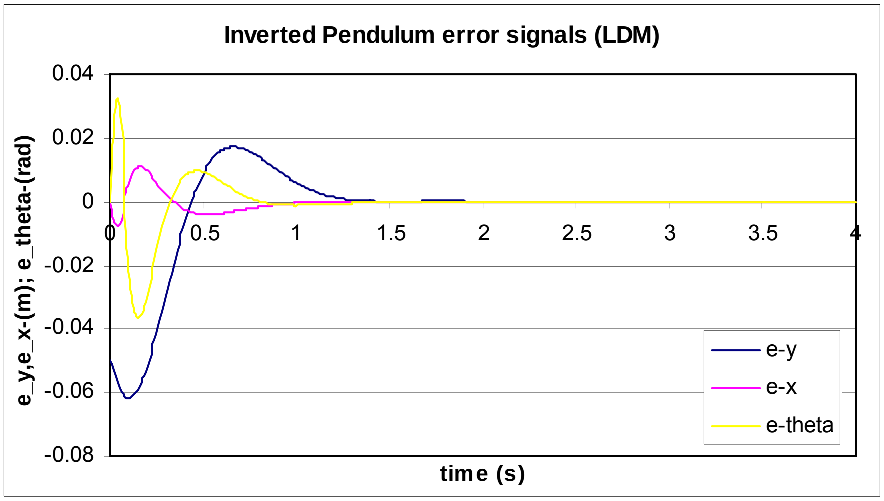


Figure 6.12: Inverted pendulum states error signals when the set point is 0m

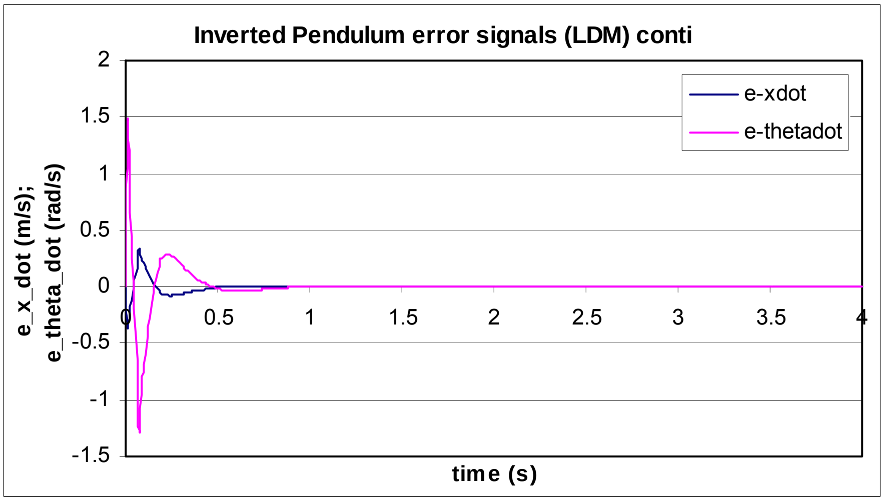


Figure 6.13: Inverted pendulum error signals when the set point is 0m

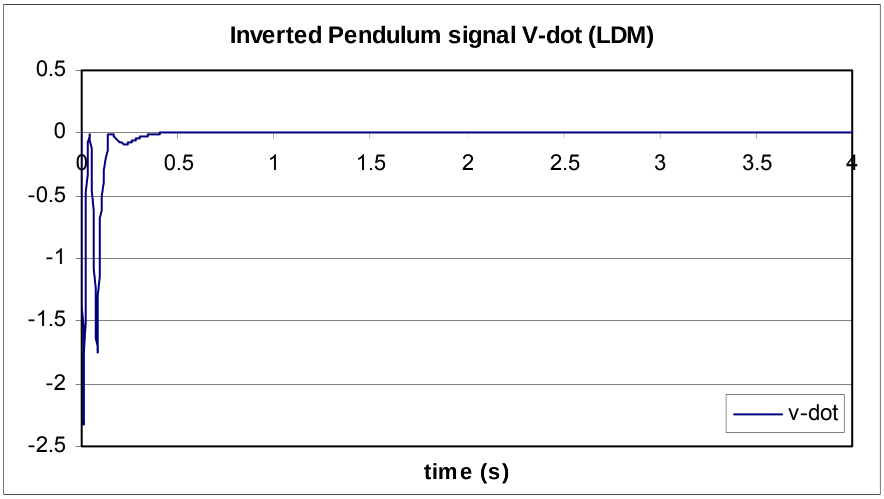


Figure 6.14: Inverted pendulum signal of Lyapunov function derivative when the set point is 0m

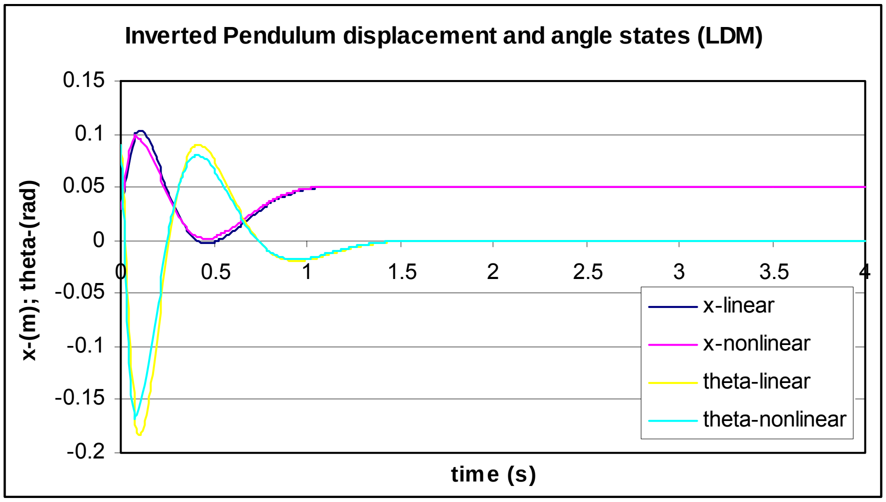


Figure 6.15: Inverted pendulum displacement and angle signals when the set point is 0.05m

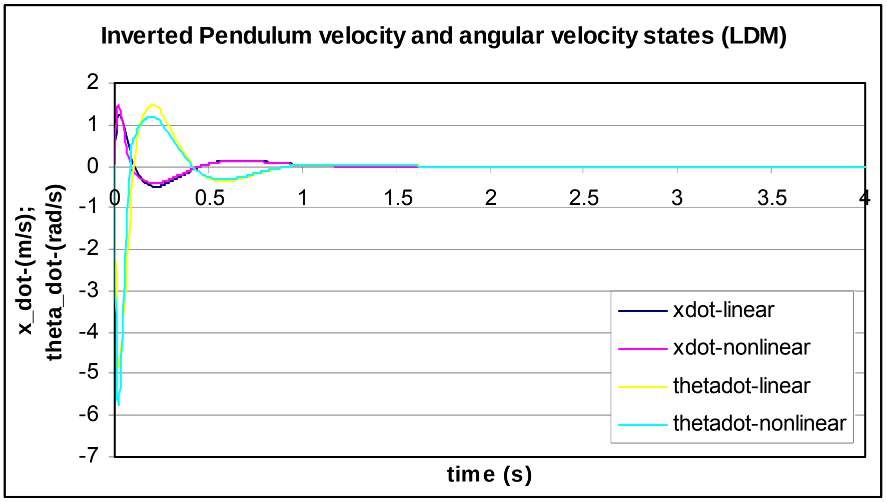


Figure 6.16: Inverted pendulum velocity and angular velocity signals when set point is 0.05m

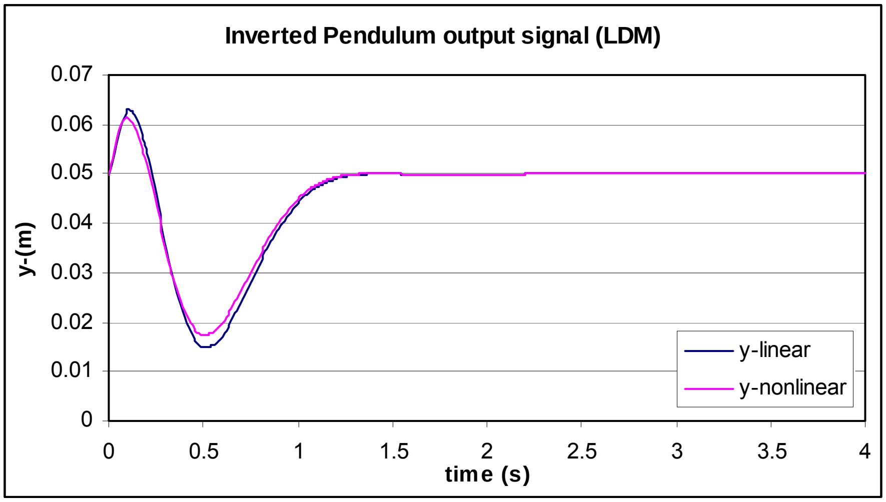


Figure 6.17: Inverted pendulum output signals when the set point is 0.05m

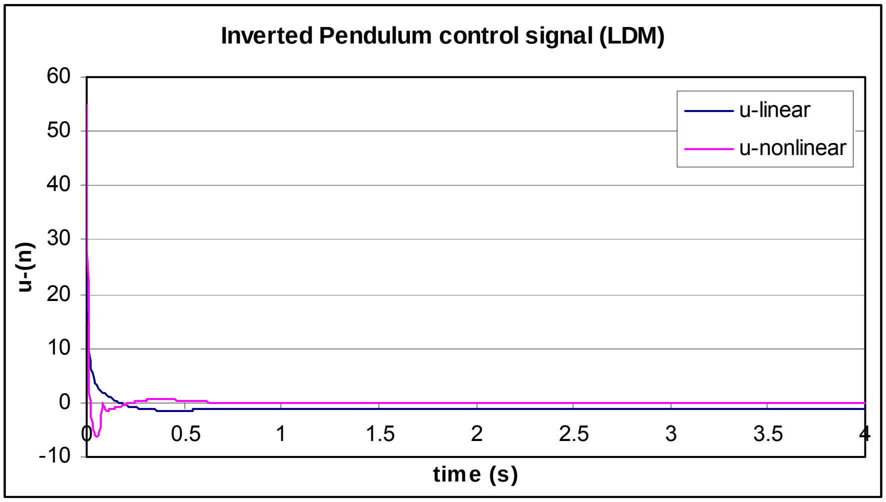


Figure 6.18: Inverted pendulum control signals when the set point is 0.05m

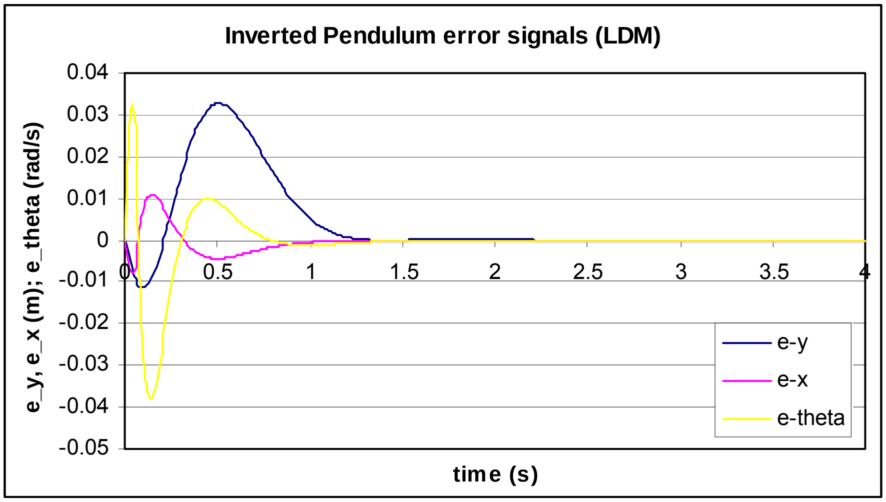


Figure 6.19: Inverted pendulum states error signals when the set point is 0.05m

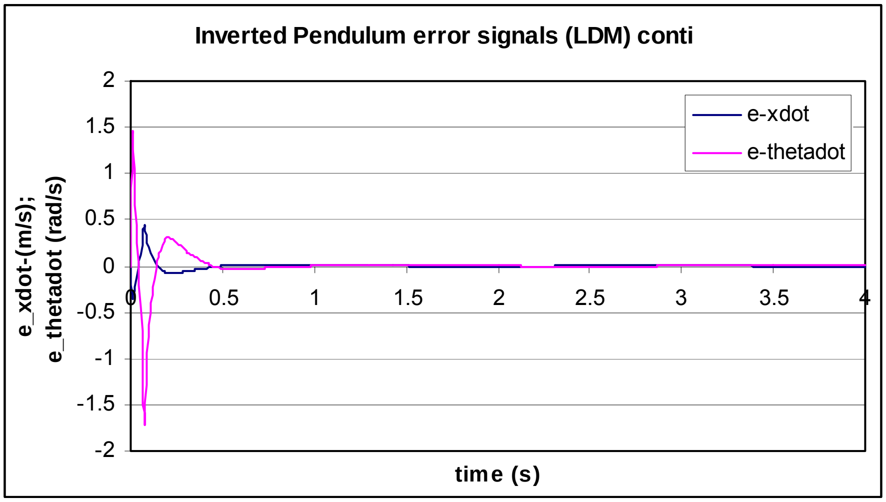


Figure 6.20: Inverted pendulum error signals when set point is 0 .05m

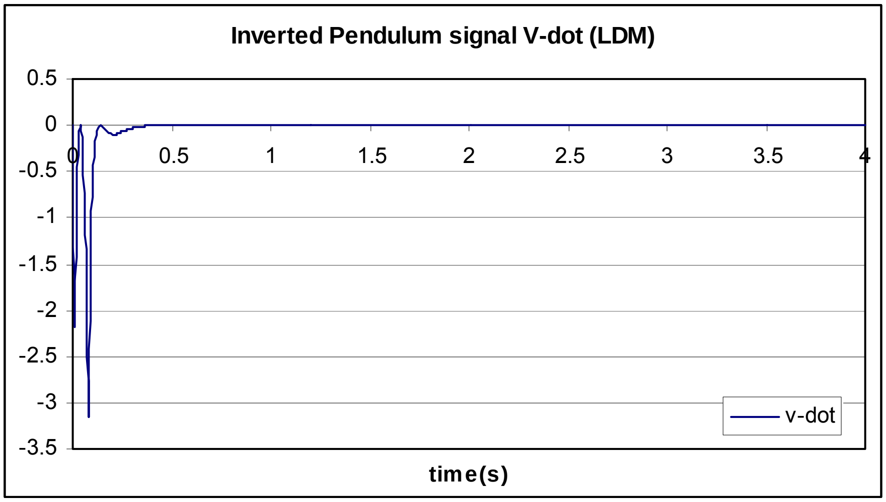


Figure 6.21: Inverted pendulum signal of Lyapunov function derivative when set point is 0.05m

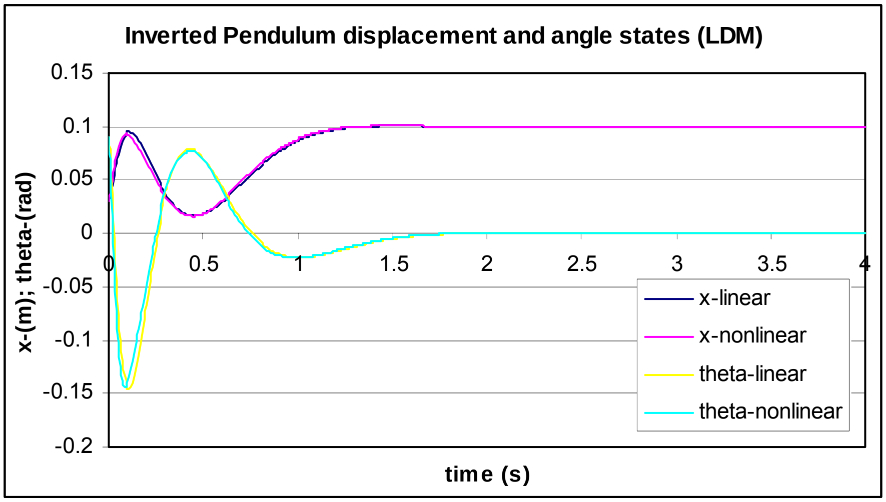


Figure 6.22: Inverted pendulum displacement and angle signals when the set point is 0.1m

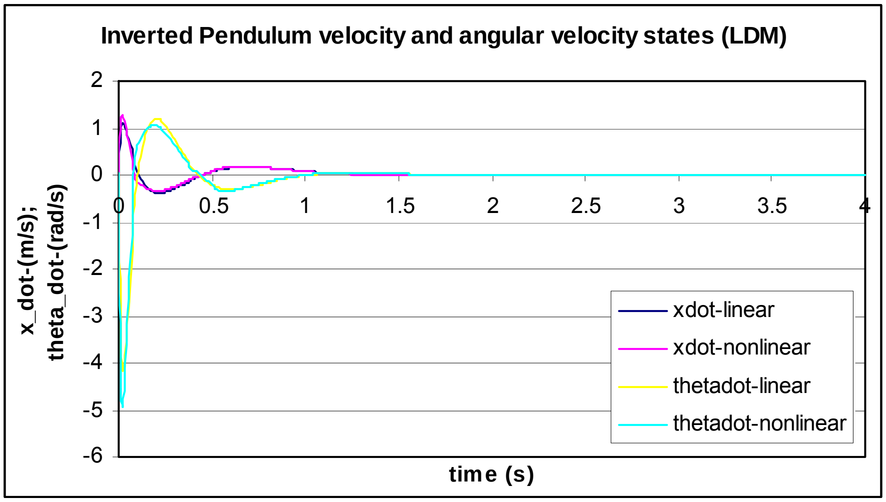


Figure 6.23: Inverted pendulum velocity and angular velocity state signals when the set point is 0.1m

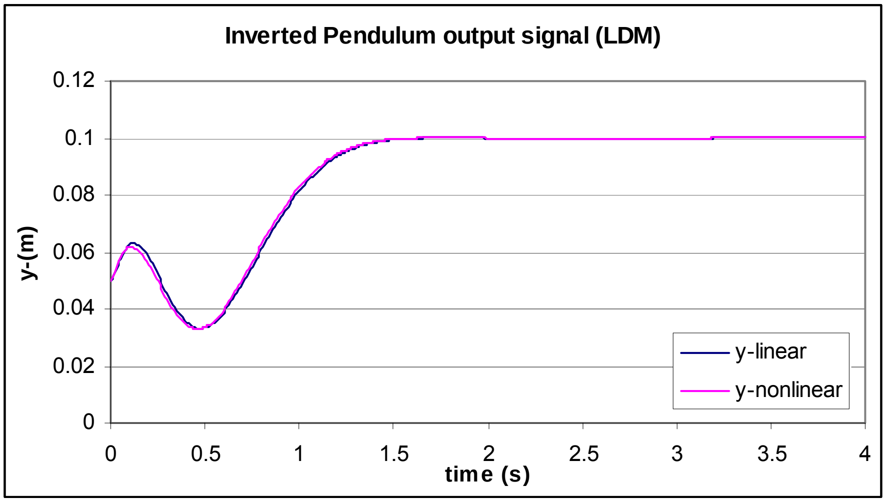


Figure 6.24: Inverted pendulum output signals when the set point is 0.1m

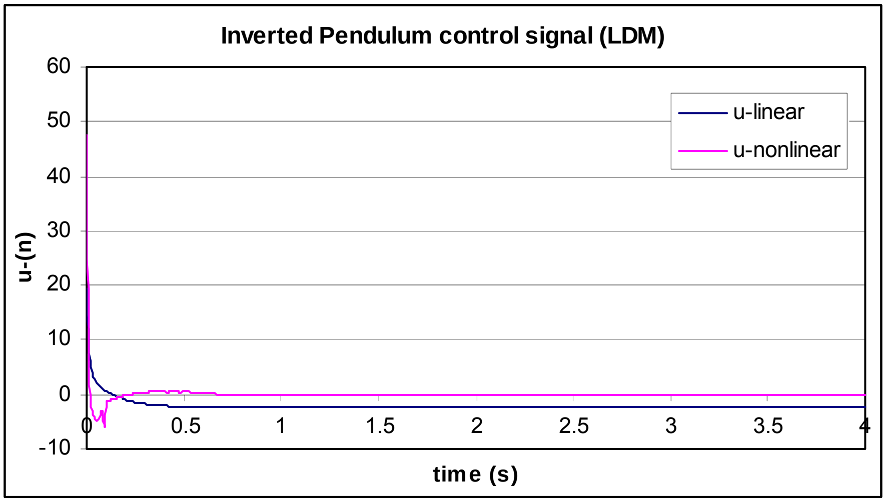


Figure 6.25: Inverted pendulum control signals when the set point is 0.1m

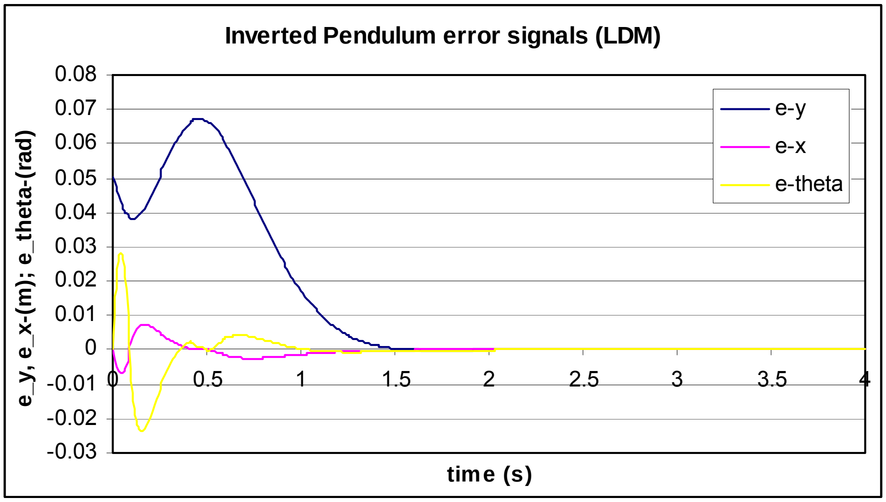


Figure 6.26: Inverted pendulum states error signals when the set point is 0.1m

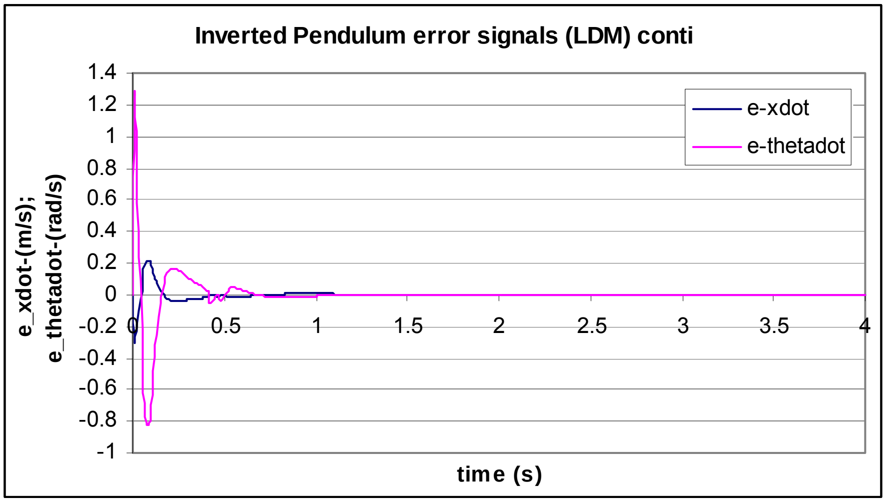


Figure 6.27: Inverted pendulum error signals when the set point is 0 .1m

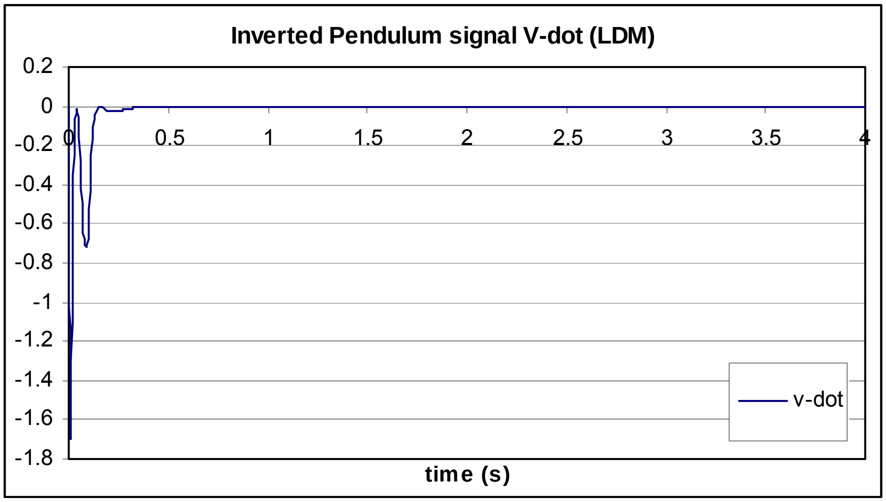


Figure 6.28: Inverted pendulum signal of Lyapunov function derivative when the set point is 0.1m

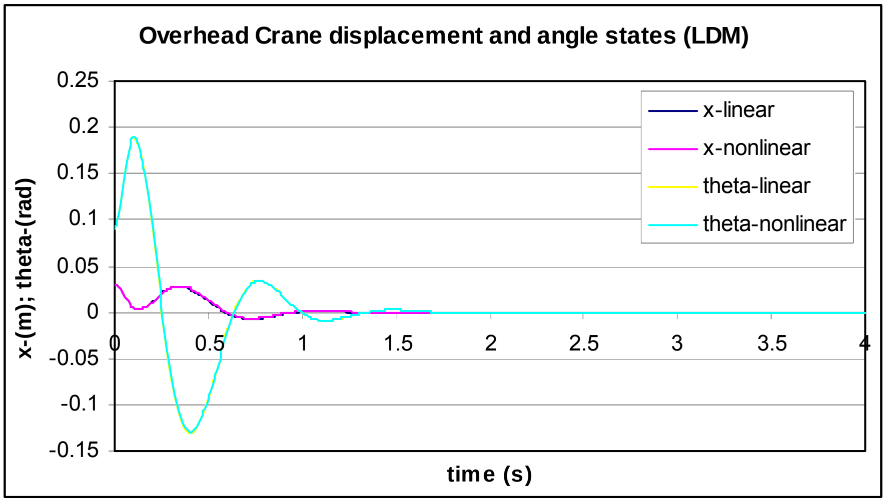


Figure 6.29: Overhead crane displacement and angle state signals when the set point is 0m

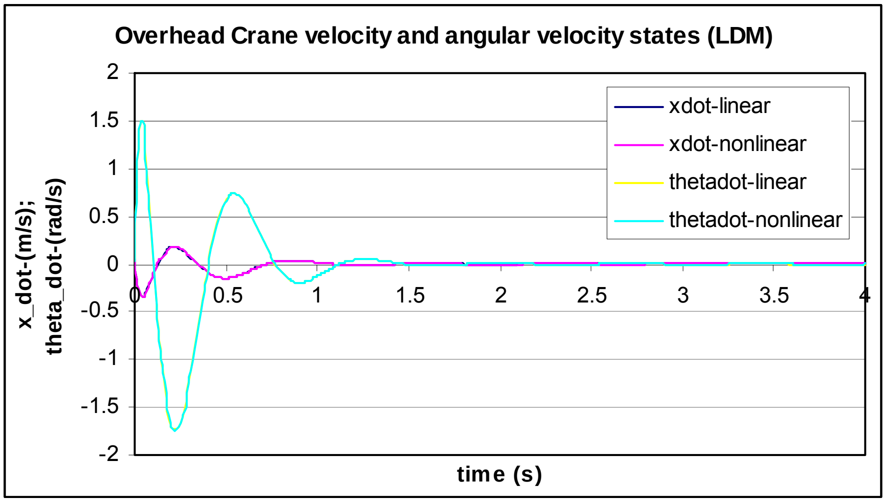


Figure 6.30: Overhead crane velocity and angular velocity state signals when the set point is 0m

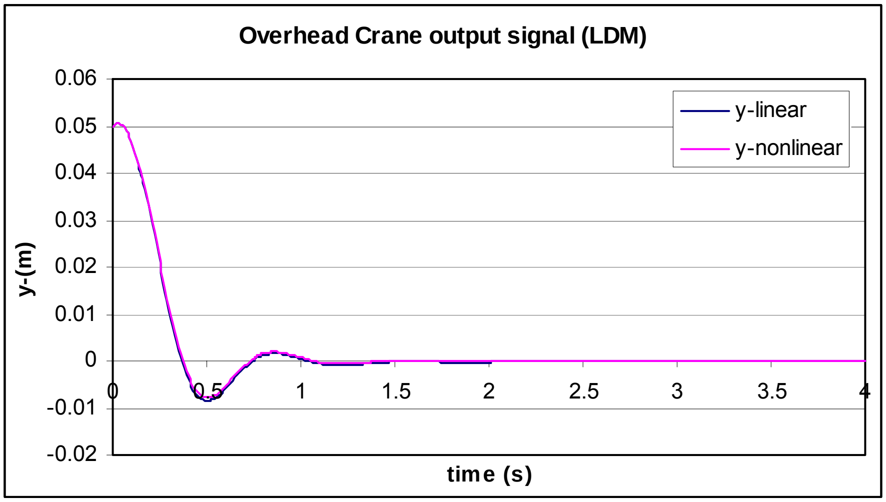


Figure 6.31: Overhead crane output signals when the set point is 0m

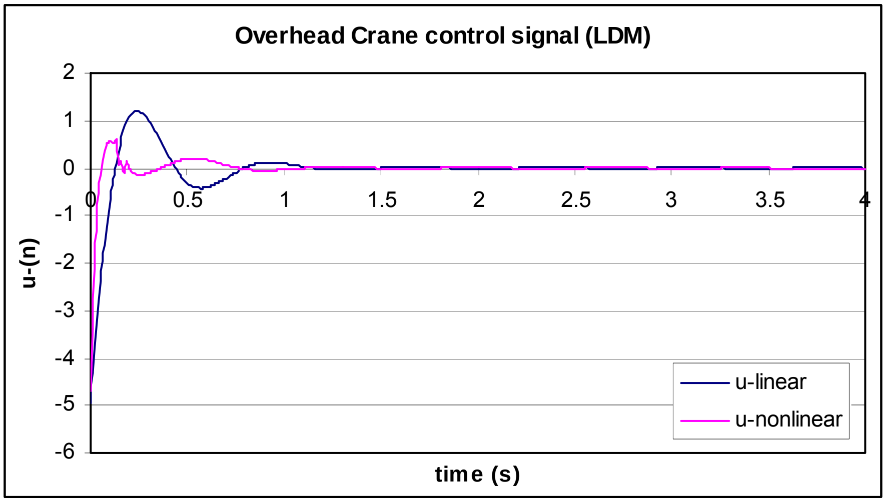


Figure 6.32: Overhead crane control signals when the set point is 0m

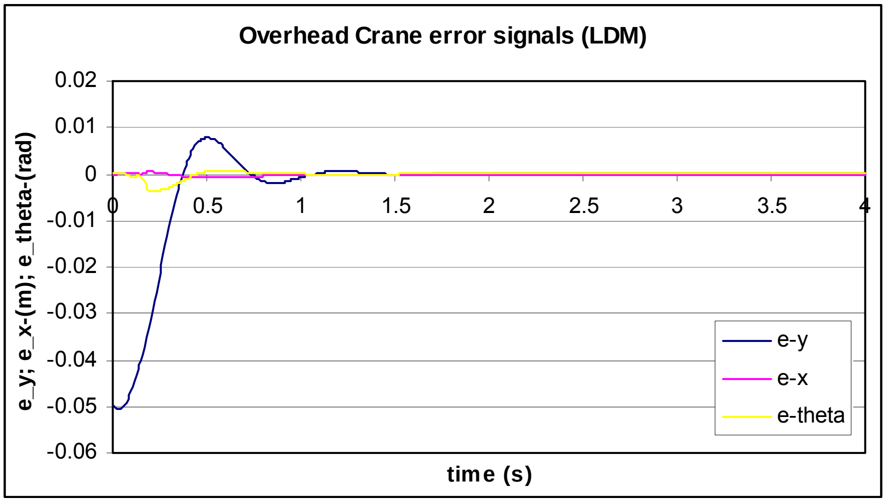


Figure 6.33: Overhead crane states error signals when the set point is 0m

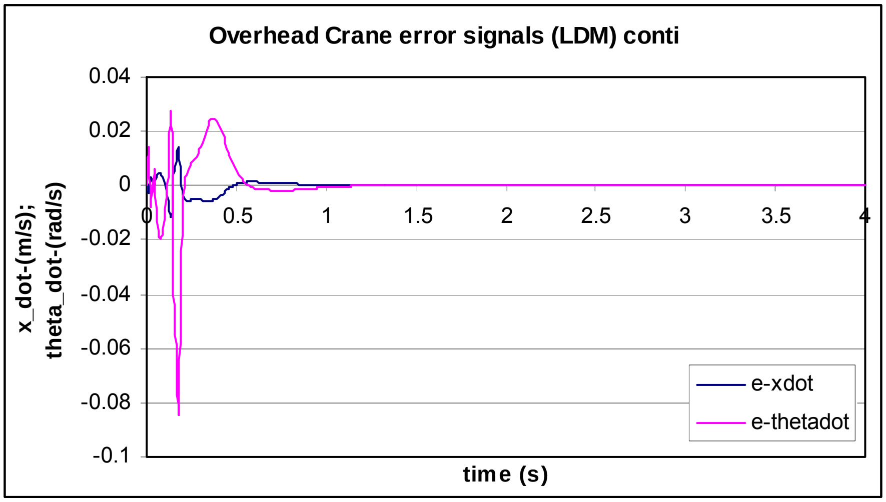


Figure 6.34: Overhead crane error signals when the set point is 0m

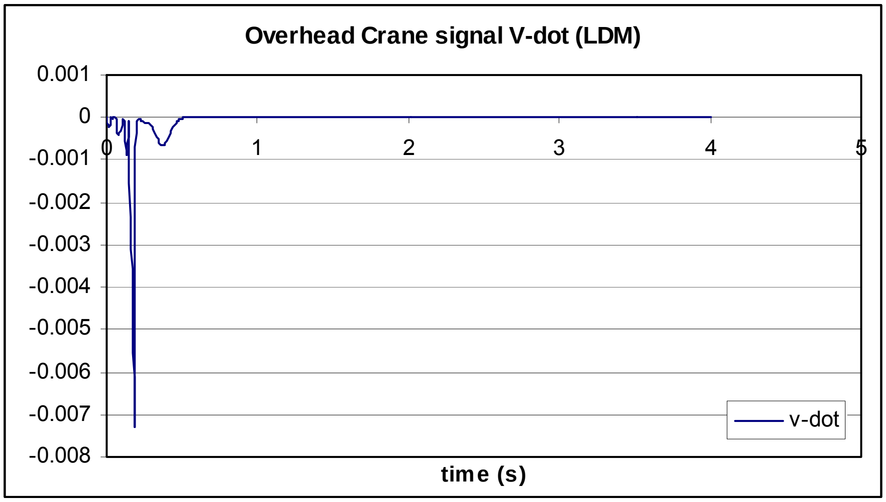


Figure 6.35: Overhead crane signal of Lyapunov function derivative when the set point is 0m

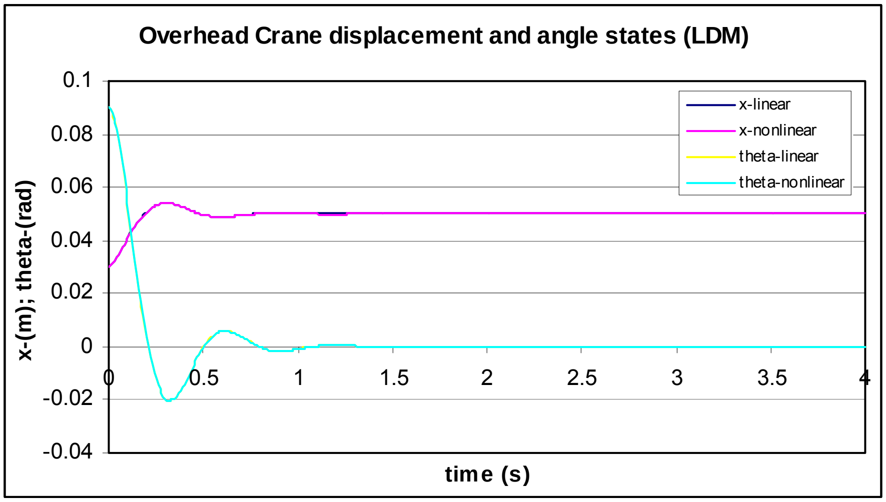


Figure 6.36: Overhead crane displacement and angle state signals when the set point is 0.05m

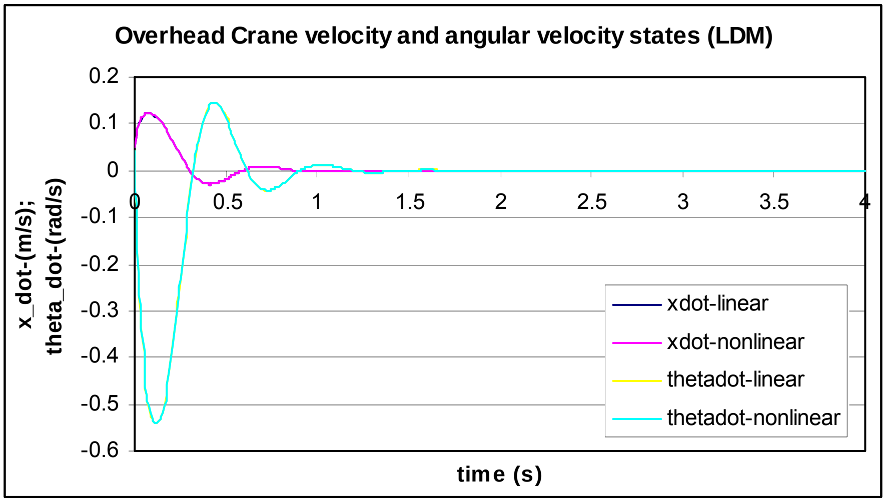


Figure 6.37: Overhead crane velocity and angular velocity state signals when the set point is 0.05m

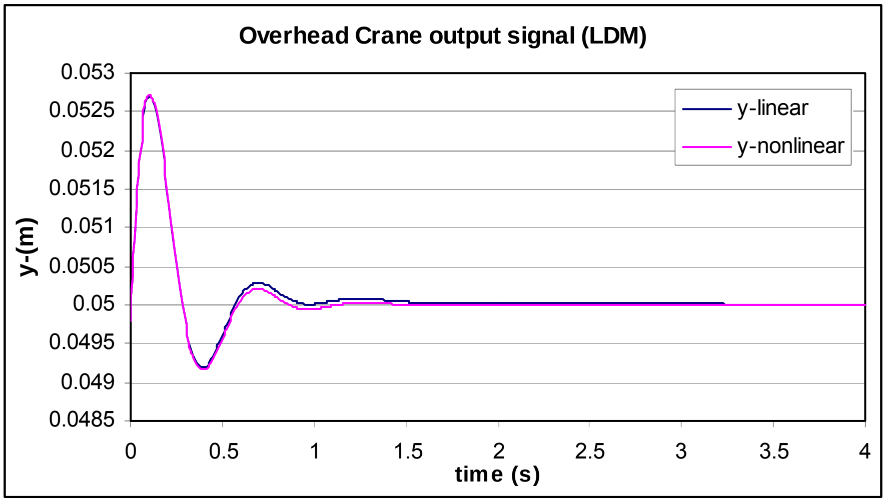


Figure 6.38: Overhead crane output signals when the set point is 0.05m

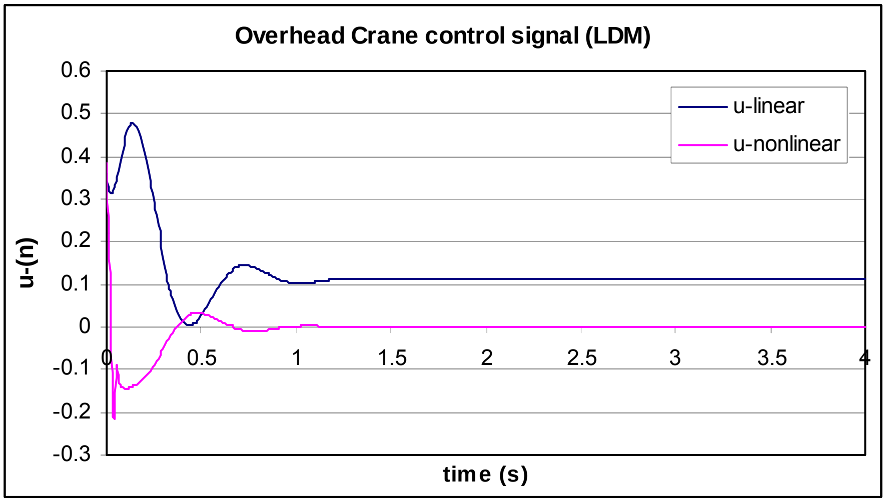


Figure 6.39: Overhead crane control signals when the set point is 0.05m

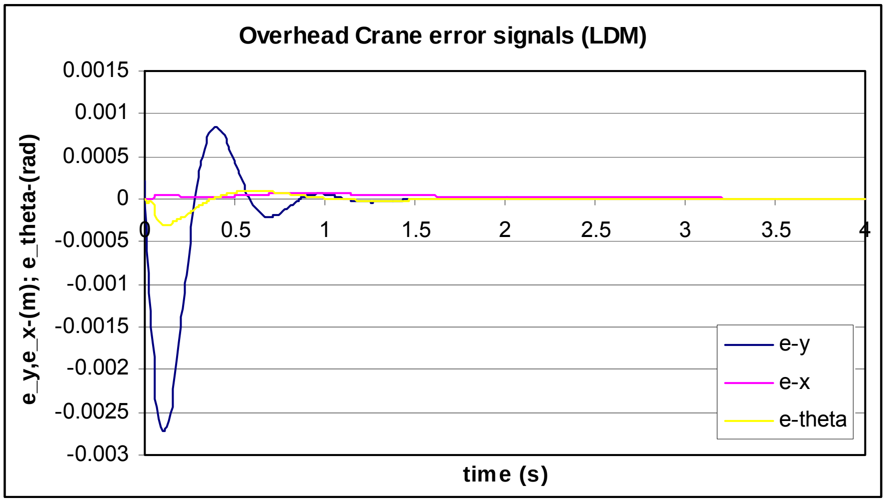


Figure 6.40: Overhead crane states error signals when the set point is 0.05m

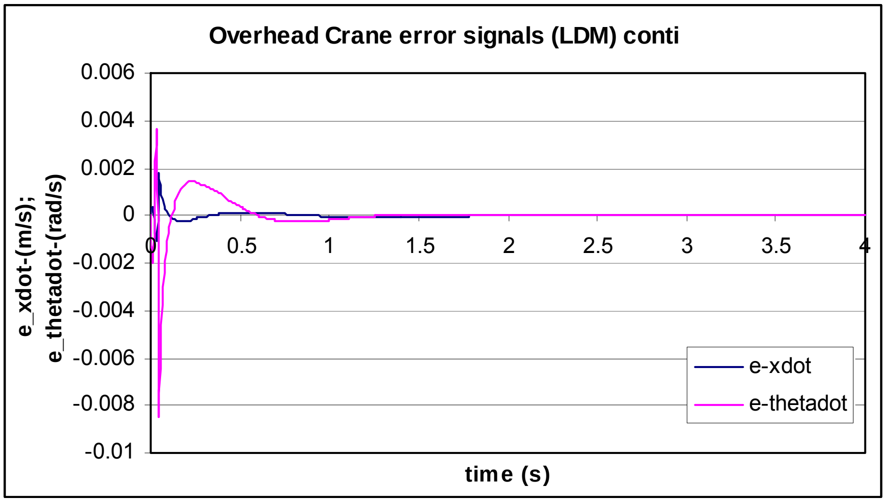


Figure 6.41: Overhead crane error signals when the set point is 0.05m

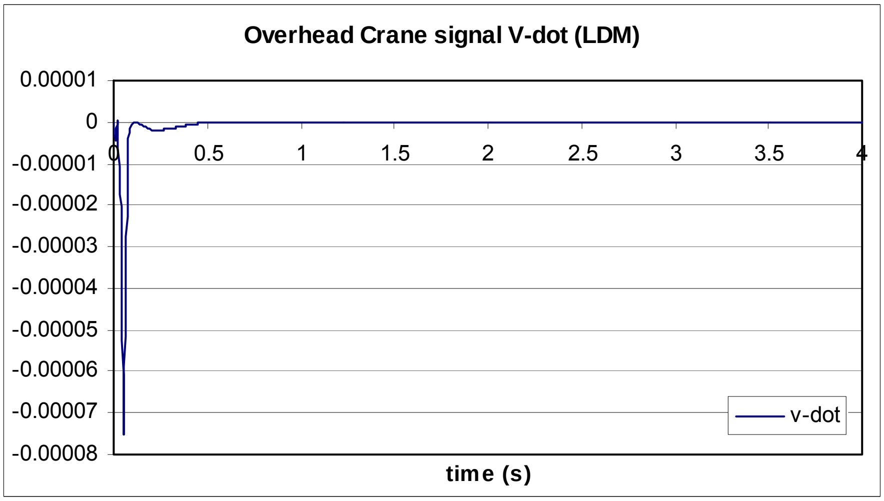


Figure 6.42: Overhead crane signal of Lyapunov function derivative when the set point is 0.05m

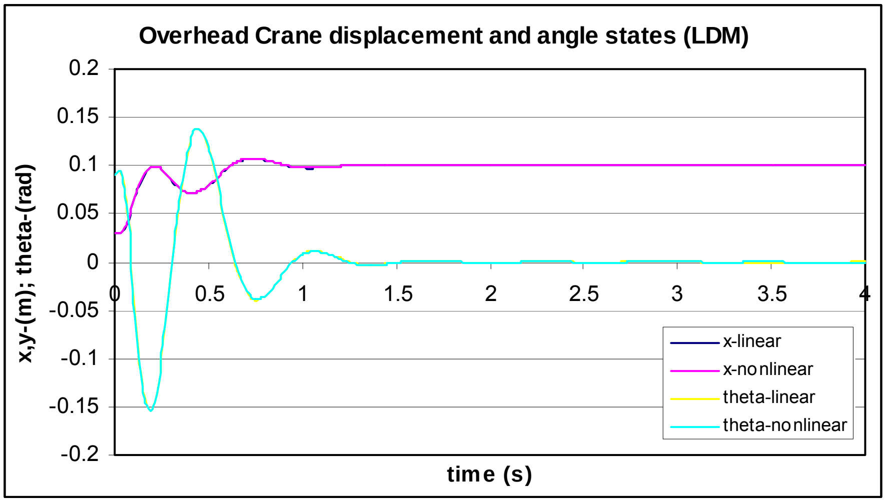


Figure 6.43: Overhead crane displacement and angle state signals when the set point is 0.1m

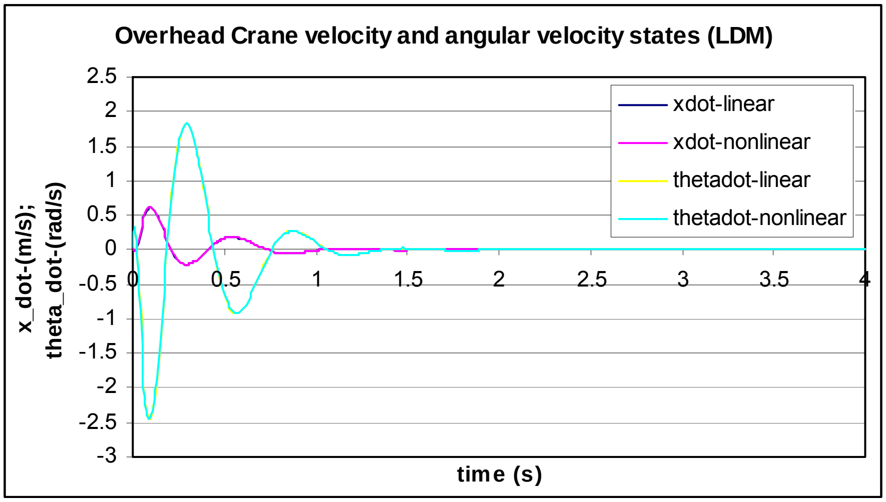


Figure 6.44: Overhead crane velocity and angular velocity state signals when the set point is 0.1m

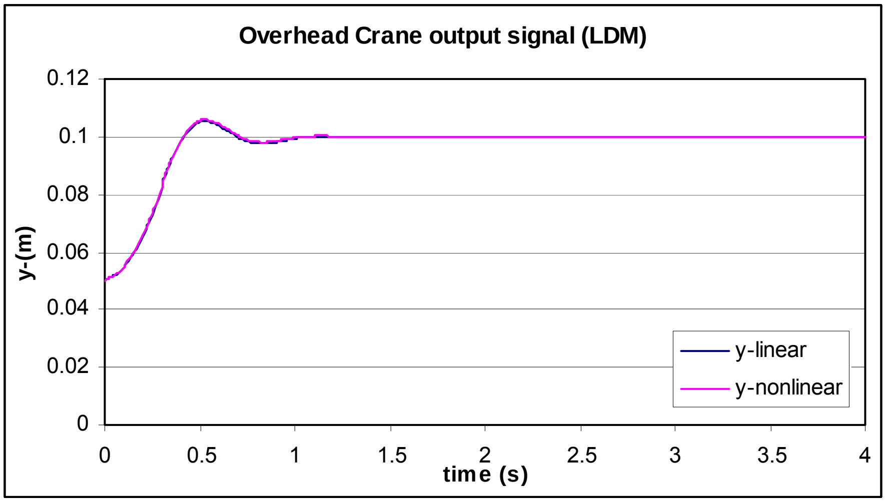


Figure 6.45: Overhead crane output signals when the set point is 0.1m

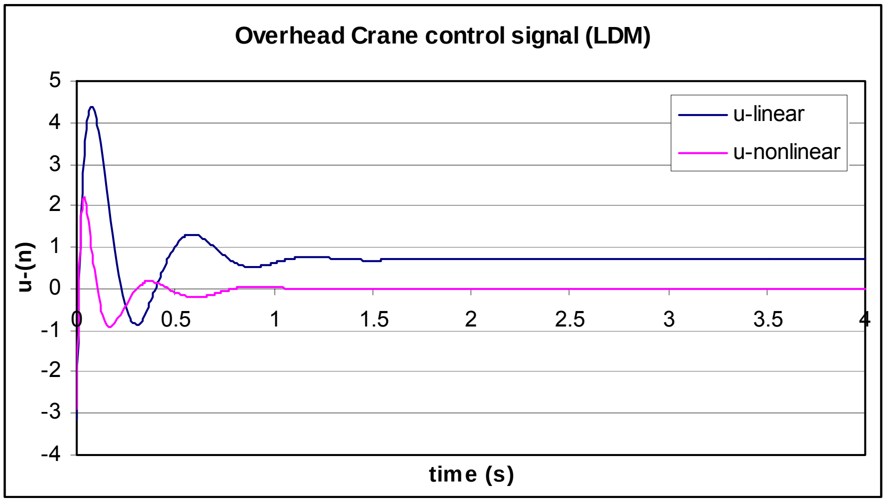


Figure 6.46: Overhead crane control signals when the set point is 0.1m

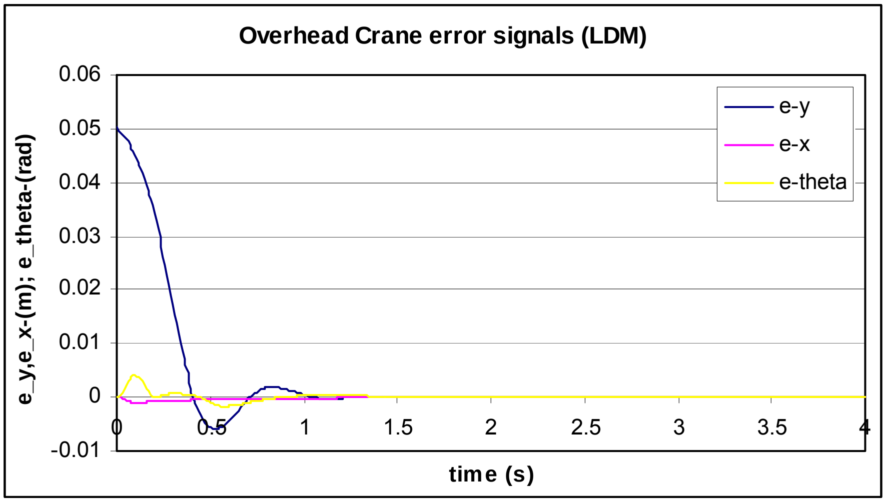


Figure 6.47: Overhead crane states error signals when the set point is 0.1m

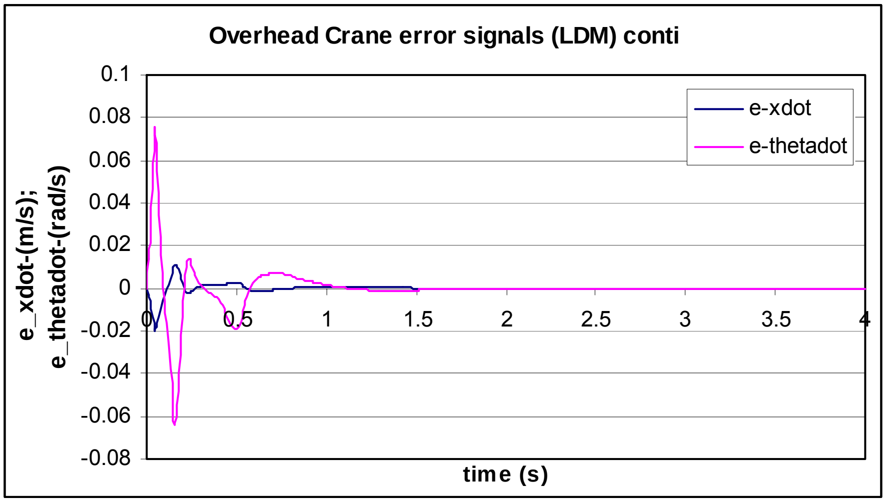


Figure 6.48: Overhead crane error signals when the set point is 0.1m

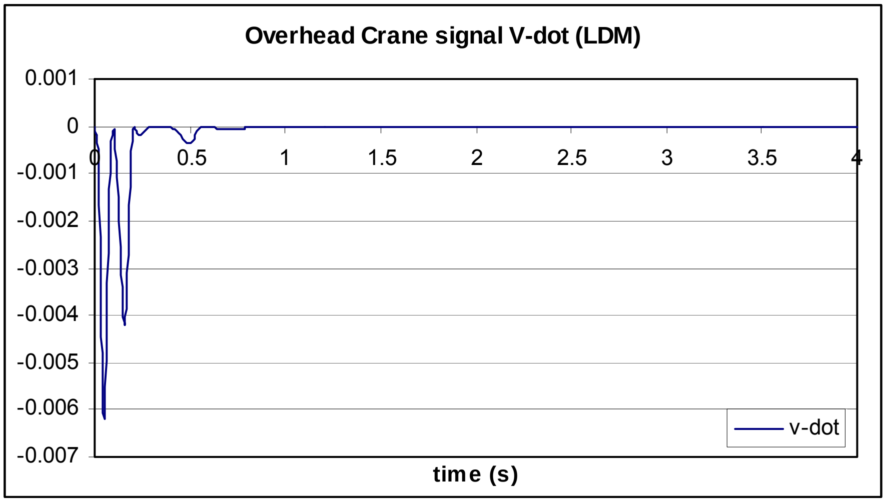


Figure 6.49: **Overhead crane signal of Lyapunov function derivative when set point is 0.1m**

6.4.4 Discussion of the results

From the simulation results, for both the inverted pendulum and the overhead crane, the following points can be summarized:

- All the states are stabilized
- The plant output follows the reference model and the set point trajectories
- The errors go to zeros
- The derivative of the Lyapunov function is smaller than 0 and when t→∞, $\dot{V} \to 0$

Table 6.3: **Table of the simulation results**

Set point	characteristics	Inverted pendulum	Overhead crane
0m	Time Delay	0	0
	Overshoot	40%	12%
	Rising Time	1s	0.02s
	Steady State error	0.001	0.001
	Settling Time	1.2s	1.3s
0.05m	Time Delay	0	0
	Overshoot	26%	5%
	Rising Time	0.8s	0.01
	Steady State error	0.001	0.001
	Settling Time	1,2s	0.8s
0.1m	Time Delay	0	0
	Overshoot	2%	3%
	Rising Time	1.2s	0.4s
	Steady State error	0.002	0.001
	Settling Time	1.3s	0.6s

Compared to the linear control designed in Chapter 5 for the linear model, the results obtained for the nonlinear control system show better achievement of the required specifications. The performance of system is formed in Table 6.3 for different set points according to requirements for overshoot rising time, steady state error, settling time. It can be seen that the results are better than the given at the beginning requirements.

6.5 Conclusion

The general procedures for developing the nonlinear controller based on the Lyapunov direct method is given in this chapter. It can be applied on various systems. The reference model plays a heavy role in the system. It must be designed optimally. In this chapter, there are two pairs of weighting matrices Q and R. The first pair is used to design the feedback gain for the liner control. The second pair is used to develop the reference model system. In order to get the best closed-loop simulation results, these two pairs of weighting matrices are adjusted separately. The simulation results demonstrate that the successful nonlinear controllers have been designed based on the Lyapunov direct method. The designed controllers are used in Chapter 8 to control of the inverted pendulum and the overhead crane.

CHAPTER SEVEN
FEEDBACK LINEARIZATION METHOD BASED MODEL REFERENCE CONTROL SYSTEM DESIGN

7.1 Introduction

As the feedback linearization theory is studied generally in Chapter 3. In this chapter, the feedback linearization theory is applied and developed for Single-Input-Single-Output (SISO) systems, Multi-Inputs-Multi-Outputs (MIMO) systems and Single-Input-Multi-Outputs (SIMO) systems. The feedback linearization is a principle of control which has drawn much attention. The central idea is to algebraically transform nonlinear systems dynamics into (fully or partly) linear one, so that the linear control techniques can be applied. This completely differs from conventional (Jacobian) linearization, because feedback linearization is achieved by exact state transformation and feedback, rather than by linear approximations of the dynamics. The advantages of feedback linearization are that the design of control can be used generally, in the sense that the same principle can be used on all systems of the right type. Moreover, extensions have been developed to take into account possible model inaccuracies. The successful applications of the feedback linearization method include control systems for helicopters, high-performance aircraft, industrial robots, biomedical devices and vehicle control.

In this thesis, the controller is related to the model reference controller because the nonlinear controller is designed to make the closed-loop system to behave linearly according to a specific transfer function model. This has the consequence that feedback linearization is basically subject to the same limitations as direct inverse control and shares the same characteristics, such as:

The advantages:

- Implementation is simple
- Only a model of the system to be controlled is required
- Tuning of the closed-loop response can be made without retraining of the model. An outer feedback can be introduced containing a linear pole placement controller.
- The linearization is exact not like the approximate linearization of the nonlinear model equations

and the disadvantages:

- Restricted to a particular class of systems. Difficult to resolve whether an unknown system actually belongs to this class
- Lack of design parameters for tuning of the controller
- Problems when the inverse system (locally) is unstable or near the stability border
- The linearizing effect of the controller depends on the values of the model parameters
- Very big control efforts sometimes have to be realized to linearize the closed-loop system

The nonlinear linearizing control linearizes the closed-loop system consisting of the nonlinear plant and the nonlinear controller with output u(t). This type of control is very sensitive to variations in the model parameters. In order to overcome these difficulties an additional linear control v(t) designed for the linearized system is needed. The basic idea of the feedback linearization approach is to use a control which consists of two components: one component cancels out the plant's nonlinearities and the other component controls the resulting linear system. The basic structure of the feedback linearization is shown in Figure 7.1. The method is limited to a class of dynamic systems or plants, whose models are efficiently smooth, that is, the plants whose right hand sides of the modelling differential equations are sufficiently many times differentiable.

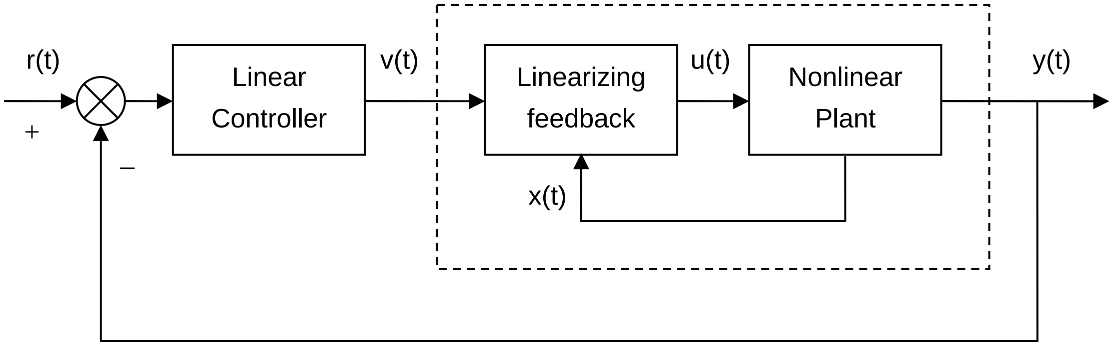


Figure 7.1: **Feedback linearized system with external linear controller**

This chapter is organized in the following way: In the section 7.2, the nonlinear system model as the plant to be controlled is presented. In the section 7.3, the feedback linearization theory is applied to different nonlinear systems, such as SISO systems, MIMO systems and SIMO system. Based on the study of the feedback linearization, the pendulum system is linearized in the section 7.4. The suitable linear controllers for controlling of the inverted pendulum and the overhead crane are designed in the section 7.5. The closed-loop system simulation results are represented in the section 7.6.

7.2 Nonlinear plant model

For a nonlinear plant model, the dynamic systems that are considered are of the type:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \quad x(0) = x_0 \tag{7.1}$$

$$y(t) = h(x,t) \tag{7.2}$$

where $f(x(t)) \in R^n$, $g(x(t)) \in R^{n \times m}$, $h(x,t) \in R^l$ are vector and matrix functions. g(x) can be written as: $g(x(t))=[g_1(x(t)) \quad g_2(x(t)) \cdots g_m(x(t))]$ where $g_i(x(t)) \in R^n$. It is assumed that f(0)=0 and h(0)=0.

The single input nonlinear system can be expressed as:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \quad x(0) = x_0 \tag{7.3}$$

$$y = h(x(t)) \tag{7.4}$$

A LTI model can be expressed as:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad x(0) = x_0 \tag{7.5}$$

$$y(t) = Cx(t) \tag{7.6}$$

From the analogy between the linear and nonlinear descriptions it can be seen that the linear model is a special case of the above nonlinear model in Equation (7.1) and (7.2).

The aim of control of the nonlinear plant is to design a state-feedback stabilizing control for the system in Equation (7.1) and (7.2). To construct such a controller it is necessary according to the theory to work with an equivalent system model rather than with the original one. The first step then is to reduce a nonlinear system model into one equivalent form that is a generalization of the controllable and observable form known from linear system theory. In order to obtain this form a special type of derivatives called Lie derivative are introduced.

Some related terminologies and definitions include relative degree ρ, internal dynamics, zero dynamics, diffeomorphisms and input-state linearization have been given in Chapter 3. In the feedback linearization theory, there are Input-State Linearization and Input-Output feedback linearization. In the input-state feedback linearization, the input is related with the state directly. The Input-State linearization is realized by a combination of a state transformation and an input transformation. It achieves the linearization of the input and state. Different from the input-state linearization, in the input-output feedback linearization, the output is indirectly related to the input. The relationship between the input and output is created by differentiating the output function until the input appears, and then designing the control to cancel the nonlinearity of the system. From the relative degree point of view, when the relative degree ρ equals to the system order n, the input-output linearization can be considered as the input-state linearization. In other words, the input-state linearization is a special case of the input-output linearization. For this project, since the inverted pendulum is a under actuated system, the input-output linearization is studied and applied to the pendulum system.

7.3 Study of the Feedback linearization approach

The input-output linearization theory can be applied to the stabilizing and tracking problems. For this particular project, the reference model generates the desired trajectory, the feedback linearization approach is used for the tracking the desired trajectory. Consider the following nonlinear system:

$$\dot{x}(t) = f(x(t), u(t)) \tag{7.7}$$

$$y(t) = h(x(t)) \tag{7.8}$$

where y is the system output. The objective of the tracking control is to keep the output y(t) follows the desired output trajectory $y_d(t)$. The difficulty is that the output y is indirectly related to the input u, through the state variable x and a nonlinear equation. A direct and simple relation between the system output y and the control input u can be created by continuously differentiating the output y. The approach that generates a linear input-output relation first and then formulating a controller based on linear control techniques is known as the input-output linearization approach. The approach can also be extended to the stabilizing control problem. In the following subsections, the input-output linearization approach is studied applied to SISO systems, MIMO systems and SIMO systems.

7.3.1 Single-Input-Single-Output (SISO) systems

Consider the nonlinear SISO system expressed as follows:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \; x(0) = x_0 \tag{7.9}$$

$$y = h(x(t)) \tag{7.10}$$

where f, g and h are with continuous behaviour and have continuous derivative of high order in some domain Ω, where $\Omega \in R^n$ and $x \in \Omega \in R^n$. The mapping (transformations) f: $0 \rightarrow R^n$ and g: $0 \rightarrow R^n$ are called vector fields on Ω.

It is necessary to find a control input:

$$u(t)=\alpha(x)+\beta(x)v \tag{7.11}$$

which can linearize and stabilize the nonlinear system expressed in Equation (7.9) and (7.10) according to a given reference trajectory $y^{sp}(t)$.

The input-output linearization means the dependence between the output y(t) and the input u(t) of the linearized system is a linear dependence. In order to find such dependence, it is necessary to express y(t) as a function of u(t) in the nonlinear system expressed in Equation (7.9) and (7.10). It can be seen that u(t) only appears in Equation (7.9), it does not appear in Equation (7.10). But if derivatives of y(t) according to time are taken, then Equation (7.9) can be substituted in these expressions for the derivatives, and u(t) can appear in the output equation. From this such obtained dependence u(t) can be selected to linearize the closed-loop system. Then the following steps are done in order to obtain the linearizing control u(t).

7.3.1.1 Differentiation of the output

The differentiation of the output Equation (7.10) is taken. The time t is omitted for clarity in the reading of the equations.

$$\dot{y} = \frac{\partial h}{\partial x} \cdot \dot{x} = \frac{\partial h}{\partial x}\left[f(x) + g(x)u\right] = L_f h(x) + L_g h(x)u \tag{7.12}$$

where $L_f h(x) = \frac{\partial h}{\partial x} f(x)$ is called the Lie derivative of h with respect to f, or along f, and $L_g h(x)$ is called the Lie derivative of h with respect to g, or along g.

The Equation (7.12) expresses the notion of the derivative h along the trajectory of the system $\dot{x} = f(x) + g(x)u$. The new notation is convenient when the repetition of differentiation of y=h(x) is done. The following rules are used:

$$\begin{aligned} L_g L_f h(x) &= \frac{\partial (L_f h)}{\partial x} g(x) \\ L_f^2 h(x) &= L_f L_f h(x) = \frac{\partial (L_f h)}{\partial x} f(x) \\ L_f^k h(x) &= L_f L_f^{k-1} h(x) = \frac{\partial \left(L_f^{k-1} h\right)}{\partial x} f(x) \\ L_f^0 h(x) &= h(x) \end{aligned} \tag{7.13}$$

If it appears in Equation (7.12) that $L_g h(x)=0$, then:

$$\dot{y} = L_f h(x) \tag{7.14}$$

that means $\dot{y}$ is independent on the control input u. New differentiation is necessary to be taken in order to find dependence between y and u. The second derivative of Equation (7.14) is:

$$\ddot{y} = y^{(2)} = \frac{\partial (L_f h)}{\partial x} \dot{x} = \frac{\partial (L_f h)}{\partial x}\left[f(x) + g(x)u\right] = L_f^2 h(x) + L_g L_f h(x)u \tag{7.15}$$

Once again if $L_g L_f h(x)=0$, then $y^{(2)}$ is independent on u and is:

$$y^{(2)} = L_f^2 h(x) \tag{7.16}$$

If the process of differentiation is repeated ρ-1 times and if:

$L_g L_f^{i-1} h(x) = 0$ for i=1, 2, ···, ρ-1

which means that u(t) does not appear in the equations for $y, \dot{y}, \ddot{y}, \cdots y^{(\rho-1)}$, and if finally it appears in the equation of $y^{(\rho)}$, then

$L_g L_f^{\rho-1} h(x) \neq 0$ and the ρ-th derivative is:

$$y^{(\rho)} = L_f^{\rho} h(x) + L_g L_f^{\rho-1} h(x) u \tag{7.17}$$

Equation (7.17) shows dependence between y and u, which means that the system expressed in Equation (7.9) and (7.10) is input-output linearizable. Equation (7.17) is obtained after ρ times differentiating of y according to time. The integer ρ is the relative degree of the system expressed in Equation (7.9) and (7.10). This means y(t) can be obtained by integrating the Equation (7.17) ρ times as shown in Figure 7.2. It can be written that the output y is obtained after ρ times integrating of the signal:

$$y^{(\rho)} = v \tag{7.18}$$

where v is a new input which is called the synthetic input or synthetic control. A linear relation between y and v is established.

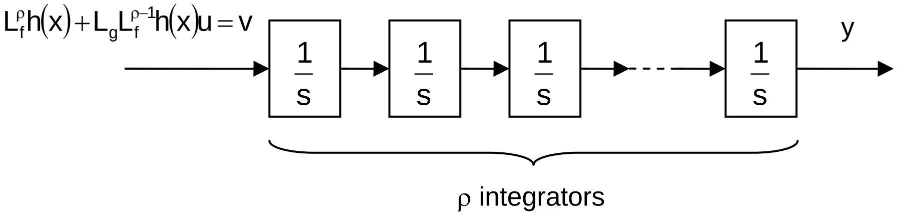


Figure 7.2: **Relationship between the new input and output**

If this signal v is known then the control can be calculated in order to cancel the nonlinearities $L_f^{\rho} h(x)$ and $L_g L_f^{\rho-1} h(x)$.

$$y^{(\rho)} = v = L_f^{\rho} h(x) + L_g L_f^{\rho-1} h(x) u \tag{7.19}$$

7.3.1.2 Linearization controller design for the SISO system

From Equation (7.17) and (7.18), the control input u can be expressed as:

$$u = \frac{1}{L_g L_f^{\rho-1} h(x)} \left[-L_f^{\rho} h(x) + v \right] \tag{7.20}$$

It can be verified that if Equation (7.20) is substituted in Equation (7.19), the Equation (7.18) is obtained as shown in Equation (7.21).

$$y^{(\rho)} = L_f^{\rho} h(x) + L_g L_f^{\rho-1} h(x) \cdot \frac{v - L_f^{\rho} h(x)}{L_g L_f^{\rho-1} h(x)} = L_f^{\rho} h(x) + v - L_f^{\rho} h(x) = v \qquad (7.21)$$

The equation $y^{(\rho)}=v$ is a linear equation, which means that if the system (7.19) is controlled by a control function which is expressed in Equation (7.20), the resultant closed-loop system will have linear behaviour.

The nonlinear system expressed in Equation (7.19) with its corresponding controller expressed in Equation (7.20) is graphically represented in Figure 7.3. The cancellation of the nonlinearities is exact and the input output behaviour of the linearized system is given by a chain of ρ integrators.

It can be seen that the nonlinear plant can be represented by a new set vector of state space:

$$\zeta = \left[y, \dot{y}, \ddot{y}, \cdots, y^{(\rho-1)}\right]^T \qquad (7.22)$$

or written by the Lie derivatives:

$$\zeta = \left[h(x) \quad L_f h(x) \quad \cdots \quad L_f^{\rho-1} h(x)\right]^T \qquad (7.23)$$

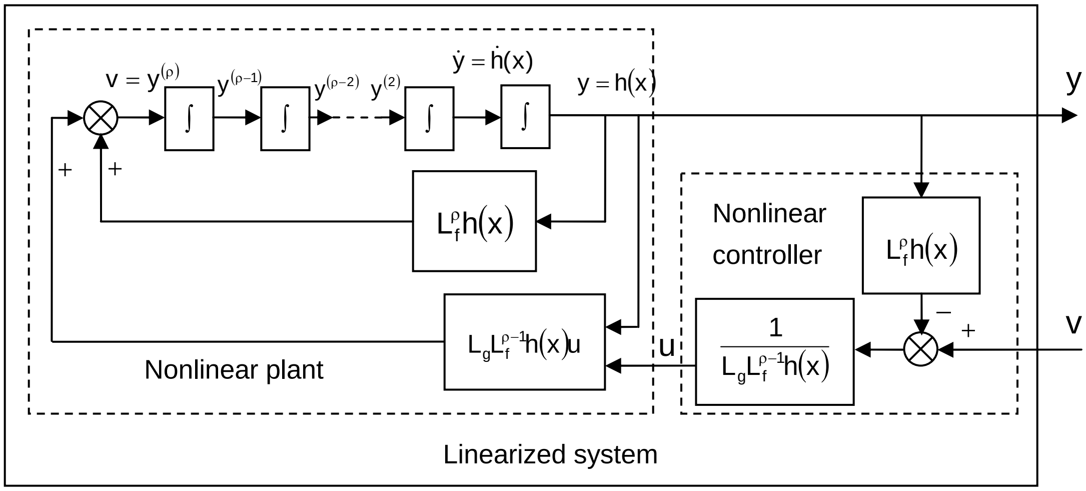


Figure 7.3: The linearizing loop

As the order of the system is $n \rightarrow x \in R^n$, then in order fully to represent the system in Equation (7.9) and (7.10) by Equation (7.17), the value of the relative degree ρ has to be equal to the order of the system, which is $\rho=n$. Then the system is fully controllable. If the relative degree is smaller than the system order, which is $\rho<n$, then there are dynamics that are unobservable in the input-output linearization. They cannot be influenced by the nonlinear control u. In such case the nonlinear control can only partly linearize the system, but the closed-loop system will be stable only if these unobservable dynamics are stable. As it is mentioned before, because these dynamics cannot be seen from the external input-output relationship, they are so called "internal dynamics". The study of the internal dynamics stability can be simplified locally by studying that of the zero-dynamics instead. If the zero-dynamics are unstable, different control strategies should be sought, only simplified by the fact that the transformed dynamics is partly linear.

7.3.1.3 The state transformation

A continuous differentiable map with a continuously differentiable inverse is known as a diffeomorphism. The definition of diffeomorphism has been given in Chapter 3. From the definition of the diffeomorphism, Lemma 7.1 is given as: Let $T(x)$ be a smooth function defined in a region Ω in R^n. If the Jacobian matrix ∇T is non-singular at a point $x=x_0$ of Ω, then $T(x)$ defines a local diffeomorphism in a sub-region of Ω. (Slotine and Li, 1991:233)

A diffeomorphism is used to transform a nonlinear system into another nonlinear system with new set of states. Referring to the nonlinear system expressed in Equation (7.9) and (7.10), the change of the states x to the new set of states ζ can be represented by a transformation T, then the new set of states ζ can be defined by:

$$\zeta = T(x) \tag{7.24}$$

Differentiation of ζ yields,

$$\dot{\zeta} = \frac{\partial T}{\partial x}\dot{x} = \frac{\partial T}{\partial x}\left[f(x) + g(x)u\right]$$

Then the new state space representation of the system can be expressed as:

$$\dot{\zeta} = f'(\zeta) + g'(\zeta)u \tag{7.25}$$

$$y = h'(\zeta) \tag{7.26}$$

The transformation must be invertible, which is, it must have an invertible map $T^{-1}(x)$ such that $x=T^{-1}(\zeta)$ for all $\zeta\in T(\Omega)$ where Ω is the domain of T, f, g. Moreover, because the derivatives of ζ and x should be continuous, it is necessary both T and T^{-1} to be continuously differentiable.

7.3.1.4 The normal forms

When the relative degree ρ is smaller than the system order n, the nonlinear system can be transformed, using $y, \dot{y}, \cdots, y^{(r-1)}$ as part of the new state components, into a so called normal form. The normal form provides the convenience of studying the internal dynamics and zero-dynamics, because the normal form makes part of the system dynamic linear. Consider a nonlinear system which is expressed in state space notation as:

$$\begin{aligned} \dot{y}_1 &= \dot{y} = y_2 \\ \dot{y}_2 &= \ddot{y} = y_3 \\ &\vdots \\ \dot{y}_{\rho-1} &= y^{(\rho-1)} = y_\rho \\ \dot{y}_\rho &= y^{(\rho)} = v \end{aligned} \tag{7.27}$$

If the nonlinear plant in state space notation is expressed by the vector ζ, then:

$$\begin{aligned} \dot{\zeta}_1 &= \zeta_2 \\ \dot{\zeta}_2 &= \zeta_3 \\ &\vdots \\ \dot{\zeta}_{\rho-1} &= \zeta_\rho \\ \dot{\zeta}_\rho &= \zeta^{(\rho)} \end{aligned} \tag{7.28}$$

From Equation (7.22), (7.27) and (7.28),

$$\zeta = [\zeta_1, \zeta_2, \zeta_3, \cdots \zeta_\rho]^T = [y, \dot{y}, \ddot{y}, \cdots, y^{(\rho-1)}]^T \tag{7.29}$$

The normal form of the system then can be expressed as:

$$\begin{aligned} \dot{\zeta}_1 &= \zeta_2 \\ \dot{\zeta}_2 &= \zeta_3 \\ &\vdots \\ \dot{\zeta}_{\rho-1} &= \zeta_\rho \\ \dot{\zeta}_\rho &= a(\zeta, \eta) + b(\zeta, \eta)u \end{aligned} \tag{7.30}$$

or $\dot{\eta} = \varphi(\zeta, \eta)$, where $\varphi(\zeta, \eta) \in R^n$ (7.31)

with the output defined as:

$$y = \zeta_1 \tag{7.32}$$

where $a(\zeta, \eta) = L_f^\rho h(x) = L_f^\rho h[T^{-1}(\zeta, \eta)]$ and $b(\zeta, \eta) = L_g L_f^{\rho-1} h(x) = L_g L_f^{\rho-1} h[T^{-1}(\zeta, \eta)]$, the $\zeta_i \in R^\rho$, $\eta_j \in R^{n-\rho}$ are referred to as normal coordinates or normal states at x_0.

In order to show that the system expressed in Equation (7.9) and (7.10) can be transformed in to the normal form, it is necessary to prove that a diffeomorphism T(x) exists such that Equation (7.30) and (7.31) are verified. T(x) is in the form of:

$$T(x) = \left[\zeta_1, \zeta_2, \cdots \zeta_\rho, \cdots \eta_1, \eta_2, \cdots \eta_{n-\rho}\right]^T \tag{7.33}$$

or $$T(x) = \begin{bmatrix} \zeta \\ \eta \end{bmatrix} \tag{7.34}$$

From Lemma 7.1, it is known that if the gradients $\nabla\zeta_i$ and $\nabla\eta_j$ are all linearly independent, then T is diffeomorphism. The gradients $\nabla\zeta_i$ are linearly independent. This can be shown by Lemma 7.2 which states that: If the relative degree of the system expressed in (7.9) and (7.10) is ρ in the region Ω, then the gradients $\nabla\zeta_1$, $\nabla\zeta_2$, …, $\nabla\zeta_\rho$ are linearly independent in Ω. (Slotine and Li, 1991:250)

Because the relative degree ρ is always smaller than the system order n, which means in order to complete the coordinate transformation, it is necessary to prove there are n-ρ more functions η_j exist and the gradients $\nabla\eta_j$ are all linearly independent as well. This can be proved in the same way of proving the gradients $\nabla\zeta_i$ are linearly independent by using Lemma 7.2. The complementary condition for the gradients $\nabla\eta_j$ to be linearly independent of the gradients $\nabla\zeta_I$ is to verify that:

$$\frac{\partial\eta_i}{\partial x} g(x) = 0 \text{ for } 1 \le i \le n\text{-}\rho \tag{7.35}$$

which ensures that:

$$\dot{\eta} = \frac{\partial\varphi}{\partial x}\dot{x} = \frac{\partial\varphi}{\partial x}\left[f(x) + g(x)u\right] = \frac{\partial\varphi}{\partial x} f(x) \tag{7.36}$$

In other words, Equation (7.36) describes the dynamics of the system do not depend on the control u.

At this point, it has been proven that the gradients $\nabla\zeta_i$ and $\nabla\eta_j$ are all linearly independent. As stated at the beginning, if the gradients $\nabla\zeta_i$ and $\nabla\eta_j$ are all linearly independent, then diffeomorphism T exists. Furthermore, if T is proven to exist, then the nonlinear system stated in Equation (7.9) and (7.10) can be expressed in the normal form as in Equation (7.30), (7.31) and (7.32).

7.3.1.5 The zero-dynamics

The normal form in Equation (7.30) decomposes the system described in Equation (7.9) and (7.10) into an external part ζ and an internal part η. The external part is linearized by the state feedback control $u=\alpha(x)+\beta(x)v$ and it represents the linear relation between y and v. The input v can be designed to have desired output y. The internal part is unobservable for this control. It is important to determine the behaviour of the internal dynamics. The stability of the system internal-dynamics can be studied by examining the stability of the zero-dynamics. The definition of the zero-dynamics is given in Chapter 3.

Here, the zero-dynamics is studied referring to the system stability. Setting $\zeta=0$ in Equation (7.36), it gives the equation:

$$\dot{\eta} = f_o(\eta, 0) \tag{7.37}$$

which is called zero dynamics. It is an intrinsic feature of a nonlinear system and does not depend on the choice of control law or the desired trajectories. The system is said to be asymptotically minimum phase if Equation (7.34) has an asymptotically stable equilibrium point in the domain of interest. In the case $n=\rho$, the system has not zero dynamics and, by default, is said to be minimum phase.

7.3.1.6 Linear controller design and the study of the system stability

By using the feedback linearization method, the nonlinearity of the nonlinear system is cancelled. Therefore, the linear control techniques, such as the most popular ones: pole-placement method and LQR method can be used to design the linear controller based on the linear input-output relation and then the internal dynamics stability is checked. For the stabilization problem, the controller for the system expressed in Equation (7.9) and (7.10) is stated as:

$$v = -k_{\rho-1}y^{(\rho-1)} - \cdots - k_1\dot{y} - k_0 y \tag{7.38}$$

If k_i are chosen carefully such that the polynomial

$$K'(p) = p^{\rho} + k_{\rho-1}p^{r-1} + \cdots + k_1 p + k_0 \tag{7.39}$$

has all its roots strictly in the left-half plane, then the complete controller can be constructed by using Equation (7.20) and (7.39) as:

$$u = \frac{1}{L_g L_f^{\rho-1}h(x)}\left[-L_f^{\rho}h(x) + v\right] = \frac{1}{L_g L_f^{\rho-1}h(x)}\left[-L_f^{\rho}h(x) - k_{\rho-1}y^{(\rho-1)} - \cdots - k_1\dot{y} - k_0 y\right] \tag{7.40}$$

The control law expressed in Equation (7.40) stabilizes the whole system locally.

If the system relative degree equals to its order, the control law expressed in Equation (7.40) can guarantee the closed loop system global asymptotic stability, because there is no internal dynamics. But when the system relative degree is smaller than its order, the system can only be partial feedback linearized, the stability of the zero-dynamics can only ensure the local stability of a control system based on input-output linearization. In order to obtain the control law that can achieve the global stability for the partial feedback linearization, a standard Lyapunov method is used. The basic idea or this approach is to put the nonlinear system into the normal form which makes part of the dynamics linear first. Then, refereeing to the normal form Equation (7.30), its input is ζ and output is η. The internal dynamics are stabilized by a control expressed in Equation (7.41):

$$\zeta_0 = \zeta_0(\eta) \tag{7.41}$$

An associated Lyapunov function V_0 is used to show the internal dynamics stability. The internal dynamic has lower order than the original system, so the control law is generally easier than finding a stabilizing control law for the original system. Thirdly, back to the global control problem, a Lyapunov function candidate V is appropriately defined as a modified version of V_0. By carefully choosing the control input v, the newly defined Lyapunov function candidate V can meet the requirement of being the Lyapunov function for the whole closed-loop system. The obtained control law based on partial linearization can guarantee the global stability of the whole system.

For the tracking problem, the pole-placement method and LQR method are used. Consider the nonlinear system in Equation (7.9) and (7.10), and the problem of tracking a given desired trajectory $y_d(t)$. Let

$$\zeta_d = \begin{bmatrix} y_d & \dot{y}_d & \cdots & y_d^{\rho-1} \end{bmatrix}^T \tag{7.42}$$

and define the tracking error vector as:

$$\tilde{\zeta}(t) = \zeta(t) - \zeta_d(t) \tag{7.43}$$

Then the following result is obtained:

Theorem 7.1: Assume that the system in Equation (7.9) and (7.10) has relative degree ρ (defined and constant over the region of interest), that ζ_d is smooth and bounded, and that the solution η_d of the equation

$$\dot{\eta}_d = \varphi(\zeta_d, \eta_d),\ \ \eta_d(0) = 0 \tag{7.44}$$

exists, is bounded, and is uniformly asymptotically stable. Choose constant k_i such that the polynomial in Equation (7.39) has all its roots strictly in the left-half plane. Then, by using the control law:

$$u = \frac{1}{L_g L_f^{\rho-1}\zeta_1(x)}\left[-L_f^{\rho}\zeta_1 + y_d^{(r)} - k_{\rho-1}\tilde{\zeta}_r - \cdots - k_1\dot{y} - k_0\tilde{\zeta}_1\right] \tag{7.45}$$

The whole state remains bounded and the tracking error $\tilde{\zeta}$ converges to zero exponentially. (Slotine and Li, 1991:256)

7.3.2 Multi-Input-Multi-Output (MIMO) systems

The concepts used in the SISO system, such as input-output linearization, zero-dynamics, internal-dynamics and so on, can be extended to the MIMO systems. The MIMO system considered here is a square system in a neighbourhood of point x_0, which has the same number of inputs and outputs:

$$\begin{aligned} &\dot{x} = f(x) + g_1(x)u_1 + g_2(x)u_2 + \cdots + g_l(x)u_l \\ &y_1 = h_1(x) \\ &y_2 = h_2(x) \\ &\vdots \\ &y_l = h_l(x) \end{aligned} \tag{7.46}$$

where $x \in R^n$, $u \in R^l$, $y \in R^l$, f and g_l are assumed to be smooth vector fields and h_l are smooth functions. The problem for design of control is to find a static state feedback that linearizes the system in Equation (7.46).

7.3.2.1 Differentiation of the j-th output of the system

The derivatives of the j-th output of the system expressed in Equation (7.46) are taken:

$$\dot{y}_j = L_f h_j + \sum_{i=1}^{l}\left(L_{g_i} h_j\right)u_l \tag{7.47}$$

If each of the $L_{g_i} h_j(x) = 0$, then the inputs u_j do not appear in the Equation (7.47).

If χ_j is the smallest derivative integer, such that at least one of the inputs appears in $y_j^{\chi_j}$, that is:

$$y_j^{\chi_j} = L_f^{\chi_j} h_j + \sum_{i=1}^{l} L_{g_i}\left(L_f^{\chi_j - 1} h_j\right)u_i\ ,\ \ j = \overline{1,l} \tag{7.48}$$

with at least one of the $L_{g_i}\left(L_f^{\chi_j - 1} h_j\right) \neq 0$, for some x. After differentiation of all outputs, the similar Equation (7.48) can be written.

If Equation (7.48) is written in a vector matrix form, it will have the following representation:

$$\begin{bmatrix} y_1^{\chi_1} \\ y_2^{\chi_2} \\ \vdots \\ y_j^{\chi_l} \end{bmatrix} = \begin{bmatrix} L_f^{\chi_1}h_1 \\ L_f^{\chi_2}h_2 \\ \vdots \\ L_f^{\chi_l}h_2 \end{bmatrix} + \begin{bmatrix} L_{g_1}L_f^{\chi_1-1}h_1 & L_{g_2}L_f^{\chi_1-1}h_1 & \cdots & L_{g_l}L_f^{\chi_1-1}h_1 \\ L_{g_1}L_f^{\chi_2-1}h_2 & L_{g_2}L_f^{\chi_2-1}h_2 & \cdots & L_{g_l}L_f^{\chi_2-1}h_2 \\ \vdots & & & \\ L_{g_1}L_f^{\chi_l-1}h_l & L_{g_2}L_f^{\chi_l-1}h_l & \cdots & L_{g_l}L_f^{\chi_l-1}h_l \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_l \end{bmatrix} \tag{7.49}$$

The matrix in front of the control vector u is represented by the notation E(x).

7.3.2.2 Calculation of the nonlinear control

Equation (7.49) is used for linearization of the closed-loop system. It can be seen that matrix E(x) must be non-singular one which means that det[E(x)]≠0 for some given point. From Equation (7.49), the nonlinear control u can be expressed as:

$$u = -E^{-1}(x)\begin{bmatrix} L_f^{\chi_1}h_1 \\ L_f^{\chi_2}h_2 \\ \vdots \\ L_f^{\chi_l}h_2 \end{bmatrix} + E^{-1}(x)v \tag{7.50}$$

yields a linear closed-loop system:

$$\begin{bmatrix} y_1^{\chi_1} \\ y_2^{\chi_2} \\ \vdots \\ y_l^{\chi_l} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_l \end{bmatrix} \tag{7.51}$$

The control and closed-loop system can be expressed by Equation (7.50) and (7.51), only if the system (7.46) has a relative degree vector $[\chi_1, \chi_2 \cdots \chi_l]^T$, where $\rho = \sum_{i=1}^{l} \chi_i$ at a point $x=x_0$, and if:

$$L_{g_i}L_f^k h_i(x) = 0 \text{ for } 0 \le k \le \chi_j\text{-}2 \tag{7.52}$$

and $L_{gi}L_f^{\chi_j-1}h_i(x) \neq 0$, which means that the matrix $A(x_0)$ is not singular. If ρ=n, there is no internal dynamics, with the control law in Equation (7.50), an input-state linearization of the original system is obtained and the system expressed in Equation (7.46) is fully linearizable.

7.3.2.3 Decoupling of the linearized system

From Equation (7.51) can be seen that the closed-loop system can be decoupled to l separate subsystems. Thus, decoupling is a by-product of linearization. Equation (7.51) is called a decoupling control law. The invertible matrix E(x) is called the decoupling matrix of the system. The MIMO system in Equation (7.46) then has relative degree of (ρ_1, …, ρ_l) at x_0, and the scalar $\rho=\rho_1+ \ldots +\rho_l$ is called the total relative degree of the system at x_0.

7.3.2.4 Normal form of the system

The normal form of the MIMO system is written for every of the outputs in the same way as for SISO system where the system state has two parts – the controllable part and non-controllable part.

7.3.3 System with Single-Input-Multi-Outputs (SIMO)

The SIMO system can be represented in the following form of:

$$\begin{aligned} &\dot{x} = f(x) + g(x)u \\ &y_1 = h_1(x) \\ &y_2 = h_2(x) \\ &\vdots \\ &y_l = h_l(x) \end{aligned} \tag{7.53}$$

where $x\in R^n$, $u\in R^l$, $y\in R^l$ and f, g are assumed to be smooth vector fields and h_j are smooth functions.

7.3.3.1 Differentiation of the j-th output of the system

Following the same procedure in SISO and MIMO system, in order to find the relation between output y and control input u, differentiation of the j-th output of the system Equation (7.53) is taken:

$$\dot{y}_j = L_f h_j + L_g h_j u \tag{7.54}$$

If $L_g h_j = 0$, the Equation (7.54) does not depend on u.

If χ_j is the smallest derivative for which u appears in the $y_j^{\chi_j}$, it can be written as:

$$y_j^{\chi_j} = L_f^{\chi_j} h_j + L_g L_f^{\chi_{j-1}} h_j(x)u\,, \;\; j = \overline{1,l} \tag{7.55}$$

where $L_g L_f^{\chi_{j-1}} h_j(x) \neq 0$.

After differentiation of all outputs as in Equation (7.55) the common presentation of the linearizable system can be written in a vector-matrix form:

$$\begin{bmatrix} y_1^{\chi_1} \\ y_2^{\chi_2} \\ \vdots \\ y_l^{\chi_l} \end{bmatrix} = \begin{bmatrix} L_f^{\chi_1} h_1(x) \\ L_f^{\chi_2} h_2(x) \\ \vdots \\ L_f^{\chi_l} h_l(x) \end{bmatrix} + \begin{bmatrix} L_g L_f^{\chi_1 -1} h_1(x) \\ L_g L_f^{\chi_2 -1} h_2(x) \\ \vdots \\ L_g L_f^{\chi_l -1} h_l(x) \end{bmatrix} u = b(x) + a(x)u = v \qquad (7.56)$$

Equation (7.56) is used for linearization of the closed-loop system. The vector in front of u can be written as a vector a(x).

7.3.3.2 Calculation of the nonlinear control

In this case, in order to obtain full linearization of the closed-loop system, it is necessary the system expressed in Equation (7.56) to have full relative degree $\rho = \chi_1 + \chi_2 + \cdots + \chi_l = n$. Then Equation (7.56) can be solved according to the control u.

$$b(x) + a(x)u = v \qquad (7.57)$$

$$a(x)u = v - b(x) \qquad (7.58)$$

$$a^T(x)a(x)u = a^T(x)v - a^T(x)b(x) \qquad (7.59)$$

Every term of the Equation (7.59) multiply by $\left[a^T(x)a(x)\right]^{-1}$, then:

$$\left[a^T(x)a(x)\right]^{-1} a^T(x)a(x)u = \left[a^T(x)a(x)\right]^{-1} a^T(x)v - \left[a^T(x)a(x)\right]^{-1} a^T(x)b(x) \qquad (7.60)$$

The control law is obtained by simplifying Equation (7.60).

$$u = \left[a^T(x)a(x)\right]^{-1} a^T(x)v - \left[a^T(x)a(x)\right]^{-1} a^T(x)b(x) \qquad (7.61)$$

The control given by Equation (7.61) is linearizing the closed-loop system which can be presented as:

$$\begin{bmatrix} y_1^{\chi_1} \\ y_2^{\chi_2} \\ \vdots \\ y_l^{\chi_l} \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_l \end{bmatrix} \qquad (7.62)$$

The condition for existence of (7.61) and (7.62) is that all $L_g L_f^{\chi_j -1} \neq 0$.

Referring to nonlinear SISO, MIMO and SIMO system, the input-output feedback linearization are studied comprehensively till now in this chapter. Based on the study of the feedback linearization theory, the input-output feedback linearization approach is applied to control the inverted pendulum and the overhead crane. The detailed work is presented in the following sections.

7.4 Design of an input-output linearizing controller for the pendulum system

The pendulum system can be configured into the inverted pendulum mode and the overhead crane mode. The mathematical models of the pendulum system have been derived in Chapter 4. The derived mathematical models include the inverted pendulum, the overhead crane and the servo system. The inverted pendulum and the overhead crane are the essential plants that need to be controlled. The servo system is used to convert the mechanical signal force to the electrical signal u. It is considered separately from the inverted pendulum and the overhead crane. The feedback linearization approach is applied to the inverted pendulum and the overhead crane individually without the servo system involved. Although the input-output linearizing controllers are designed for the inverted pendulum and the overhead crane individually, there are many aspects in which they are very similar. The studies of the input-output linearization for the overhead crane are done by using the inverted pendulum as the reference.

7.4.1 Design of an input-output linearizing controller for the inverted pendulum

7.4.1.1 The inverted pendulum mathematical model

In this project, the considered equilibrium point for the inverted pendulum is in the range of $-5^{\circ}<\theta<5^{\circ}$. Since the angle is varying in a very small range, the output $y=z_1+L\sin z_3$ can be linearized as: $y=z_1+Lz_3$. Then the inverted pendulum can be expressed as:

$$\begin{bmatrix}\dot z_1\\ \dot z_2\\ \dot z_3\\ \dot z_4\end{bmatrix}=\begin{bmatrix} z_2\\ \dfrac{\begin{bmatrix}-\left(J+mL^2\right)\cdot b\cdot z_2-m^2\cdot L^2\cdot g\cdot\sin z_3\cdot\cos z_3+\\ +m\cdot L\cdot b_t\cdot z_4\cdot\cos z_3+\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot\sin z_3\end{bmatrix}}{\sigma}\\ z_4\\ \dfrac{\begin{bmatrix}\left(M+m\right)\cdot m\cdot g\cdot L\cdot\sin z_3-m^2\cdot L^2\cdot z_4^2\cdot\sin z_3\cdot\cos z_3\\ -\left(M+m\right)\cdot b_t\cdot z_4+m\cdot L\cdot\cos z_3\cdot b\cdot z_2\end{bmatrix}}{\sigma}\end{bmatrix}+\begin{bmatrix}0\\ \dfrac{J+m\cdot L^2}{\sigma}\\ 0\\ \dfrac{-m\cdot L\cdot\cos z_3}{\sigma}\end{bmatrix}\cdot F$$

$$y=z_1+Lz_3$$

where $\sigma=\left(J+mL^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot\cos^2 z_3$

All four states equations of the inverted pendulum can be represented in the standard affine form as:

$$\dot z_1=f_1(z)+g_1(z)u \qquad (7.63)$$

$$\dot{z}_2 = f_2(z) + g_2(z)u \tag{7.64}$$

$$\dot{z}_3 = f_3(z) + g_3(z)u \tag{7.65}$$

$$\dot{z}_4 = f_4(z) + g_4(z)u \tag{7.66}$$

The output is:

$$y=z_1+Lz_3=[1 \quad 0 \quad L \quad 0]z \tag{7.67}$$

where $z=[z_1 \quad z_2 \quad z_3 \quad z_4]^T$; $u=F$; $f_1(z)=z_2$; $g_1(z)=0$; $f_3(z)=z_4$; $g_3(z)=0$;

$$f_2(z) = \frac{\begin{bmatrix} -\left(J+mL^2\right)\cdot b\cdot z_2 - m^2\cdot L^2\cdot g\cdot \sin z_3\cdot \cos z_3 + \\ +m\cdot L\cdot b_t\cdot z_4\cdot \cos z_3 + \left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot \sin z_3 \end{bmatrix}}{\sigma}; \; g_2(z) = \frac{J+m\cdot L^2}{\sigma};$$

$$f_4(z) = \frac{\begin{bmatrix} (M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 - m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 \\ -(M+m)\cdot b_t\cdot z_4 + m\cdot L\cdot \cos z_3\cdot b\cdot z_2 \end{bmatrix}}{\sigma}; \; g_4(z) = \frac{-m\cdot L\cdot \cos z_3}{\sigma}.$$

From the mathematical model of the inverted pendulum, it can be seen the inverted pendulum is a fourth order system.

7.4.1.2 Differentiation of the output

In order to find the linear dependence between the input and output, the differentiation of the output is taken. The first derivative of the output is:

$$\begin{aligned} \dot{y}(t) &= \dot{z}_1 + L\dot{z}_3 \\ &= \dot{f}_1(z) + \dot{g}_1(z)u(t) + L\dot{f}_3(z) + L\dot{g}_3(z)u(t) \end{aligned}$$

Because $g_1(z)=0$ and $g_3(z)=0$, then:

$$\dot{y}(t) = f_1(z) + Lf_3(z) = z_2 + Lz_4 \tag{7.68}$$

The control input u(t) does not appear in the expression of the first derivative in Equation (7.68). The relationship between input u(t) and output y(t) is not found. Take the second derivative of y(t) then:

$$\begin{aligned} \ddot{y}(t) &= \ddot{z}_1 + L\ddot{z}_3 \\ &= \dot{z}_2 + L\dot{z}_4 \\ &= f_2(z) + g_2(z)u + Lf_4(z) + Lg_4(z)u \\ &= f_2(z) + Lf_4(z) + [g_2(z) + Lg_4(z)]u \end{aligned} \tag{7.69}$$

It can be seen from Equation (7.69) that the second derivative of output y(t) depends explicitly on the control input u(t). This means that the inverted pendulum has relative degree of two.

7.4.1.3 The linearized equation of the inverted pendulum

Let $\ddot{y}(t)=v(t)$ (7.70)

where v(t) is the new control signal which represents two integrators in series as shown in Figure 7.4.

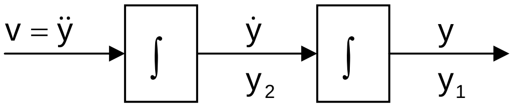


Figure 7.4: **Output integration**

As shown in the inverted pendulum mathematical model, its output is $y=z_1+Lz_3$. Based on the system output, Figure 7.4 and Equations (7.70), the following equations are obtained:

$$\dot{y}_1(t)=\dot{y}(t)=y_2=z_2+Lz_4$$

$$\dot{y}_2(t)=\ddot{y}(t)=f_2(z)+Lf_4(z)+[g_2(z)+Lg_4(z)]u=v(t)$$

In a matrix form the model for linearization is:

$$\begin{bmatrix}\dot{y}_1\\ \dot{y}_2\end{bmatrix}=\begin{bmatrix}z_2+Lz_4\\ f_2(z)+Lf_4(z)\end{bmatrix}+\begin{bmatrix}0\\ g_2(z)+Lg_4(z)\end{bmatrix}u=\begin{bmatrix}0\\ 1\end{bmatrix}v(t) \tag{7.71}$$

7.4.1.4 Calculation of the control signal for the inverted pendulum

The control signal u(t) can be calculated from the second equation of the linearization model in Equation (7.71).

Since: $f_2(z)+Lf_4(z)+[g_2(z)+Lg_4(z)]u(t)=v(t)$

Then: $u(t)=\dfrac{v(t)-f_2(z)-Lf_4(z)}{g_2(z)+Lg_4(z)}$ (7.72)

From Equation (7.72), it can be seen that the control signal u(t) exists only when $g_2(z)+Lg_4(z)\neq 0$. First of all, find the region of $g_2(z)+Lg_4(z)\neq 0$.

From the inverted pendulum mathematical model, it is known that $g_2(z)=\dfrac{J+m\cdot L^2}{\sigma}$ and

$g_4(z)=\dfrac{-m\cdot L\cdot\cos z_3}{\sigma}$. Then:

$$g_2(z)+Lg_4(z)=\frac{J+mL^2}{\sigma}+L\left(\frac{-mL\cos z_3}{\sigma}\right)=\frac{J+mL^2-mL^2\cos z_3}{\sigma}=\frac{J+mL^2(1-\cos z_3)}{\sigma} \tag{7.73}$$

In order to guarantee $g_2(z)+Lg_4(z)\neq 0$, the numerator and the denominator of the Equation (7.73) must not equal to zero at the same time, where:

$$J+mL^2(1-\cos z_3)\neq 0 \tag{7.74}$$

$$\sigma=(J+mL^2)\cdot(M+m)-m^2\cdot L^2\cdot\cos^2 z_3\neq 0 \tag{7.75}$$

Equation (7.75) can be rewritten as:

$$\sigma=J(M+m)+mML^2+m^2L^2(1-\cos z_3)\neq 0 \tag{7.76}$$

Since $-1\leq\cos z_3\leq 1$, it is easy to get $-1\leq -\cos z_3\leq 1$. Further more, the limitation of $1-\cos z_3$ can be obtained as: $0\leq 1-\cos z_3\leq 2$.

Then the limitation of the Equation (7.74) and (7.76) can be found as:

$$J\leq J+mL^2(1-\cos z_3)\leq J+2mL^2 \tag{7.77}$$

$$J(M+m)+mML^2\leq J(M+m)+mML^2+m^2L^2(1-\cos z_3)\leq J(M+m)+mML^2+2m^2L^2 \tag{7.78}$$

Since all the parameters involved in the Equation (7.77) and (7.78) are positive numbers, the values of the expression in these two equations are always a positive values, then it can be proven that Equation (7.74) and (7.76) never equal to zero in the range of $0\leq z_3\leq 2\pi$. So, the control signal exists all the time when z_3 varies in this interval.

7.4.1.5 Investigation of the inverted pendulum zero-dynamics

The control signal $u(t)$ is obtained. It is necessary to check the stability of the system internal-dynamics by studying the system zero-dynamics. The zero-dynamics is characterized by a set of variables where the output is identically equal to zero. This means:

$$y=x+L\theta=0 \tag{7.79}$$

Based on this equality, the inverted pendulum zero-dynamics are studied.

7.4.1.5.1 Controller at the zero-dynamics

The inverted pendulum is a fourth order system. As the relative degree for the output $y=z_1+Lz_3$ is two, the closed-loop system will be locally stable only if the zero-dynamics have stable behaviour. The zero-dynamics can be obtained from the model for linearization:

$y=z_1+Lz_3=0$

$$y_1=y=z_1+Lz_3=0 \tag{7.80}$$

then, $\dot{y}_1=0=z_2+Lz_4=0$ (7.81)

Equation (7.80) and (7.81) can be written in the following form that:

z_1=-Lz_3 or z_3=-z_1/L

z_2=-Lz_4 or z_4=-z_2/L

The control that provides these conditions is:

$$u_{zd} = \left.\frac{v - f_2(z) - Lf_4(z)}{g_2(z) + Lg_4(z)}\right|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} \tag{7.82}$$

where v=0 as only the linearizing part is considered.

The zero-dynamic model of the inverted pendulum is the system model in which the control u_{zd} is applied and the conditions z_1=-Lz_3, z_2=-Lz_4 are fulfilled.

7.4.1.5.2 The zero dynamics of the cart sub-system

The "cart" subsystem mathematical model of the inverted pendulum is:

$$\dot{z}_1 = z_2 \tag{7.83}$$

$$\dot{z}_2 = f_2\Big|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} + g_2\Big|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} \cdot u_{zd} \tag{7.84}$$

After substitution of u_{zd}, it is obtained that:

$$\dot{z}_1 = z_2 \tag{7.85}$$

$$\dot{z}_2 = f_2 + g_2 \left.\frac{v - f_2(z) - Lf_4(z)}{g_2(z) + Lg_4(z)}\right|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} = \left.\frac{f_2g_2 + Lf_2g_4 + g_2v - g_2f_2 - g_2Lf_4}{g_2 + Lg_4}\right|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}}$$

$$= \left.\frac{g_2v + L(f_2g_4 - g_2f_4)}{g_2 + Lg_4}\right|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} \tag{7.86}$$

By substituting the expressions of f_2, f_4, g_2 and g_4, the linearized cart model can be derived as:

$$f_2g_4 = \frac{\begin{bmatrix} -\left(J + mL^2\right)\cdot b \cdot z_2 - m^2 \cdot L^2 \cdot g \cdot \sin z_3 \cdot \cos z_3 \\ + m \cdot L \cdot b_t \cdot z_4 \cdot \cos z_3 + \left(J + mL^2\right)\cdot m \cdot L \cdot z_4^2 \cdot \sin z_3 \end{bmatrix} \times \left[- m \cdot L \cdot \cos z_3\right]}{\left[\left(J + mL^2\right)\cdot(M + m) - m^2 \cdot L^2 \cdot \cos^2 z_3\right]^2} \tag{7.87}$$

$$f_4g_2 = \frac{\begin{bmatrix} (M + m)\cdot m \cdot g \cdot L \cdot \sin z_3 - m^2 \cdot L^2 \cdot z_4^2 \cdot \sin z_3 \cdot \cos z_3 \\ -(M + m)\cdot b_t \cdot z_4 + m \cdot L \cdot \cos z_3 \cdot b \cdot z_2 \end{bmatrix} \times \left(J + m \cdot L^2\right)}{\left[\left(J + mL^2\right)\cdot(M + m) - m^2 \cdot L^2 \cdot \cos^2 z_3\right]^2} \tag{7.88}$$

$$f_2g_4 - f_4g_2 = \frac{\begin{bmatrix} \left(J+mL^2\right)\cdot b\cdot z_2\cdot m\cdot L\cdot \cos z_3 + m^3\cdot L^3\cdot g\cdot \sin z_3\cdot \cos z_3^2 \\ -m^2\cdot L^2\cdot b_t\cdot z_4\cdot \cos z_3^2 - \left(J+mL^2\right)\cdot m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 \\ -\left(M+m\right)\cdot m\cdot g\cdot L\cdot \sin z_3\cdot \left(J+m\cdot L^2\right) \\ +m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3\cdot \left(J+m\cdot L^2\right) \\ +\left(M+m\right)\cdot b_t\cdot z_4\cdot \left(J+m\cdot L^2\right) - m\cdot L\cdot \cos z_3\cdot b\cdot z_2\cdot \left(J+m\cdot L^2\right) \end{bmatrix}}{\left[\left(J+mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos^2 z_3\right]^2} \tag{7.89}$$

Simplify the expression in Equation (7.89), then Equation (7.90) is obtained.

$$f_2g_4 - f_4g_2 = \frac{\begin{bmatrix} m^3\cdot L^3\cdot g\cdot \sin z_3\cdot \cos z_3^2 - \left(M+m\right)\cdot m\cdot g\cdot L\cdot \sin z_3\cdot \left(J+m\cdot L^2\right) \\ -m^2\cdot L^2\cdot b_t\cdot z_4\cdot \cos z_3^2 + \left(M+m\right)\cdot b_t\cdot z_4\cdot \left(J+m\cdot L^2\right) \end{bmatrix}}{\left[\left(J+mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_2g_4 - f_4g_2 = \frac{\begin{matrix} -\left[\left(J+m\cdot L^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos z_3^2\right]\cdot m\cdot g\cdot L\cdot \sin z_3 \\ +\left[\left(J+m\cdot L^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos z_3^2\right]\cdot b_t\cdot z_4 \end{matrix}}{\left[\left(J+mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_2g_4 - f_4g_2 = \frac{\left[\left(J+m\cdot L^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos z_3^2\right]\cdot\left(b_t\cdot z_4 - m\cdot g\cdot L\cdot \sin z_3\right)}{\left[\left(J+mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_2g_4 - f_4g_2 = \left.\frac{b_t\cdot z_4 - m\cdot g\cdot L\cdot \sin z_3}{\left(J+mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos^2 z_3}\right|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} \tag{7.90}$$

and It is easy to get,

$$g_2 + Lg_4 = \frac{J + mL^2 - mL^2\cos z_3}{\left(J+mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos^2 z_3} \tag{7.91}$$

Then the zero-dynamics model of the “cart” can be expressed as:

$$\dot{z}_1 = z_2 \tag{7.92}$$

$$\dot{z}_2 = \left.\frac{L\left(b_t\cdot z_4 - m\cdot g\cdot L\cdot \sin z_3\right)}{J + mL^2 - mL^2\cos z_3}\right|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} = \frac{-b_t z_2 + mgL^2 \sin\dfrac{z_1}{L}}{J + mL^2 - mL^2\cos\dfrac{z_1}{L}} \tag{7.93}$$

The stability of the inverted pendulum’s “cart” subsystem zero-dynamics model is investigated according to the Lyapunov function:

$$V = \frac{1}{2}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix}^T \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \frac{1}{2}\left[z_1^2 + z_2^2\right] \tag{7.94}$$

Its first derivative is:

$$\frac{dV}{dt} = \dot{V} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}^T \begin{pmatrix} \dot{z}_1 \\ \dot{z}_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}^T \begin{pmatrix} z_2 \\ \dfrac{-b_t z_2 + mgL^2 \sin\dfrac{z_1}{L}}{J + mL^2 - mL^2 \cos\dfrac{z_1}{L}} \end{pmatrix} = z_1 z_2 + \frac{-b_t z_2^2 + mgL^2 z_2 \sin\dfrac{z_1}{L}}{J + mL^2 - mL^2 \cos\dfrac{z_1}{L}}$$

$$= \frac{z_1 z_2 \left(J + mL^2 - mL^2 \cos\dfrac{z_1}{L} \right) - b_t z_2^2 + mgL^2 z_2 \sin\dfrac{z_1}{L}}{J + mL^2 - mL^2 \cos\dfrac{z_1}{L}} \qquad (7.95)$$

It can be easily seen that the selected Lyapunov function is positive definite. So, if $\dot{V}$ can be proven to be negative definite, the system is stable; if $\dot{V}$ equals to zero, the system is at the margin of the stabilization. Otherwise, the system is unstable.

According to the inverted pendulum's “cart” subsystem zero-dynamics model, the stability of the cart system is investigated in Matlab/Simulink environment. First, the cart zero-dynamics model is built graphically in the Simulink as shown in Figure 7.5. Based on the linearized cart model, the Lyapunov function and its derivative are built as shown in Figure 7.6. An associated m-file in Appendix A. 6 called “zero_dyn.m” is used to provide all the parameters that are used in the Simulink files. Based on the system zero-dynamics and the simulation results, the stability of the linearized cart system is investigated.

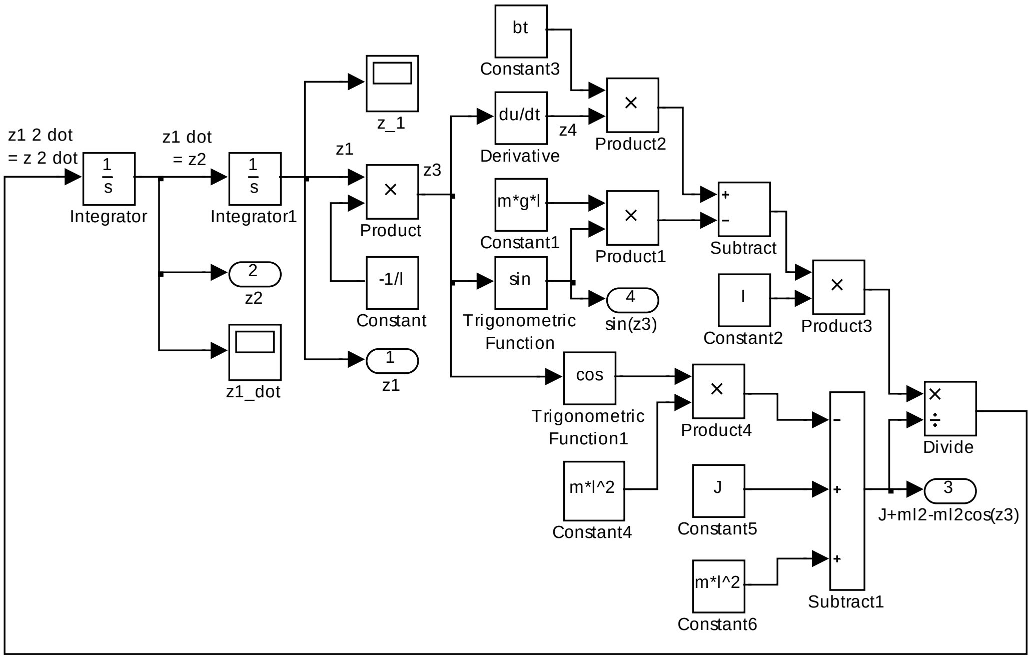


Figure 7.5: The linearized inverted pendulum cart model

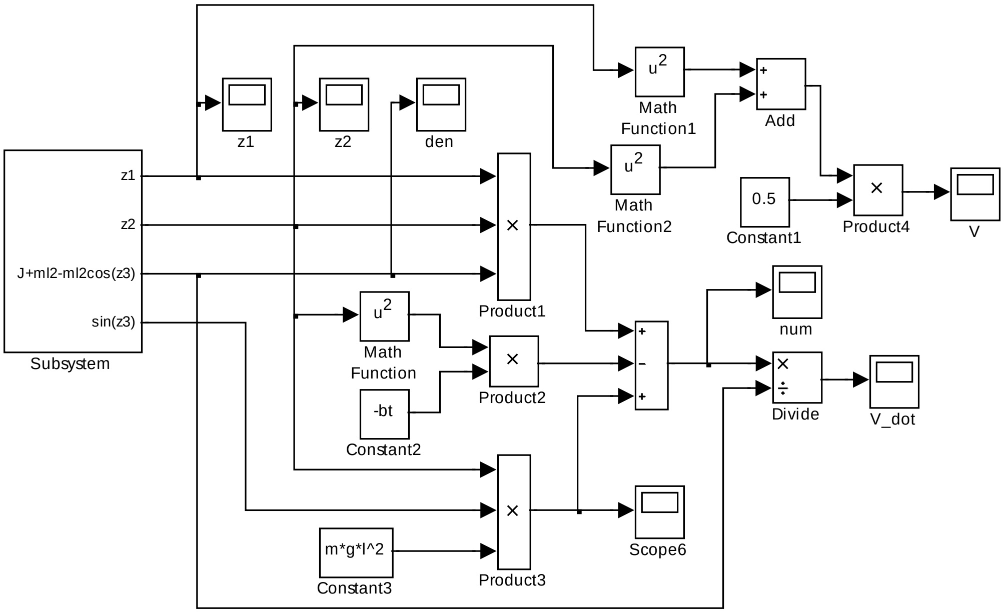


Figure 7.6: **The Lyapunov function of the cart model**

The simulation results are presented in Figure 7.7 to Figure 7.10. In Figure 7.7 and Figure 7.8, the states z_1 and z_2 are shown respectively. It can be seen that both states convergence to constant values. The oscillation amplitudes of the states decrease with time. The Lyapunov function V is positive definite, as proved by Figure 7.9. In Figure 7.10, the derivative of the Lyapunov function $\dot{V}$ is shown. Firstly, it can be seen that $\dot{V}$ is not negative definite, secondly $\dot{V}$ oscillates and its amplitude is decreasing with time. Eventually, it settles at zero. This means the system is at the margin of the stabilization. The external controller is needed to stabilize the system.

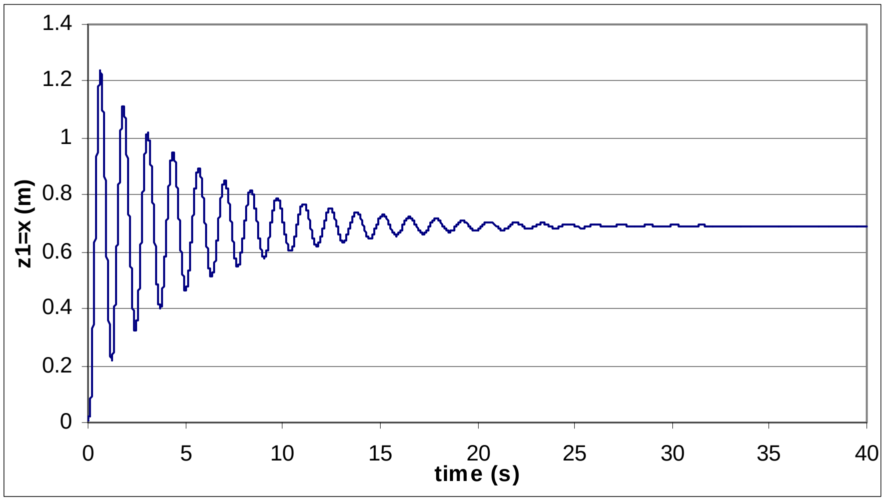


Figure 7.7: **The linearized inverted pendulum cart model state z_1**

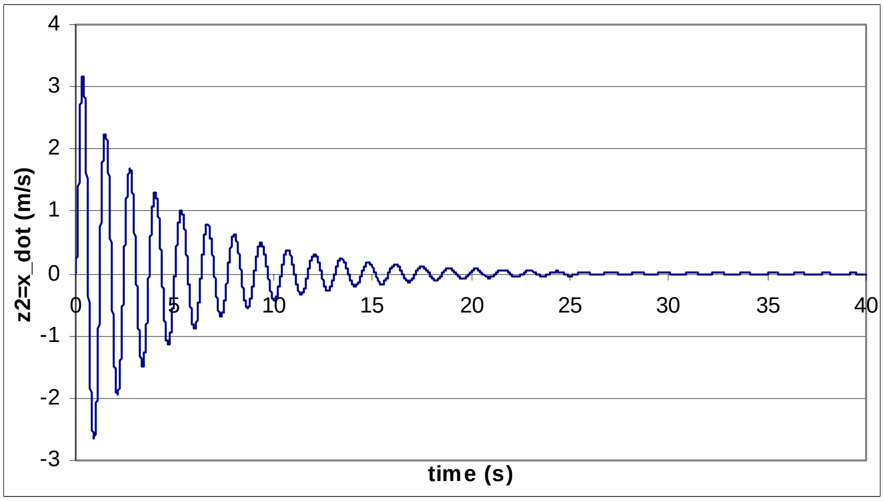


Figure 7.8: The linearized inverted pendulum cart model state z_2

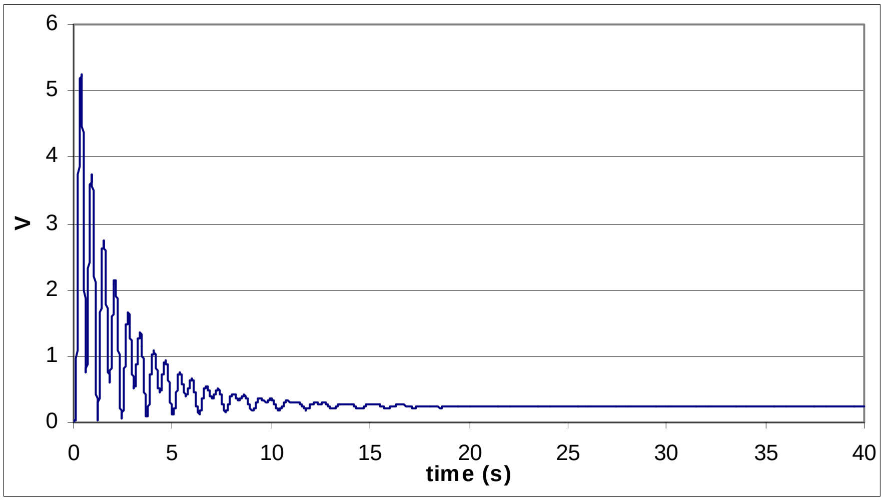


Figure 7.9: The Lyapunov function

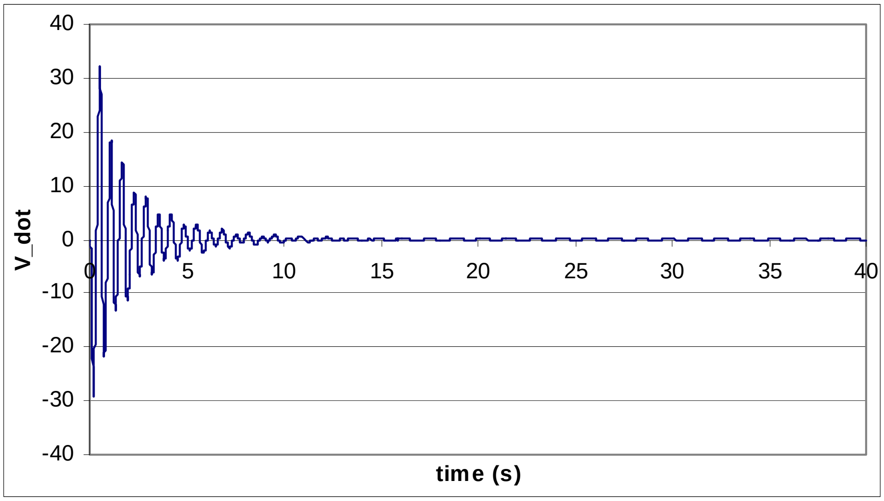


Figure 7.10: The derivative of the Lyapunov function

7.4.1.5.3 The zero dynamics of the pendulum

The “pendulum” subsystem mathematical model of the inverted pendulum is:

$$\dot{z}_3 = z_4 \tag{7.96}$$

$$\dot{z}_4 = f_4\Big|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} + g_4\Big|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} \cdot u_{zd} \tag{7.97}$$

After substitution of u_{zd}, the model is:

$$\dot{z}_3 = z_4 \tag{7.98}$$

$$\dot{z}_4 = f_4 + g_4 \frac{-f_2 - Lf_4}{g_2 + Lg_4}\Bigg|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} = \frac{f_4 g_2 + Lf_4 g_4 - f_2 g_4 - Lf_4 g_4}{g_2 + Lg_4} = \frac{f_4 g_2 - f_2 g_4}{g_2 + Lg_4}\Bigg|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} \tag{7.99}$$

By substituting the expression of f_2, f_4, g_2 and g_4, the linearized cart model can be derived as shown in Equation (7.100) and (7.101).

$$f_4 g_2 - f_2 g_4 = \frac{\left[\begin{array}{l} (M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 \cdot \left(J + m\cdot L^2\right) \\ -m^2\cdot L^2\cdot z_4^2 \cdot \sin z_3 \cdot \cos z_3 \cdot \left(J + m\cdot L^2\right) \\ -(M+m)\cdot b_t \cdot z_4 \cdot \left(J + m\cdot L^2\right) + m\cdot L\cdot \cos z_3 \cdot b\cdot z_2 \cdot \left(J + m\cdot L^2\right) \\ -\left[\begin{array}{l} \left(J + mL^2\right)\cdot b\cdot z_2 \cdot m\cdot L\cdot \cos z_3 + m^3\cdot L^3\cdot g\cdot \sin z_3 \cdot \cos z_3^2 \\ -m^2\cdot L^2\cdot b_t\cdot z_4\cdot \cos z_3^2 - \left(J + mL^2\right)\cdot m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 \end{array}\right] \end{array}\right]}{\left[\left(J + mL^2\right)\cdot (M+m) - m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_4 g_2 - f_2 g_4 = \frac{\left[\begin{array}{l} (M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 \cdot \left(J + m\cdot L^2\right) \\ -m^2\cdot L^2\cdot z_4^2 \cdot \sin z_3 \cdot \cos z_3 \cdot \left(J + m\cdot L^2\right) \\ -(M+m)\cdot b_t \cdot z_4 \cdot \left(J + m\cdot L^2\right) + m\cdot L\cdot \cos z_3 \cdot b\cdot z_2 \cdot \left(J + m\cdot L^2\right) \\ -\left(J + mL^2\right)\cdot b\cdot z_2 \cdot m\cdot L\cdot \cos z_3 - m^3\cdot L^3\cdot g\cdot \sin z_3 \cdot \cos z_3^2 \\ +m^2\cdot L^2\cdot b_t\cdot z_4\cdot \cos z_3^2 + \left(J + mL^2\right)\cdot m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 \end{array}\right]}{\left[\left(J + mL^2\right)\cdot (M+m) - m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_4 g_2 - f_2 g_4 = \frac{\left[\begin{array}{l} (M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 \cdot \left(J + m\cdot L^2\right) - m^3\cdot L^3\cdot g\cdot \sin z_3\cdot \cos z_3^2 \\ -(M+m)\cdot b_t \cdot z_4 \cdot \left(J + m\cdot L^2\right) + m^2\cdot L^2\cdot b_t\cdot z_4\cdot \cos z_3^2 + \end{array}\right]}{\left[\left(J + mL^2\right)\cdot (M+m) - m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_4 g_2 - f_2 g_4 = \frac{\left[\begin{array}{l} \left[(M+m)\cdot \left(J + m\cdot L^2\right) - m^2\cdot L^2\cdot \cos z_3^2\right]\cdot m\cdot g\cdot L\cdot \sin z_3 \\ \left[-(M+m)\cdot \left(J + m\cdot L^2\right) + m^2\cdot L^2\cdot \cos z_3^2\right]\cdot b_t\cdot z_4 \end{array}\right]}{\left[\left(J + mL^2\right)\cdot (M+m) - m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_4 g_2 - f_2 g_4 = \frac{\left[(M+m)\cdot \left(J + m\cdot L^2\right) - m^2\cdot L^2\cdot \cos z_3^2\right]\cdot \left[m\cdot g\cdot L\cdot \sin z_3 - b_t\cdot z_4\right]}{\left[\left(J + mL^2\right)\cdot (M+m) - m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_4 g_2 - f_2 g_4 = \left.\frac{-b_t \cdot z_4 + m \cdot g \cdot L \cdot \sin z_3}{\left(J + mL^2\right) \cdot \left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2 z_3}\right|_{\substack{z_1 = -Lz_3 \\ z_2 = -Lz_4}}$$

$$g_2 + Lg_4 = \frac{J + mL^2 - mL^2 \cos z_3}{\left(J + mL^2\right) \cdot \left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2 z_3}$$

Then the model of the zero dynamics of the pendulum is:

$$\dot{z}_3 = z_4 \tag{7.100}$$

$$\dot{z}_4 = \left.\frac{-b_t \cdot z_4 + m \cdot g \cdot L \cdot \sin z_3}{J + mL^2 - mL^2 \cos z_3}\right|_{\substack{z_1 = -Lz_3 \\ z_2 = -Lz_4}} \tag{7.101}$$

The stability of the inverted pendulum's "pendulum" subsystem zero-dynamics is studied according to the Lyapunov function:

$$V = \frac{1}{2}\begin{pmatrix} z_3 \\ z_4 \end{pmatrix}^T \begin{pmatrix} z_3 \\ z_4 \end{pmatrix} \tag{7.102}$$

$$\frac{dV}{dt} = \dot{V} = \begin{pmatrix} z_3 \\ z_4 \end{pmatrix}^T \begin{pmatrix} \dot{z}_3 \\ \dot{z}_4 \end{pmatrix} = \begin{pmatrix} z_3 \\ z_4 \end{pmatrix}^T \left(\begin{array}{c} z_4 \\ \left.\dfrac{-b_t \cdot z_4 + m \cdot g \cdot L \cdot \sin z_3}{J + mL^2 - mL^2 \cos z_3}\right| \end{array} \right) = z_3 z_4 + \frac{-b_t \cdot z_4^2 + m \cdot g \cdot L \cdot z_4 \cdot \sin z_3}{J + mL^2 - mL^2 \cos z_3}$$

$$= \frac{z_3 z_4 \left(J + mL^2 - mL^2 \cos \dfrac{z_1}{L} \right) - b_t \cdot z_4^2 + m \cdot g \cdot L \cdot z_4 \cdot \sin z_3}{J + mL^2 - mL^2 \cos \dfrac{z_1}{L}} \tag{7.103}$$

The simulations are done in Matlab environment. In the Figure 7.11, the pendulum zero-dynamics model is built. Based on this model, the Lyapunov function V and its derivative $\dot{V}$ are built as shown in Figure 7.12. From the simulation results which are shown in Figure 7.13 to Figure 7.16, the stability of the system is determined. In Figure 7.13, it can be seen the pendulum swings and stops at the position of 2π where the angle θ=180°. That means the feedback control signal u can not stabilize the pendulum at position θ=0°. If the pendulum is at the position of θ=180°, the pendulum stops swinging $\dot{\theta} = 0$, as shown in Figure 7.14. The chosen Lyapunov function is positive definite, it can be seen from Figure 7.15. The derivative of the Lyapunov function $\dot{V}$ converges to zero as shown in Figure 7.16. The pendulum is not stable at θ=0° and stable at θ=180°. The external controller is needed to stabilize the system.

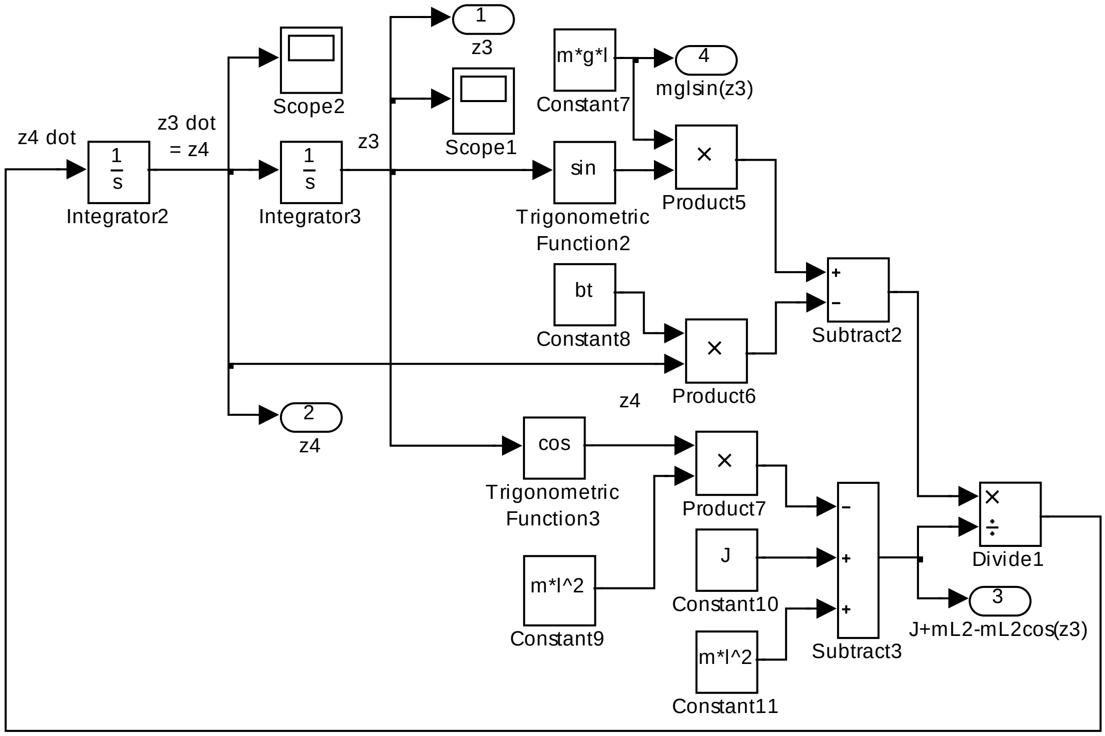


Figure 7.11: The linearized pendulum model of the inverted pendulum

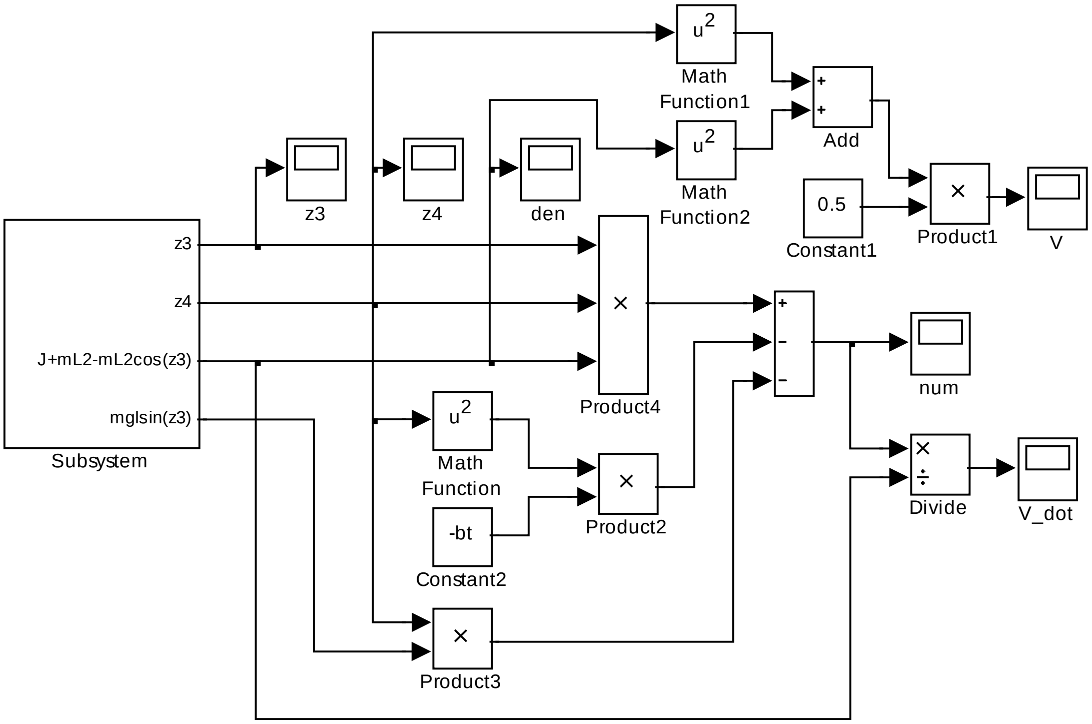


Figure 7.12: The Lyapunov function of the pendulum model

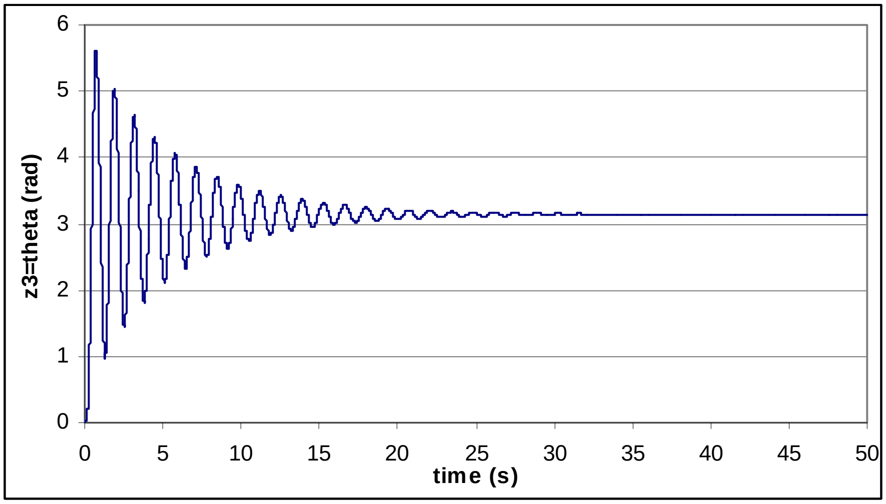


Figure 7.13: The linearized inverted pendulum “pendulum” model state z_1

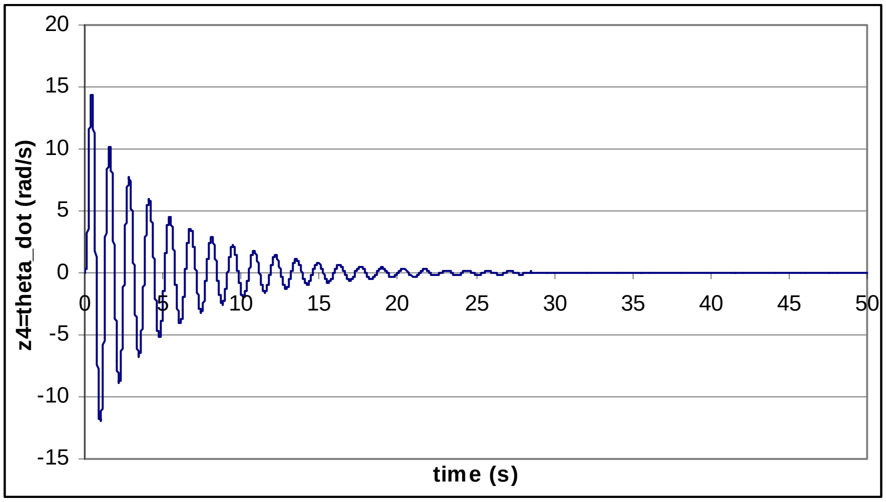


Figure 7.14: The linearized inverted pendulum “pendulum” model state z_2

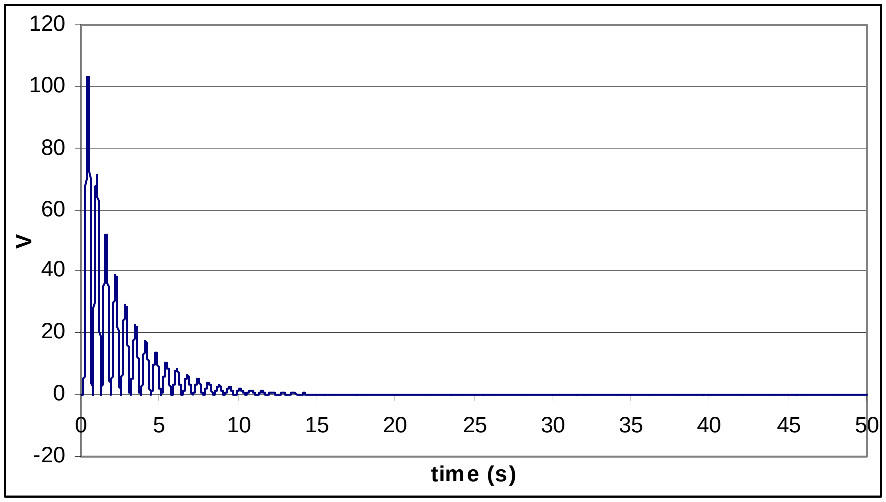


Figure 7.15: The Lyapunov function

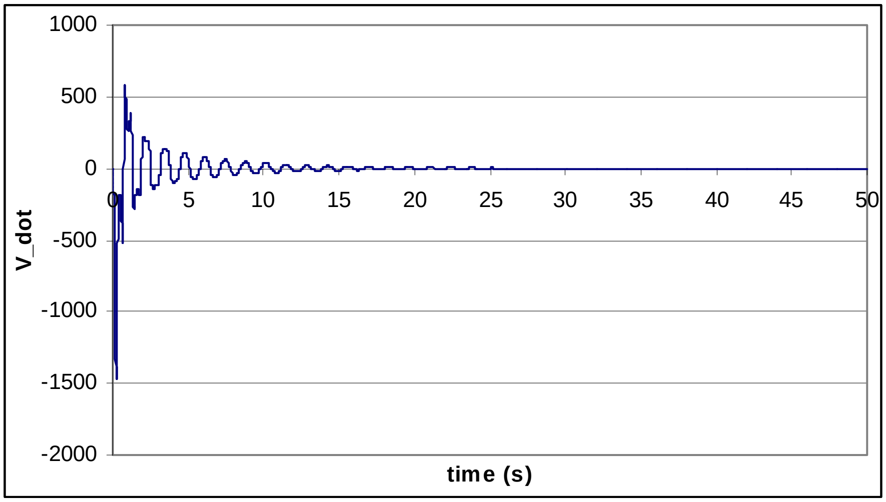


Figure 7.16: The derivative of the Lyapunov function

By studying of the cart subsystem zero-dynamics and the pendulum subsystem zero-dynamics of the inverted pendulum, it can be seen both system are at the margin of the stabilization but for θ=180°. That means the linearizing control u is not sufficient to stabilize the system. The external control is necessary to be designed. In this project, the model reference control based linear quadratic regulator is used.

7.4.1.6 The linearized inverted pendulum model

The linearized inverted pendulum system is obtained after substitution of the obtained control u in Equation (7.72) into the nonlinear model of the inverted pendulum:

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\\ \dot{z}_3\\ \dot{z}_4\end{bmatrix}=\begin{bmatrix}f_1(z)\\ f_2(z)\\ f_3(z)\\ f_4(z)\end{bmatrix}+\begin{bmatrix}0\\ g_2(z)\\ 0\\ g_4(z)\end{bmatrix}\cdot\frac{v-f_2(z)-Lf_4(z)}{g_2(z)+Lg_4(z)}=\begin{bmatrix}f_1(z)\\ f_2(z)\\ f_3(z)\\ f_4(z)\end{bmatrix}+\begin{bmatrix}0\\ g_2(z)\cdot\dfrac{v-f_2(z)-Lf_4(z)}{g_2(z)+Lg_4(z)}\\ 0\\ g_4(z)\cdot\dfrac{v-f_2(z)-Lf_4(z)}{g_2(z)+Lg_4(z)}\end{bmatrix}$$

$$=\begin{bmatrix}f_1(z)\\ f_2(z)+g_2(z)\cdot\dfrac{v-f_2(z)-Lf_4(z)}{g_2(z)+Lg_4(z)}\\ f_3(z)\\ f_4(z)+g_4(z)\cdot\dfrac{v-f_2(z)-Lf_4(z)}{g_2(z)+Lg_4(z)}\end{bmatrix} \tag{7.104}$$

Then, the inverted pendulum nonlinear mathematical model can be written in the linear form with the new input v:

$$A_{lin}(z)+B_{lin}(z)\cdot v=\begin{bmatrix} f_1(z) \\ f_2(z)+\dfrac{-g_2(z)\cdot f_2(z)-Lg_2(z)\cdot f_4(z)}{g_2(z)+Lg_4(z)} \\ f_3(z) \\ f_4(z)+\dfrac{-g_4(z)\cdot f_2(z)-Lg_4(z)\cdot f_4(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}+\begin{bmatrix} 0 \\ \dfrac{g_2(z)}{g_2(z)+Lg_4(z)} \\ 0 \\ \dfrac{g_4(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}v \qquad (7.105)$$

Simplify the expression in Equation (7.105) and its simplest form is obtained as shown in Equation (7.106):

$$A_{lin}(z)+B_{lin}(z)\cdot v=\begin{bmatrix} f_1(z) \\ \dfrac{g_2(z)\cdot f_2(z)+Lf_2(z)g_4(z)-g_2(z)\cdot f_2(z)-Lg_2(z)\cdot f_4(z)}{g_2(z)+Lg_4(z)} \\ f_3(z) \\ \dfrac{g_2(z)\cdot f_4(z)+Lf_4(z)g_4(z)-g_4(z)\cdot f_2(z)-Lg_4(z)\cdot f_4(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}+\begin{bmatrix} 0 \\ \dfrac{g_2(z)}{g_2(z)+Lg_4(z)} \\ 0 \\ \dfrac{g_4(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}v$$

$$A_{lin}(z)+B_{lin}(z)\cdot v=\begin{bmatrix} f_1(z) \\ \dfrac{Lf_2(z)g_4(z)-Lg_2(z)\cdot f_4(z)}{g_2(z)+Lg_4(z)} \\ f_3(z) \\ \dfrac{g_2(z)f_4(z)-g_4(z)\cdot f_2(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}+\begin{bmatrix} 0 \\ \dfrac{g_2(z)}{g_2(z)+Lg_4(z)} \\ 0 \\ \dfrac{g_4(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}v$$

$$A_{lin}(z)+B_{lin}(z)\cdot v=\begin{bmatrix} z_2 \\ \dfrac{L[f_2(z)g_4(z)-g_2(z)\cdot f_4(z)]}{g_2(z)+Lg_4(z)} \\ z_4 \\ \dfrac{g_2(z)f_4(z)-g_4(z)\cdot f_2(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}+\begin{bmatrix} 0 \\ \dfrac{g_2(z)}{g_2(z)+Lg_4(z)} \\ 0 \\ \dfrac{g_4(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}v \qquad (7.106)$$

where $\dfrac{L[f_2(z)g_4(z)-g_2(z)\cdot f_4(z)]}{g_2(z)+Lg_4(z)}=\dfrac{Lb_t z_4-mgL^2\sin z_3}{J+mL^2-mL^2\cos z_3}$;

$$\frac{g_2(z)f_4(z)-g_4(z)\cdot f_2(z)}{g_2(z)+Lg_4(z)}=\frac{-b_t\cdot z_4+m\cdot g\cdot L\cdot\sin z_3}{J+mL^2-mL^2\cos z_3};$$

$$\frac{g_2(z)}{g_2(z)+Lg_4(z)}=\frac{J+mL^2}{J+mL^2-mL^2\cos z_3};$$

$$\frac{g_4(z)}{g_2(z)+Lg_4(z)}=\frac{-mL\cos z_3}{J+mL^2-mL^2\cos z_3};$$

The obtained closed-loop system in Equation (7.106) is not fully linearized by the designed linearizing controller u, because the relative degree of the linearizing system is two and the nonlinear system has order four. In order to improve the performance of the obtained closed-loop system, a linear controller is necessary to be designed. In order to obtain it, the closed-loop system is linearized around the point $\theta=z_3\approx 0$, for which $\sin z_3=z_3$ and $\cos z_3=1$ and the value $\dot{\theta}^2$ is approximately equal to zero.

Then: $\dfrac{L[f_2(z)g_4(z)-g_2(z)\cdot f_4(z)]}{g_2(z)+Lg_4(z)}=\dfrac{Lb_t z_4-mgL^2 z_3}{J+mL^2-mL^2}=\dfrac{Lb_t z_4-mgL^2 z_3}{J}$,

$$\frac{g_2(z)f_4(z)-g_4(z)\cdot f_2(z)}{g_2(z)+Lg_4(z)}=\frac{-b_t\cdot z_4+m\cdot g\cdot L\cdot z_3}{J+mL^2-mL^2}=\frac{-b_t\cdot z_4+m\cdot g\cdot L\cdot z_3}{J}$$

$$\frac{g_2(z)}{g_2(z)+Lg_4(z)}=\frac{J+mL^2}{J+mL^2-mL^2}=\frac{J+mL^2}{J};$$

$$\frac{g_4(z)}{g_2(z)+Lg_4(z)}=\frac{-mL\cos z_3}{J+mL^2-mL^2}=\frac{-mL}{J};$$

After substitution of the obtained transformations the linearized system can be written in the state space form as shown in Equation (7.107):

$$\begin{bmatrix}\dot{z}_1\\ \dot{z}_2\\ \dot{z}_3\\ \dot{z}_4\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & -\dfrac{mgL^2}{J} & \dfrac{Lb_t}{J}\\ 0 & 0 & 0 & 1\\ 0 & 0 & \dfrac{mgL}{J} & -\dfrac{b_t}{J}\end{bmatrix}\begin{bmatrix}z_1\\ z_2\\ z_3\\ z_4\end{bmatrix}+\begin{bmatrix}0\\ \dfrac{J+mL^2}{J}\\ 0\\ \dfrac{-mL}{J}\end{bmatrix}v \tag{7.107}$$

with the output: $y=z_1+Lz_3$

The linearized inverted pendulum mathematical model in Equation (7.107) is used to design the inverted pendulum linear closed-loop control v.

7.4.2 Design of an input-output linearizing controller for the overhead crane

From Chapter 4, it can be seen that the overhead crane has very similar mathematical model with the inverted pendulum. Because of the mathematical model similarity, the input-output linearizing controller design for the overhead crane is simplified dramatically.

By following the procedure of the feedback linearization approach applied to the inverted pendulum, the input-output linearizing controller for the overhead crane is designed and the zero-dynamics are studied for the overhead crane. The linearized mathematical model of the overhead crane is presented at the end of this section. In this section, an anti-swing controller is designed. (Park et al, 2007:379)

7.4.2.1 The overhead crane mathematical model

The overhead crane mathematical model with linear output is:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} z_2 \\ \dfrac{\begin{bmatrix} -\left(J+mL^2\right)\cdot b\cdot z_2 + m^2\cdot L^2\cdot g\cdot \sin z_3\cdot \cos z_3 + \\ +\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot \sin z_3 + m\cdot L\cdot b_t\cdot z_4\cdot \cos z_3 \end{bmatrix}}{\sigma} \\ z_4 \\ \dfrac{\begin{bmatrix} +m\cdot L\cdot \cos z_3\cdot b\cdot z_2 - \left(M+m\right)\cdot m\cdot g\cdot L\cdot \sin z_3 - \\ -m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 - \left(M+m\right)\cdot b_t\cdot z_4 \end{bmatrix}}{\sigma} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{J+m\cdot L^2}{\sigma} \\ 0 \\ \dfrac{-m\cdot L\cdot \cos z_3}{\sigma} \end{bmatrix}\cdot F$$

$$y = z_1 + Lz_3$$

where $\sigma = \left(J+mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos^2 z_3$

The above overhead crane mathematical model can also be written in the form of:

$$\dot{z}_1 = f_1(z) + g_1(z)u \tag{7.108}$$

$$\dot{z}_2 = f_2(z) + g_2(z)u \tag{7.109}$$

$$\dot{z}_3 = f_3(z) + g_3(z)u \tag{7.110}$$

$$\dot{z}_4 = f_4(z) + g_4(z)u \tag{7.111}$$

The output is

$$y=z_1+Lz_3=[1 \quad 0 \quad L \quad 0]z \tag{7.112}$$

where $z=[z_1 \quad z_2 \quad z_3 \quad z_4]^T$; $u=F$; $f_1(z)=z_2$; $g_1(z)=0$; $f_3(z)=z_4$; $g_3(z)=0$;

$$f_2(z) = \frac{\begin{bmatrix} -\left(J+mL^2\right)\cdot b\cdot z_2 + m^2\cdot L^2\cdot g\cdot \sin z_3\cdot \cos z_3 + \\ +\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot \sin z_3 + m\cdot L\cdot b_t\cdot z_4\cdot \cos z_3 \end{bmatrix}}{\sigma};\ g_2(z) = \frac{\left(J+m\cdot L^2\right)}{\sigma};$$

$$f_4(z) = \frac{\begin{bmatrix} +m\cdot L\cdot \cos z_3\cdot b\cdot z_2 - \left(M+m\right)\cdot m\cdot g\cdot L\cdot \sin z_3 - \\ -m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 - \left(M+m\right)\cdot b_t\cdot z_4 \end{bmatrix}}{\sigma};\ g_4(z) = \frac{-m\cdot L\cdot \cos z_3}{\sigma}.$$

7.4.2.2 Calculation of the control

The control signal u is:

$$u(t)=\frac{v(t)-f_2(z)-Lf_4(z)}{g_2(z)+Lg_4(z)} \tag{7.113}$$

It exists for $0 \leq z_3 \leq 2\pi$.

7.4.2.3 Investigation of the zero dynamics

The zero-dynamics of the overhead crane are studied by applying the same principle used to investigate the inverted pendulum stability which is let $y=0$.

7.4.2.3.1 The zero dynamics of the cart sub-system

The “cart” subsystem mathematical model of the overhead crane is:

$$\dot{z}_1 = z_2 \tag{7.114}$$

$$\dot{z}_2 = f_2\Big|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} + g_2\Big|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} \cdot u_{zd} \tag{7.115}$$

After substitution of u_{zd}, it is obtained that:

$$\dot{z}_1 = z_2 \tag{7.116}$$

$$\dot{z}_2 = f_2 + g_2 \left.\frac{v-f_2(z)-Lf_4(z)}{g_2(z)+Lg_4(z)}\right|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} = \frac{f_2g_2+Lf_2g_4+g_2v-g_2f_2-g_2Lf_4}{g_2+Lg_4}$$

$$= \left.\frac{g_2v+L(f_2g_4-g_2f_4)}{g_2+Lg_4}\right|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} \tag{7.117}$$

By substituting the expression of f_2, f_4, g_2 and g_4, the cart zero-dynamics model can be derived as shown in Equation (7.123) and (7.124).

$$f_2g_4 = \frac{\begin{bmatrix} -\left(J+mL^2\right)\cdot b\cdot z_2 + m^2\cdot L^2\cdot g\cdot \sin z_3\cdot \cos z_3 + \\ +\left(J+mL^2\right)\cdot m\cdot L\cdot z_4^2\cdot \sin z_3 + m\cdot L\cdot b_t\cdot z_4\cdot \cos z_3 \end{bmatrix} \times \left[-m\cdot L\cdot \cos z_3\right]}{\left[\left(J+mL^2\right)\cdot(M+m)-m^2\cdot L^2\cdot \cos^2 z_3\right]^2} \tag{7.118}$$

$$f_4g_2 = \frac{\begin{bmatrix} +m\cdot L\cdot \cos z_3\cdot b\cdot z_2 - (M+m)\cdot m\cdot g\cdot L\cdot \sin z_3 - \\ -m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 - (M+m)\cdot b_t\cdot z_4 \end{bmatrix} \times \left(J+m\cdot L^2\right)}{\left[\left(J+mL^2\right)\cdot(M+m)-m^2\cdot L^2\cdot \cos^2 z_3\right]^2} \tag{7.119}$$

$$f_2g_4 - f_4g_2 = \frac{\begin{bmatrix} \left(J+mL^2\right)\cdot b\cdot z_2\cdot m\cdot L\cdot \cos z_3 - m^3\cdot L^3\cdot g\cdot \sin z_3\cdot \cos z_3^2 - \\ -\left(J+mL^2\right)\cdot m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3 - m^2\cdot L^2\cdot b_t\cdot z_4\cdot \cos z_3^2 + \\ +\left(M+m\right)\cdot m\cdot g\cdot L\cdot \sin z_3\cdot \left(J+m\cdot L^2\right)+ \\ +m^2\cdot L^2\cdot z_4^2\cdot \sin z_3\cdot \cos z_3\cdot \left(J+m\cdot L^2\right)- \\ -m\cdot L\cdot \cos z_3\cdot b\cdot z_2\cdot \left(J+m\cdot L^2\right)+\left(M+m\right)\cdot b_t\cdot z_4\cdot \left(J+m\cdot L^2\right) \end{bmatrix}}{\left[\left(J+mL^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

(7.120)

Simply the expression in Equation (7.120), then Equation (7.121) is obtained.

$$f_2g_4 - f_4g_2 = \frac{\begin{bmatrix} -m^3\cdot L^3\cdot g\cdot \sin z_3\cdot \cos z_3^2 - m^2\cdot L^2\cdot b_t\cdot z_4\cdot \cos z_3^2 + \\ +\left(M+m\right)\cdot m\cdot g\cdot L\cdot \sin z_3\cdot \left(J+m\cdot L^2\right)+\left(M+m\right)\cdot b_t\cdot z_4\cdot \left(J+m\cdot L^2\right) \end{bmatrix}}{\left[\left(J+mL^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_2g_4 - f_4g_2 = \frac{\begin{array}{l}\left[\left(J+m\cdot L^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot \cos z_3^2\right]\cdot m\cdot g\cdot L\cdot \sin z_3 \\ +\left[\left(J+m\cdot L^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot \cos z_3^2\right]\cdot b_t\cdot z_4\end{array}}{\left[\left(J+mL^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_2g_4 - f_4g_2 = \frac{\left[\left(J+m\cdot L^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot \cos z_3^2\right]\cdot\left(b_t\cdot z_4 + m\cdot g\cdot L\cdot \sin z_3\right)}{\left[\left(J+mL^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot \cos^2 z_3\right]^2}$$

$$f_2g_4 - f_4g_2 = \left.\frac{b_t\cdot z_4 + m\cdot g\cdot L\cdot \sin z_3}{\left(J+mL^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot \cos^2 z_3}\right|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} \qquad (7.121)$$

and it is easy to get,

$$g_2 + Lg_4 = \frac{J+mL^2-mL^2\cos z_3}{\left(J+mL^2\right)\cdot\left(M+m\right)-m^2\cdot L^2\cdot \cos^2 z_3} \qquad (7.122)$$

Then the model of the zero dynamics of the cart is:

$$\dot{z}_1 = z_2 \qquad (7.123)$$

$$\dot{z}_2 = \left.\frac{L\left(b_t\cdot z_4 + m\cdot g\cdot L\cdot \sin z_3\right)}{J+mL^2-mL^2\cos z_3}\right|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} = \frac{-b_t z_2 - mgL^2\sin\dfrac{z_1}{L}}{J+mL^2-mL^2\cos\dfrac{z_1}{L}} \qquad (7.124)$$

It can be seen that the only difference between the inverted pendulum “cart” subsystem zero-dynamics model and the overhead crane “cart” subsystem zero-dynamics model is the sign of the term $mgL^2\sin(z_1/L)$. Based on the overhead crane “cart” subsystem zero-dynamics model, the stability of the system is investigated according to the Lyapunov function:

$$V=\frac{1}{2}\begin{pmatrix}z_1\\z_2\end{pmatrix}^T\begin{pmatrix}z_1\\z_2\end{pmatrix}=\frac{1}{2}\left[z_1^2+z_2^2\right] \tag{7.125}$$

Its derivative can be expressed as:

$$\frac{dV}{dt}=\dot{V}=\begin{pmatrix}z_1\\z_2\end{pmatrix}^T\begin{pmatrix}\dot{z}_1\\\dot{z}_2\end{pmatrix}=\begin{pmatrix}z_1\\z_2\end{pmatrix}^T\begin{pmatrix}z_2\\ \dfrac{-b_t z_2-mgL^2\sin\frac{z_1}{L}}{J+mL^2-mL^2\cos\frac{z_1}{L}}\end{pmatrix}=z_1z_2+\frac{-b_t z_2^2-mgL^2 z_2\sin\frac{z_1}{L}}{J+mL^2-mL^2\cos\frac{z_1}{L}}$$

$$=\frac{z_1z_2\left(J+mL^2-mL^2\cos\frac{z_1}{L}\right)-b_t z_2^2-mgL^2z_2\sin\frac{z_1}{L}}{J+mL^2-mL^2\cos\frac{z_1}{L}} \tag{7.126}$$

The overhead crane “cart” subsystem zero-dynamics model is built based on the inverted pendulum “cart” zero-dynamics model. The simulation results are presented in Figure 7.17 to Figure 7.20. Figure 7.17 and Figure 7.18 show that the system states convergence to zero. Figure 7.19 shows the chosen Lyapunov function is positive definite. Its derivative is not negative definite as shown in Figure 7.20, but it convergence to zero, which means the system is at the margin of the stabilization. The external controller is needed to stabilize the system.

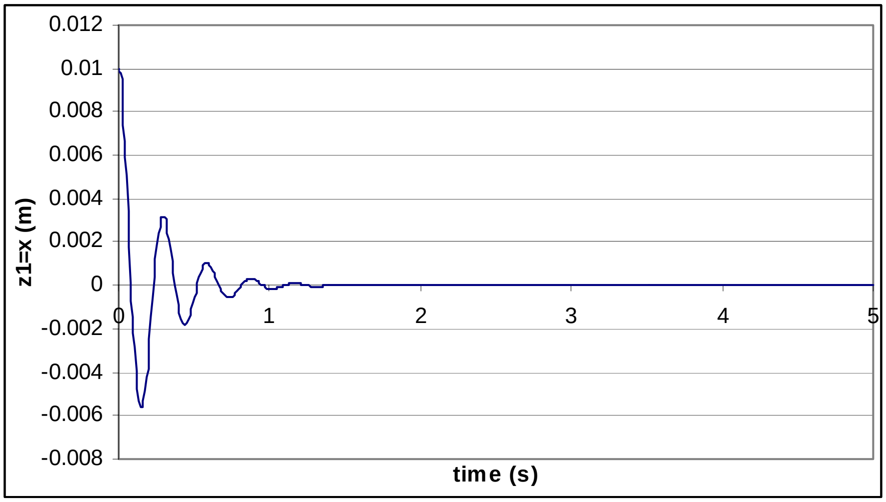


Figure 7.17: **The linearized overhead crane cart model state z_1**

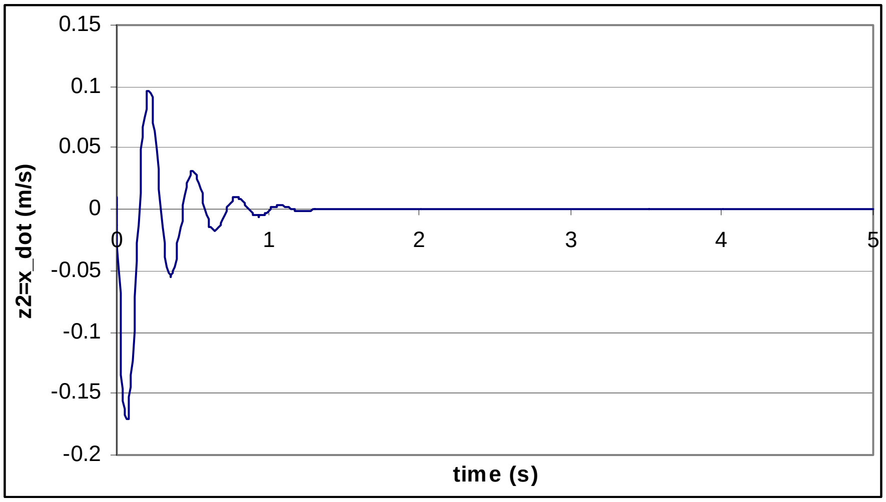


Figure 7.18: The overhead crane linearized cart model state z_2

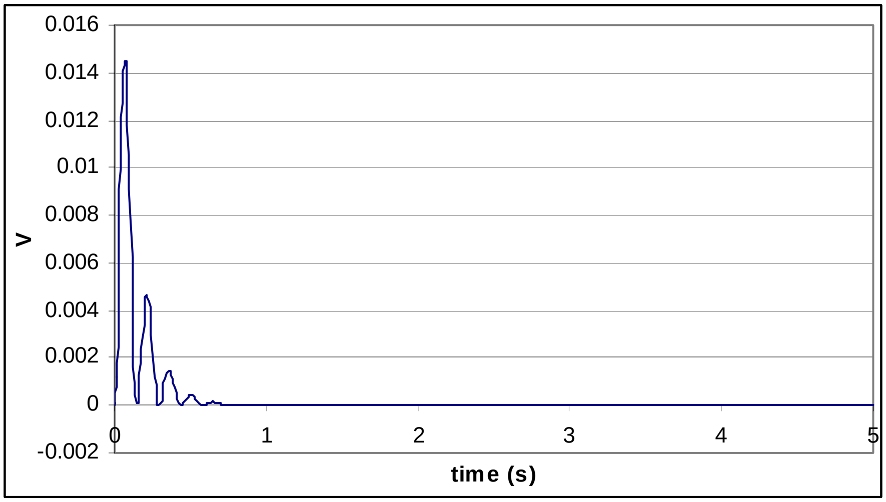


Figure 7.19: The Lyapunov function

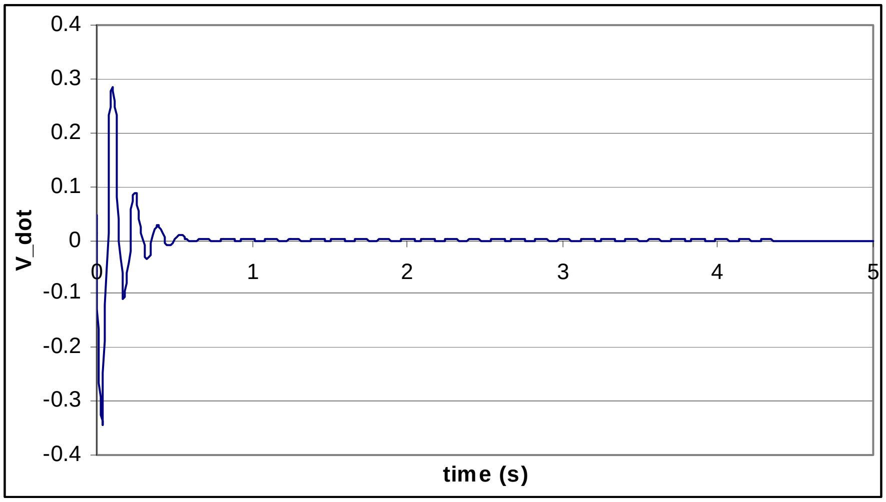


Figure 7.20: The derivative of the Lyapunov function

7.4.2.3.2 The zero dynamics of the pendulum

The “pendulum” subsystem mathematical model of the overhead crane is:

$$\dot{z}_3 = z_4 \tag{7.127}$$

$$\dot{z}_4 = f_4\Big|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} + g_4\Big|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} \cdot u_{zd} \tag{7.128}$$

After substitution of u_{zd}, the model is:

$$\dot{z}_3 = z_4 \tag{7.129}$$

$$\dot{z}_4 = f_4 + g_4 \frac{-f_2 - Lf_4}{g_2 + Lg_4}\Bigg|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} = \frac{f_4 g_2 + Lf_4 g_4 - f_2 g_4 - Lf_4 g_4}{g_2 + Lg_4} = \frac{f_4 g_2 - f_2 g_4}{g_2 + Lg_4}\Bigg|_{\substack{z_1=-Lz_3 \\ z_2=-Lz_4}} \tag{7.130}$$

By substituting the expression of f_2, f_4, g_2 and g_4, the linearized cart model can be derived as shown in Equation (7.136) and (7.137).

$$f_2 g_4 = \frac{\begin{bmatrix} -\left(J + mL^2\right)\cdot b \cdot z_2 + m^2 \cdot L^2 \cdot g \cdot \sin z_3 \cdot \cos z_3 + \\ + \left(J + mL^2\right)\cdot m \cdot L \cdot z_4^2 \cdot \sin z_3 + m \cdot L \cdot b_t \cdot z_4 \cdot \cos z_3 \end{bmatrix} \times \left[-m \cdot L \cdot \cos z_3\ \right]}{\left[\left(J + mL^2\right)\cdot\left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2 z_3\right]^2} \tag{7.131}$$

$$f_4 g_2 = \frac{\begin{bmatrix} + m \cdot L \cdot \cos z_3 \cdot b \cdot z_2 - \left(M + m\right)\cdot m \cdot g \cdot L \cdot \sin z_3 - \\ - m^2 \cdot L^2 \cdot z_4^2 \cdot \sin z_3 \cdot \cos z_3 - \left(M + m\right)\cdot b_t \cdot z_4 \end{bmatrix} \times \left(J + m \cdot L^2\right)}{\left[\left(J + mL^2\right)\cdot\left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2 z_3\right]^2} \tag{7.132}$$

$$f_4 g_2 - f_2 g_4 = \frac{\begin{bmatrix} -\left(M+m\right)\cdot m \cdot g \cdot L \cdot \sin z_3 \cdot \left(J + m \cdot L^2\right) \\ - m^2 \cdot L^2 \cdot z_4^2 \cdot \sin z_3 \cdot \cos z_3 \cdot \left(J + m \cdot L^2\right) \\ + m \cdot L \cdot \cos z_3 \cdot b \cdot z_2 \cdot \left(J + m \cdot L^2\right) - \left(M+m\right)\cdot b_t \cdot z_4 \cdot \left(J + m \cdot L^2\right) \\ - \left(J + mL^2\right)\cdot b \cdot z_2 \cdot m \cdot L \cdot \cos z_3 + m^3 \cdot L^3 \cdot g \cdot \sin z_3 \cdot \cos z_3^2 + \\ + \left(J + mL^2\right)\cdot m^2 \cdot L^2 \cdot z_4^2 \cdot \sin z_3 \cdot \cos z_3 + m^2 \cdot L^2 \cdot b_t \cdot z_4 \cdot \cos z_3^2 \end{bmatrix}}{\left[\left(J + mL^2\right)\cdot\left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2 z_3\right]^2} \tag{7.133}$$

Simplify the expression in Equation (7.133), then Equation (7.134) is obtained.

$$f_4 g_2 - f_2 g_4 = \frac{\begin{bmatrix} -\left(M+m\right)\cdot m \cdot g \cdot L \cdot \sin z_3 \cdot \left(J + m \cdot L^2\right) - \left(M+m\right)\cdot b_t \cdot z_4 \cdot \left(J + m \cdot L^2\right) \\ + m^3 \cdot L^3 \cdot g \cdot \sin z_3 \cdot \cos z_3^2 + m^2 \cdot L^2 \cdot b_t \cdot z_4 \cdot \cos z_3^2 \end{bmatrix}}{\left[\left(J + mL^2\right)\cdot\left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2 z_3\right]^2}$$

$$f_4 g_2 - f_2 g_4 = \frac{\begin{array}{l} -\left[\left(M+m\right)\cdot\left(J + m \cdot L^2\right) - m^2 \cdot L^2 \cdot \cos z_3^2\right]\cdot m \cdot g \cdot L \cdot \sin z_3 \\ -\left[\left(M+m\right)\cdot\left(J + m \cdot L^2\right) - m^2 \cdot L^2 \cdot \cos z_3^2\right]\cdot b_t \cdot z_4 \end{array}}{\left[\left(J + mL^2\right)\cdot\left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2 z_3\right]^2}$$

$$f_4 g_2 - f_2 g_4 = \frac{-\left[\left(M+m\right)\cdot\left(J + m \cdot L^2\right) - m^2 \cdot L^2 \cdot \cos z_3^2\right]\cdot\left[m \cdot g \cdot L \cdot \sin z_3 + b_t \cdot z_4\right]}{\left[\left(J + mL^2\right)\cdot\left(M + m\right) - m^2 \cdot L^2 \cdot \cos^2 z_3\right]^2}$$

$$f_4 g_2 - f_2 g_4 = \frac{-\left[m\cdot g\cdot L\cdot \sin z_3 + b_t \cdot z_4\right]}{\left(J+mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos^2 z_3} \tag{7.134}$$

and it is easy to get,

$$g_2 + Lg_4 = \frac{J+mL^2 - mL^2\cos z_3}{\left(J+mL^2\right)\cdot\left(M+m\right) - m^2\cdot L^2\cdot \cos^2 z_3} \tag{7.135}$$

Then the model of the zero dynamics of the pendulum is:

$$\dot{z}_3 = z_4 \tag{7.136}$$

$$\dot{z}_4 = \left.\frac{-b_t\cdot z_4 - m\cdot g\cdot L\cdot \sin z_3}{J+mL^2 - mL^2\cos z_3}\right|_{\substack{z_1=-Lz_3\\ z_2=-Lz_4}} \tag{7.137}$$

Similar with the difference between the inverted pendulum’s “cart” subsystem zero-dynamics model and the overhead crane’s “cart” subsystem zero-dynamics model, the difference between the inverted pendulum’s “pendulum” subsystem zero-dynamics and the overhead crane’s “pendulum” subsystem zero-dynamics is the sign of the term mgLsin(z_3). Based on the study of the overhead crane “pendulum” subsystem zero-dynamics model, the stability of the system is investigated according to the Lyapunov function:

$$V = \frac{1}{2}\begin{pmatrix} z_3 \\ z_4 \end{pmatrix}^T \begin{pmatrix} z_3 \\ z_4 \end{pmatrix} \tag{7.138}$$

Its derivative can be expressed as:

$$\frac{dV}{dt} = \dot{V} = \begin{pmatrix} z_3 \\ z_4 \end{pmatrix}^T \begin{pmatrix} \dot{z}_3 \\ \dot{z}_4 \end{pmatrix} = \begin{pmatrix} z_3 \\ z_4 \end{pmatrix}^T \begin{pmatrix} z_4 \\ \left.\dfrac{-b_t\cdot z_4 - m\cdot g\cdot L\cdot \sin z_3}{J+mL^2 - mL^2\cos z_3}\right| \end{pmatrix} = z_3 z_4 + \frac{-b_t\cdot z_4^2 - m\cdot g\cdot L\cdot z_4\cdot \sin z_3}{J+mL^2 - mL^2\cos z_3}$$

$$= \frac{z_3 z_4\left(J+mL^2 - mL^2\cos\dfrac{z_1}{L}\right) - b_t\cdot z_4^2 - m\cdot g\cdot L\cdot z_4\cdot \sin z_3}{J+mL^2 - mL^2\cos\dfrac{z_1}{L}} \tag{7.139}$$

The overhead crane’s “pendulum” subsystem zero-dynamics model is obtained by modifying the inverted pendulum’s “pendulum” subsystem zero-dynamics model. The simulation results are presented in Figure 7.21 to Figure 7.24. Figure 7.21 and Figure 7.22 show that the system states convergence to zero. Figure 7.23 shows the chosen Lyapunov function is positive definite. Its derivative is not negative definite as shown in Figure 7.24. The derivative of the Lyapunov function converges to zero means the system is at the margin of the stabilization. The external controlle is needed to stabilize the system.

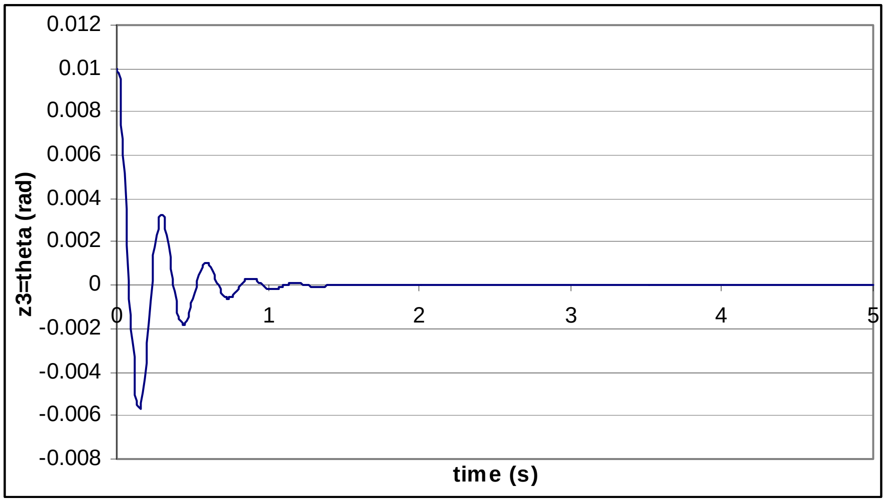


Figure 7.21: The linearized overhead crane “pendulum” model state z_1

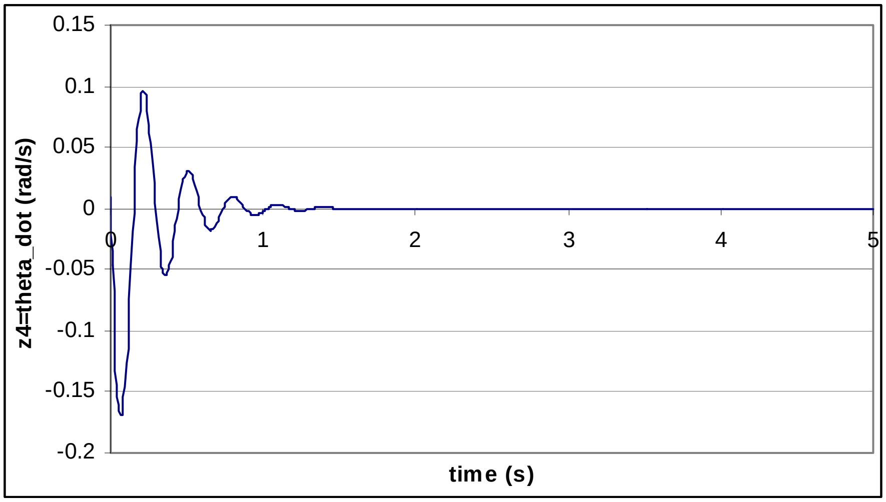


Figure 7.22: The linearized overhead crane “pendulum” model state z_2

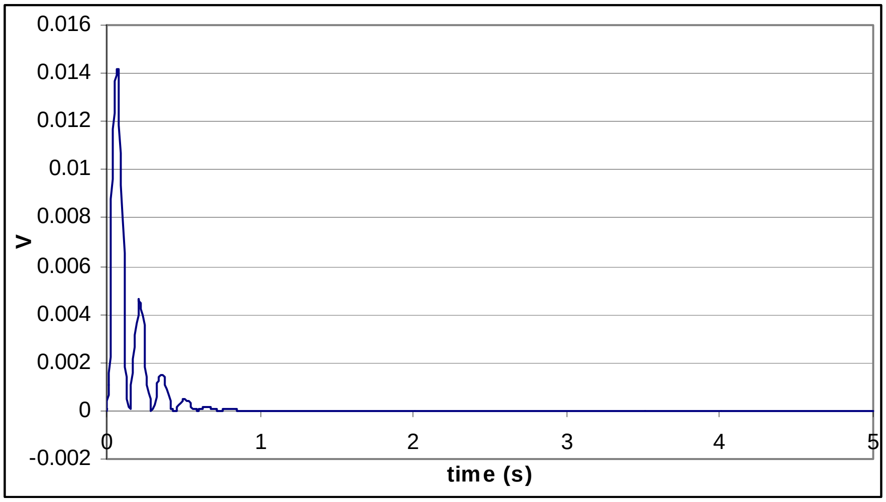


Figure 7.23: The Lyapunov function

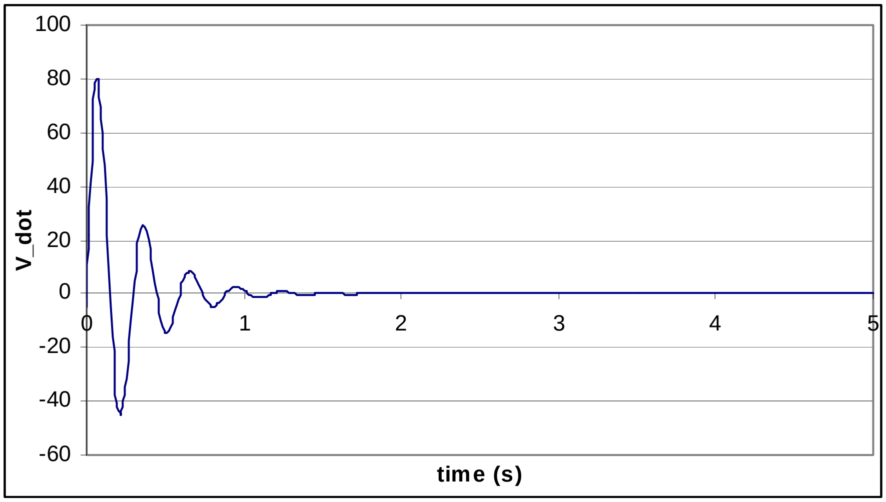


Figure 7.24: The derivative of the Lyapunov function

7.4.2.3.3 The linearized overhead crane model

From Equation (7.106) the linearized overhead carne model can be derived. Equation (7.140) is the linearized closed-loop system.

$$A_{lin}(z)+B_{lin}(z)\cdot v=\begin{bmatrix} z_2 \\ \dfrac{L[f_2(z)g_4(z)-g_2(z)\cdot f_4(z)]}{g_2(z)+Lg_4(z)} \\ z_4 \\ \dfrac{g_2(z)f_4(z)-g_4(z)\cdot f_2(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}+\begin{bmatrix} 0 \\ \dfrac{g_2(z)}{g_2(z)+Lg_4(z)} \\ 0 \\ \dfrac{g_4(z)}{g_2(z)+Lg_4(z)} \end{bmatrix}v \tag{7.140}$$

where $\dfrac{L[f_2(z)g_4(z)-g_2(z)\cdot f_4(z)]}{g_2(z)+Lg_4(z)}=\dfrac{Lb_t z_4+mgL^2\sin z_3}{J+mL^2-mL^2\cos z_3}$;

$$\frac{g_2(z)f_4(z)-g_4(z)\cdot f_2(z)}{g_2(z)+Lg_4(z)}=\frac{-b_t\cdot z_4-m\cdot g\cdot L\cdot\sin z_3}{J+mL^2-mL^2\cos z_3};$$

$$\frac{g_2(z)}{g_2(z)+Lg_4(z)}=\frac{J+mL^2}{J+mL^2-mL^2\cos z_3};$$

$$\frac{g_4(z)}{g_2(z)+Lg_4(z)}=\frac{-mL\cos z_3}{J+mL^2-mL^2\cos z_3};$$

Same as Equation (7.106), the obtained closed-loop system in Equation (7.140) is not fully linearized by the designed linearizing controller u, because the relative degree of the linearizing system is two and the nonlinear system has order four. In order to improve the performance of the obtained closed-loop system, a linear controller is necessary to be designed. In order to obtain it, the closed-loop system is linearized around the point $\theta = z_3 \approx 0$, for which $\sin z_3 = z_3$ and $\cos z_3 = 1$ and the value $\dot{\theta}^2$ is approximately equal to zero.

Then: $\dfrac{L\left[f_2(z)g_4(z) - g_2(z)\cdot f_4(z)\right]}{g_2(z) + Lg_4(z)} = \dfrac{-Lb_t z_4 - mgL^2 z_3}{J + mL^2 - mL^2} = \dfrac{-Lb_t z_4 - mgL^2 z_3}{J}$;

$$\frac{g_2(z)f_4(z) - g_4(z)\cdot f_2(z)}{g_2(z) + Lg_4(z)} = \frac{-b_t \cdot z_4 - m \cdot g \cdot L \cdot z_3}{J + mL^2 - mL^2} = \frac{-b_t \cdot z_4 - m \cdot g \cdot L \cdot z_3}{J};$$

$$\frac{g_2(z)}{g_2(z) + Lg_4(z)} = \frac{J + mL^2}{J + mL^2 - mL^2} = \frac{J + mL^2}{J};$$

$$\frac{g_4(z)}{g_2(z) + Lg_4(z)} = \frac{-mL\cos z_3}{J + mL^2 - mL^2} = \frac{-mL}{J};$$

After substitution of the obtained transformations the linearized system can be written in the state space form as shown in Equation (7.141):

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & \dfrac{mgL^2}{J} & \dfrac{Lb_t}{J} \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\dfrac{mgL}{J} & -\dfrac{b_t}{J} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{J + mL^2}{J} \\ 0 \\ \dfrac{-mL}{J} \end{bmatrix} v \qquad (7.141)$$

with the output: $y = z_1 + Lz_3$

The system in Equation (7.141) is used to design the overhead crane linear closed-loop control v in the following section.

7.5 Design of the linear controller

The system stability is examined by studying the system's zero-dynamics. It can be seen that the nonlinear controller is not sufficient to control the system. The systems are at the margin of the stability, so the external linear controllers are needed to stabilize the whole system. The external linear controller is design based on the linear quadratic regulator (LQR) technique. In this chapter, the LQR technique is applied in the same manner as it is implemented in Chapter 5, but the linearized mathematical models used in Chapter 5 are replaced by the input-output feedback linearized models. As shown in Figure 7.25, the linear control v is used to stabilize the linearized nonlinear system. Here the linearized nonlinear system is not simply linearized by the approximation of the trigonometric function, such as: if $-5^\circ < \theta < 5^\circ$, $\sin\theta \approx 0$ and $\cos\theta \approx 1$, but it is linearized by the nonlinear control u.

LQR technique is presented and applied in Chapter 5. In a LQR design, the gain matrix $\hat{K}$ for a linear state feedback control law $u(t) = -\hat{K}e(t)$ is found by minimizing the performance index in the form of:

$$J = \int_0^\infty e(t)^T Qe(t) + u(t)^T Ru(t)dt \tag{7.142}$$

where $Q \in R^{5\times5}$ is a positive definite (or positive semidefinite) Hermitian or real symmetric matrix. $R \in R^1$ is a positive definite Hermitian or real symmetric matrix. The matrices Q and R are weighting parameters that penalize certain states or control inputs and $y^{sp}=0$. In Matlab, the command “lqr” is used to solve the continuous-time, linear, quadratic regulator problem. This command computes the feedback gain matrix $\hat{K}$. The weighting matrices Q and R used for controlling the inverted pendulum and the overhead crane are:

$$Q_{ip}=\text{diag}\,\{9000 \quad 130 \quad 10 \quad 3 \quad 220000\} \qquad R_{ip}=1 \tag{7.143}$$

$$Q_{oc}=\text{diag}\,\{150 \quad 150 \quad 100 \quad 100 \quad 3000\} \qquad R_{oc}=1 \tag{7.144}$$

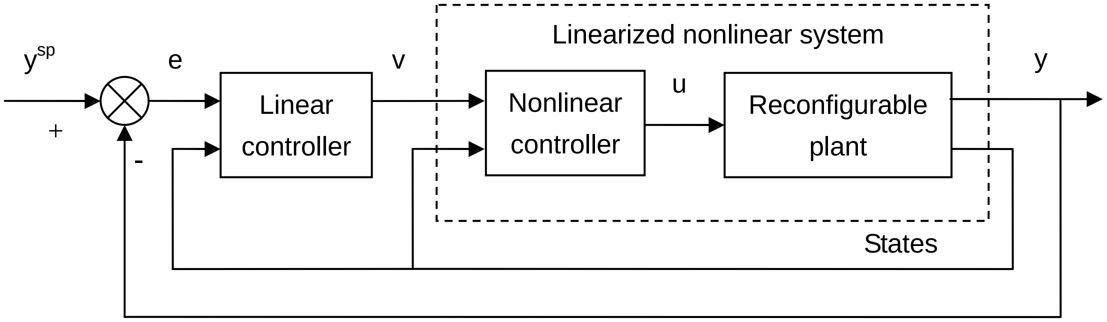


Figure 7.25: The structure block diagram of the feedback linearization system

By using the selected weighting matrices, the LQR controllers are designed by using Matlab. The controller gain matrix $\hat{K}$ for the inverted pendulum and the overhead crane are:

$$\hat{K} = [-266.0967 \quad -64.4596 \quad -129.8868 \quad -17.6466 \quad 469.0416] \tag{7.145}$$

$$\hat{K} = [56.6956 \quad 26.1802 \quad -38.2601 \quad -2.1790 \quad -54.7723] \tag{7.146}$$

7.6 Simulation

Based on the block diagram of the feedback linearization control system in Figure 7.25, the system is built in Simulink as shown in Figure 7.26. The Simulink files for the inverted pendulum and the overhead crane are called “FBL_IP.mdl” and “FBL_OC.mdl” respectively. The Simulink block diagrams for the inverted pendulum and the overhead crane have the same structure. The differences are the actual controlled plant and the parameters used to construct the controllers. The subsystem contained in Figure 7.26 consists of the linear controller and the nonlinear controller which are shown in Figure 7.27 and Figure 7.28 respectively. The associated m-files: "*FBL_IP.m*" and "*FBL_OC.m*" are used to declare the necessary parameters and calculate the feedback control gain matrices. These two m-files are presented in Appendix A. 7 and A. 8.

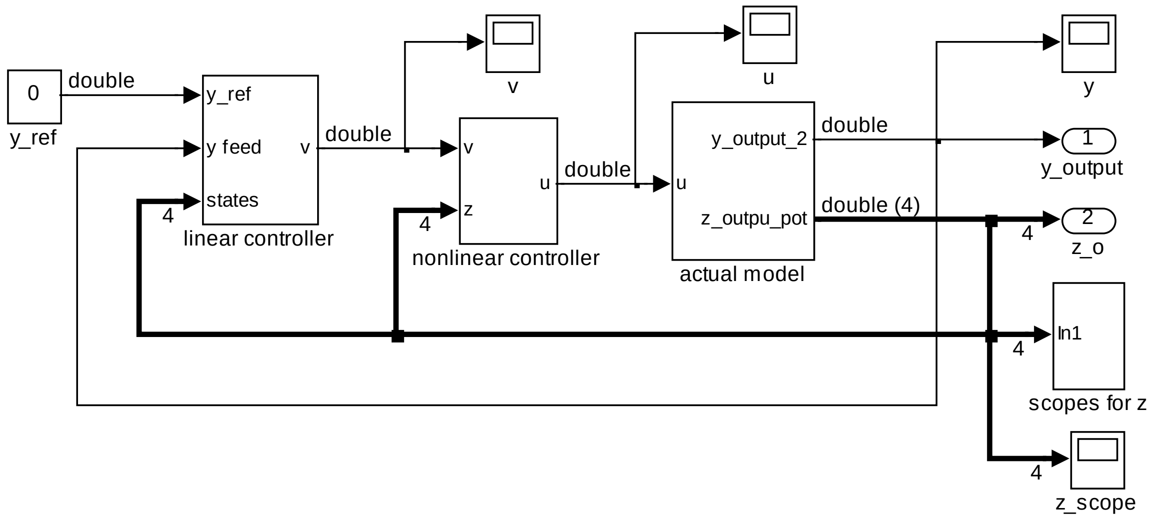


Figure 7.26: The Simulink block diagram of the feedback linearization system

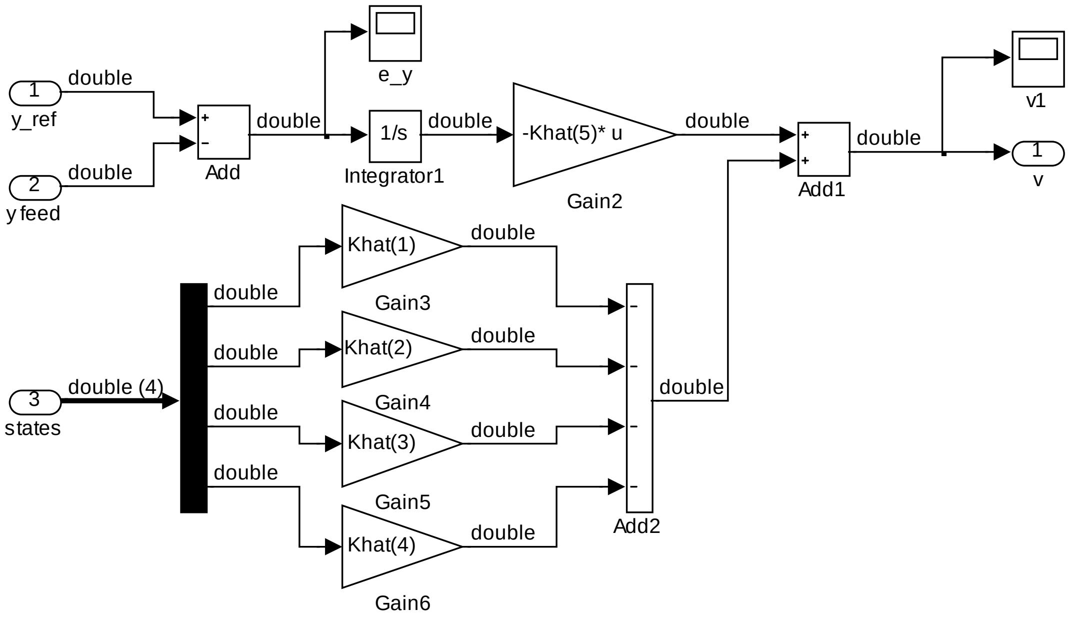


Figure 7.27: The Simulink block diagram of the linear controller

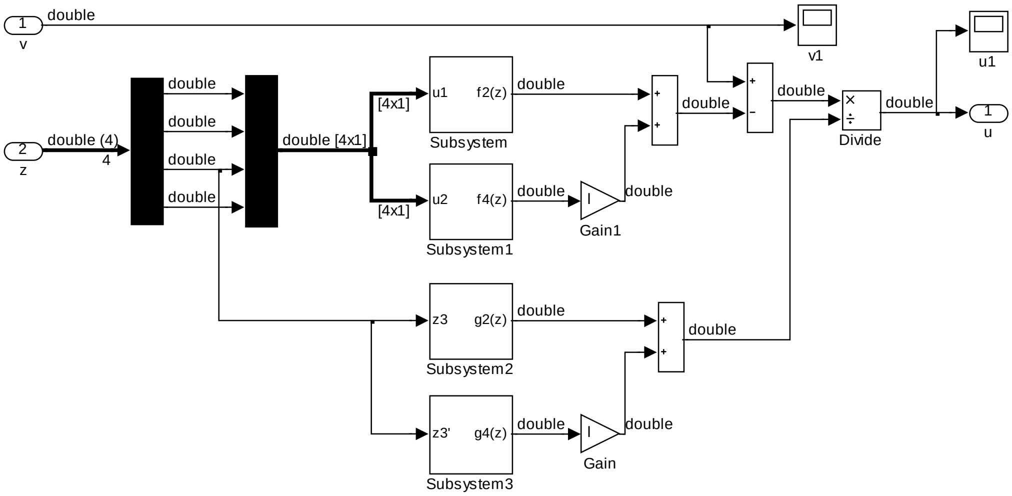


Figure 7.28: **The Simulink block diagram of the nonlinear controller**

The simulation results are presented in Figure 7.29 to Figure 7.52. The presented signals include the position signal x, the angle signal θ, the velocity signal $\dot{x}$, the angular velocity signal $\dot{\theta}$, the output signal y, the linear and nonlinear control signals v and u. The simulations are done for different set points 0m, 0.05m and 0.1m for both of the inverted pendulum and the overhead crane. The initial conditions for both the inverted pendulum and the overhead crane are z(0)=[0.03 0.05 0.09 0.04].

From the simulation results, the following points can be concluded:

- All the states are stabilized
- The plant output follows the set point trajectories
- The errors go to zeros

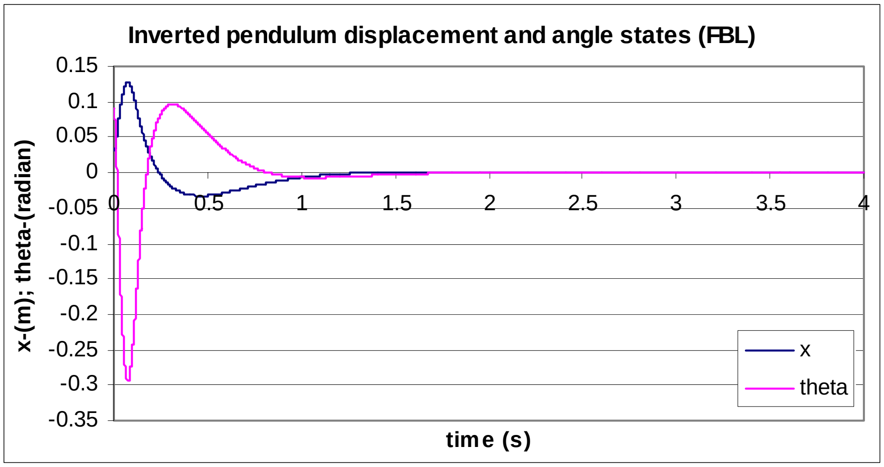


Figure 7.29: **Inverted pendulum displacement and angle state signals when set point is 0m**

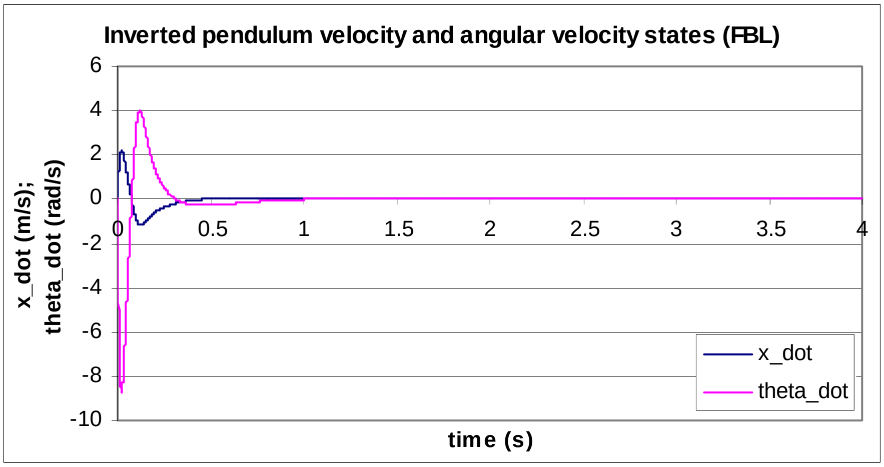


Figure 7.30: Inverted pendulum velocity and angular velocity state signals when the set point is 0m

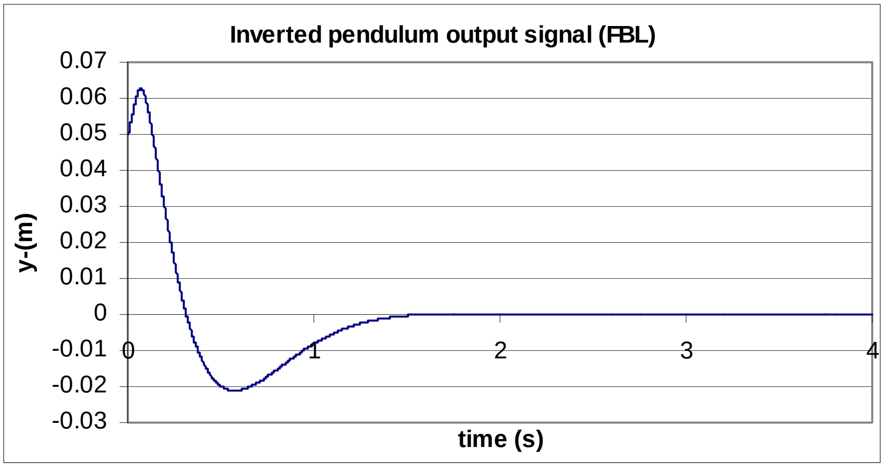


Figure 7.31: Inverted pendulum output signal when the set point is 0m

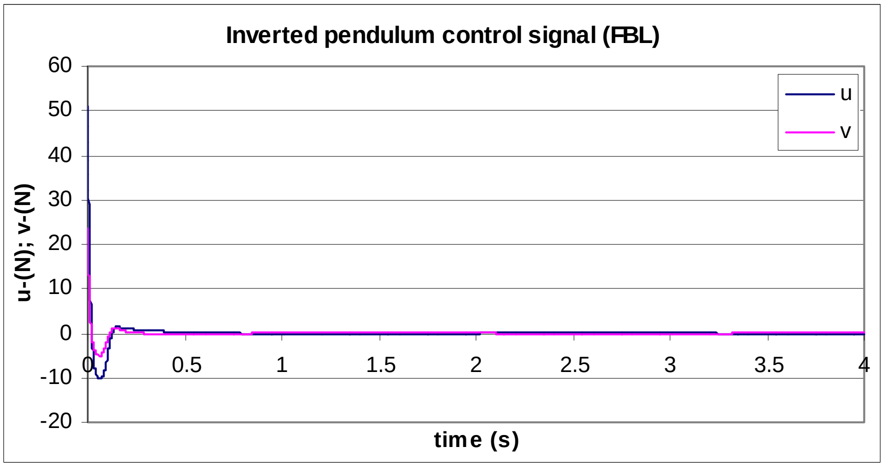


Figure 7.32: Inverted pendulum control signals when the set point is 0m

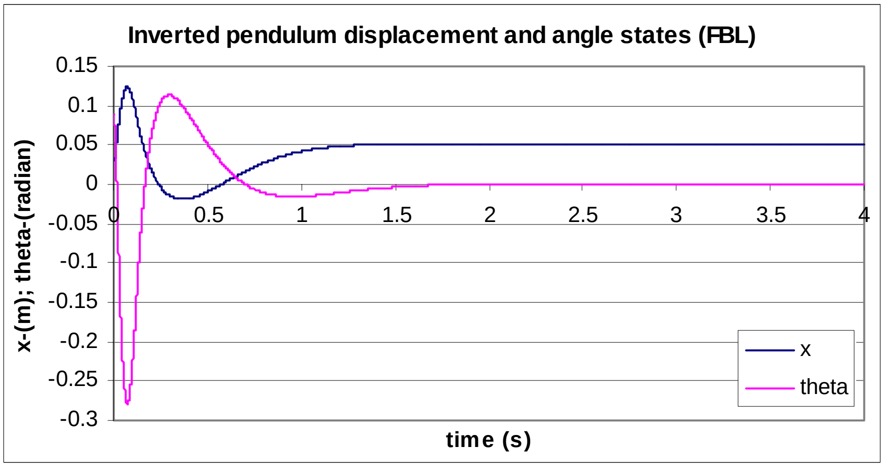


Figure 7.33: Inverted pendulum displacement and angle signals when the set point is 0.05m

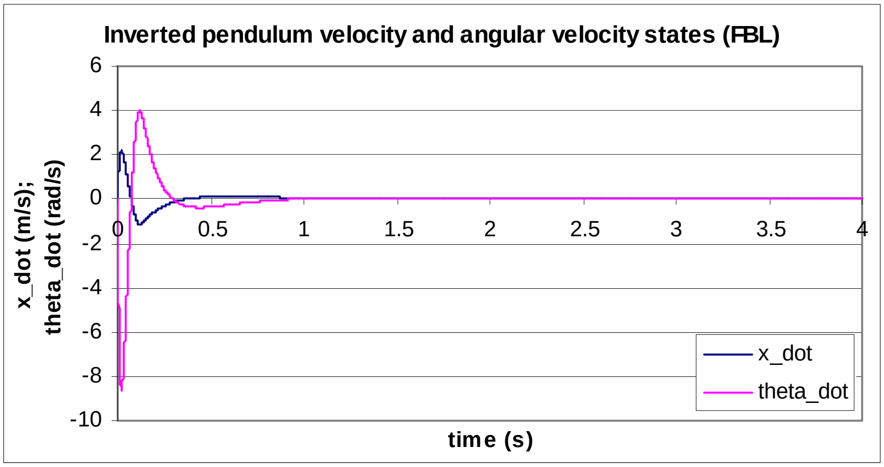


Figure 7.34: Inverted pendulum velocity and angular velocity signals when set point is 0.05m

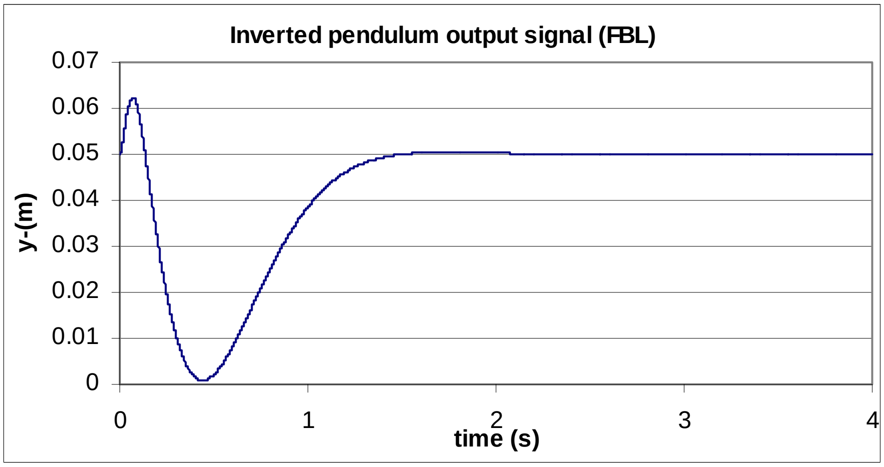


Figure 7.35: Inverted pendulum output signal when the set point is 0.05m

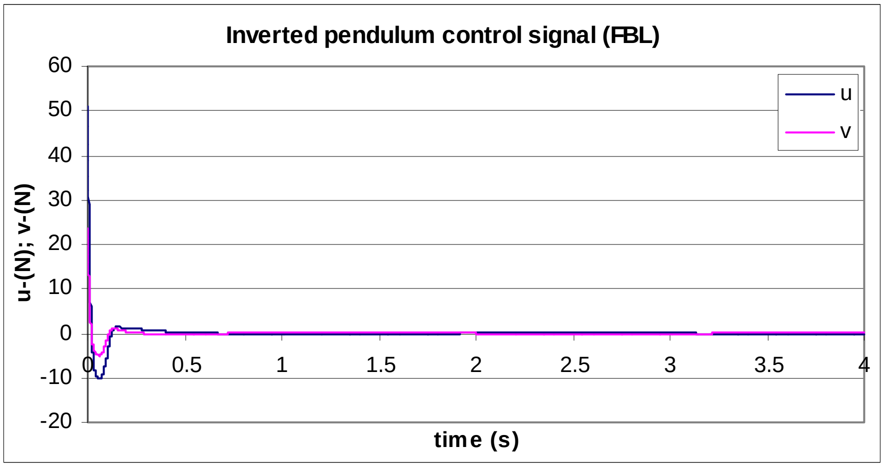


Figure 7.36: Inverted pendulum control signals when the set point is 0.05m

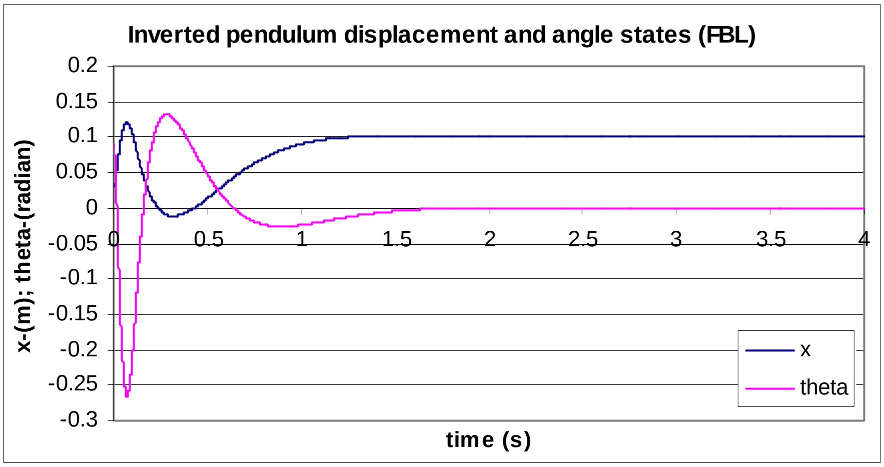


Figure 7.37: Inverted pendulum displacement and angle signals when the set point is 0.1m

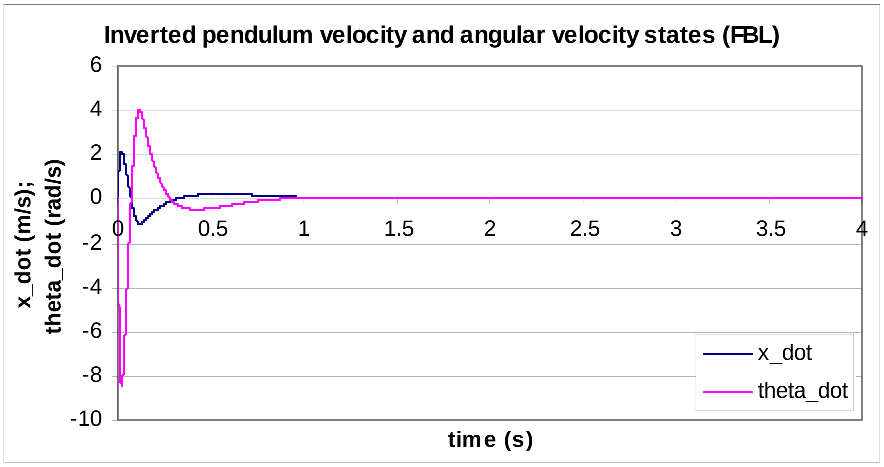


Figure 7.38: Inverted pendulum velocity and angular velocity state signals when the set point is 0.1m

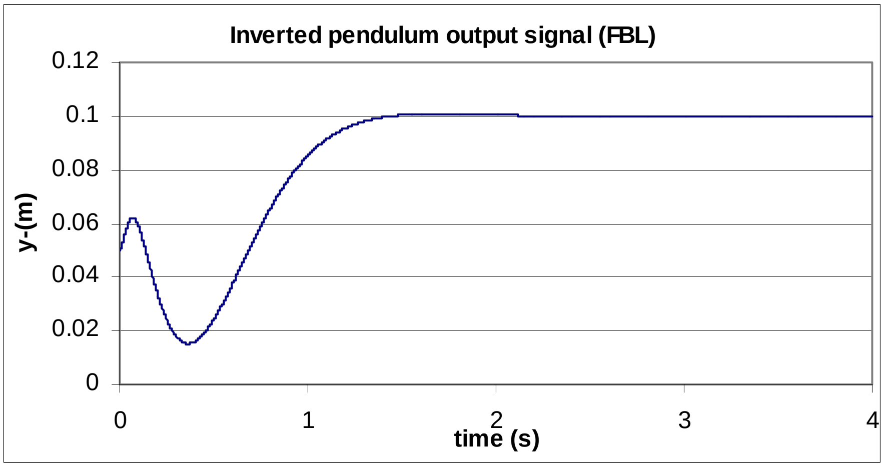


Figure 7.39: Inverted pendulum output signal when the set point is 0.1m

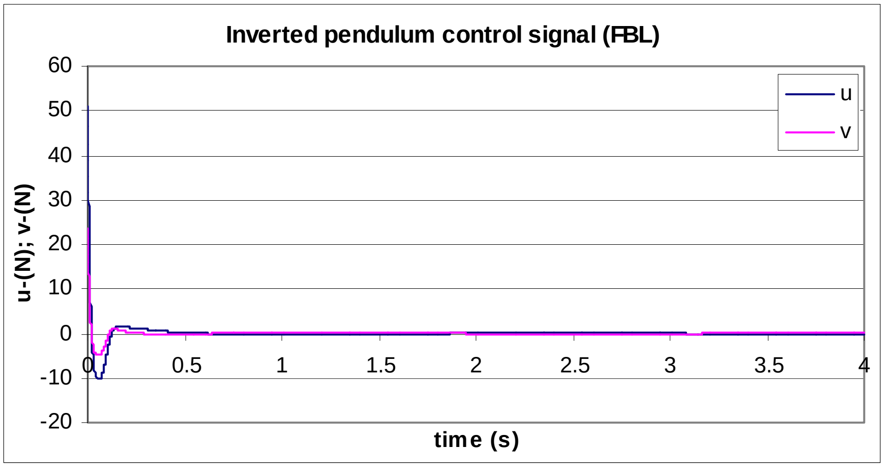


Figure 7.40: Inverted pendulum control signals when the set point is 0.1m

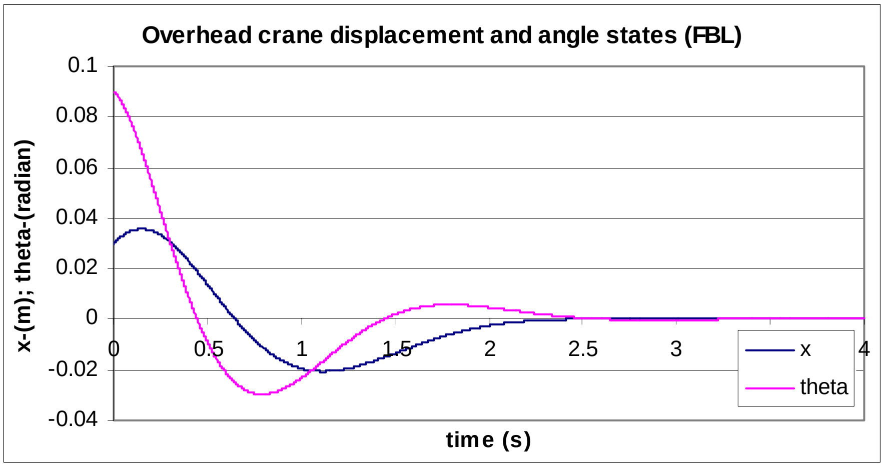


Figure 7.41: Overhead crane displacement and angle state signals when the set point is 0m

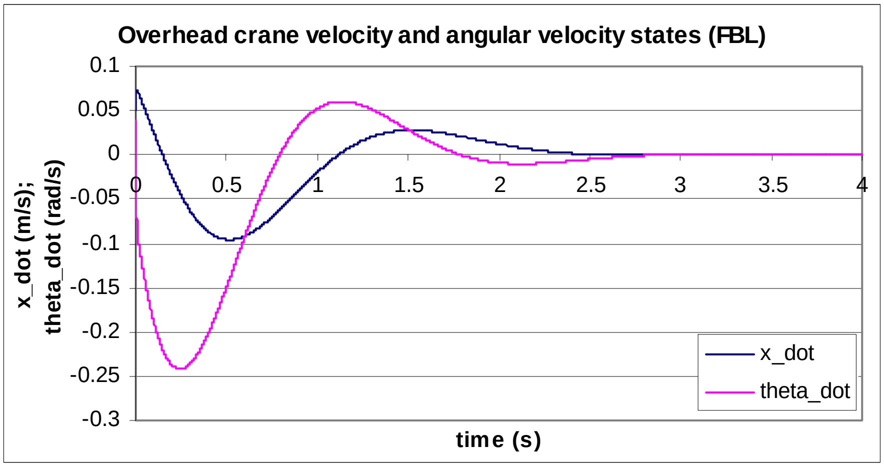


Figure 7.42: Overhead crane velocity and angular velocity state signals when the set point is 0m

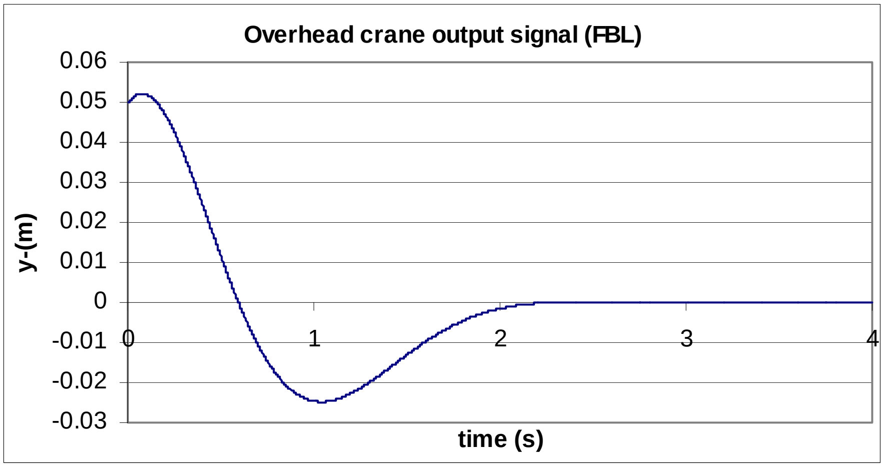


Figure 7.43: Overhead crane output signal when the set point is 0m

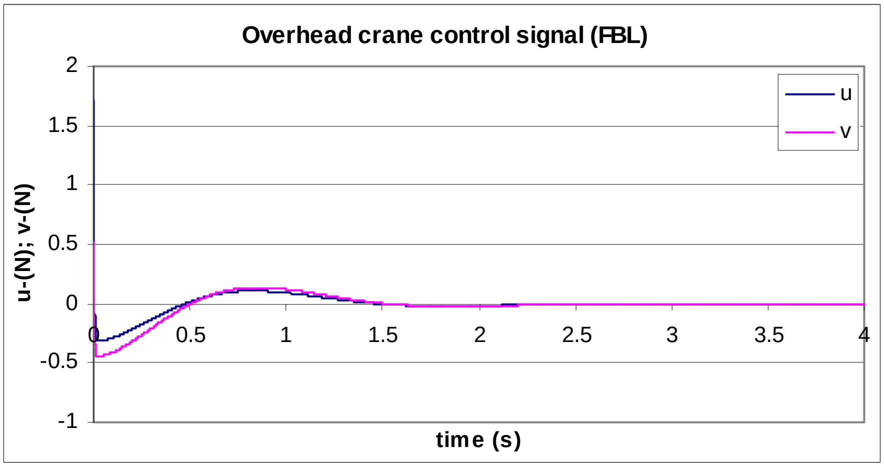


Figure 7.44: Overhead crane control signals when the set point is 0m

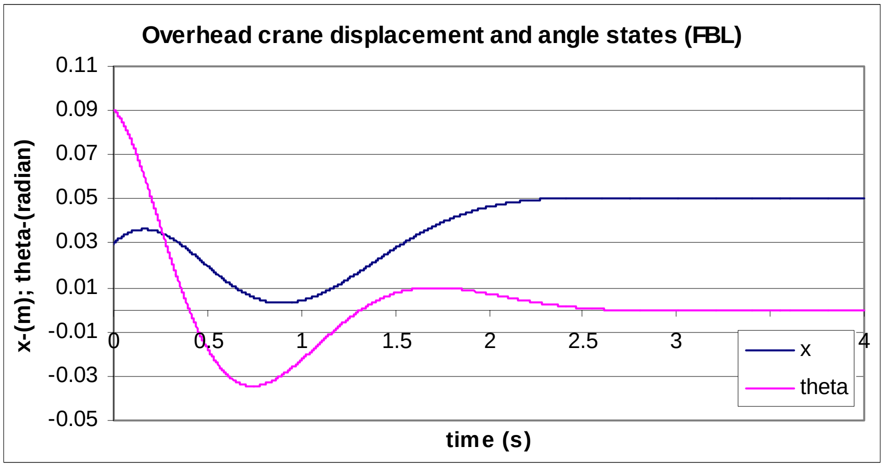


Figure 7.45: Overhead crane displacement and angle state signals when the set point is 0.05m

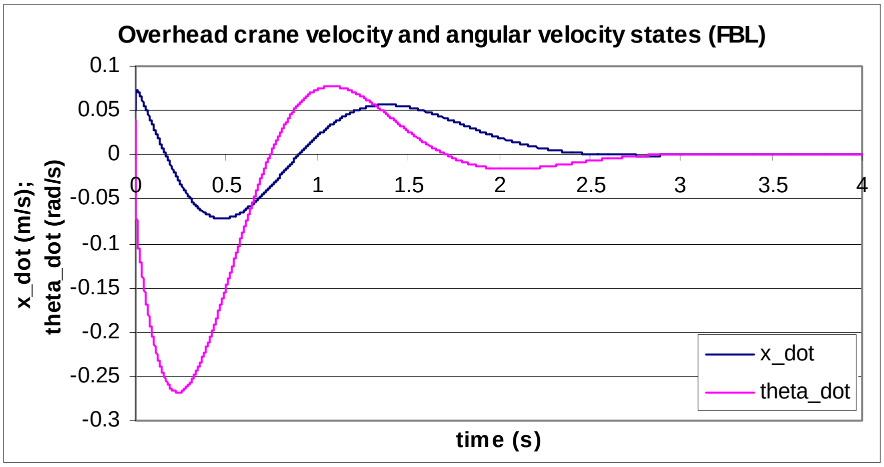


Figure 7.46: Overhead crane velocity and angular velocity state signals when the set point is 0.05m

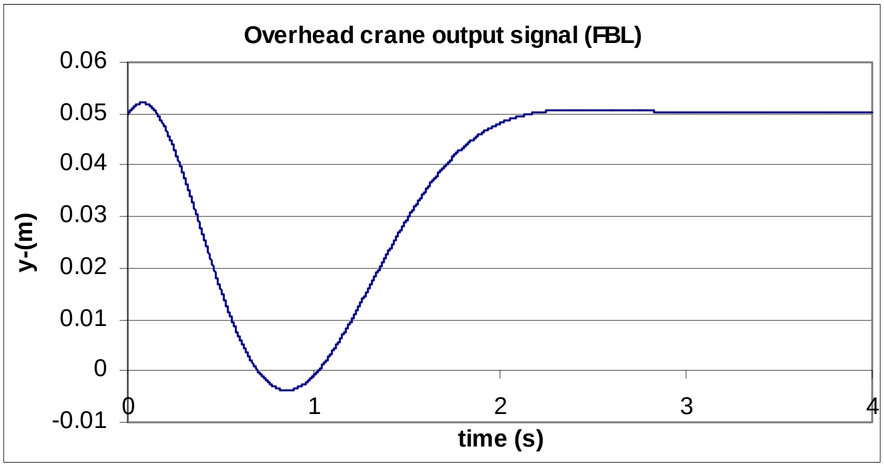


Figure 7.47: Overhead crane output signal when the set point is 0.05m

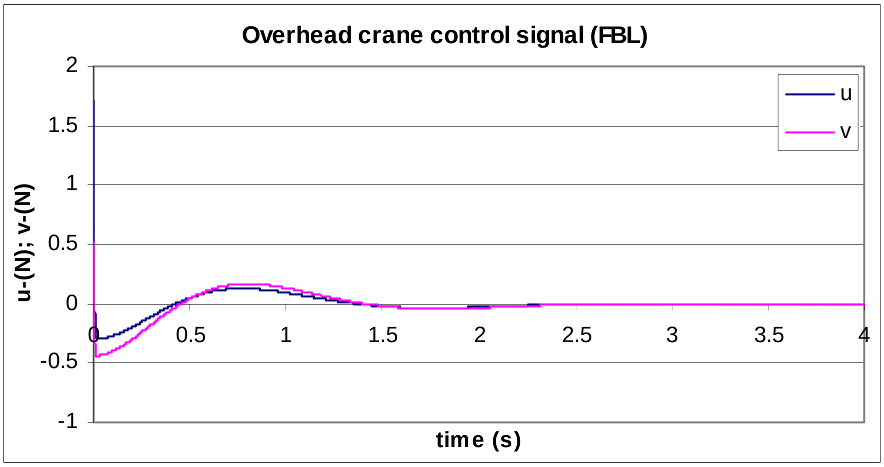


Figure 7.48: Overhead crane control signals when the set point is 0.05m

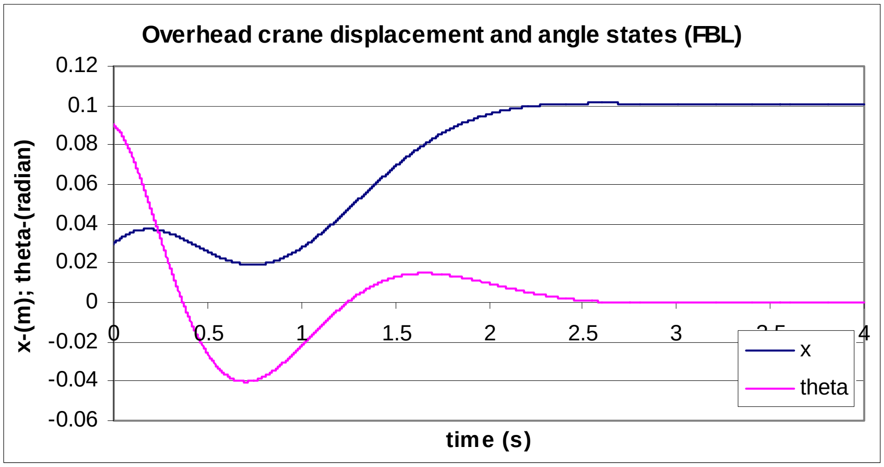


Figure 7.49: Overhead crane displacement and angle state signals when the set point is 0.1m

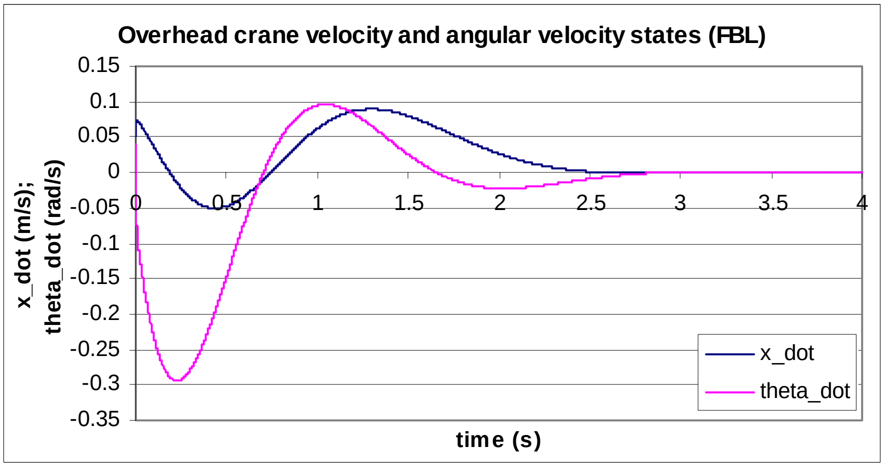


Figure 7.50: Overhead crane velocity and angular velocity state signals when the set point is 0.1m

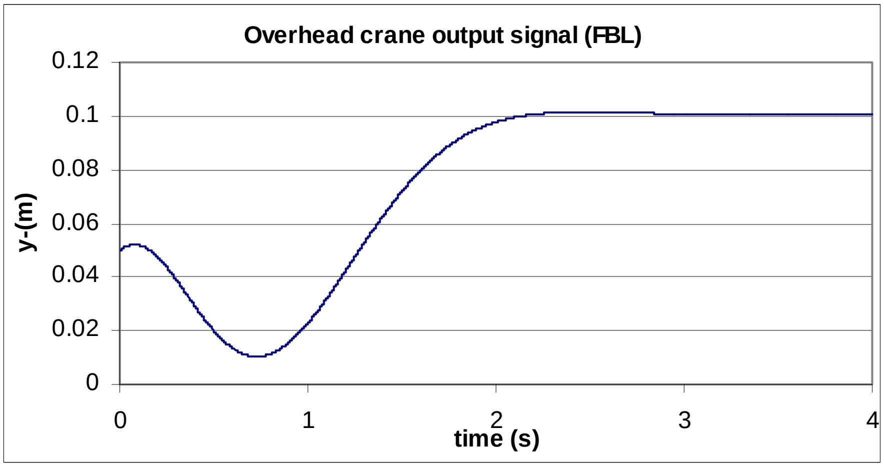


Figure 7.51: Overhead crane output signals when the set point is 0.1m

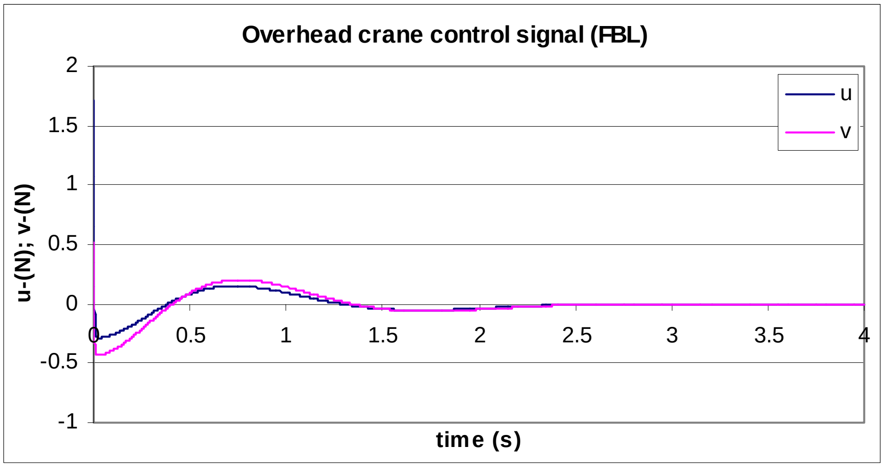


Figure 7.52: Overhead crane control signals when the set point is 0.1m

7.7 Conclusion

In this chapter, the input-output feedback linearization is presented in a very detail way. The nonlinear systems of the inverted pendulum and the overhead crane are linearized by using the input-output linearization. By differentiating the output, the linear dependent relationship between the input u and output y are crated. Based on this relationship, the nonlinear control u is found. By studying the system's zero-dynamics, it is proven that the nonlinear controller is not sufficient to stabilize the complete system, the external controllers are needed. In this project, the external controllers are designed based on the LQR technique. The simulation results show that the systems are stabilized by the external linear controller. Compare to the linear controller, the feedback linearization approach gives better system response in terms of the system performances. Compare with the Lyapunov stability based nonlinear control, they have very similar behaviours. In the next chapter, the Real-Time implementation is done based on the designed nonlinear controllers, including the Lyapunov stability theory based MRC and the feedback linearization method based control.

CHAPTER EIGHT

REAL-TIME IMPLEMENTATION CONFIGURATION AND RESULTS

8.1 Introduction

In order to verify the concept of the RMS, the developed nonlinear controllers are implemented in Real -Time for the lab-scaled RMS which consists of the reconfigurable plant and the controller. The plant and the reconfigurable controllers are reconfigured in parallel and synchronously. In other words, the reconfigurable plant and the reconfigurable controller are always corresponding to each other. The control designs are useful and meaningful only when they can be used to achieve certain goals, for example, to regulate or stabilize certain processes or quantities in real world. This also requires the Real-Time implementation to prove the designed controller achieves the specifications. Real-Time implementation means using the software and hardware platform to realize the control implementations in reality. Difference between the simulation and the Real-Time implementation is that in the Real-Time implementation all the mathematical functions are replaced by the actual devices or equipments. The plant mathematical model is replaced by the actual plant.

In the thesis, the closed-loop control systems are simulated in Chapter 5, Chapter 6 and Chapter 7. Good results are received. The next step is to shift the project from the simulation platform to the Real-Time implementation platform. On the Real-Time implementation platform, the Bytronic pendulum system is used as the reconfigurable plant and the National Instruments (NI) CompactRIO FPGA with the NI LabVIEW are used as the final reconfigurable controller. As all the simulations are done in the Matlab software environment, it is convenient first to transfer the program from the simulation mode to the Real-Time implementation mode in the same environment using Real-Time Windows Target (RTWT). Then, the Real-Time implementations of the reconfigurable controllers are done on the platform of the I/O DAQ device with signal conditioning, PC and Matlab first, it is considered as the first transition platform. Once the Real-Time implementations are achieved on this platform, which means the project is implementable and the plant models and the controllers are designed and developed successfully. Then, the reconfigurable controller is implemented on the platform of I/O DAQ device and signal conditioning with LabVIEW which is considered as the second transition platform. On this platform, the software of reconfigurable controllers is designed in the LabVIEW software environment. The first and second transition platforms share the same hardware structure, the controller algorithms are sitting in the PC, the I/O ports are connected through the I/O DAQ device for the signal conditioning with the plant, but they have different software environment, Matlab and LabVIEW respectively. The second transition platform is used to create a LabVIEW version of the reconfigurable controllers in term of software. Based on these LabVIEW version reconfigurable controllers, the final reconfigurable controllers based on the NI compactRIO FPGA with LabVIEW is developed.

This chapter is organized in the following manner. In the section 8.1, the introduction gives an overview of the chapter. In the section 8.2, the I/O DAQ devices, signal conditioning and PC with Matlab Real-Time implementation platform is studied. Include the hardware and software. The CompactRIO Real-Time implementation platform is studied in the section 8.3 in terms of the structures of the CompactRIO implementation platform. The Real-Time implementation platform configurations are shown in the section 8.4 for both Real-Time implementation platforms. In the section 8.5, the Real-Time implementation results are presented. These results include: a) by using the I/O DAQ devices and signal conditioning with PC platform associated with the Lyapunov stability theory based control and feedback linearization theory based control; b) the CompactRIO and PC platform with Lyapunov stability theory based control and feedback linearization theory based control. The conclusion is presented in the section 8.6.

8.2 I/O DAQ device with signal conditioning and PC Real-Time implementation platform

The I/O DAQ device with signal conditioning and PC Real-Time implementation platform consists of five basic components shown in Figure 8.1. These basic components are:

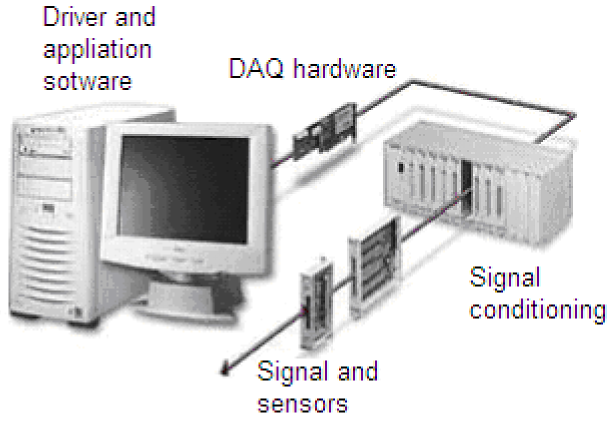


Figure 8.1: **Data Acquisition system**

(Adopted from http://zone.ni.com/devzone/cda/tut/p/id/3536, April 25th 2007)

- Transducers and sensors
- Signals
- Signal conditioning
- I/O DAQ hardware
- Drivers and Application software

These five components are introduced associate with the project individually in the following subsections.

8.2.1 Transducers and sensors

A transducer is a device that converts a physical phenomenon into a measurable electrical signal, such as voltage or current. The ability of I/O DAQ system to measure different phenomena depends on the transducers to convert the physical phenomena into signals measurable by the I/O DAQ hardware. In this project, the transducers are integrated with the reconfigurable plant. A 5K servo potentiometer is used to measure the angle signal (θ) and a 5K multi-turn potentiometer is used to measure the position signal (x).

8.2.2 Signals

The appropriate transducers convert physical phenomena into measurable signals. Signals can be classified into analog and digital ones. Analog signals have three primary characteristics including level, shape and frequency. The useful information that can be measured from a digital signal includes the state and the rate. The transducers in the reconfigurable plant convert the angle signal (θ) and the position signal (x) into the electrical voltage (V) form.

8.2.3 Signal conditioning

Sometimes the generated by the transducers signals are not suitable to be measured by the I/O DAQ device directly. Signal conditioning is essential for an effective I/O DAQ system. It maximizes the accuracy of a system, allows sensors to operate properly, and guarantees safeness of the control system. The signal conditioning used in this project is the SCC portable, low-cost, modular signal conditioning system. SCC products condition a variety of analog input and digital I/O signals. Because the transducers used in the pendulum system generate analog signals and the control signal send to the pendulum system is an analog signal, the analog input and output SCC modules are used for the data acquisition. These pods like SCC modules are plugged into NI SC-2345 connector blocks in the signal conditioning carrier shown in Figure 8.2.

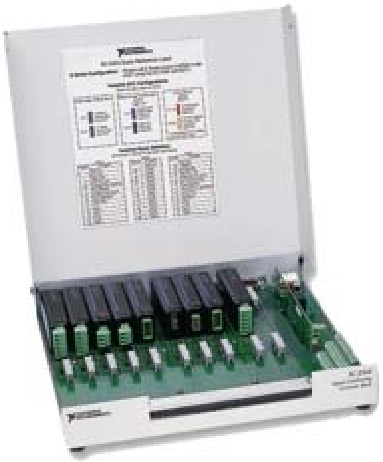


Figure 8.2: **SC 2345 signal conditioning carrier**

(Adopted from http://www.transducertechniques.com/I/O DAQ-SCC-2345.cfm, May 11th 2008)

The enclosures for SCC signal conditioning modules connect directly to 68-pin I/O DAQ devices. In this project, the used SCC modules include:

- NI SCC-A10 Voltage Attenuator
- NI SCC-FT01 Feedthrough

NI SCC-A10 Voltage Attenuator and NI SCC-FT01 Feedthrough are used as the analog input (AI) and analog output (AO) respectively shown in Figure 8.3.

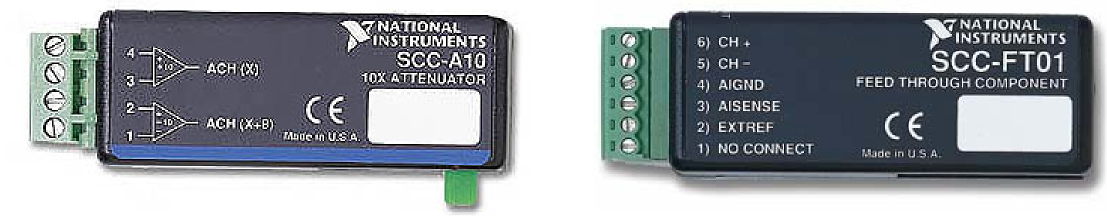


Figure 8.3: **SCC analog Input module (left) and analog output module (right)**

(Adopted from http://sine.ni.com/nips/cds/view/p/lang/en/nid/1714, May 10th 2007; Adopted from http://sine.ni.com/nips/cds/view/p/lang/en/nid/1717, May 10th 2007)

The NI SCC-A10 is a dual-channel module that accepts input voltage sources up to 60VDC. Each channel of the NI SCC-A10 includes a 10:1 attenuation circuit and differential instrumentation amplifier with low-impedance outputs for maximum scanning rates by the multifunction I/O DAQ device. The NI SCC-FT01 is a feedthrough module that offers direct connection to analog input, analog output, digital I/O, and General

Purpose Counter (GPCTR) channel of the I/O DAQ device. The data sheet of the SCC is in Appendix D. (http://www.ni.com/pdf/products/us/4daqsc253-265_194-196.pdf)

8.2.4 I/O DAQ hardware

I/O DAQ hardware acts as the interface between the computer and the outside world. The fundamental function of the I/O DAQ hardware is to digitize incoming signals so that the computer can interpret them. NI offers PCI bus based I/O DAQ boards which can plug into any desktop computer. The I/O DAQ board used in this project is the NI PCI-6035E 200kS/s, 16-Bit, 16-Analog-Input Multifunction I/O DAQ shown in Figure 8.4. This multifunction data acquisition device provides full functionality. The data sheet of this I/O DAQ board is in Appendix E.

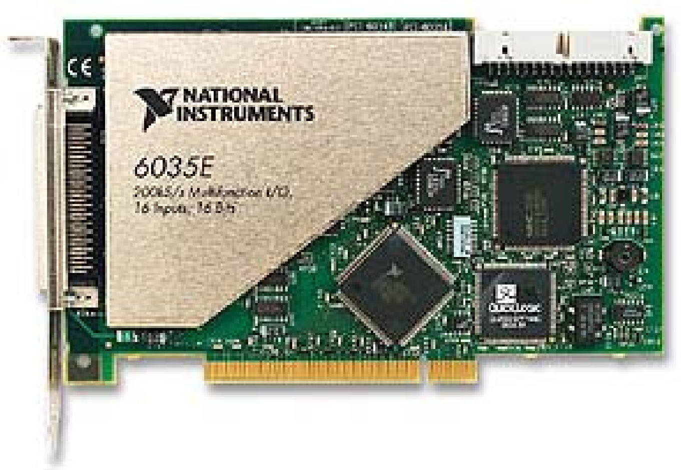


Figure 8.4: **NI PCI-6035E I/O DAQ board**

(Adopted from http://sine.ni.com/nips/cds/view/p/lang/en/nid/11915, May 6th 2008)

The I/O DAQ board and the Signal conditioning carrier are connected through a cable shown in Figure 8.5.

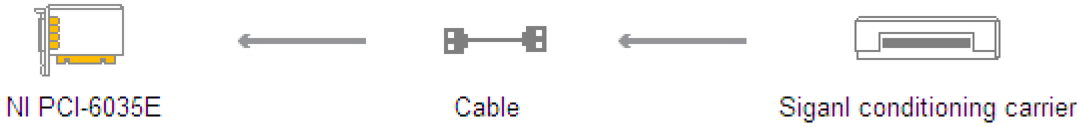


Figure 8.5: **Connection between the I/O DAQ board and the signal conditioning carrier**

(Adopted from http://sine.ni.com/nips/cds/view/p/lang/en/nid/11915, May 10th 2008)

8.2.5 Driver and application software

The software components include the driver software and the application software. Driver software is the layer of software for easily communicating with the hardware. It forms the middle layer between the application software and the hardware. The application software can be either a development environment or a configuration-based program. Application software adds analysis and presentation capabilities to driver software. The used I/O DAQ system in the project is compatible with both Matlab/Simulink and LabVIEW software environment.

Matlab is the short for “matrix laboratory”. It is a numerical computing environment and programming language. Simulink is a tool for modelling, simulating and analyzing multi-domain dynamic systems. Matlab and Simulink are integrated tightly and they have been widely used in control theory and digital signal processing for simulation and design. As shown in Chapter 5, Chapter 6 and Chapter 7, the Matlab and Simulink have been used for the system simulation. In this chapter, the Real-Time implementation is done by using Matlab/Simulink software and NI I/O DAQ system as the above mentioned I/O DAQ system is compatible with Matlab/Simulink software environments (http://www.mathworks.de/products/daq/supportedio14005.html, April 10th 2008).

LabVIEW is short for “Laboratory Virtual Instrumentation Engineering Workbench). It is a platform and development environment for a visual programming language developed by National Instruments. Above mentioned I/O DAQ system is the product of NI, it is compatible with LabVIEW (http://sine.ni.com/nips/cds/view/p/lang/en/nid/1191, April 12th 2008)

8.3 NI CompactRIO Real-Time implementation platform

The final Real-Time implementation platform is executed on NI CompactRIO control and acquisition system in order to realize the reconfigurability of the reconfigurable controller. The National Instruments CompactRIO programmable automation controller (PAC) is an advanced embedded control and data acquisition system designed for applications that require high performance and reliability. The CompactRIO has the characteristics of: system’s open, embedded architecture, small size, extreme ruggedness, and flexibility. It enables the designers to quickly build the customized embedded systems. In the word “CompactRIO”, RIO means Reconfigurable Input/Output technology. The RIO technology enables engineers to build the customized hardware circuitry using reconfigurable FPGA chips and LabVIEW graphical development tools. It is ideal for building optimized and flexible electric circuitry for I/O Communication and Control (IOCC) applications without actually building custom circuitry. The CompactRIO provides the benefit of modular FPGA-timed I/O with built-in signal conditioning and direct signal connectivity.

8.3.1 CompactRIO architecture overview

From the architecture point of view, CompactRIO combines an embedded Real-Time processor, a high-performance FPGA and hot-swappable I/O modules. Each I/O module is connected directly to the FPGA, providing low-level customization of timing and I/O signal processing. The FPGA is connected to the embedded Real-Time processor via a high-speed PCI bus. This represents a low-cost architecture with open access to low-level hardware resources. LabVIEW contains built-in data transfer mechanisms to pass data from the I/O modules to the FPGA and also from the FPGA to the embedded processor for Real-Time analysis, post-processing, data logging, or communication to a networked host computer shown in Figure 8.6. (http://www.ni.com/compactrio/whatis.htm, July 29th 2008)

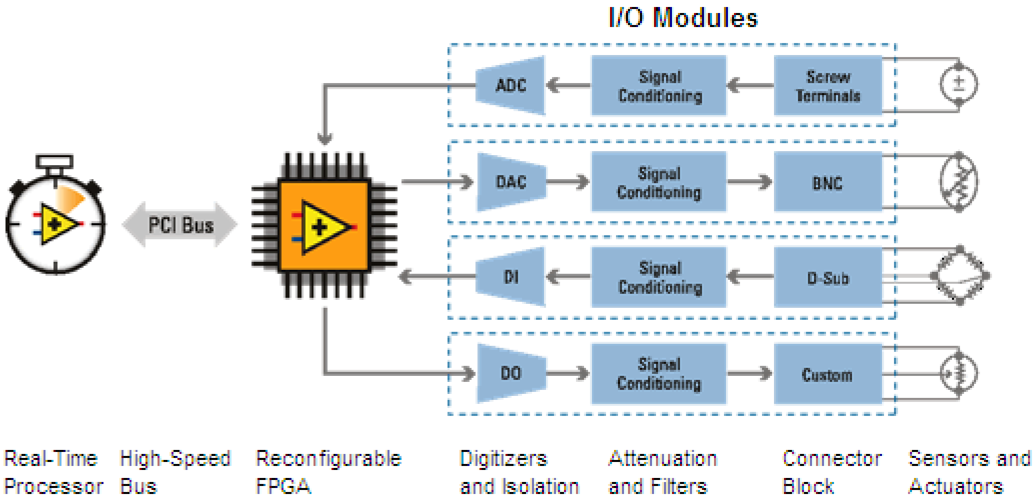


Figure 8.6: NI CompactRIO Control and Acquisition System

(Adopted from http://www.ni.com/compactrio/whatis.htm, 22nd April 2008)

The CompactRIO PAC combines the ruggedness and reliability of Programmable Logic Controllers (PLCs) with the software capabilities and the flexibility of PCs. PACs provide a new level of performance and flexibility for industrial control and acquisition. CompactRIO is an open embedded system that gives engineers a unique, low-cost architecture containing a reconfigurable FPGA for complete control over low-level hardware and software resources. From Figure 8.6, it can be seen that the CompactRIO consists of three basic components, the I/O Modules, FPGA and Real-Time Processor. These three components are introduced briefly in the following subsections.

8.3.1.1 CompactRIO I/O modules

The I/O modules: A variety of I/O types are available including voltage, current, thermocouple, RTD, accelerometer, and strain gauge inputs; up to ±60V simultaneous-sampling analog I/O; 12, 24, and 48 V industrial digital I/O; 5 V/TTL digital I/O; counter/timers; pulse generation; and high voltage/current relays. There are built-in signal conditioning contained in the modules, the wires can be connected directly from the C series modules to the sensors and actuators.

8.3.1.2 CompactRIO FPGA

The FPGA: FPGA is abbreviation of Field Programmable Gate Array. It is a semiconductor device containing Configurable Logic Block (CLB), Interconnect Resource (IR) and Input/Output Block (IOB) as shown in Figure 8.7. The Configurable Logic Block can be programmed to duplicate the functionality of basic logic gates (such as AND, OR, XOR, NOT) or more complex combinatorial functions such as decoders or simple math functions. In most FPGAs, these Configurable Logic Blocks also include memory elements, which may be simple flip-flops or more complete blocks of memories. FPGA has several advantages, such as a shorter time to market and ability to re-program in the field to fix bugs, and lower non-recurring engineering costs. The FPGA makers may offer less flexible versions of their FPGAs that are cheaper. The development of these designs is made on regular FPGAs and then migrated into a fixed version that more resembles an ASIC due to lack of ability to modify the design once it is committed. Considering from the technical point of view, FPGA offer large logic capacity, exceeding several million equivalent logic gates, and include dedicated memory resources, special hardware circuitry that is often needed in digital system and supports a wide range of interconnection standards. (Zhang L. K, 2005)

For the CompactRIO, the embedded FPGA is a high-performance, reconfigurable chip, which the engineers can program with LabVIEW and LabVIEW FPGA module graphical development tools.

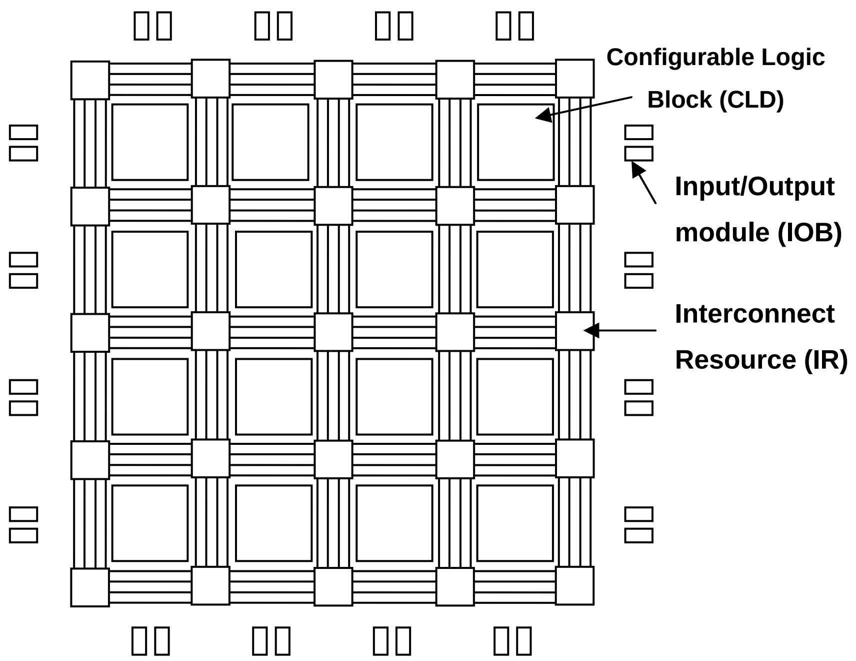

Configurable Logic Block (CLD)
Input/Output module (IOB)
Interconnect Resource (IR)

Figure 8.7: **The internal structures of the FPGA**

8.3.1.3 NI CompactRIO Real-Time processor

The CompactRIO embedded system features an industrial 400MHz Freescale MPC5200 Real-Time processor that deterministically executes the LabVIEW Real-Time applications on the reliable Wind River VxWorks Real-Time operating system. LabVIEW has built-in functions for transferring data between the FPGA and the Real-Time processor within the CompactRIO embedded system. It is possible to choose from more than 600 built-in LabVIEW functions to build the multithreaded embedded system for Real-Time control, analysis, data logging, and communication.

8.3.2 CompactRIO hardware deployment for the Real-Time implementation of the pendulum system

The CompactRIO system used in this project, consists of four basic components, which are:

- Real-Time Controllers
- Reconfigurable Chassis
- I/O Modules
- Development Software

These four components are introduced below.

8.3.2.1 CompactRIO Real-Time controller

The CompactRIO Real-Time controller used in this project is cRIO-9004 as shown in Figure 8.8. It offers stand-alone embedded execution for deterministic LabVIEW Real-Time applications (http://sine.ni.com/nips/cds/view/p/lang/en/nid/14155). It is used for Real-Time analysis, control and data logging. It features a 200MHz industrial processor that reliably and deterministically executes the LabVIEW Real-Time applications, 512 MB of non-volatile CompactFlash storage, 64 MB DRAM memory, 10/100BaseT Ethernet port with embedded Web and file servers with remote-panel user interface and RS232 serial port for connection to peripherals. The Real-Time processor also features dual 11 to 30 VDC supply inputs, a user DIP switch, LED status indicators, a Real-Time clock, watchdog timers, and other high-reliability features.

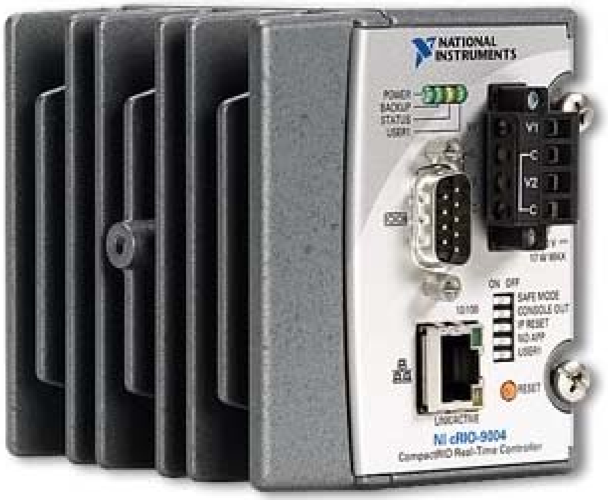


Figure 8.8: CompactRIO Real-Time controller

(Adopted from http://sine.ni.com/nips/cds/view/p/lang/en/nid/14155, April 10th 2008)

8.3.2.2 Reconfigurable Embedded Chassis

The reconfigurable embedded chassis are the heart of the CompactRIO system shown in Figure 8.9. The reconfigurable I/O (RIO) FPGA core is contained in it. This user-defined RIO FPGA is a custom hardware implementation of the control logic, input/output, timing, triggering and synchronization design. The RIO core is connected to each I/O module in a star topology for direct access to each module for precise control and unlimited flexibility in timing, triggering, and synchronization. It is programmed using easy to use elemental I/O functions to read and write signal information from each module. The RIO core is connected to the Real-Time controller as well through a local PCI bus. (http://sine.ni.com/nips/cds/view/p/lang/en/nid/14154, April 10th 2008).

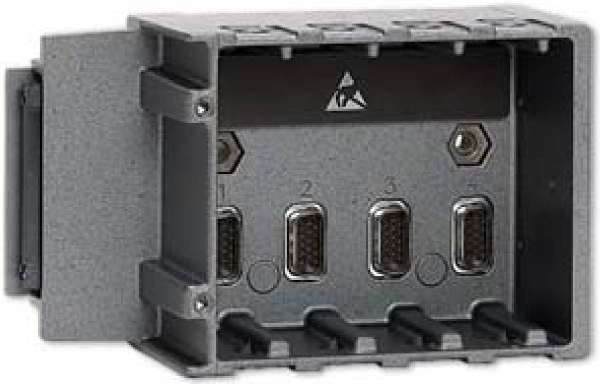


Figure 8.9: CompactRIO reconfigurable embedded chassis

(Adopted from http://sine.ni.com/nips/cds/view/p/lang/en/nid/14154, 10th April 2008)

8.3.2.3 CompactRIO I/O modules

CompactRIO I/O provides direct hardware access to the input/output circuitry of each I/O module using LabVIEW FPGA elemental I/O functions and it can also be directly connected to the sensors, actuators and communication buses. Each I/O module contains built-in signal conditioning and screw terminal. Different kinds of I/O types are available (http://sine.ni.com/nips/cds/view/p/lang/en/nid/14147). By integrating the connector junction box into the modules, the CompactRIO system significantly reduces the space requirements and cost of field wiring. In addition, by using the CompactRIO I/O Module Development Kit, engineers can create their own modules.
(ftp://ftp.ni.com/pub/devzone/pdf/tut_2856.pdf)

The I/O modules used in the project are the simultaneous-sampling analog inputs/outputs:

- NI 9201 8-Channel, 500kS/s, 12-Bit Analog Input Module
- NI 9263 4-Channel, 100kS/s, 16-bit, ±10V, Analog Output Module

as shown in Figure 8.10.

The detailed CompactRIO I/O wiring is shown in Appendix B.

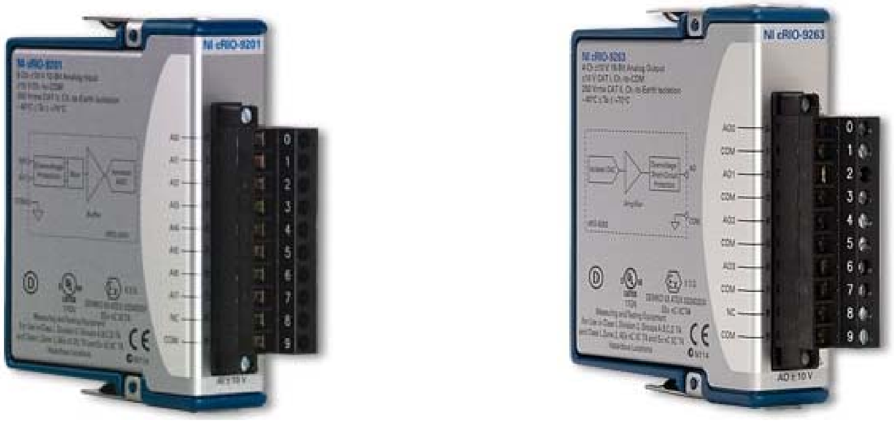


Figure 8.10: NI 9201 analog input (left) and NI 9263 analog output (right)

(Adopted from http://sine.ni.com/nips/cds/view/p/lang/en/nid/14588, 11th April 2008; Adopted from http://sine.ni.com/nips/cds/view/p/lang/en/nid/14170, 11th April 2008)

8.3.2.4 Development software

The following software is needed for building an embedded system:

- LabVIEW FPGA for specifying hardware I/O in the user-configurable FPGA core
- LabVIEW Real-Time for Building deterministic and reliable Real-Time applications

(http://sine.ni.com/nips/cds/view/p/lang/en/nid/14294)

In this project, the NI Developer Suite Core Package with the Real-Time Deployment and FPGA Deployment are used. It contains all the software that is needed.

8.4 Real-Time hardware and software configurations

Based on the learning of the Real-Time implementation hardware and software, the hardware can be configured by following the structures of the closed-loop block diagrams showed in Chapter 6 and Chapter 7. The hardware configuration basically forms the closed-loop control system in the physical hardware manner. In other words, all the components contained in the closed-loop system block diagram are replaced by the actual controller, the necessary to be controlled plant or system and the sensors involved in the feedback loops.

In this project, the plant to be controlled is the pendulum system, which can be configured into the inverted pendulum mode and the overhead crane mode. The controllers are different for different plant configurations. In this section, the hardware configurations for these two different platforms are presented respectively.

8.4.1 The I/O DAQ device with signal conditioning and PC Real-Time implementation platform hardware and software configurations

The I/O DAQ device and signal conditioning Real-Time implementation platform consists of the PC, the I/O DAQ card, the signal conditioning units and the reconfigurable plant. In this platform, the controllers are deployed in the PC via the Matlab Real-Time Windows Target. The Matlab Real-Time Windows Target is a PC solution for prototyping and testing Real-Time systems. In this environment, the PC with Matlab and Simulink is used as a host and target. The models are created in the PC by using the Simulink blocks diagrams. After creating a model and simulating it with Simulink in normal mode, the executable code can be generated with Real-Time Workshop and the Open Watcom C/C++ compiler. Then the application can be executed in Real-Time with Simulink external mode. One useful feature of the integration between Simulink external mode and the Real-Time Windows Target is that the parameters in the Simulink model can be changed while running an application in Real-Time. The designed Simulink controllers are interfaced with the plant by using the Real-Time windows target analog input/output functions. These Real-Time windows target analog input/output units acquire and send signals via I/O DAQ card and the signal conditioning modules from and to the reconfigurable plant sensors and the actuators respectively.

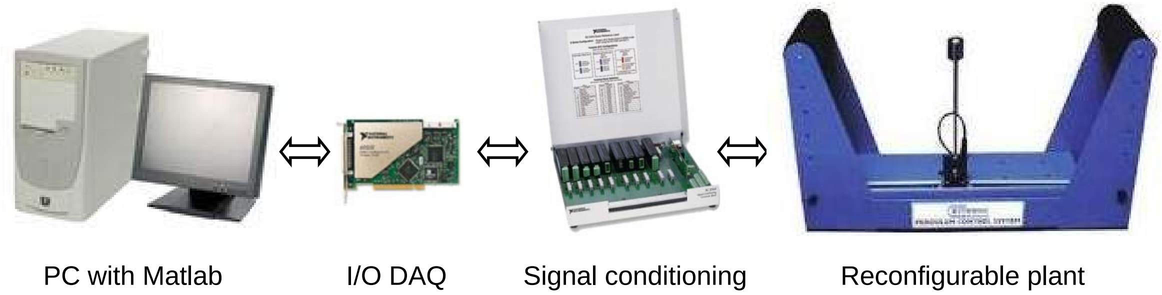


Figure 8.11: **The I/O DAQ device with signal conditioning and PC Real-Time implementation platform Hardware and software configurations**

In this project, Matlab/Simulink software environment is used. This software platform is regarded as one of the best compatible software platforms for both simulation and Real-Time implementation in automation control field.

This platform is mainly used to verify that the whether the designed control algorithms are correct. Once it is proved that the designed control algorithms are capable to control the reconfigurable plant, the hardware and the software implementation platform are shifted to the CompactRIO FPGA platform.

8.4.2 CompactRIO Real-Time implementation platform hardware and software configurations

In the CompactRIO Real-Time implementation platform three basic components are included, which are the NI CompactRIO PAC, the PC and the reconfigurable plant. Similar with the I/O DAQ with signal conditioning and PC platform, all software development is done in the host PC. The difference is that the software used in this platform is the graphical programming language LabVIEW.

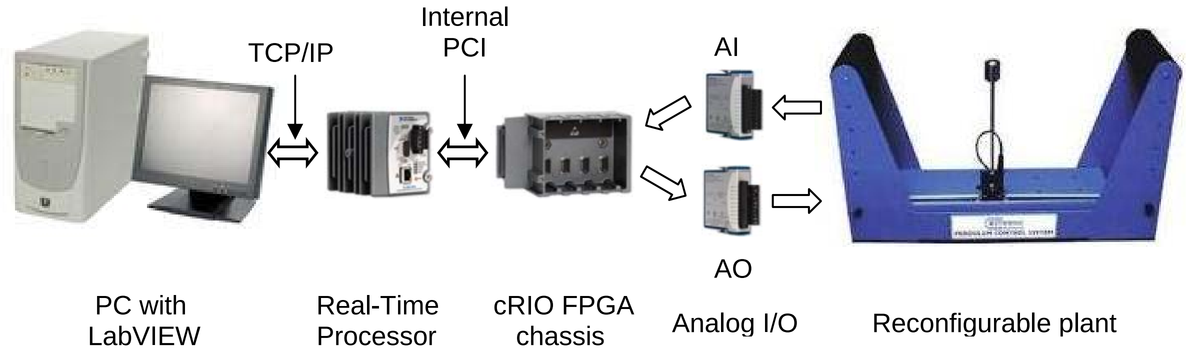


Figure 8.12: **The CompactRIO Real-Time implementation platform hardware and software configurations**

From Figure 8.11 and Figure 8.12, it can be seen that the CompactRIO Real-Time implementation platform has similar structure as the I/O DAQ device with signal conditioning and PC platform. As mentioned previously, the analog input and analog output modules are used in this project. The CompactRIO AI takes the measured signal from the sensors. According to the designed control algorithms in the cRIO FPGA, the control signal is generated and sent to the reconfigurable plant via the CompactRIO AO. But there are some essential differences between these two Real-Time implementation platforms. First of all, in the I/O DAQ device with signal conditioning platform, the PC is the host and target at the same time which means the control algorithms is developed in the PC and can only be deployed into the PC. As a result, the PC is running all the time during the control process. But for the CompactRIO Real-Time implementation platform, once the developed control algorithms are downloaded into cRIO FPGA, the CompactRIO can work as stand-alone or with a graphical user interface running on a network, the host PC. Furthermore, the CompactRIO system includes a built-in web browser to automatically publish the front panel user interface of the embedded application. CompactRIO does not have to work with the PC. This characteristic enables the CompactRIO Real-Time implementation platform to reduce the power consumption of the system and it occupies much less space. The second major difference between these two platforms is their architecture. For the I/O DAQ device with the signal conditioning and PC platform, the data process relies on the PC processor and the I/O DAQ device is directly inserted into the PC motherboard PCI slot. The control algorithm and the logging data are stored in the PC memory while the Application software is running. For the CompactRIO Real-Time implementation platform, the CompactRIO has its own processor and the data is processed by it. The control algorithm is deployed into the cRIO FPGA. From the architecture of these two platforms, it can be seen that the CompactRIO is better integrated. This reduces the system response time and enhances the system performance. Thirdly, because the CompactRIO is designed to work in the hardest industrial environment, it is reliable and suitable for the industrial applications. The I/O DAQ device with signal conditioning and PC Real-Time implementation is mainly for prototype application only.

8.5 Real-Time implementations and results

Based on the study of the I/O DAQ device with signal conditioning and PC Real-Time implementation platform and the CompactRIO Real-Time implementation platform hardware, software and the configurations, the Real-Time implementations are executed on these two platforms by using the developed control algorithms in the previous chapters, including: the Lyapunov stability based MRC and the feedback linearization method based control.

The developed Real-Time implementation software models are presented; The Real-Time implementation data results are acquired and plotted and the results are analysed and studied.

8.5.1 Real-Time implementation on the I/O DAQ device with signal conditioning and PC Real-Time implementation platform

For the I/O DAQ device with signal conditioning and PC Real-Time implementation platform, models are developed in the Matlab/Simulink software environment based on the control algorithms and Simulation block diagram developed in Chapter 6 and Chapter 7. The hardware configurations are as shown in Figure 8.11.

8.5.1.1 Lyapunov stability based MRC Real-Time implementation

8.5.1.1.1 Lyapunov stability based MRC Real-Time implementation Simulink block diagrams

Lyapunov stability based MRC Real-Time implementation Simulink block diagram is developed first. This Simulink block diagram can be obtained by using the corresponding simulation Simulink block diagrams presented in Chapter6 Figure 6.4, with some changes, such as, 1) add the Real-Time windows Target I/O units for data acquiring and sending; 2) remove the plant mathematical model and send the control signal to the actual plant; 3) add the circuit for converting the control signal from force F to voltage u given in Equation 4.112. For this project, according to the trainer manual (Bytronic international LTD, 2001: 5.14), the sampling time of the signal is equal to 0.055s.

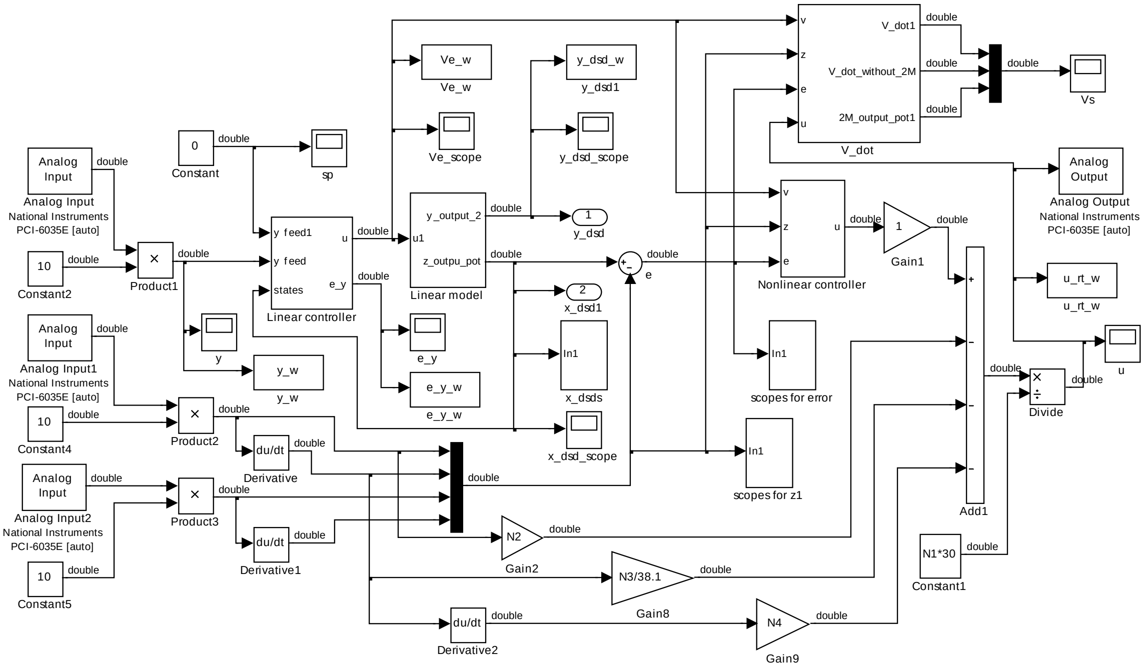


Figure 8.13: The Lyapunov stability based MRC Real-Time implementation Simulink block diagram

The Real-Time implementation Simulink block diagram is shown in Figure 8.13. The associated m-file for the Real-Time implementation only needs to provide the necessary parameters. The m-file called “*LDM_IP.m*” can be used. It is presented in Appendix A. 4. This m-file ensures the Real-Time implementation programme runs smoothly. It also provides functions to convert the data from “structure” to “double” type which enables the data to be plotted in MS-excel.

8.5.1.1.2 Lyapunov stability based MRC Real-Time implementation results

In the Lyapunov stability based MRC Real-Time implementation, the following signals including: the position signal x, the angle position signal θ, the velocity signal $\dot{x}$, the angular velocity signal $\dot{\theta}$, the output signal y, the control signal u, the error signal e and the signal of the derivative of the Lyapunov function $\dot{V}$, are taken for the inverted pendulum and the overhead crane control, respectively.

The initial conditions for all the Rea-Time implementation are specified in the following way: For the inverted pendulum, the initial cart position is at 0 m, it is held by hand in approximate vertical direction with the limit of angle position $\theta \le \pm 5^{\circ}$ away from it; for the overhead crane, the initial cart position is at 0 m as well, but the pendulum is placed at the furthest position away from the equilibrium point, approximate $\theta \approx 20^{\circ}$. Once the Real-Time implementation programming is started to run, the pendulum is released at the same time and the pendulum is controlled by the developed programme which is implemented by the associated hardware platforms. Two different set points for the output y are attempted, including 0 m and 0.1m.

Figure 8.14 to Figure 8.19 shows the Real-Time implementation results of Lyapunov stability based MRC control algorithm is applied to control the inverted pendulum at the set point of 0 m.

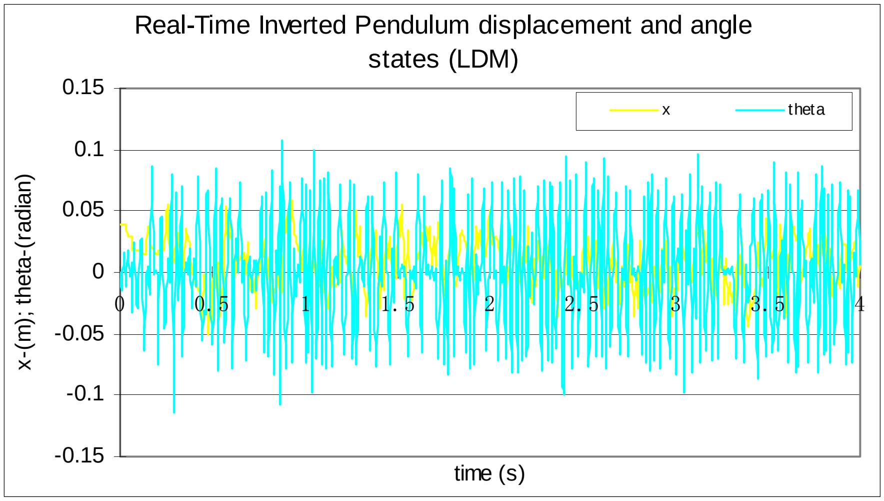


Figure 8.14: **Inverted Pendulum Real-Time displacement and angle signals when set point is 0 m**

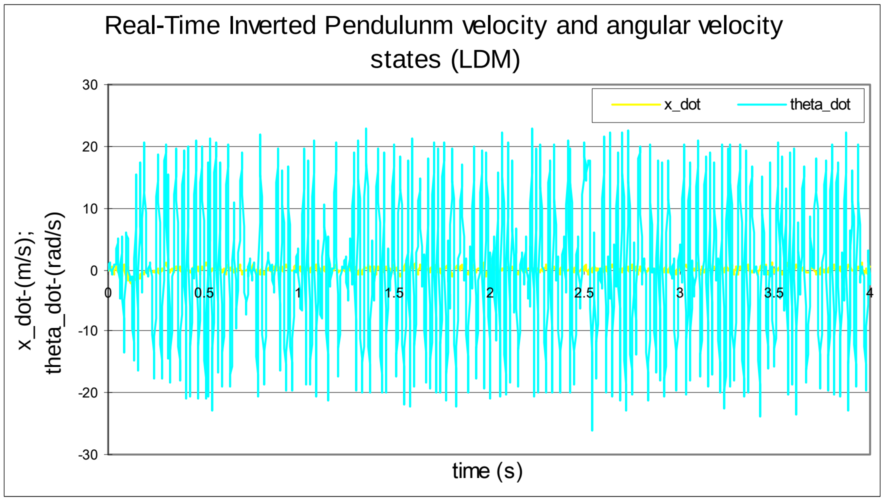


Figure 8.15: **Inverted pendulum Real-Time cart velocity and angular velocity signals when the set point is 0 m**

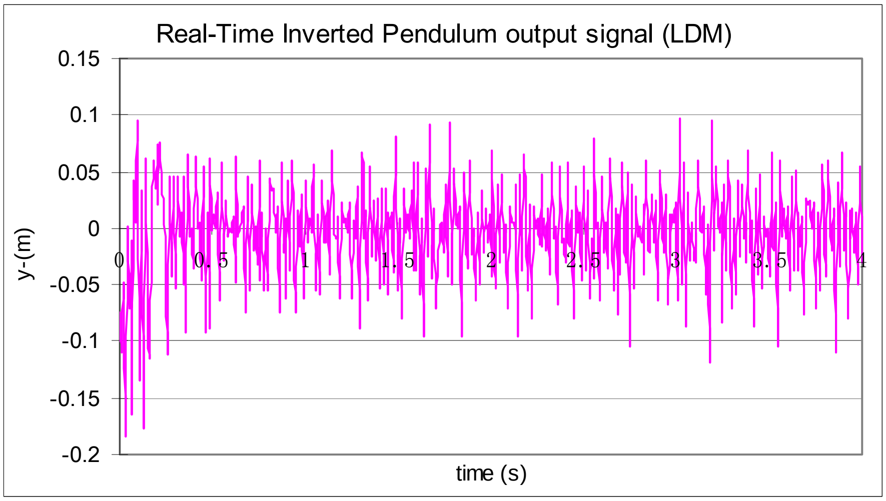


Figure 8.16: **Inverted pendulum Real-Time output signal when the set point is 0 m**

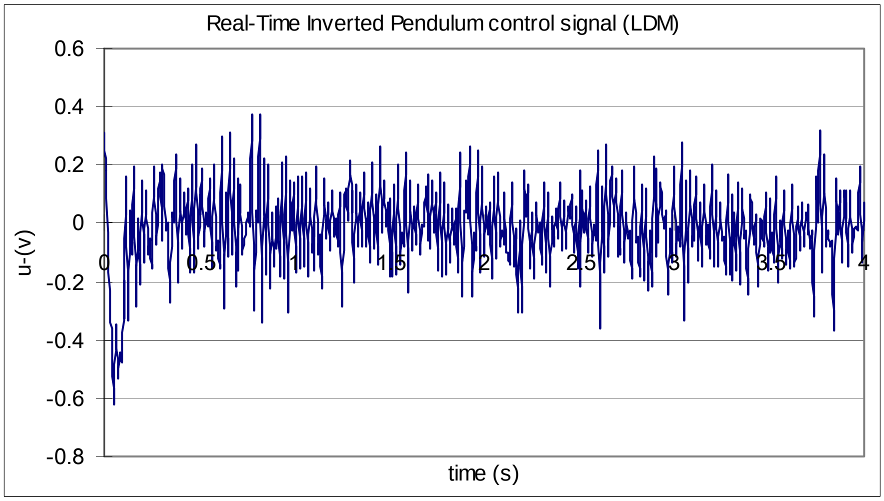


Figure 8.17: **Inverted pendulum Real-Time control signal when the set point is 0 m**

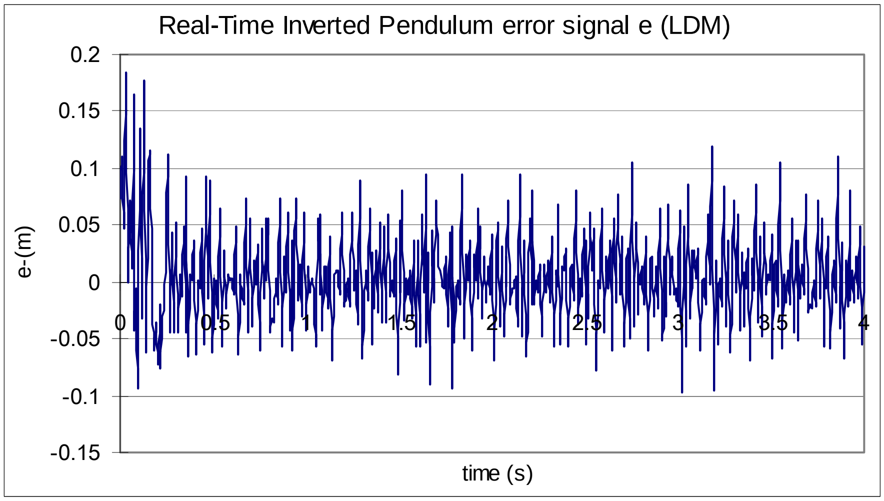


Figure 8.18: **Inverted pendulum Real-Time error signal when the set point is 0 m**

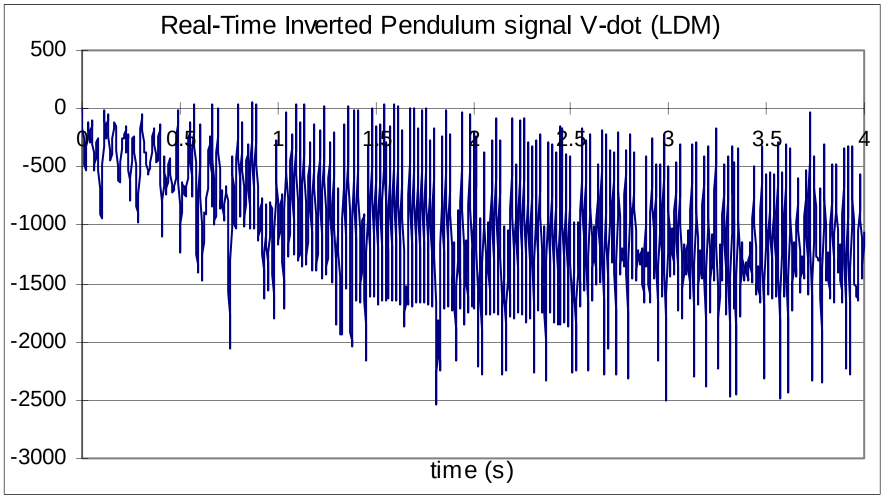


Figure8.19: **Inverted pendulum Real-Time signal of Lyapunov function derivative when the set point is 0 m**

From Figure 8.14, it can be seen that the position signal x is at around 0 m with oscillations which has amplitude of less than 5 cm. The angle signal θ is oscillates in a small range of less than ±7.5°. The output shows that the pendulum is at the position of about 0 m. The error signal turns to 0. In Figure 8.19, it can be concluded that the energy of the system is continuously dissipating. The system is bounded. From these graphs, it can be seen that all the states and output of the system are bounded and converge to zero. That means the inverted pendulum is well kept at the vertical position. The controller's designs are successful. Figure 8.20 to Figure 8.25 shows the results of Lyapunov stability based MRC are applied to control the inverted pendulum at the set point of 0.1 m.

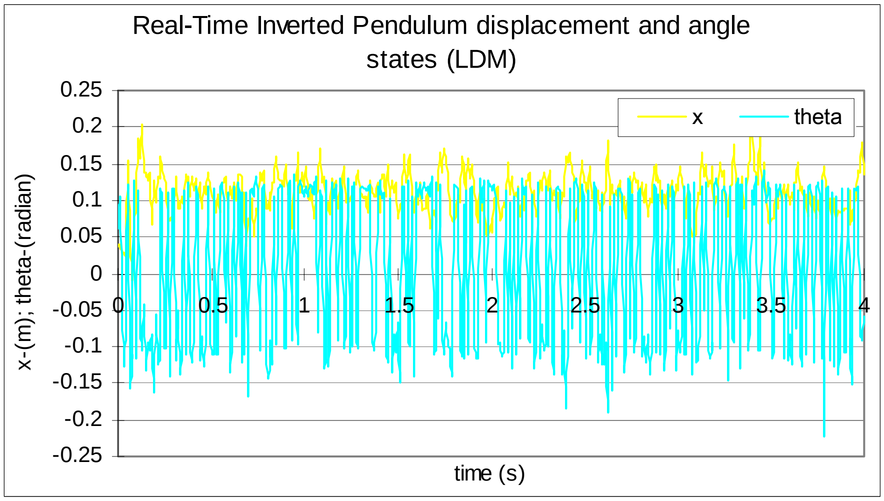


Figure 8.20: **Inverted Pendulum Real-Time displacement and angle signals when the set point is 0.1 m**

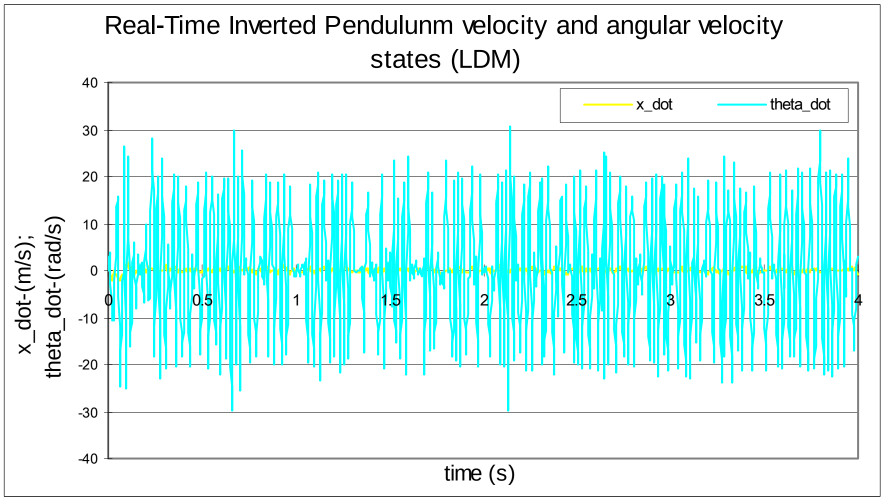


Figure 8.21: **Inverted pendulum Real-Time cart velocity and angular velocity signals when the set point is 0.1 m**

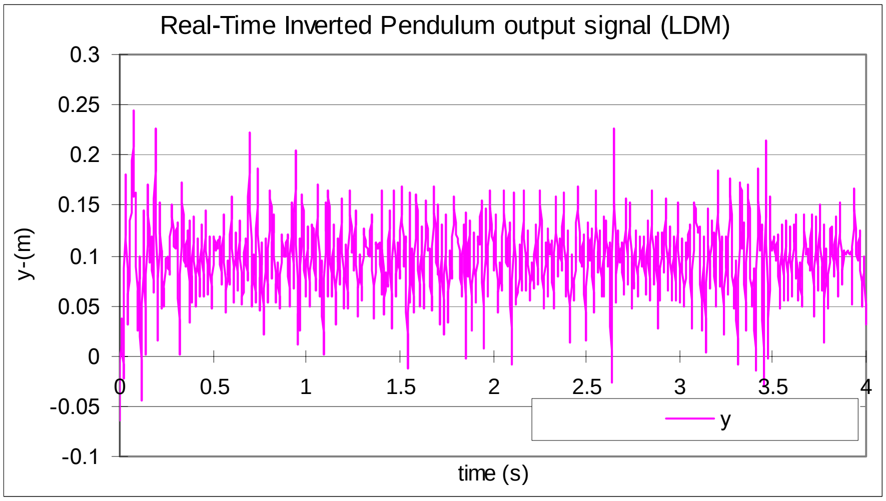


Figure 8.22: **Inverted pendulum Real-Time output signal when the set point is 0.1 m**

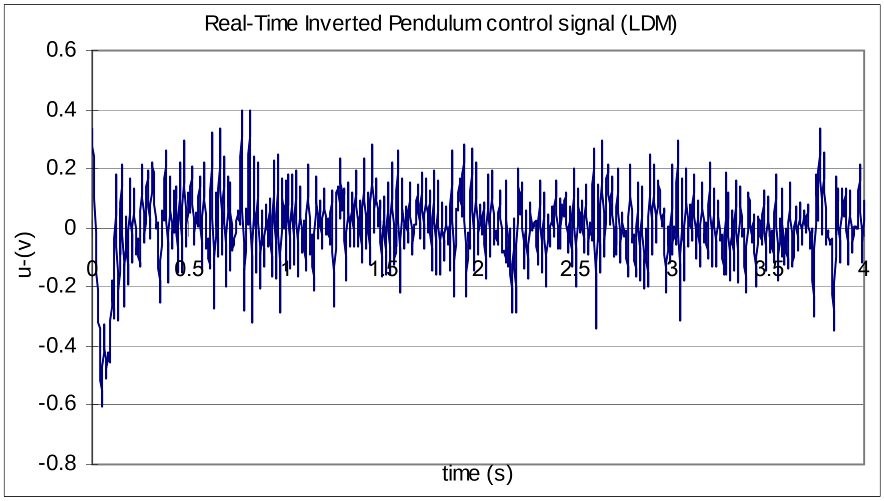


Figure 8.23: Inverted pendulum Real-Time control signal when the set point is 0.1 m

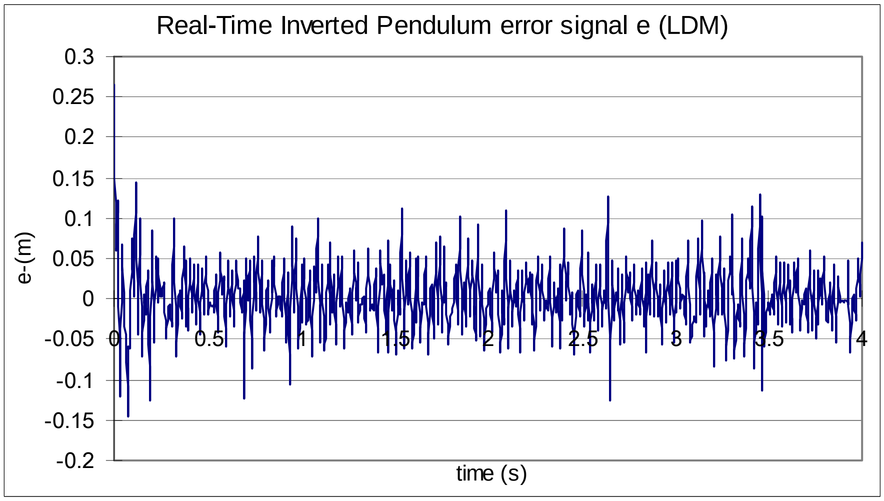


Figure 8.24: Inverted pendulum Real-Time error signal when the set point is 0.1 m

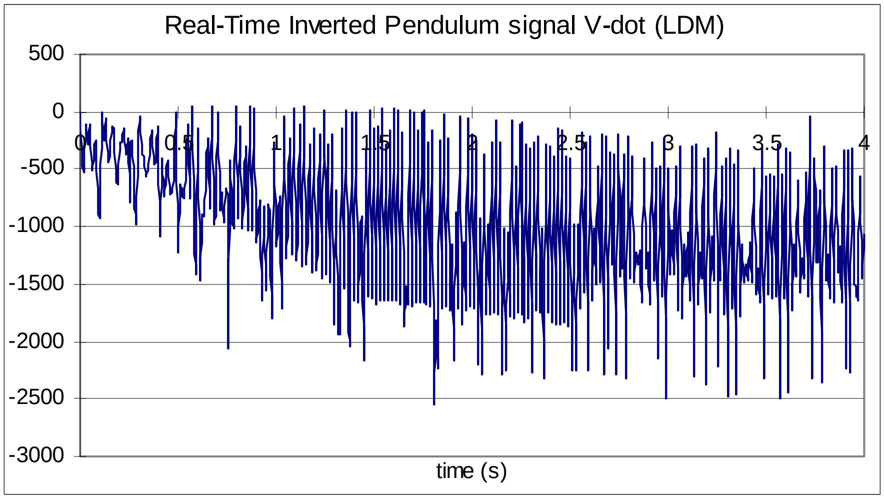


Figure 8.25: Inverted pendulum Real-Time signal of Lyapunov function derivative when the set point is 0.1 m

It can be seen that the position signal x has overshoot during the rising stage. But late converges to approximate 0.1 m and with small amplitude oscillations. The angular signal θ oscillates around zero radian. The output signal y goes to 0.1 m. The error signal e goes to 0 as well. The above mentioned result shows that the inverted pendulum is kept at the vertical direction with the displacement of 0.1 m away from the initial position. The control target is achieved. It is well know that the inverted pendulum is inherently unstable. From Figure 8.14 to Figure 8.25, it can be seen that the inverted pendulum are well controlled by the developed controller for different initial conditions and set points. The results from the Real-Time control of the crane are shown on Figure 8.26 till 8.31 for a set point for the output equal to 0 m and from Figure 3.32 till Figure 3.37 for the output set point of 0.1 m.

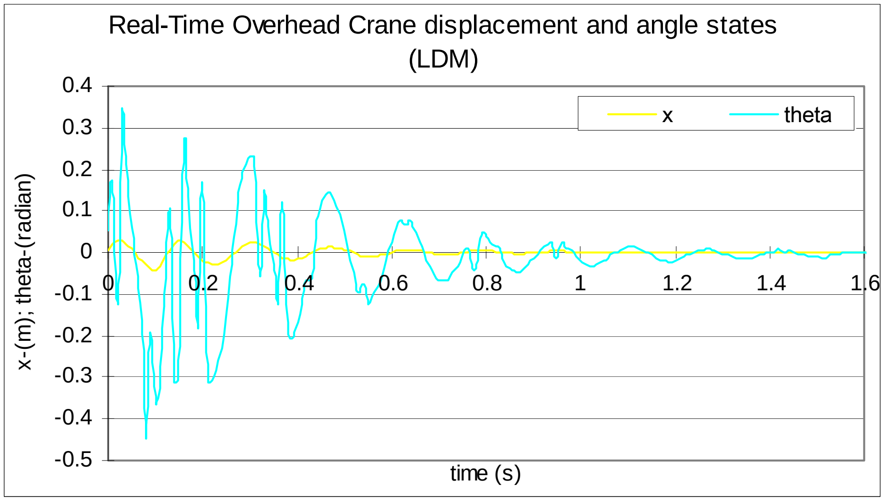


Figure 8.26: **Overhead Crane Real-Time displacement and angle signals when set point is 0 m**

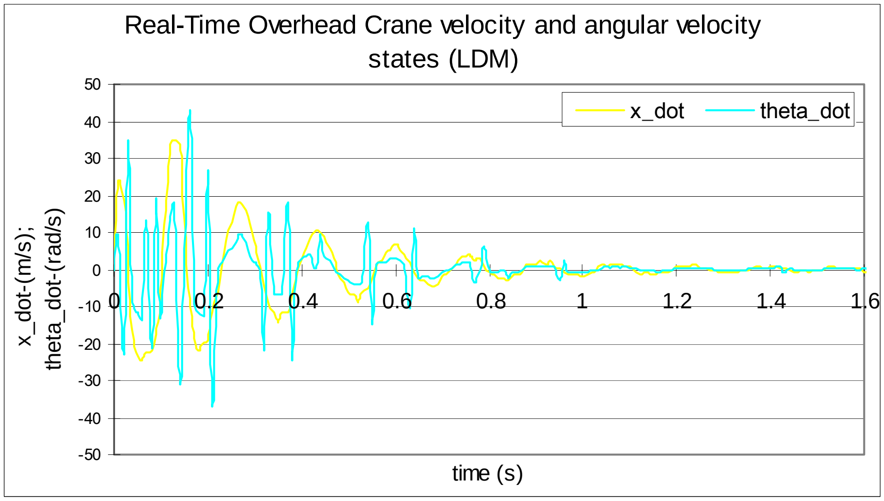


Figure 8.27: **Overhead Crane Real-Time velocity and angular velocity signals when the set point is 0 m**

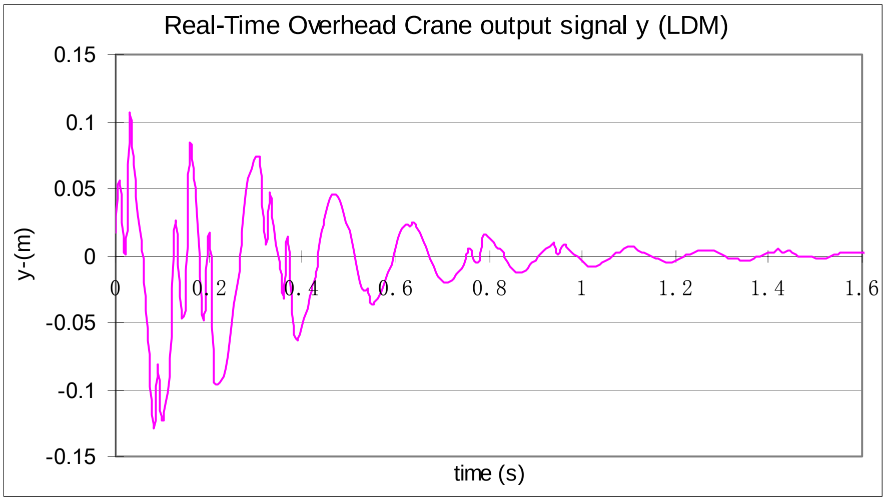


Figure 8.28: **Overhead Crane Real-Time output signal when the set point is 0 m**

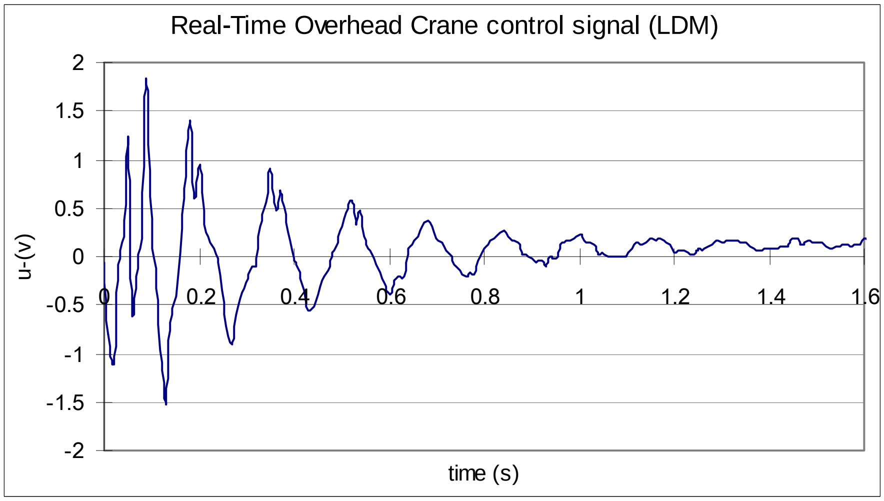


Figure 8.29: **Overhead Crane Real-Time control signal when the set point is 0 m**

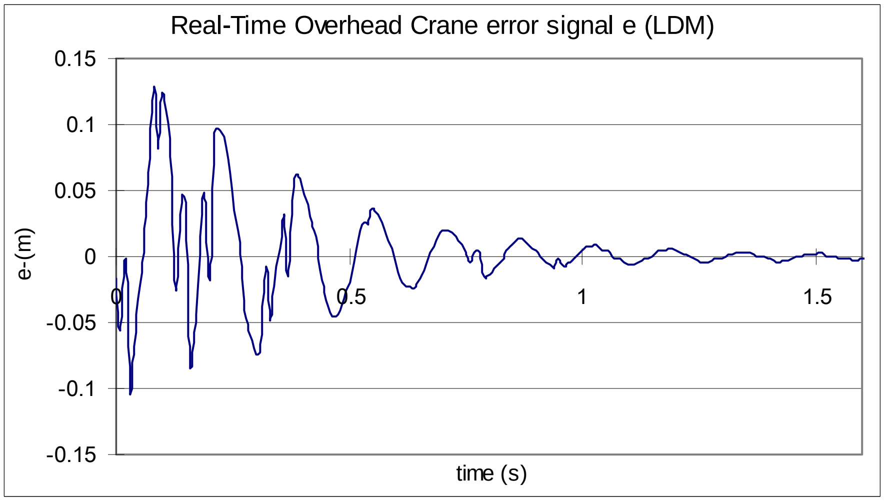


Figure 8.30: **Overhead Crane Real-Time error signal when the set point is 0 m**

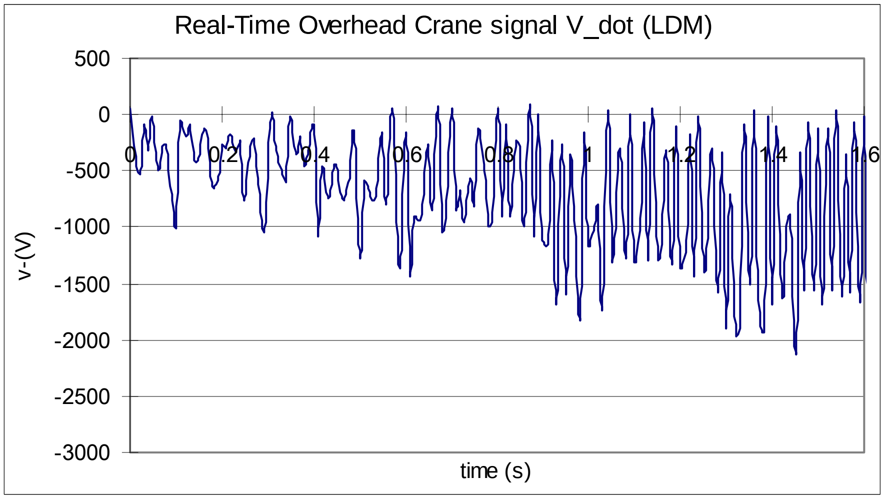


Figure 8.31: **Overhead Crane Real-Time signal of Lyapunov function derivative when the set point is 0 m**

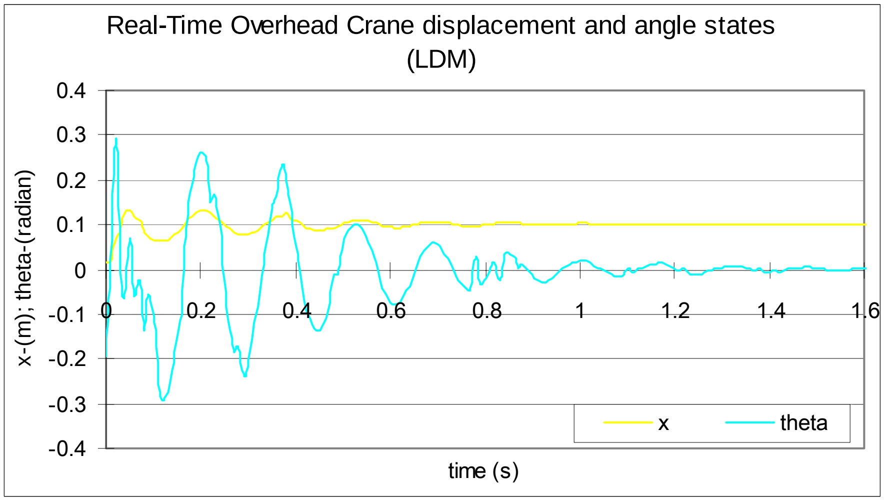


Figure 8.32.: **Overhead Crane Real-Time displacement and angle signals when set point is 0.1 m**

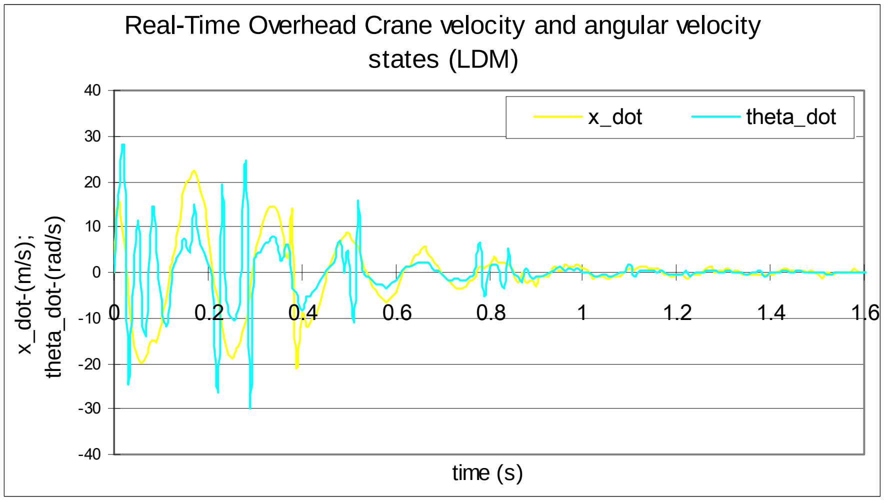


Figure 8.33: **Overhead Crane Real-Time cart velocity and angular velocity state signals when the set point is 0.1 m**

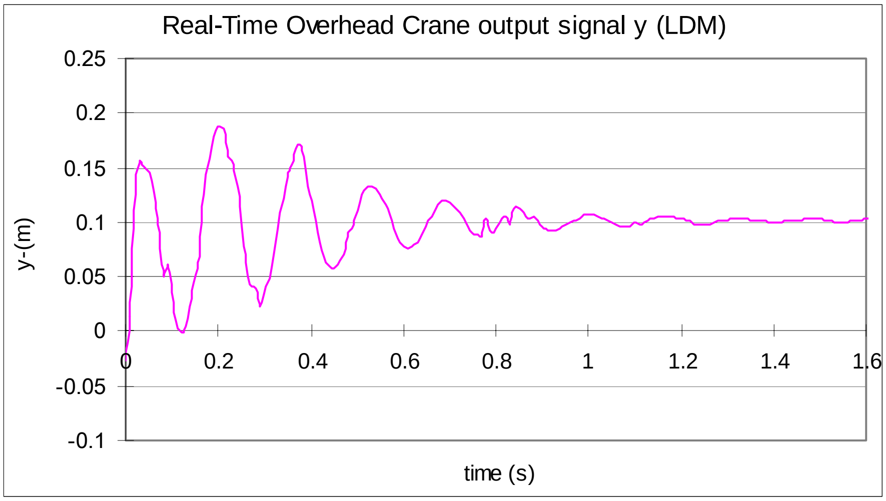


Figure 8.34: **Overhead Crane Real-Time output signal when the set point is 0.1 m**

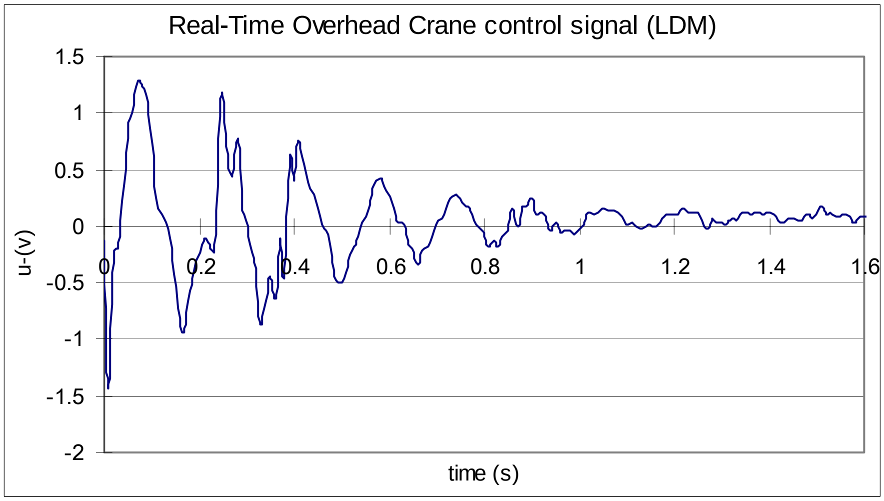


Figure 8.35: **Overhead Crane Real-Time control signal when the set point is 0.1 m**

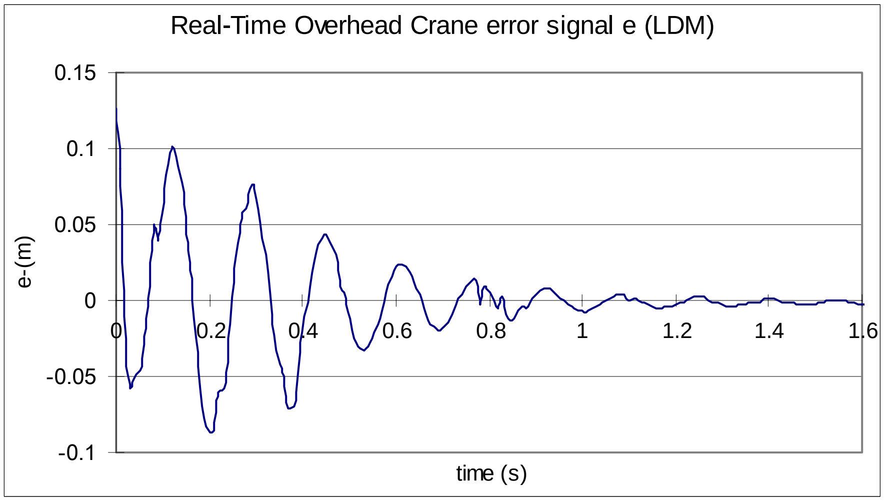


Figure 8.36: **Overhead Crane Real-Time error signal when the set point is 0.1 m**

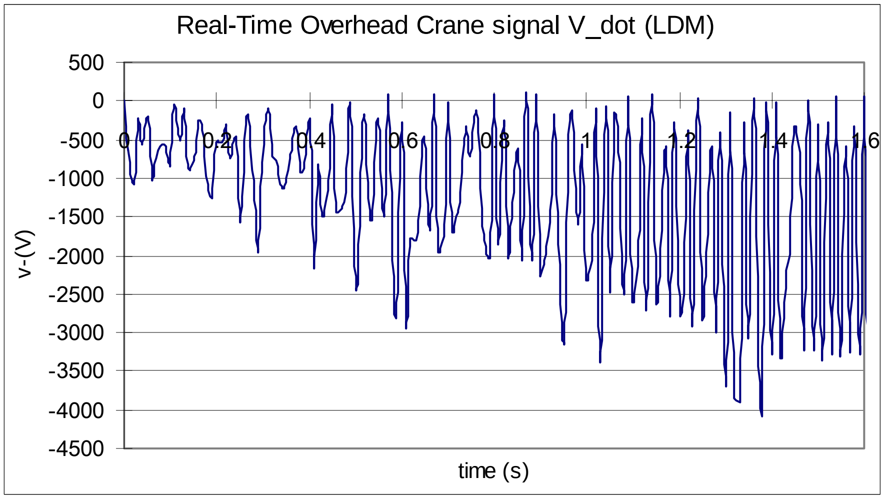


Figure 8.37: **Overhead Crane Real-Time signal of Lyapunov function derivative when the set point is 0.1 m**

For the overhead crane, the other associated m-file called "*LDM_OC.m*" is used which is presented in Appendix A. 5. It has the same functions as the previous one, but with the parameters for the overhead crane Real-Time implementation. The overhead crane is an inherently stable system, but it has the characteristic of oscillations. For this particular reconfigurable plant, the friction coefficient between the rod and the pivot is very small. If the pendulum is placed at the furthest position away from the equilibrium position, it takes about 30 seconds without control for the pendulum to settle down. But from Figure 8.26 and Figure 8.32 can be seen that under the help of the developed Lyapunov stability based MRC controller in this project, the oscillations of the overhead crane die down in a very fast way with respect to set point of 0 m and 0.1 m. From the output signals in Figure 8.28 and Figure 8.34, it can be seen the overhead crane settles down at the desired positions. The error signals converge to approximate 0 as shown in Figure 8.30 and Figure 8.36. From these graphs, it can be concluded that the oscillations of the system die very fast down, the developed controller works well and according to the specifications.

8.5.1.2 Feedback linearization method Real-Time implementation

8.5.1.2.1 Feedback linearization method Real-Time implementation Simulink block diagrams

The feedback linearization method based Real-Time implementation is done as well. In Figure 8.38, the corresponding Simulink block diagram is shown. It is derived from the corresponding simulation Simulink block diagram shown in Chapter 7, Figure 7.26 with some changes. The changes include: 1) add the Real-Time Windows Target units; 2) add the circuit to convert the signal from force F to voltage V given by Equation 4.112; 3) and remove the plant mathematical model and connect the control signal to the plant. The sampling time is 0.055s. It is specified in the Real-Time Windows Target Units. An associated m-file called "*FBL_IP.m*" is used to specify and generate the necessary parameters, including all the parameters used for the Real-Time control, the feedback control gain, the weight matrix and so on. This m-file is given in Appendix A. 7.

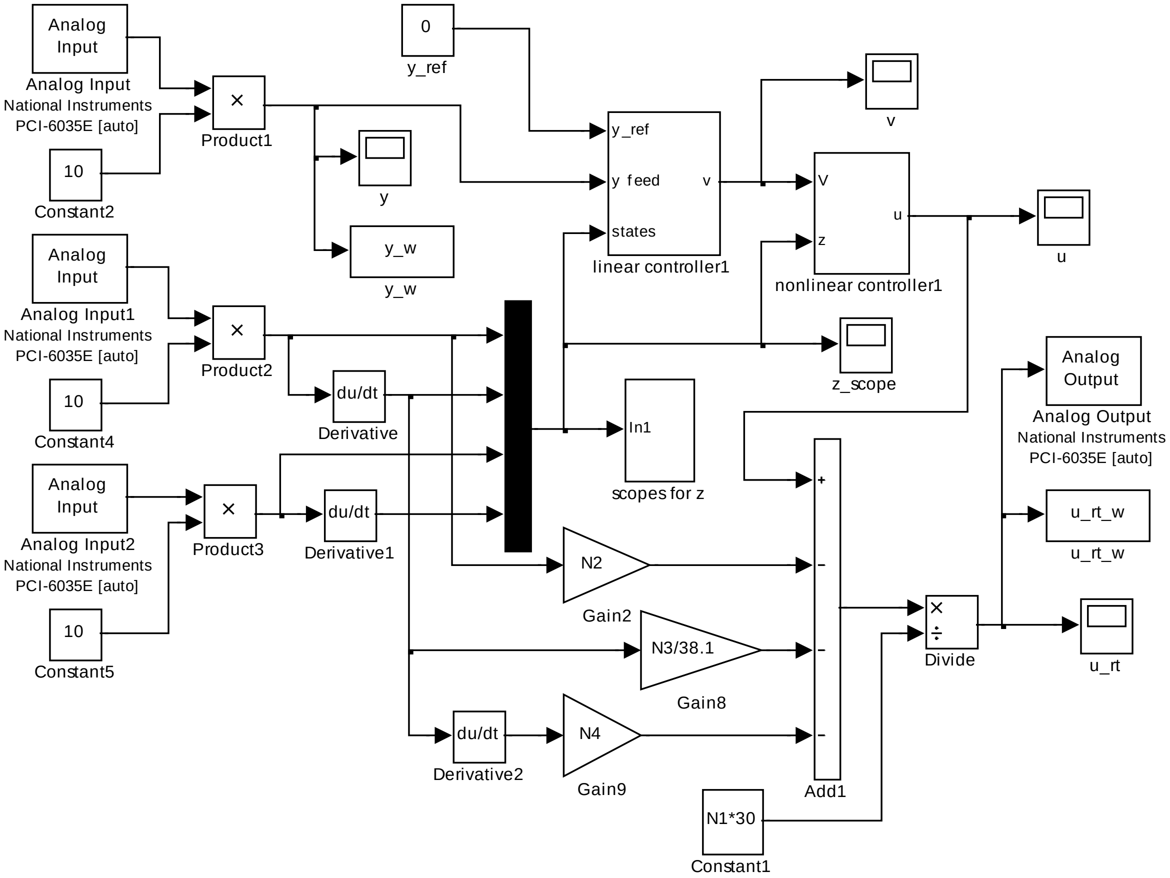


Figure 8.38: **The feedback linearization method Real-Time implementation Simulink block diagram**

8.5.1.2.2 Feedback linearization method Real-Time implementation results

For the feedback linearization method based Real-Time implementation, the following signals are taken, including: the position signal x, the angle position signal θ, the velocity signal $\dot{x}$, the angular velocity signal $\dot{\theta}$, the output signal y, the nonlinear control signal u, the linear control signal v and the error signal e. These signals are presented in the following graphs in this order for the inverted pendulum and the overhead crane respectively. Two different set points 0 m and 0.1 m are used. The initial conditions are mentioned in the previous section.

In Figure 8.39 to Figure 8.44, the relative signals of the inverted pendulum for set point at 0 m are presented. It can be seen that the pendulum position signal x is in the range of ±0.02 m. The angle signal θ is in the range of ±0.04 Radian which equal to approximate ±2.5°. The output y is also in a small range of between ±0.03 m. These values show that the inverted pendulum is kept in the vertical direction. The nonlinear control signal u, linear control signal v and the error signal e are present as well.

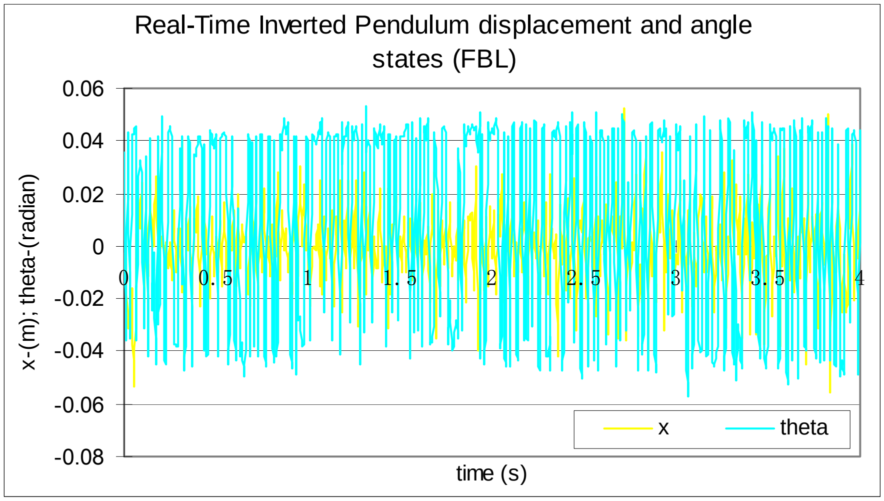


Figure 8.39: **Inverted Pendulum Real-Time displacement and angle signals when the set point is 0 m**

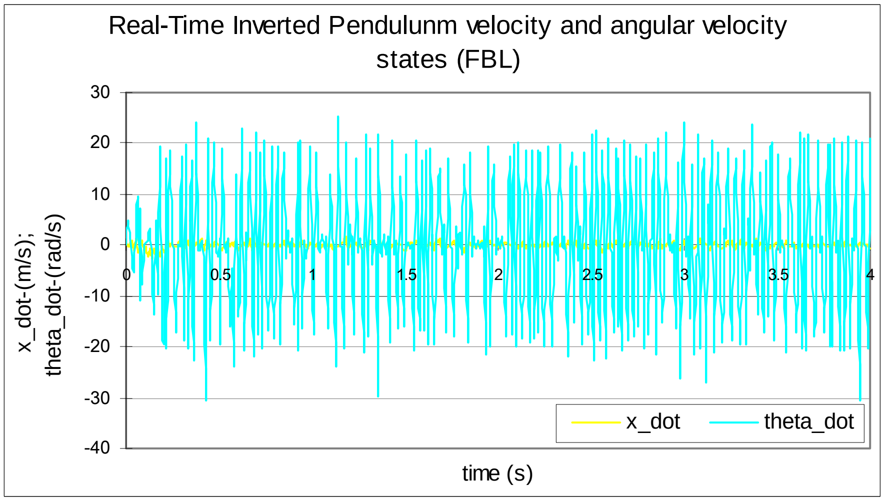


Figure8.40: **Inverted pendulum Real-Time cart velocity and angular velocity signals when the set point is 0 m**

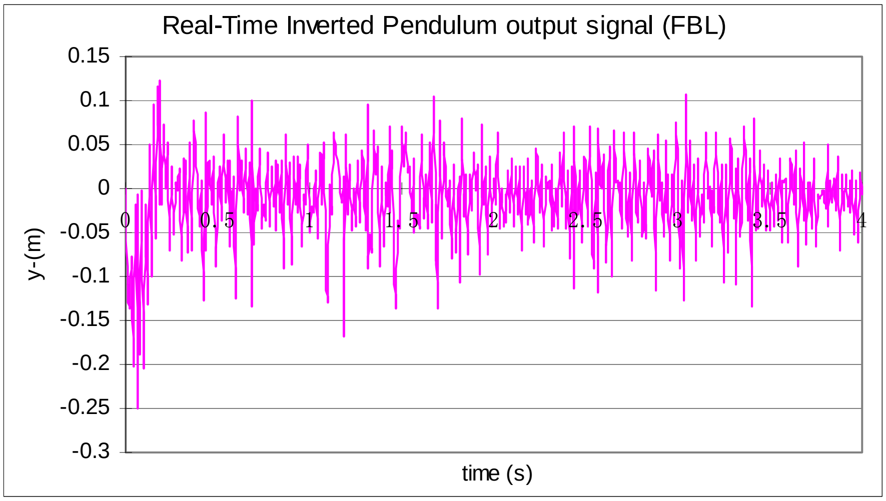


Figure 8.41: **Inverted pendulum Real-Time output signal when the set point is 0 m**

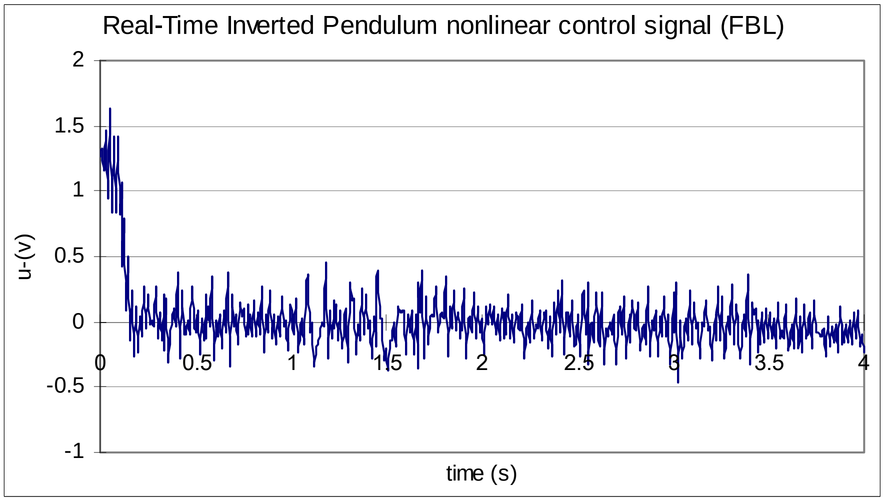


Figure 8.42: **Inverted pendulum Real-Time nonlinear control signal when the set point is 0 m**

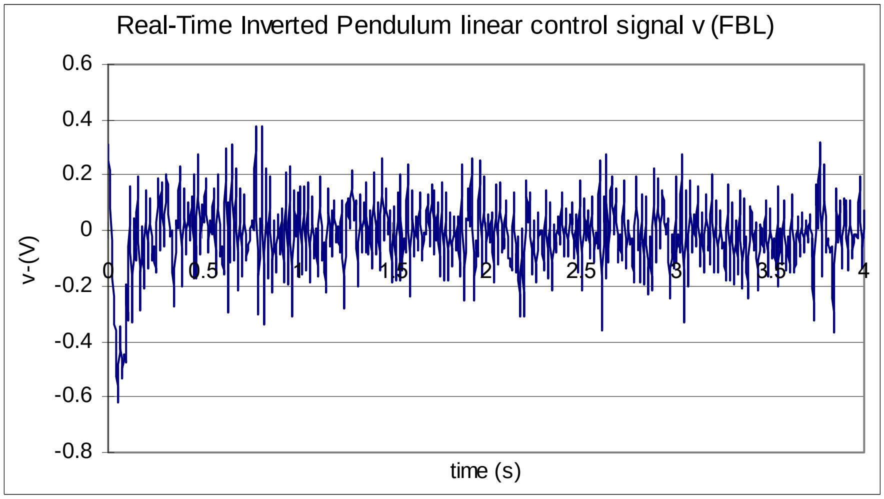


Figure 8.43: **Inverted pendulum Real-Time linear control signal when the set point is 0 m**

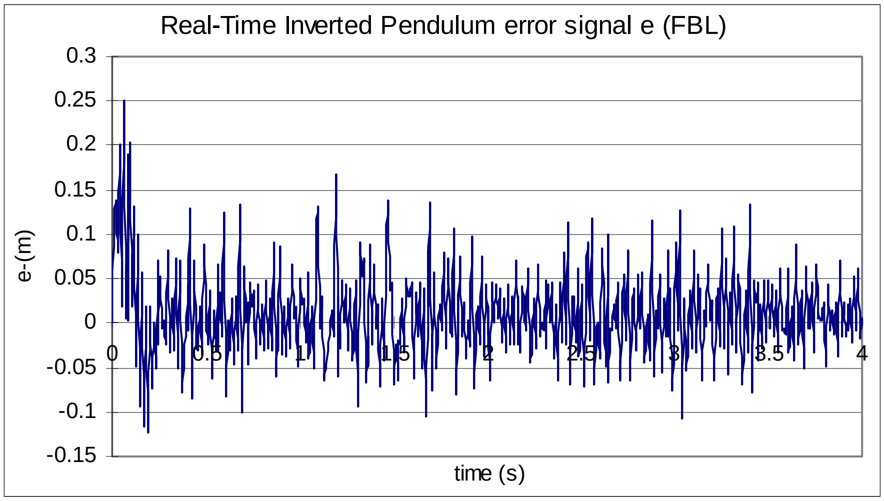


Figure 8.44: **Inverted pendulum Real-Time error signal when the set point is 0 m**

For the set point of 0.1 m, the Real-Time implementation results are presented in Figure 8.45 to Figure 8.50. Similarly, from these figures, it can be seen, the inverted pendulum is kept at the position of 0.1 m. There are overshoot at the rise stage. The cart position signal x is with oscillation of the amplitude of about ±0.03 m. The angle signal θ is limited in the range of ±0.05 Radian which is approximate 3°. The output signal y is in the range of between 0.08 m and 1.12 m. This Real-Time implementation results show that the inverted pendulum is well controlled. The developed control algorithms are suitable for controlling the inverted pendulum for different set point and initial conditions.

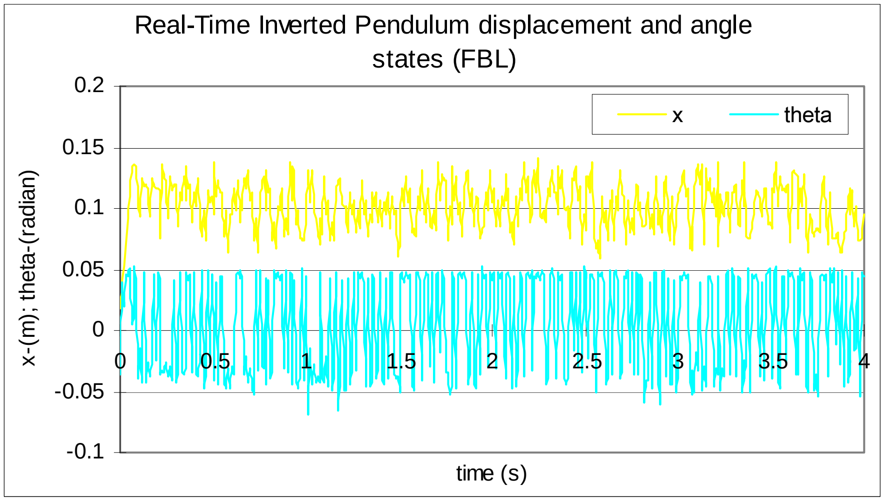


Figure 8.45: **Inverted Pendulum Real-Time displacement and angle signals when set point is 0.1 m**

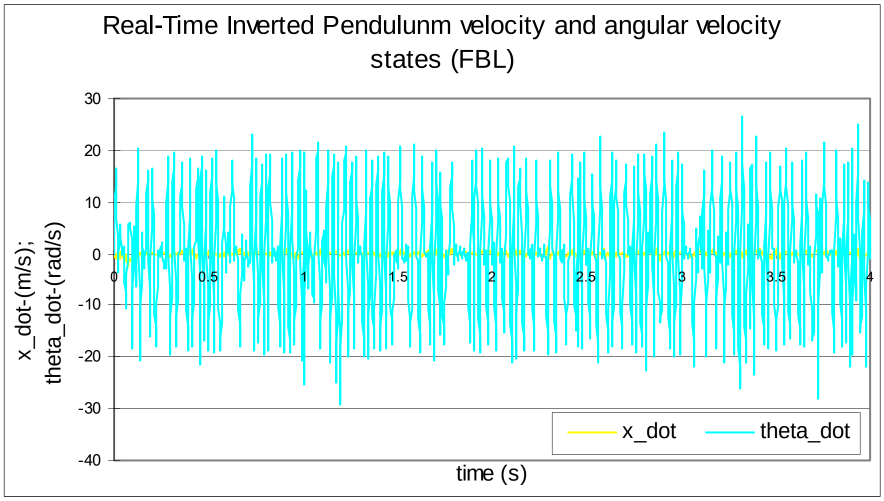


Figure 8.46: Inverted pendulum Real-Time velocity and angular velocity signals when the set point is 0.1 m

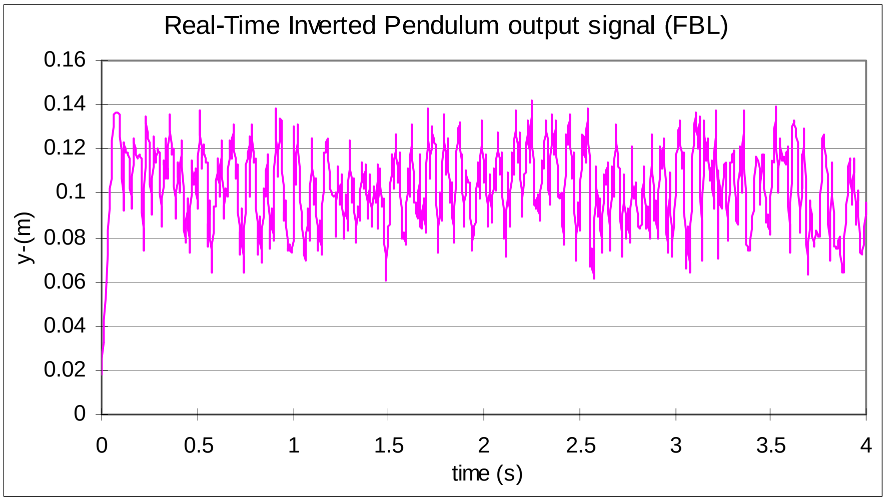


Figure 8.47: Inverted pendulum Real-Time output signal when the set point is 0.1 m

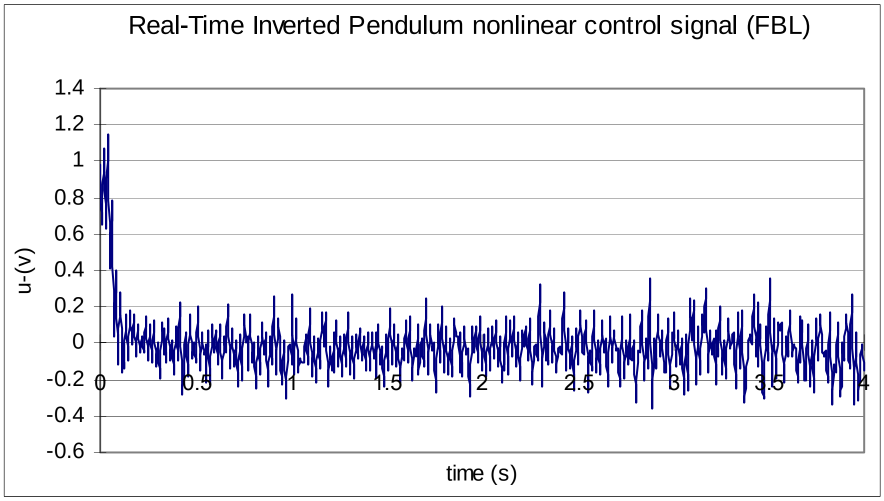


Figure 8.48: Inverted pendulum Real-Time nonlinear control signal when the set point is 0.1 m

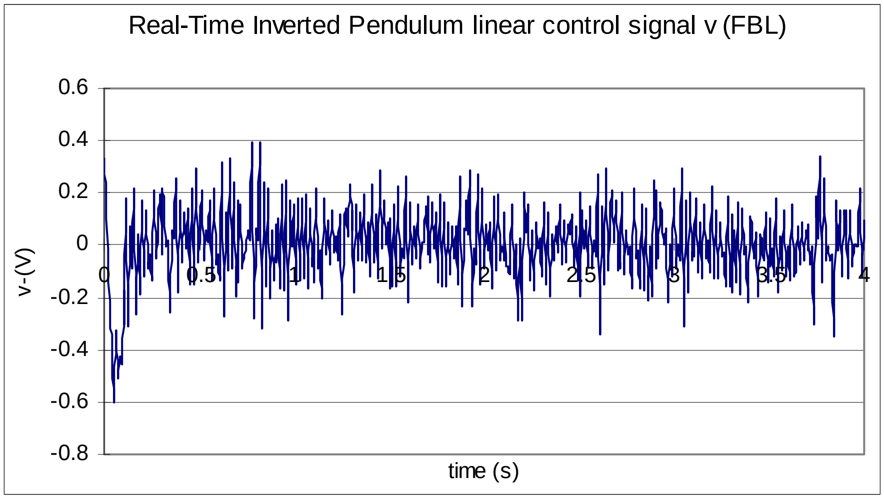


Figure 8.49: **Inverted pendulum Real-Time linear control signal when the set point is 0.1 m**

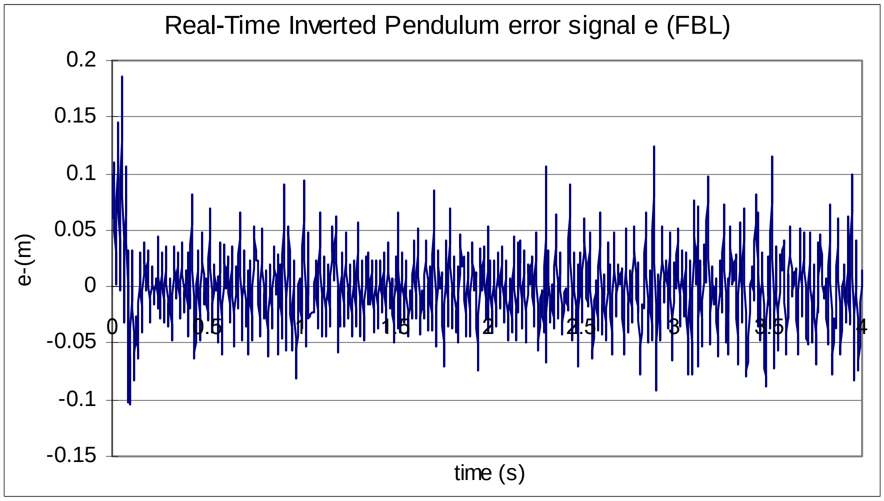


Figure 8.50: **Inverted pendulum Real-Time error signal when the set point is 0.1 m**

Between Figure 8.51 and Figure 8.62, the feedback linearization method is applied to control the overhead crane in Real-Time by using the I/O DAQ device with signal conditioning and PC for different output set points 0 m and 0.1m. The very similar Simulink block diagram is used as previous one used for controlling of the inverted pendulum as shown in Figure 8.38. The necessary change is done for the parameters in the linear control and the nonlinear control blocks. A new associated m-file called "*FBL_OC.m*" is used to generate the necessary parameters. It is give in Appendix A. 8. From the Figure 8.51 to 8.62, it can be seen that under the help of the developed controller the overhead crane goes to the equilibrium point in a very short period of time. For the different set points, the controller performs well. The control tasks are completed according to the specifications.

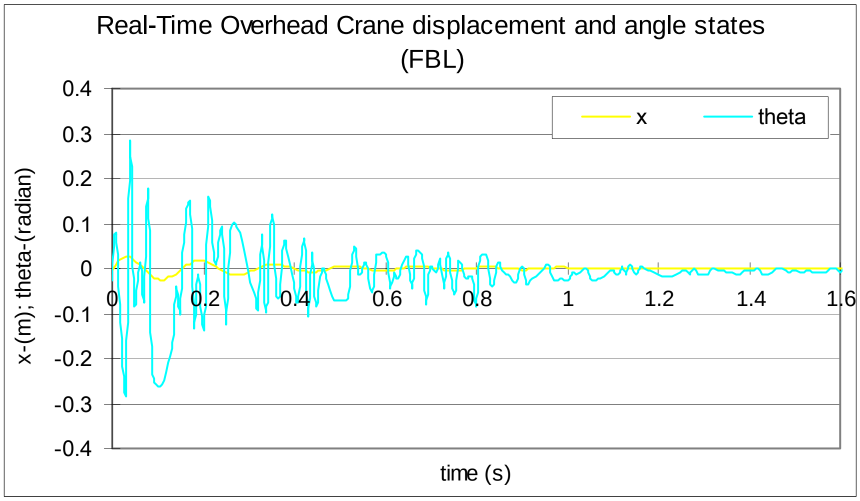


Figure 8.51: Overhead Crane Real-Time displacement and angle signals when set point is 0 m

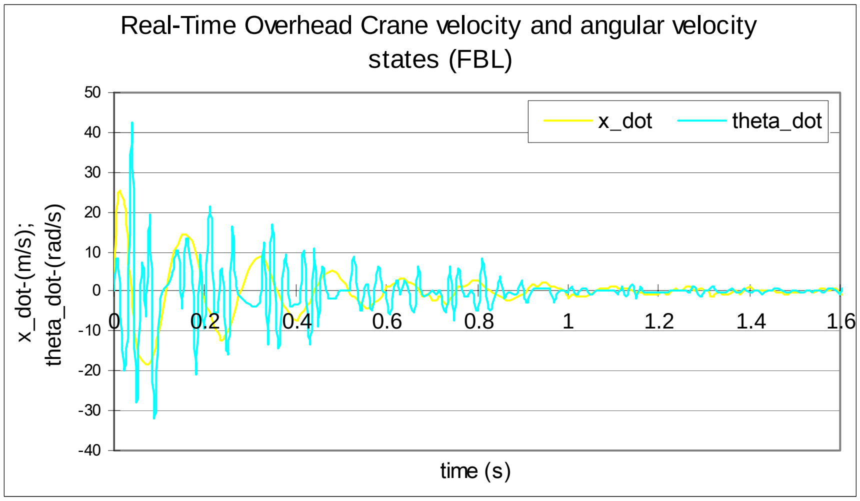


Figure 8.52: Overhead Crane Real-Time velocity and angular velocity signals when the set point is 0 m

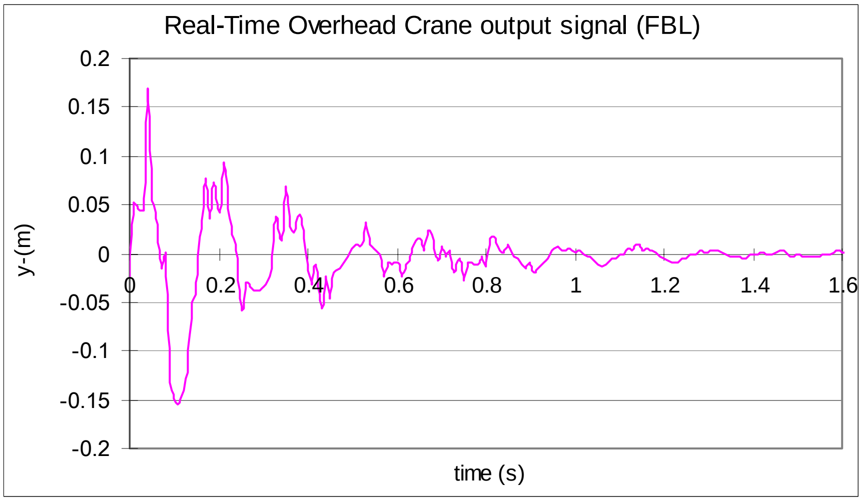


Figure 8.53: Overhead Crane Real-Time output signal when the set point is 0 m

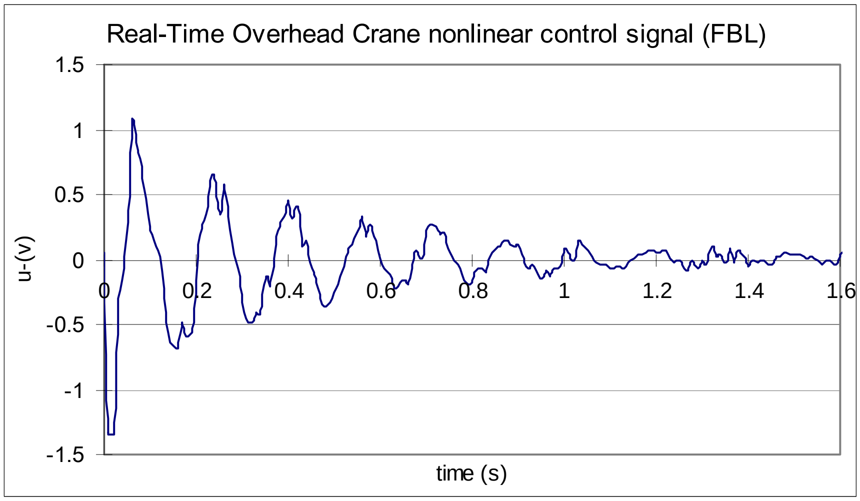


Figure 8.54: **Overhead Crane Real-Time nonlinear control signal when the set point is 0 m**

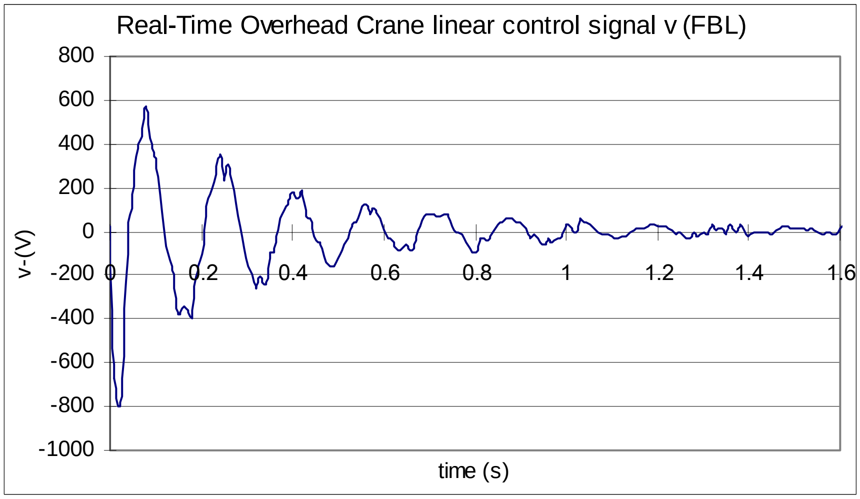


Figure 8.55: **Overhead Crane Real-Time linear control signal when the set point is 0 m**

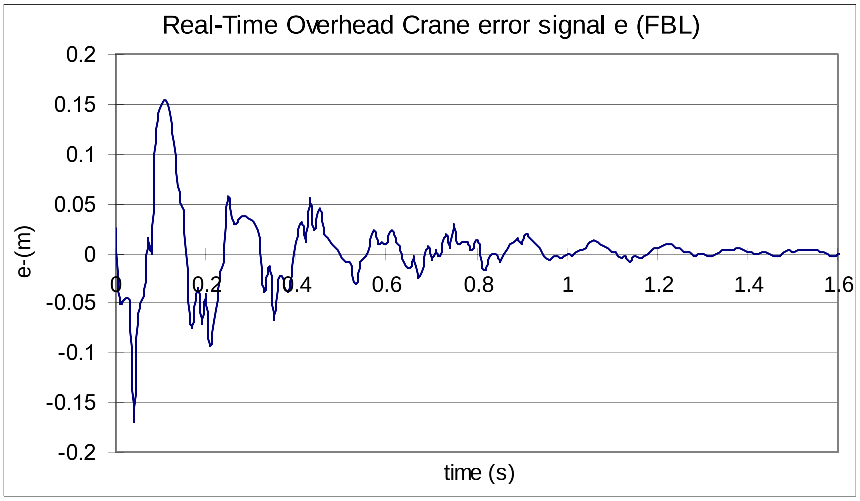


Figure 8.56: **Overhead Crane Real-Time error signal when the set point is 0 m**

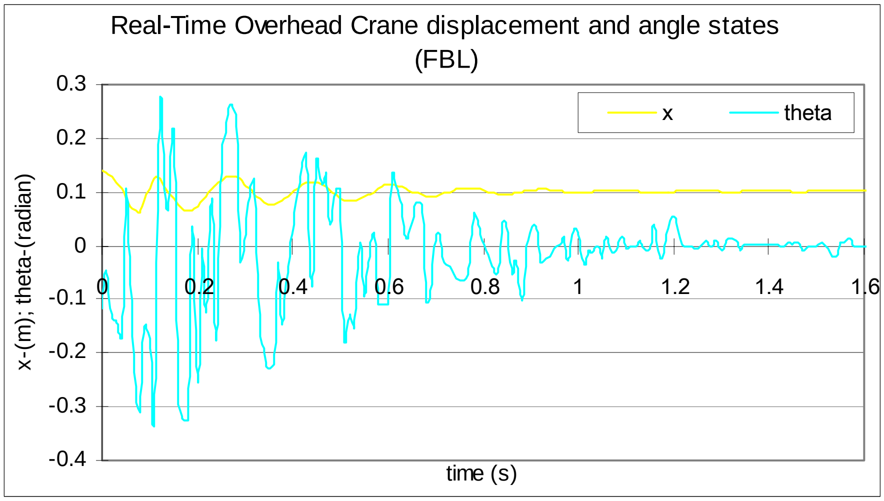


Figure 8.57: Overhead Crane Real-Time displacement and angle signals when set point is 0.1 m

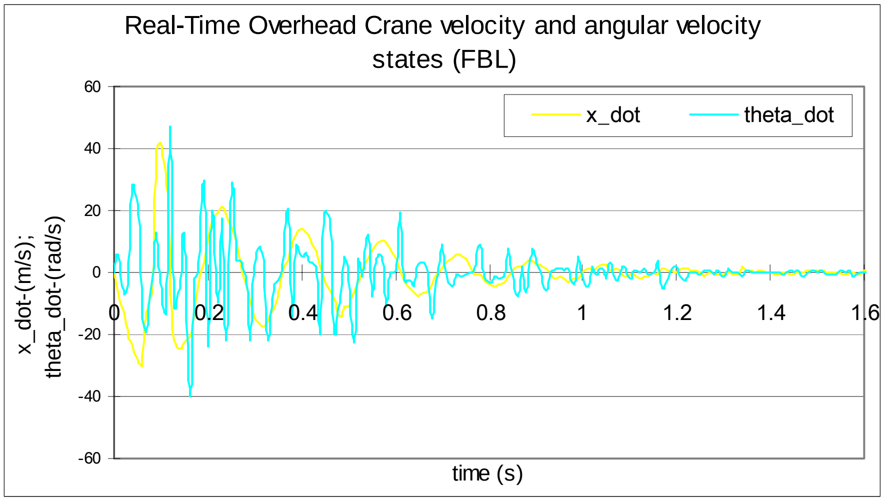


Figure 8.58: Overhead Crane Real-Time velocity and angular velocity signals when the set point is 0.1 m

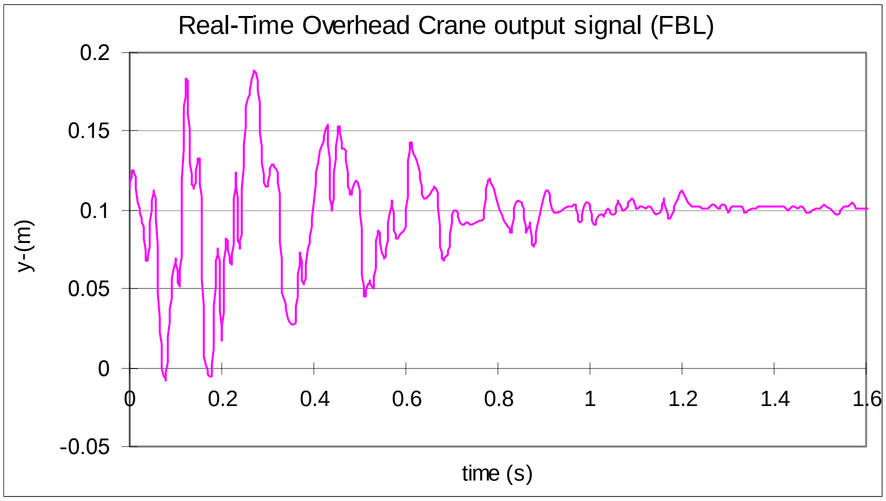


Figure 8.59: Overhead Crane Real-Time output signal when the set point is 0.1 m

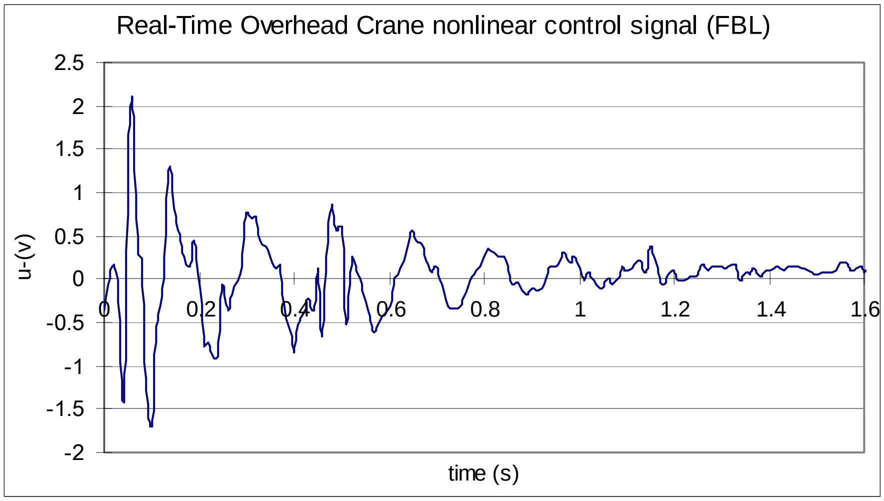


Figure 8.60: **Overhead Crane Real-Time nonlinear control signal when the set point is 0.1 m**

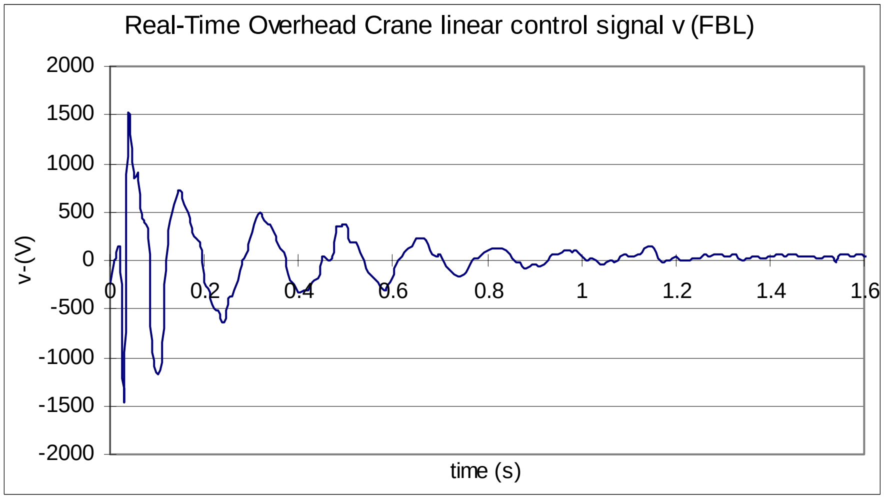


Figure 8.61: **Overhead Crane Real-Time linear control signal when the set point is 0.1 m**

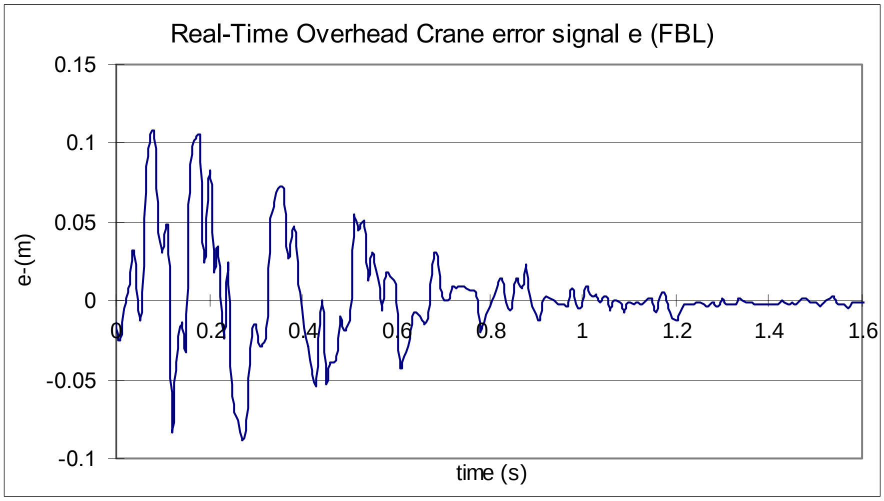


Figure 8.62: **Overhead Crane Real-Time error signal when the set point is 0.1 m**

8.5.1.3 Real-Time implementation results comparison

On the I/O DAQ device with signal conditioning and PC Real-Time implementation platform, both Lyapunov stability based MRC and feedback linearization method are deployed. It can be seen that when both methods are applied to control the inverted pendulum, the Lyapunov stability based MRC method has smaller overshot but larger oscillation amplitude. The feedback linearization method shows better steady state behaviour, such as: smaller oscillation amplitude for both position signal x and the angle signal θ. This method also shows a shorter rising time. When both methods are applied to the overhead crane, the Lyapunov stability based MRC method behaves much smoother than the feedback linearization method. Both methods take approximately the same time to settle down the overhead crane.

8.5.2 Real-Time implementation on the CompactRIO Real-Time implementation platform

The CompactRIO is the final Real-Time implementation platform. The CompactRIO realizes the concepts of the reconfigurable controller, associated with the pendulum reconfigurable plant. The lab-scale reconfigurable system is presented. In this project, both the Lyapunov stability based MRC control and the feedback linearization theory control simulation diagrams are developed in LabVIEW as well. Based on the LabVIEW simulation diagrams, the Real-Time implementation LabVIEW program is developed and implemented by using NI CompactRIO. In this section, the Real-Time implementation LabVIEW programs and the corresponding results are presented and analysed.

The reconfigurable control and acquisition system contain four major components, including:

- RIO FPGA core application for input, output, communication and control
- Time-critical loop for floating-point control, signal processing, analysis, and point-by-point decision making
- Normal-priority loop for embedded data logging, remote panel Web interface, and Ethernet/serial communication
- Networked host PC for remote graphical user interface, historical data logging, and postprocessing

Referring to this project, the reconfigurable control and acquisition system are configured in the following way, Figure 8.63:

- RIO FPGA core is configurable to read and write from the input-output ports only. The benefit of such configuration is the HOST VI and the FPGA VI are separated. FPGA VI specifies the input-output ports that are used in the project. The control algorithms are deployed in the HOST VI. Each time of changing the control algorithms code, only the HOST VI is downloaded to the CompactRIO, but the FPGA VI does not need be compiled again. This kind of the configuration is meaningful because normally compilation a FPGA VI takes more than 10 minutes. During the compilation time the LabVIEW programs are locked, no software development can be done. In this configuration, the system development time is saved.

- The signals are taken from the system and the data are logged into MS-Excel
- One laptop is used as the host PC, the Real-Time signal can be monitored. It is also used to download the HOST VI to the CompactRIO

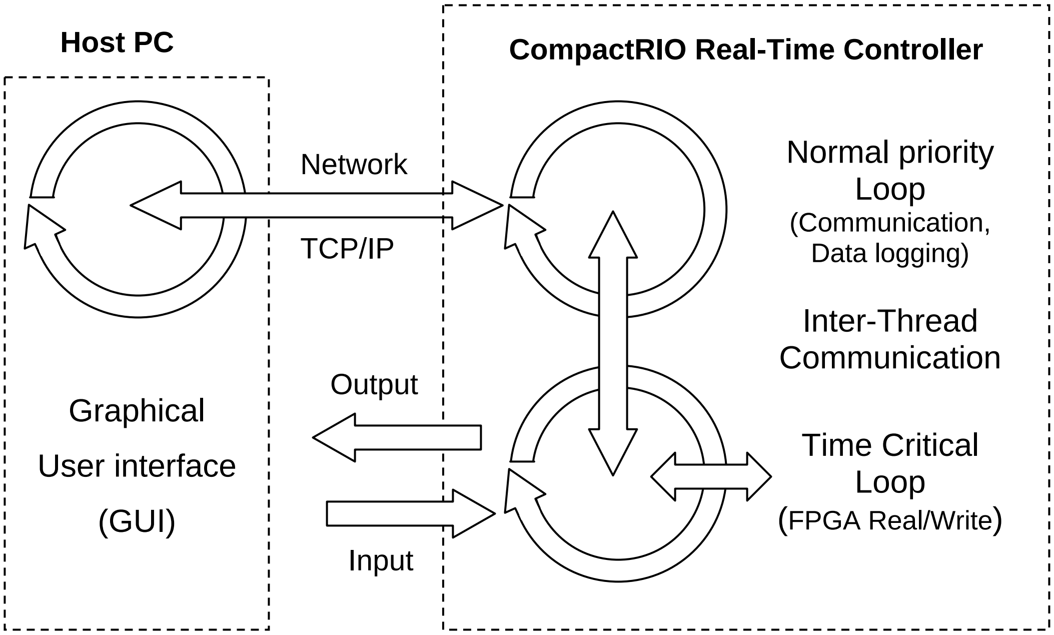

Host PC
CompactRIO Real-Time Controller
Network
TCP/IP
Normal priority Loop
(Communication, Data logging)
Inter-Thread Communication
Output
Graphical User interface (GUI)
Input
Time Critical Loop
(FPGA Real/Write)

Figure 8.63: Structure of the reconfigurable control and acquisition system

8.5.2.1 LabVIEW based subsystem for Simulation and Real-Time implementation

There are some basic subsystems built in LabVIEW according to the developed Matlab/Simulink block diagrams. Based on these subsystems, the completed LabVIEW simulation diagrams and the LabVIEW and CompactRIO based Real-Time implementations platform are constructed. The basic subsystems for the Lyapunov stability based MRC includes the linear controller, linear model, nonlinear controller and the nonlinear model. The subsystems built for the feedback linearization theory are the linear controller, nonlinear controller and the nonlinear models. These subsystems are different for the inverted pendulum and the overhead crane by means of the parameters and coefficients. But they have the same structures. In the following figures, these LabVIEW diagrams are presented. The I/O structure diagram and the shared FPGA VI are presented as well.

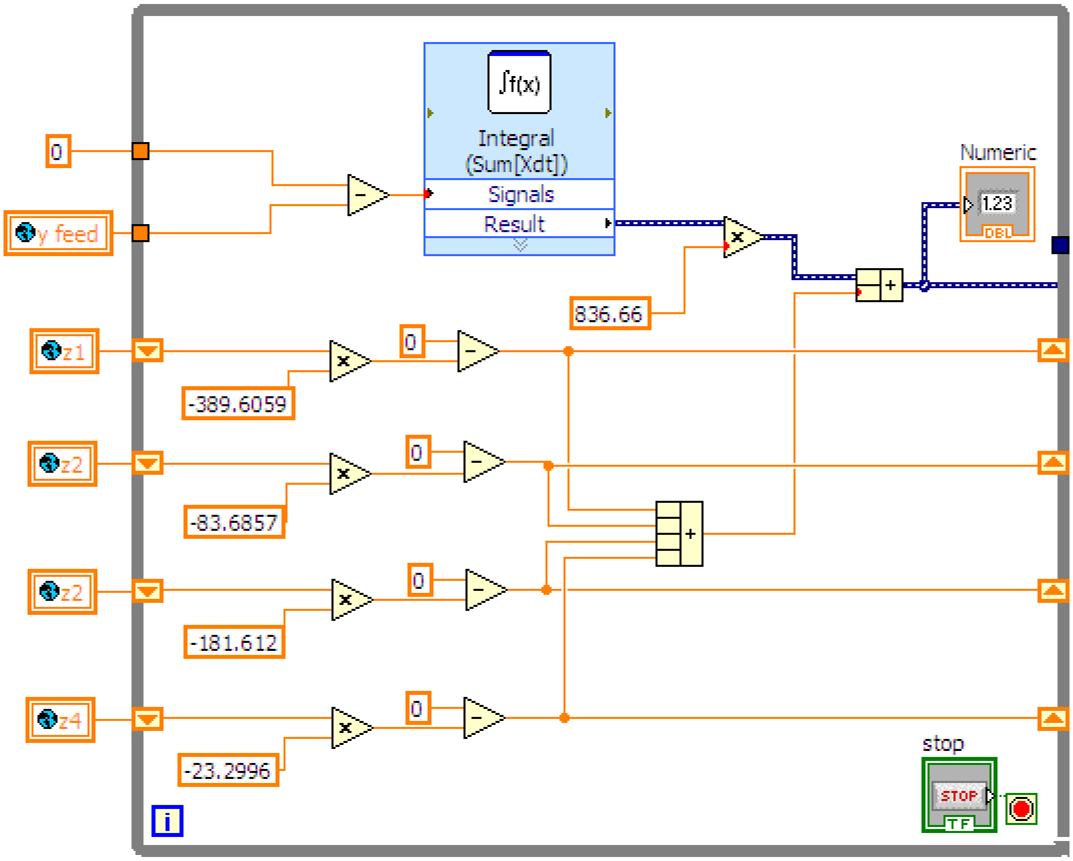


Figure 8.64: Linear controller diagram for both the Lyapunov stability based MRC and the feedback linearization approach

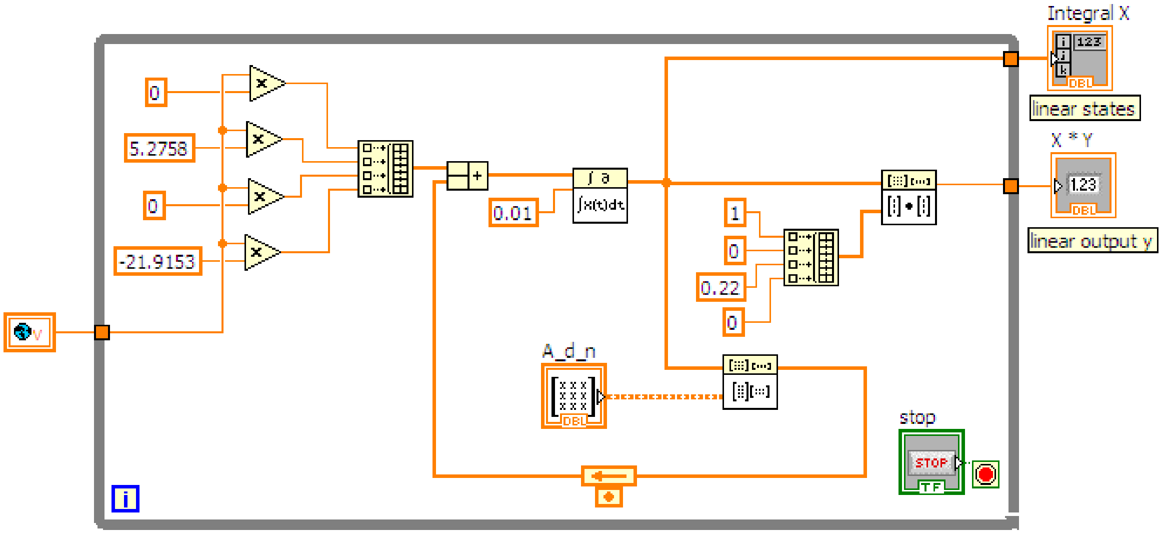
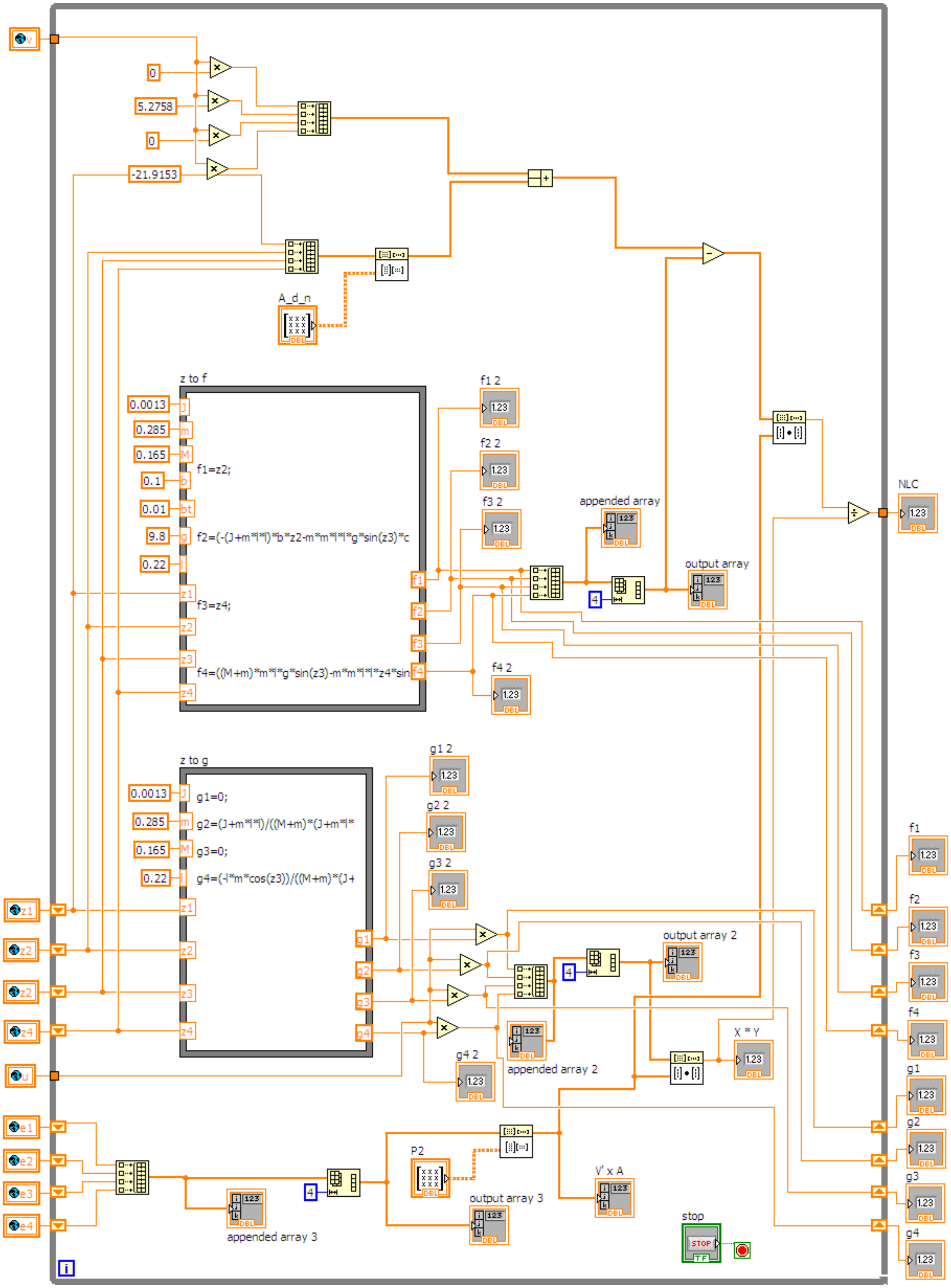


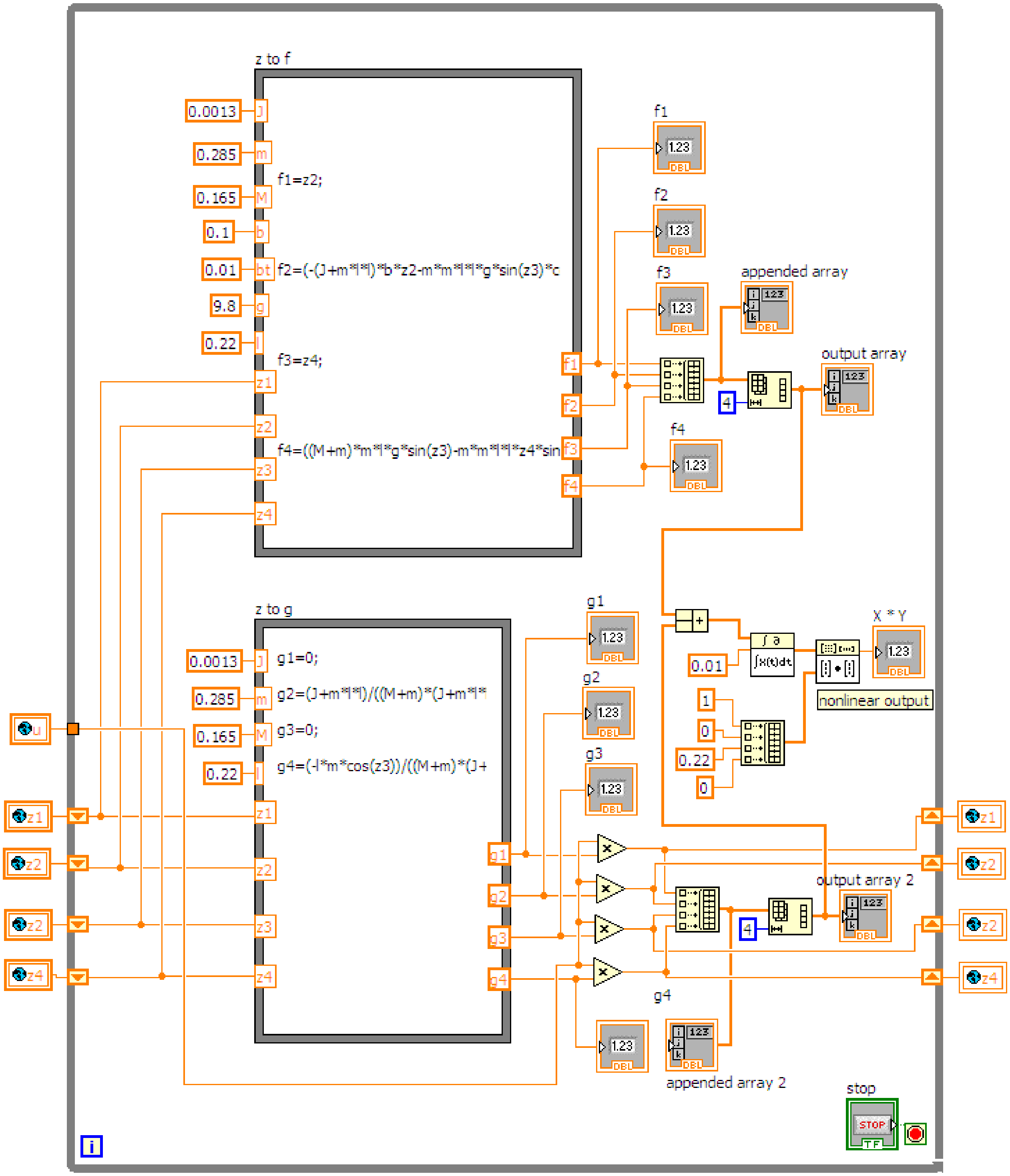
Figure 8.65: Linear model of the inverted pendulum



```
f1=z2;
f2=(-(J+m*l*l)*b*z2-m*m*l*l*g*sin(z3)*c
f3=z4;
f4=((M+m)*m*l*g*sin(z3)-m*m*l*l*z4*sin
```

```
g1=0;
g2=(J+m*l*l)/((M+m)*(J+m*l*
g3=0;
g4=(-l*m*cos(z3))/((M+m)*(J+
```

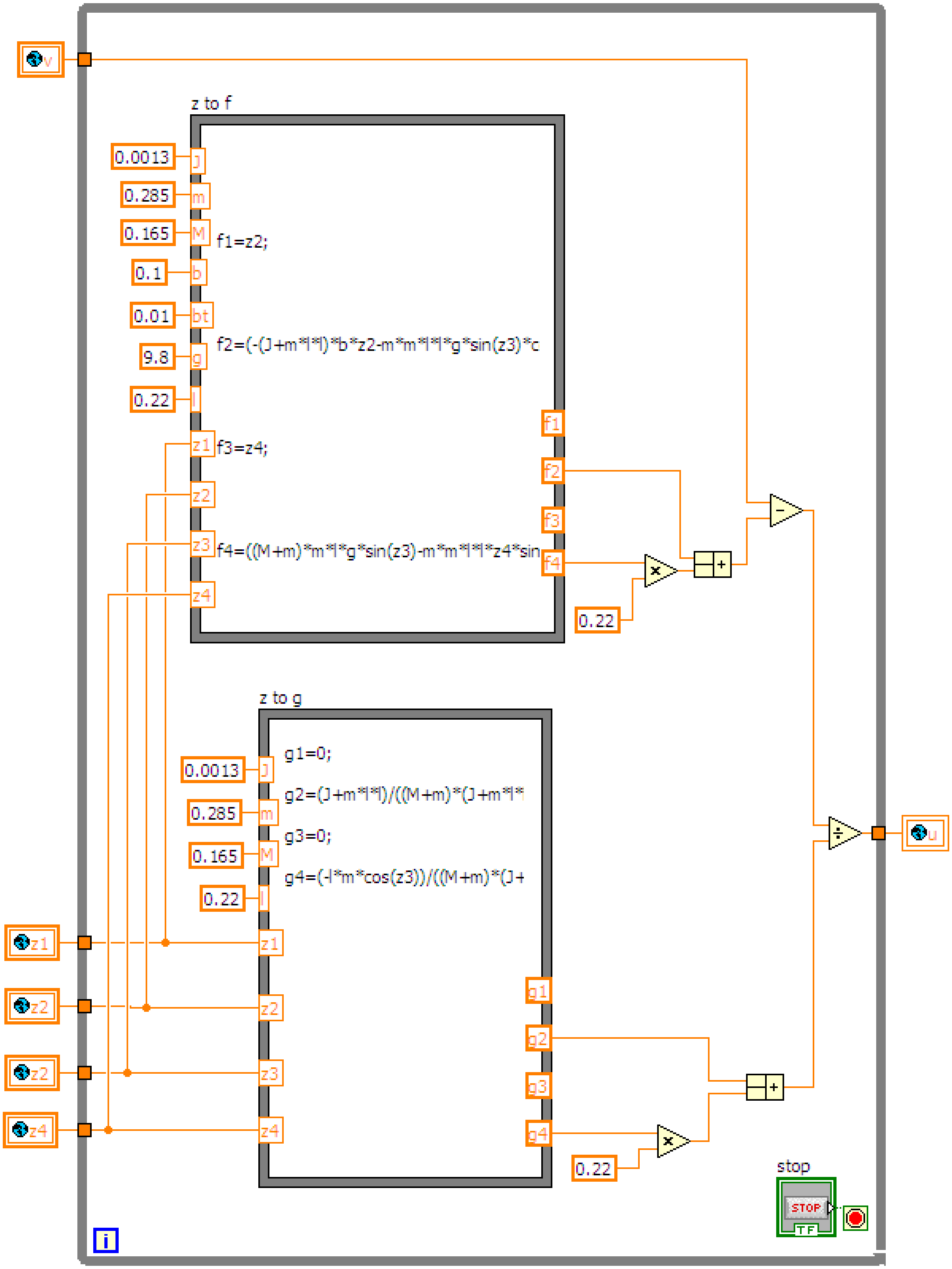
Figure 8.66: Nonlinear controller of the Lyapunov stability based MRC



```
f1=z2;

f2=(-(J+m*l*l)*b*z2-m*m*l*l*g*sin(z3)*c

f3=z4;

f4=((M+m)*m*l*g*sin(z3)-m*m*l*l*z4*sin
```

```
g1=0;
g2=(J+m*l*l)/((M+m)*(J+m*l*
g3=0;
g4=(-l*m*cos(z3))/((M+m)*(J+
```

Figure 8.67: Nonlinear model of the inverted pendulum



```
f1=z2;

f2=(-(J+m*l*l)*b*z2-m*m*l*l*g*sin(z3)*c

f3=z4;

f4=((M+m)*m*l*g*sin(z3)-m*m*l*l*z4*sin
```

```
g1=0;

g2=(J+m*l*l)/((M+m)*(J+m*l*

g3=0;

g4=(-l*m*cos(z3))/((M+m)*(J+
```

Figure 8.68: Nonlinear controller of the feedback linearization approach

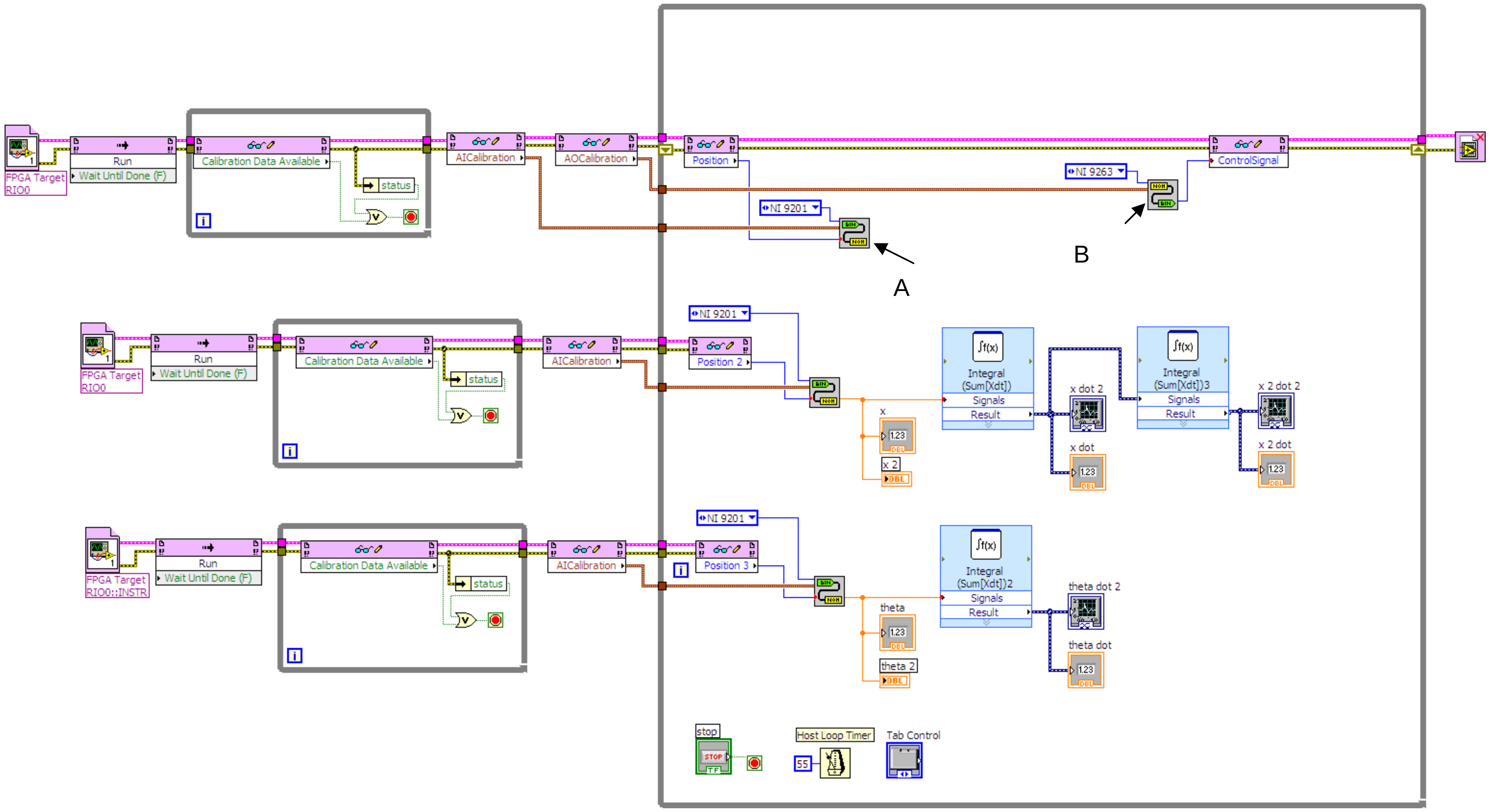


Figure 8.69: The input-output structure

In the above diagram, Figure 8.69, the input-output structure is set up. The HOST VI is developed based on this diagram. This VI acquires the output signal y, states signal x, θ and sends the control signal u to the actual plant. But there is no control algorithm between the inputs and the output. As shown in Figure 8.69, between point A and B, the proper control algorithm should be inserted. In this project, the design controllers are based on the Lyapunov stability and feedback linearization theory. The diagram below shows how the FPGA VI is configured. This FPGA VI is only used for data acquiring and to send control signal.

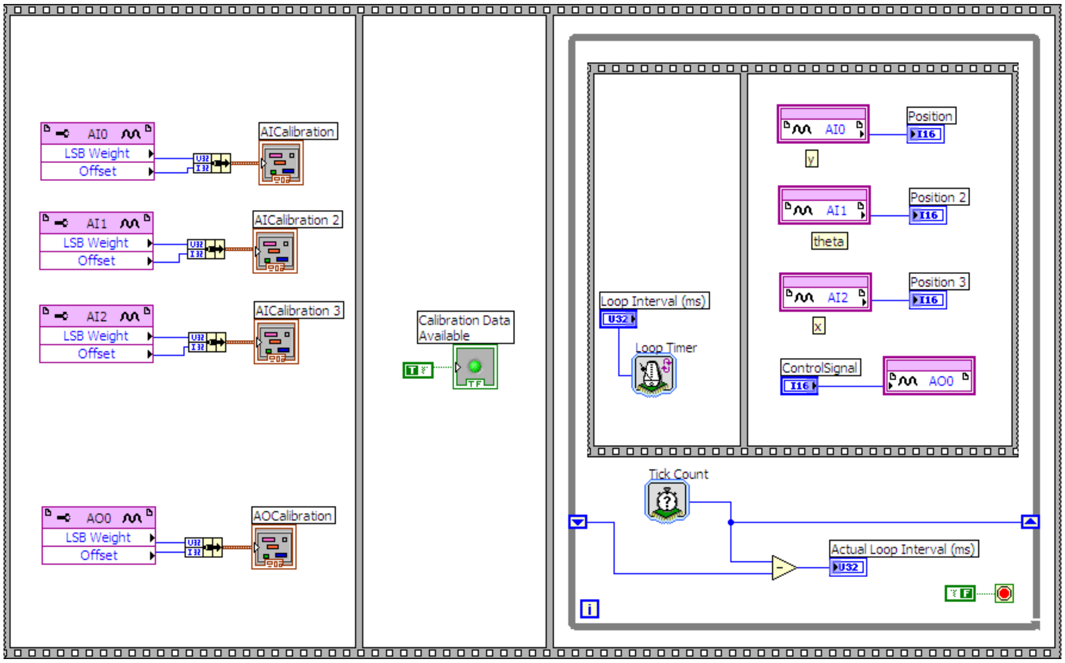


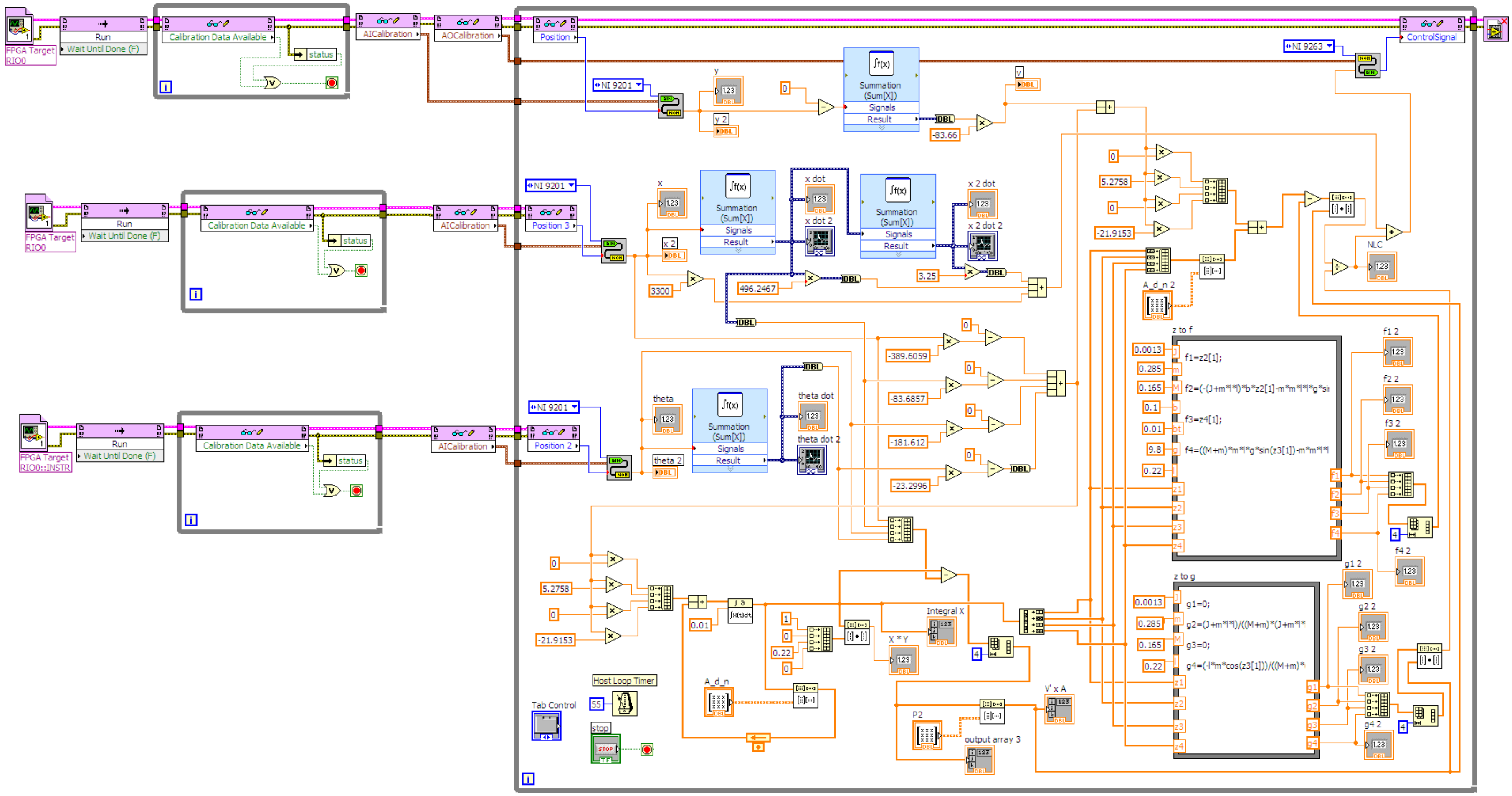
Figure 8.70: The shared FPGA VI

Based on above presented subsystems, the LabVIEW Real-Time implementation diagrams and their Real-Time implementation results are presented in the following sections.

8.5.2.2 Lyapunov stability based MRC Real-Time implementation

8.5.2.2.1 Lyapunov stability based MRC Real-Time implementation LabVIEW diagrams

In the Figure 8.71, the complete Lyapunov stability based MRC Real-Time implementation LabVIEW diagram for the inverted pendulum is shown. In this diagram, the FPGA I/O, the linear control, linear model, nonlinear controller and the nonlinear model are contained.



```
f1=z2[1];
f2=(-(J+m*l*l)*b*z2[1]-m*m*l*l*g*si
f3=z4[1];
f4=((M+m)*m*l*g*sin(z3[1])-m*m*l*l
```

```
g1=0;
g2=(J+m*l*l)/((M+m)*(J+m*l*
g3=0;
g4=(-l*m*cos(z3[1]))/((M+m)*
```

Figure 8.71: The input-output structure

The same LabVIEW diagram is used to control the overhead crane in Real-Time by changing the corresponding parameters. The Real-Time implementation results are presented in the next section.

8.5.2.2.2 Lyapunov stability based MRC Real-Time implementation results

On the CompactRIO Real-Time platform, the very similar results are received as the I/O I/O DAQ with signal conditioning and PC platform. The received results are presented below.

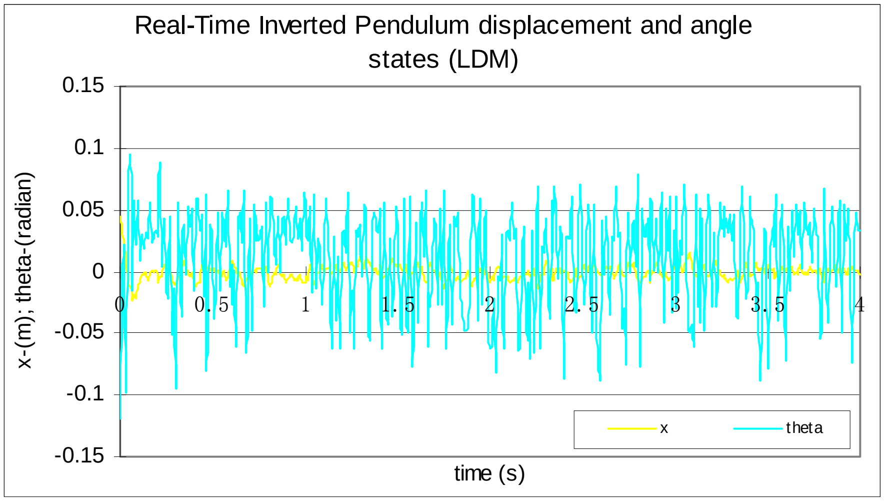


Figure 8.72: Inverted Pendulum Real-Time displacement and angle state signals when set point is 0 m

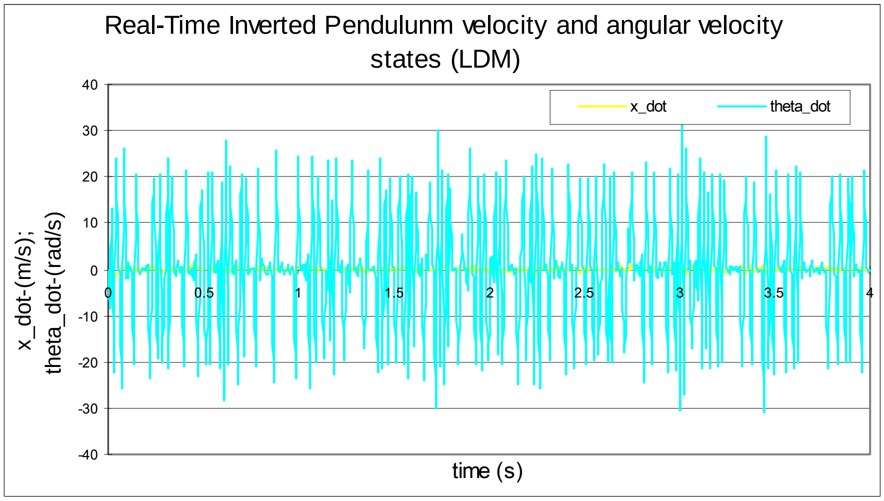


Figure 8.73: Inverted pendulum Real-Time velocity and angular velocity state signals when the set point is 0 m

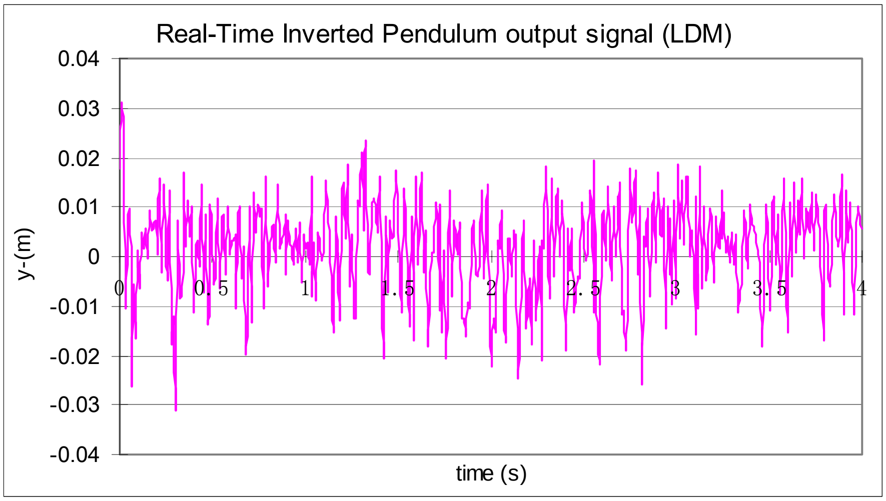


Figure 8.74: Inverted pendulum Real-Time output signal when the set point is 0 m

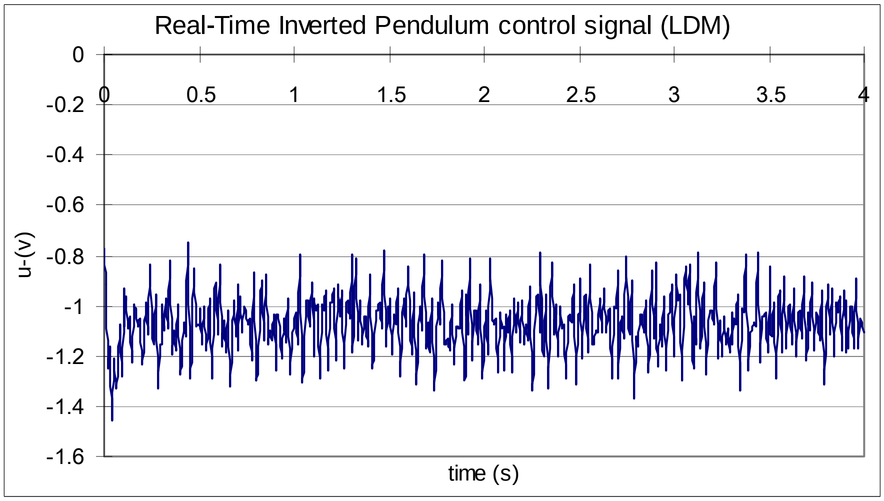


Figure 8.75: Inverted pendulum Real-Time control signal when the set point is 0 m

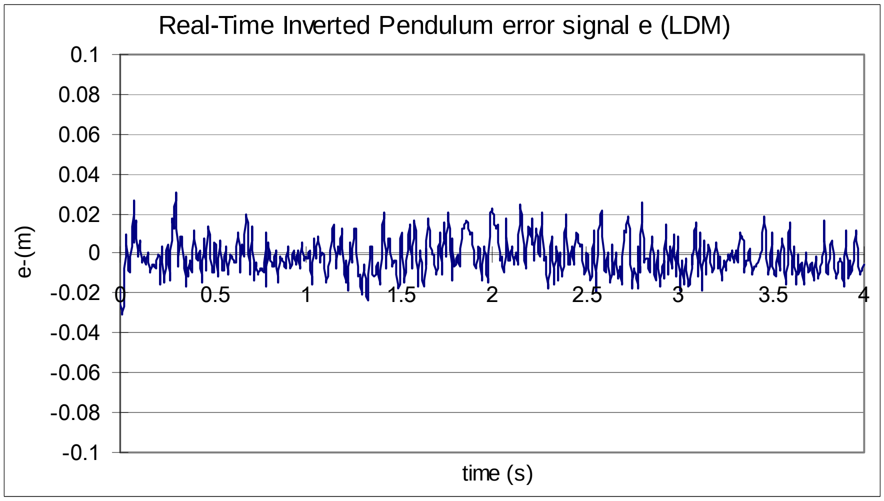


Figure 8.76: Inverted pendulum Real-Time error signal when the set point is 0 m

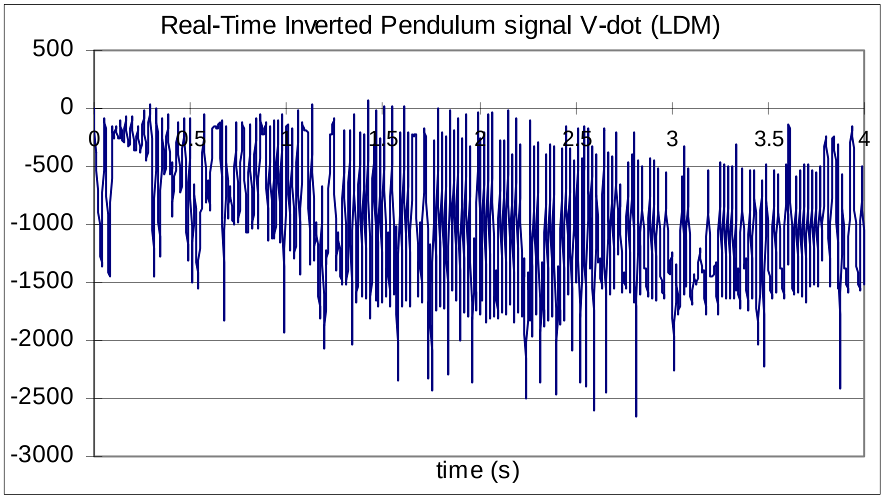


Figure8.77: **Inverted pendulum Real-Time signal of Lyapunov function derivative when the set point is 0 m**

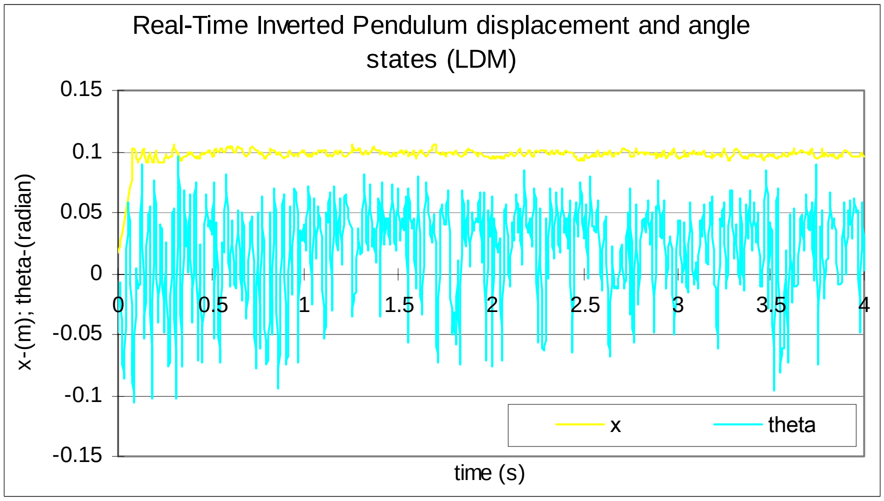


Figure 8.78: **Inverted Pendulum Real-Time displacement and angle state signals when set point is 0.1 m**

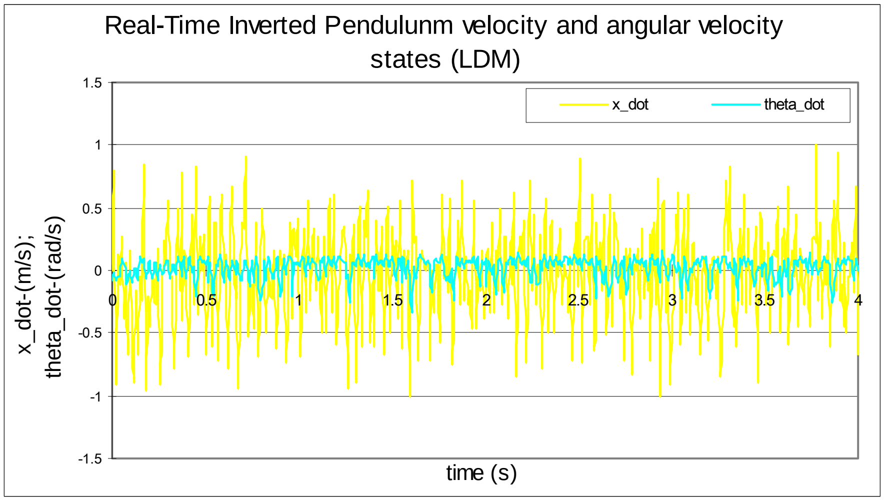


Figure 8.79: **Inverted pendulum Real-Time velocity and angular velocity state signals when the set point is 0.1 m**

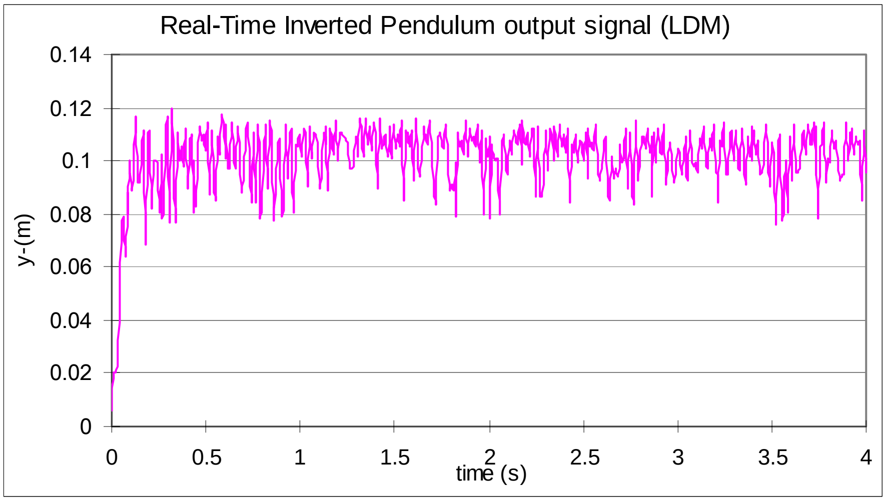


Figure 8.80: **Inverted pendulum Real-Time output signal when the set point is 0.1 m**

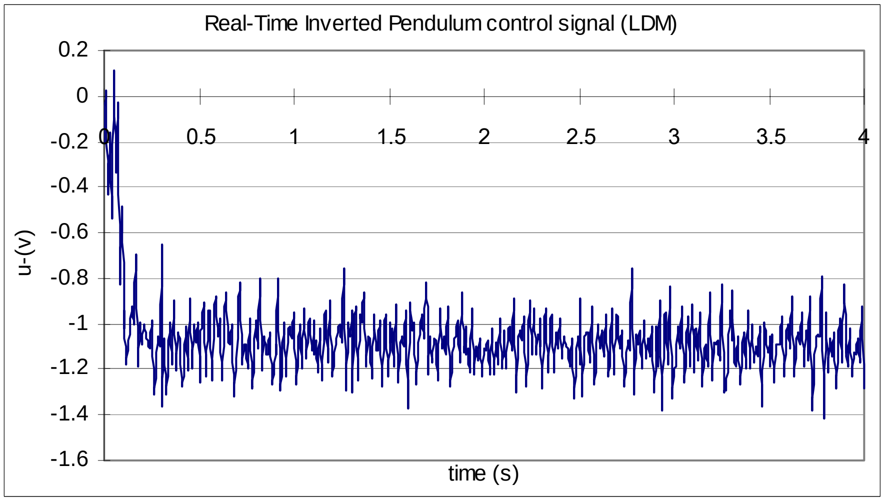


Figure 8.81: **Inverted pendulum Real-Time control signal when the set point is 0.1 m**

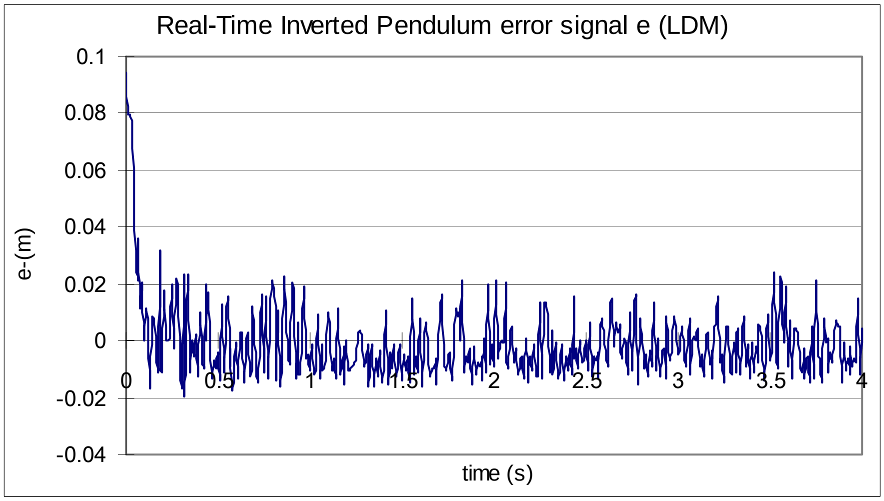


Figure 8.82: **Inverted pendulum Real-Time error signal when the set point is 0.1 m**

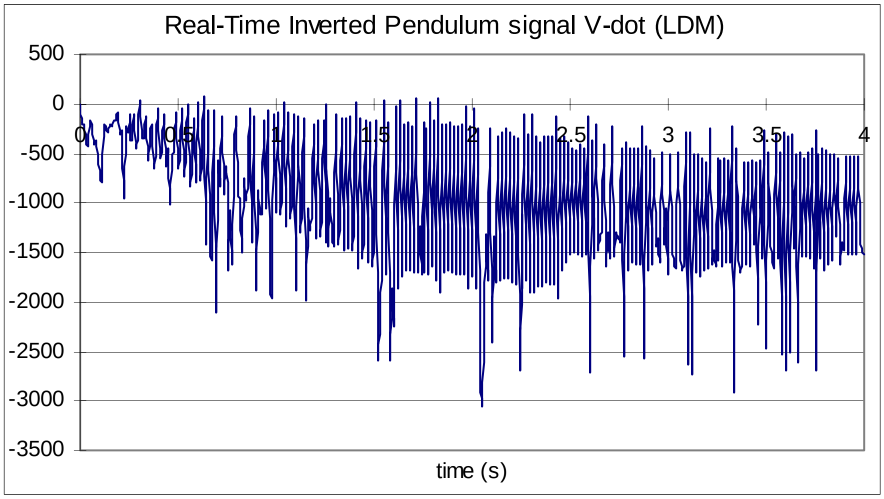


Figure 8.85: **Inverted pendulum Real-Time signal of Lyapunov function derivative when the set point is 0.1 m**

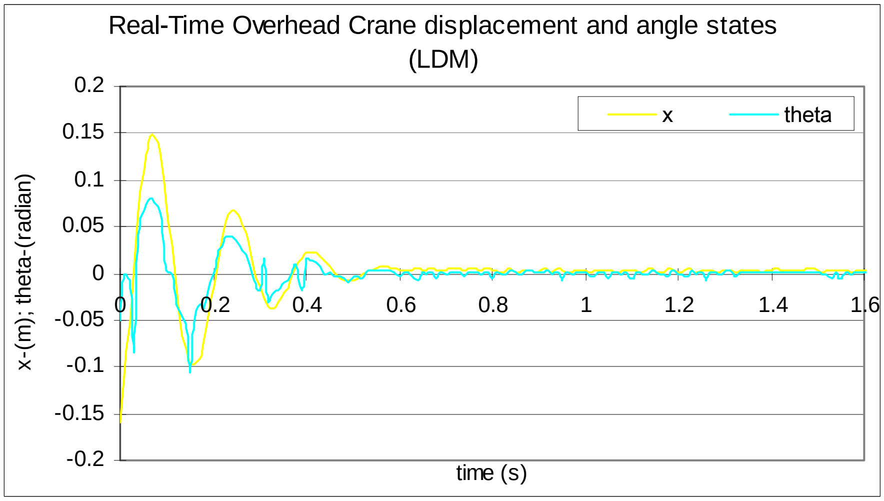


Figure 8.84: **Overhead Crane Real-Time displacement and angle state signals when set point is 0 m**

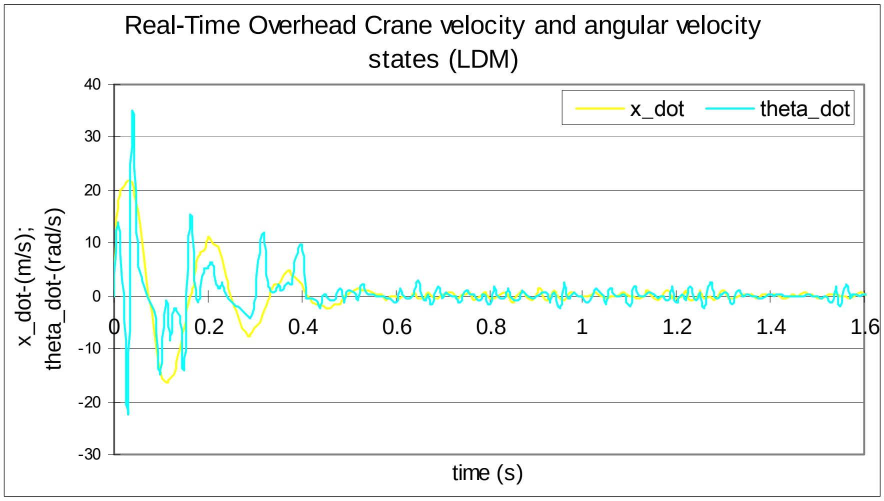


Figure 8.85: **Overhead Crane Real-Time velocity and angular velocity state signals when the set point is 0 m**

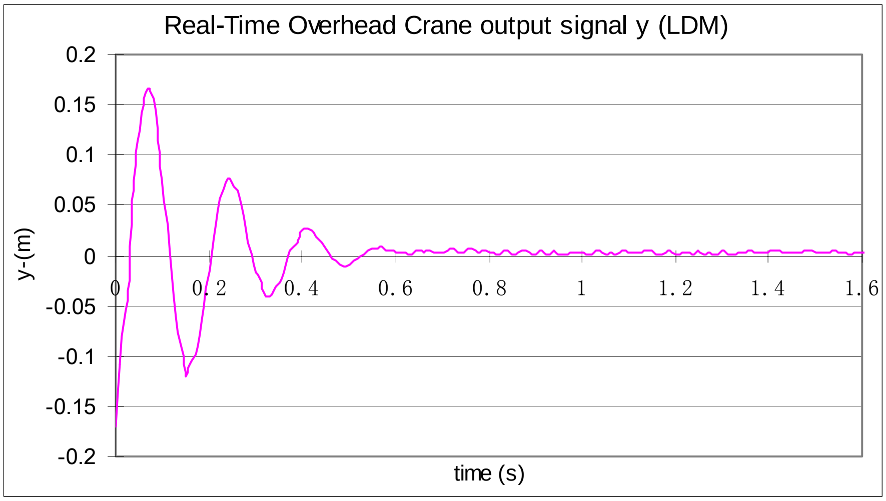


Figure 8.86: Overhead Crane Real-Time output signal when the set point is 0 m

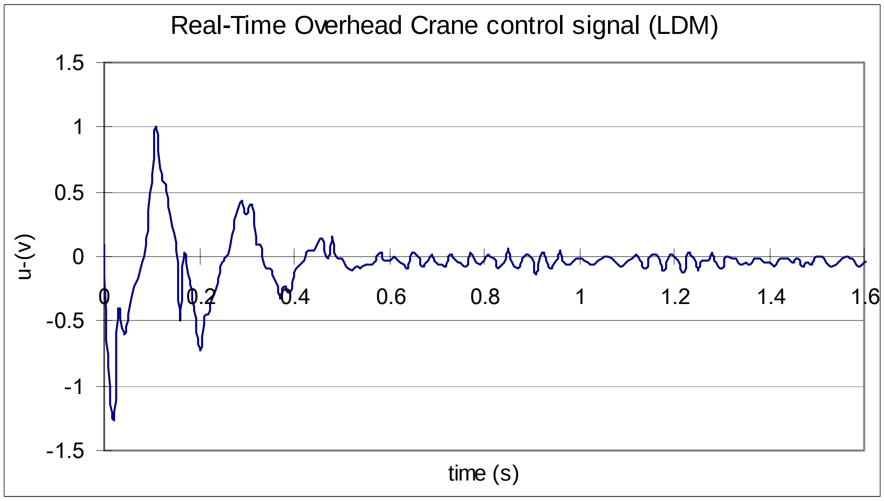


Figure 8.87: Overhead Crane Real-Time control signal when the set point is 0 m

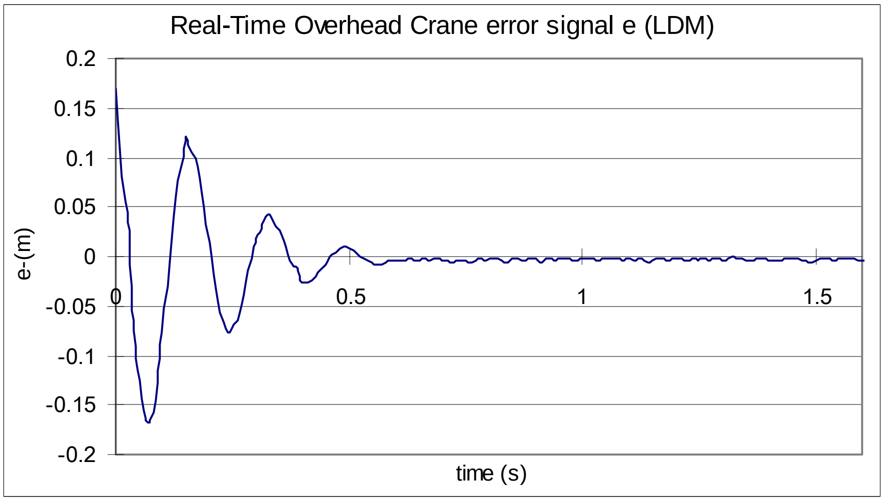


Figure 8.88: Overhead Crane Real-Time error signal when the set point is 0 m

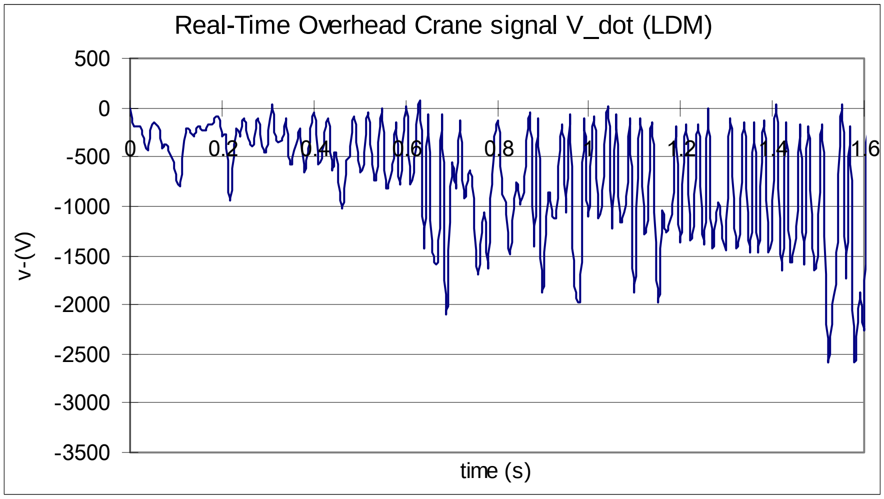


Figure 8.89: **Overhead Crane Real-Time signal of Lyapunov function derivative when the set point is 0 m**

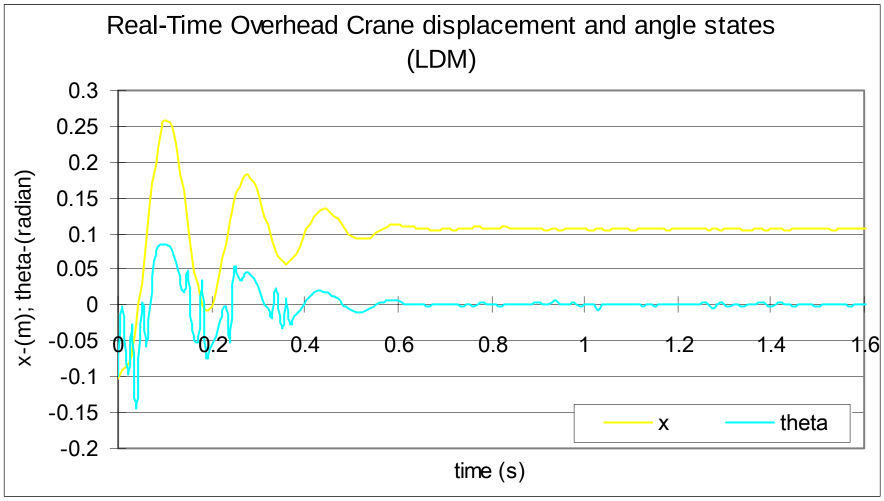


Figure 8.90.: **Overhead Crane Real-Time displacement and angle state signals when set point is 0.1 m**

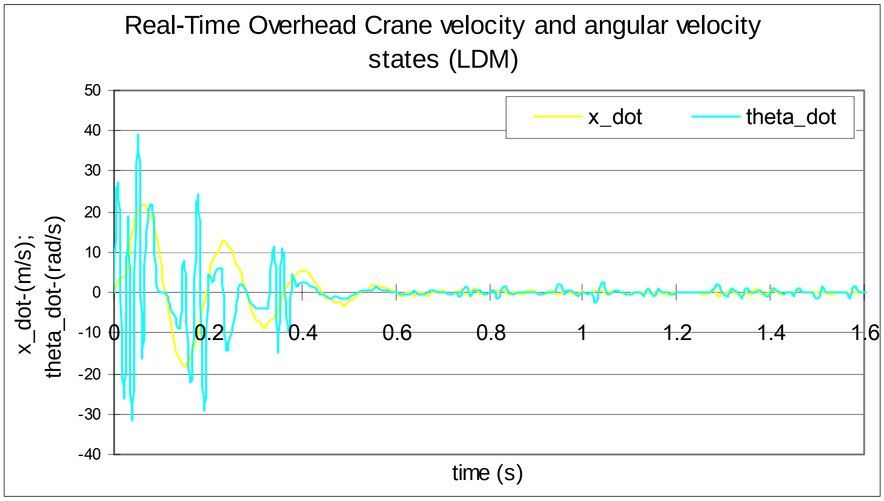


Figure 8.91: **Overhead Crane Real-Time velocity and angular velocity state signals when the set point is 0.1 m**

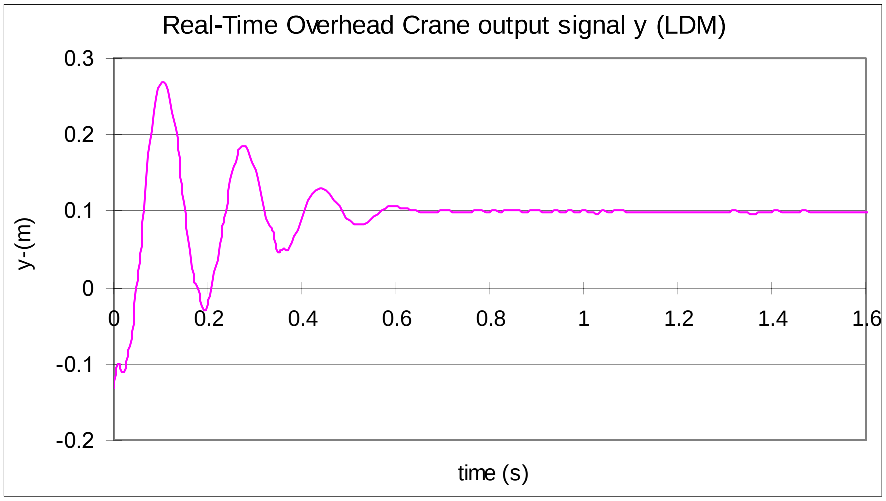


Figure 8.92: Overhead Crane Real-Time output signal when the set point is 0.1 m

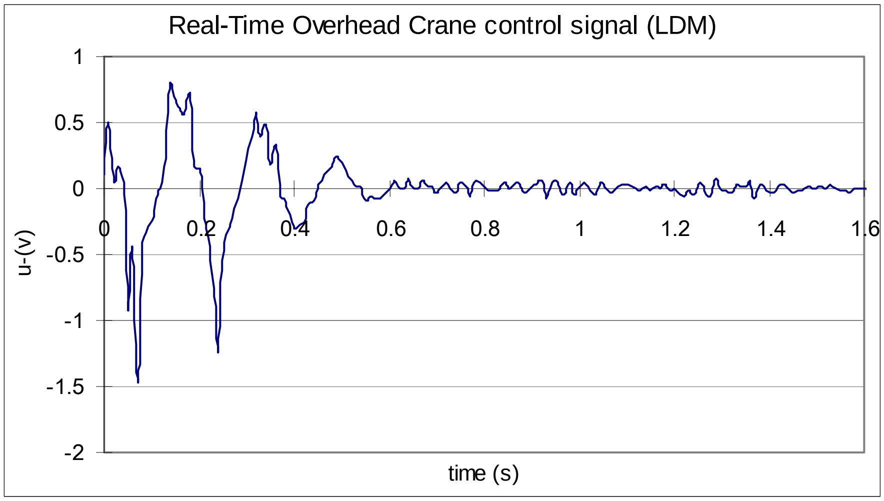


Figure 8.93: Overhead Crane Real-Time control signal when the set point is 0.1 m

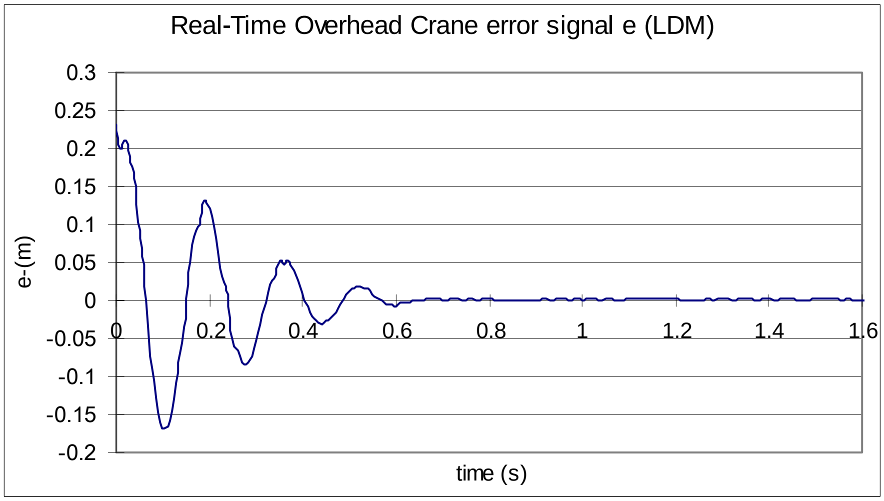


Figure 8.94: Overhead Crane Real-Time error signal when the set point is 0.1 m

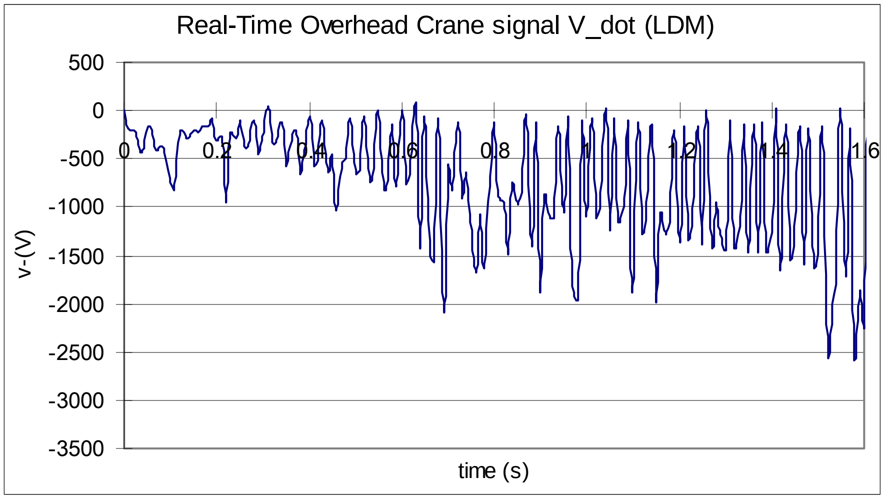


Figure 8.95: Overhead Crane Real-Time signal of Lyapunov function derivative when the set point is 0.1 m

8.5.2.3 Feedback linearization method Real-Time implementation

8.5.2.3.1 Feedback linearization method Real-Time implementation LabVIEW block diagrams

Similarly, based on the subsystems from Figure 8.64 to Figure 8.70, the feedback linearization theory Real-Time implementation LabVIEW diagram is constructed as Shown in Figure 8.96.

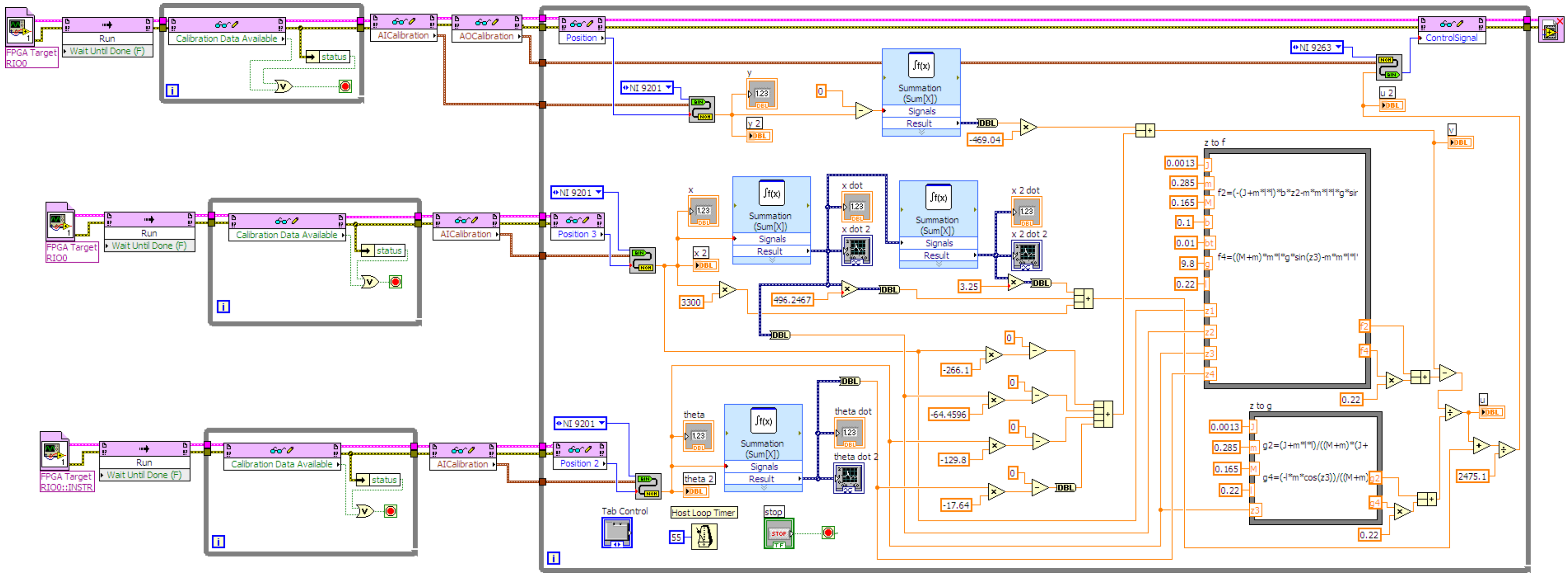


Figure 8.96: The input-output structure

8.5.2.3.2 Feedback linearization method Real-Time implementation results

The Real-Time implementation results of the feedback linearization theory method control on the CompactRIO are presented.

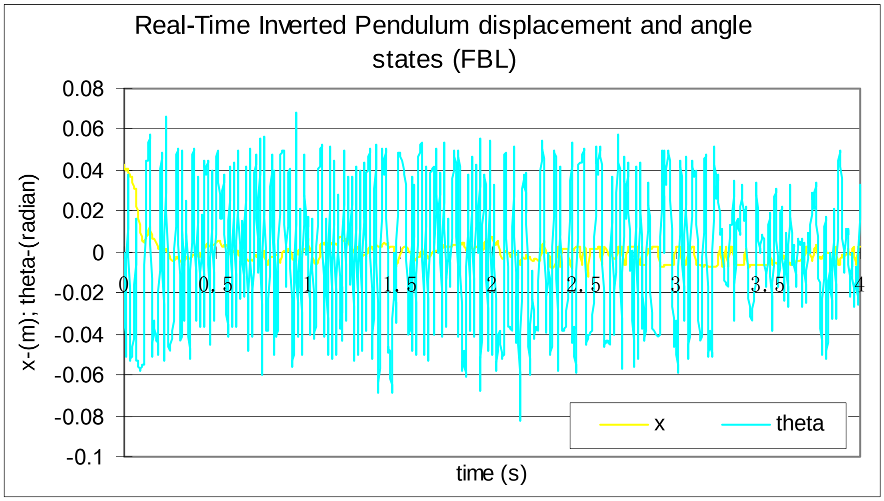


Figure 8.97: **Inverted Pendulum Real-Time displacement and angle state signals when set point is 0 m**

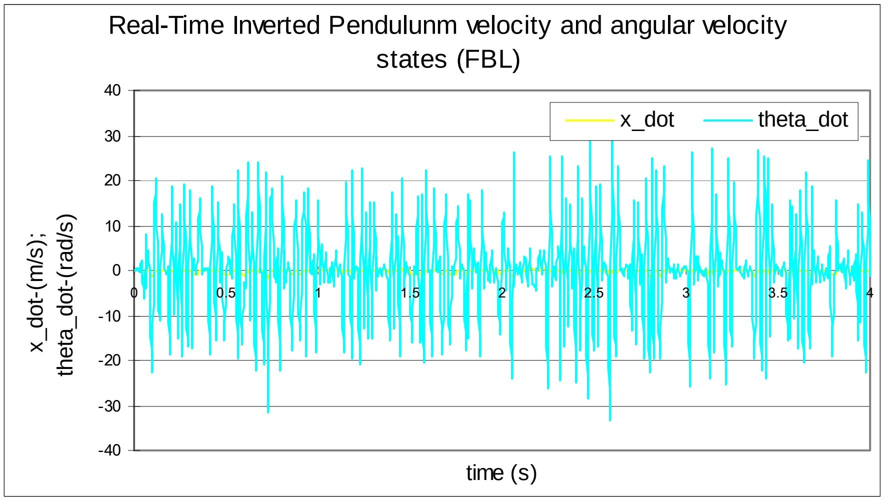


Figure8.98: **Inverted pendulum Real-Time velocity and angular velocity state signals when the set point is 0 m**

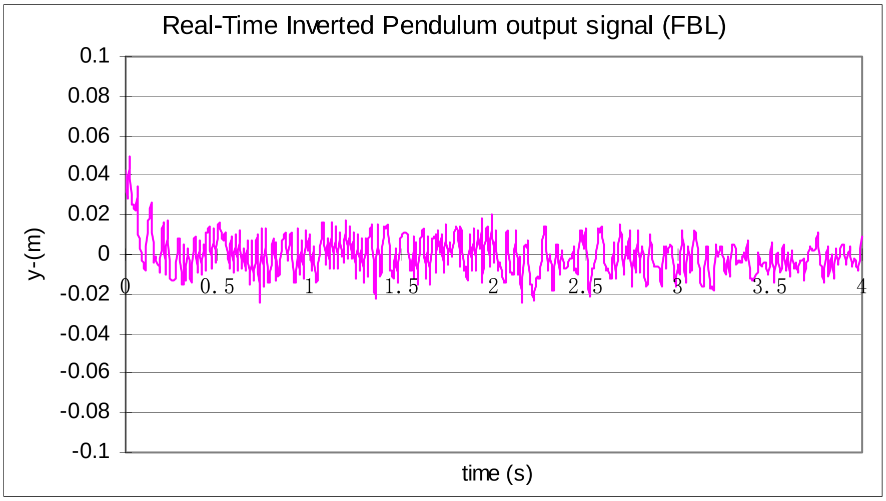


Figure 8.99: **Inverted pendulum Real-Time output signal when the set point is 0 m**

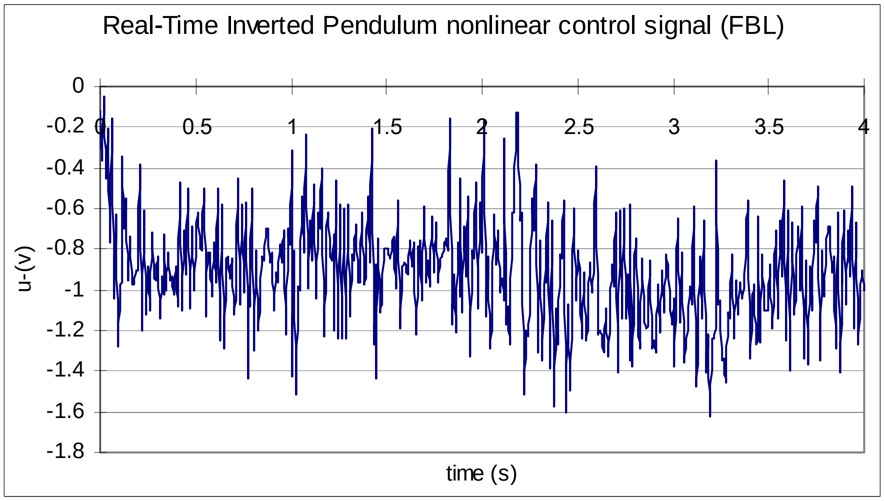


Figure 8.100: **Inverted pendulum Real-Time nonlinear control signal when the set point is 0 m**

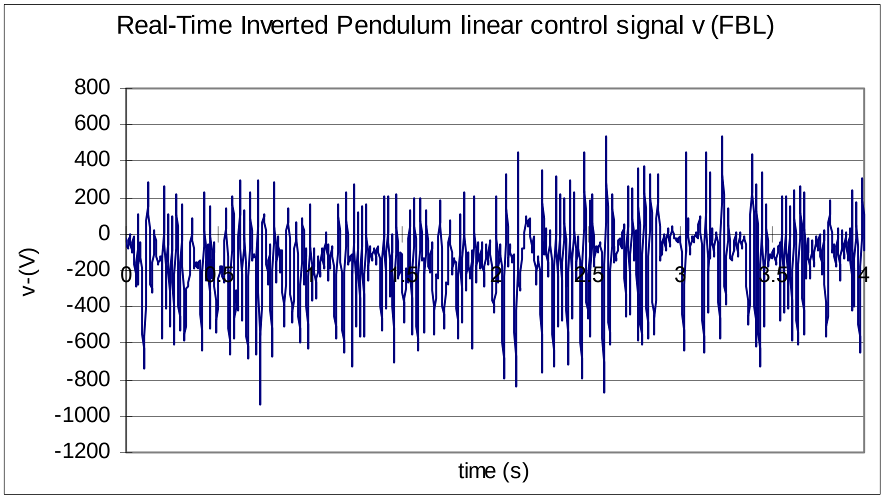


Figure 8.101: **Inverted pendulum Real-Time linear control signal when the set point is 0 m**

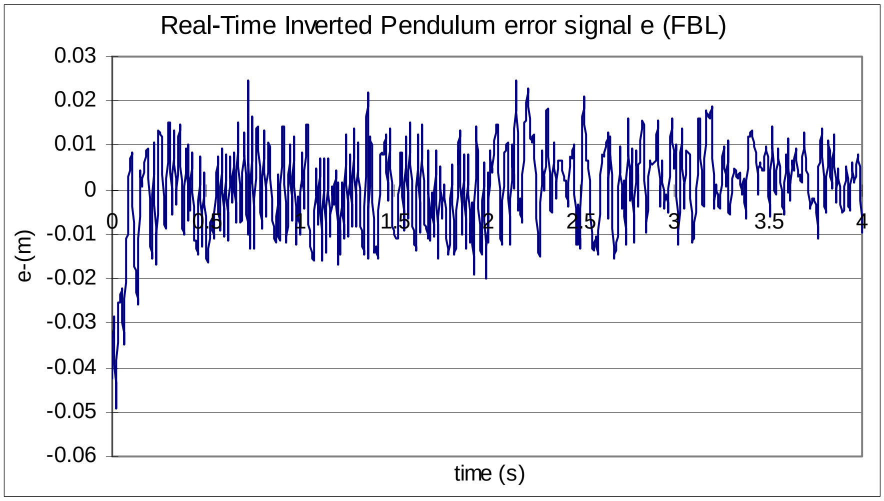


Figure 8.102: Inverted pendulum Real-Time error signal when the set point is 0 m

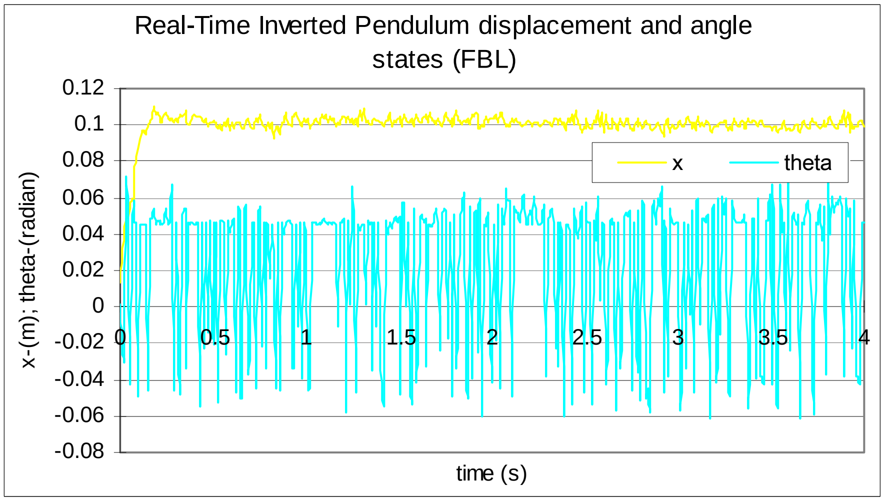


Figure 8.103: Inverted Pendulum Real-Time displacement and angle state signals when set point is 0.1 m

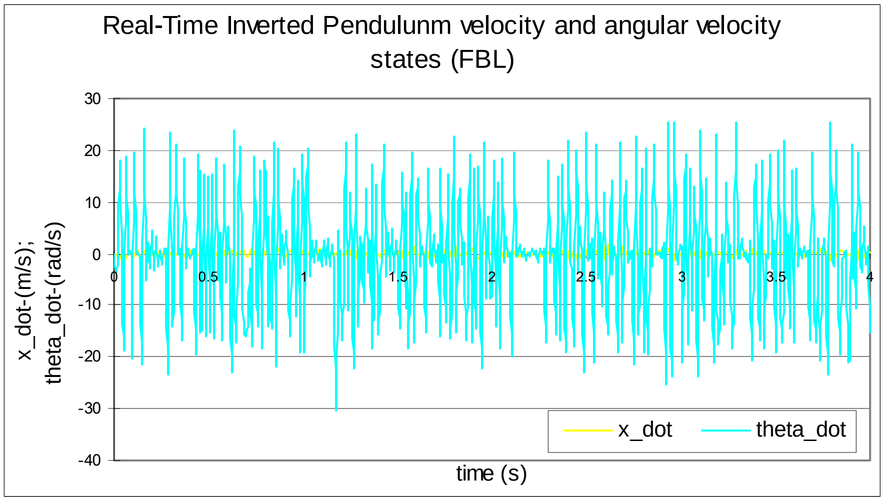


Figure 8.104: Inverted pendulum Real-Time velocity and angular velocity state signals when the set point is 0.1 m

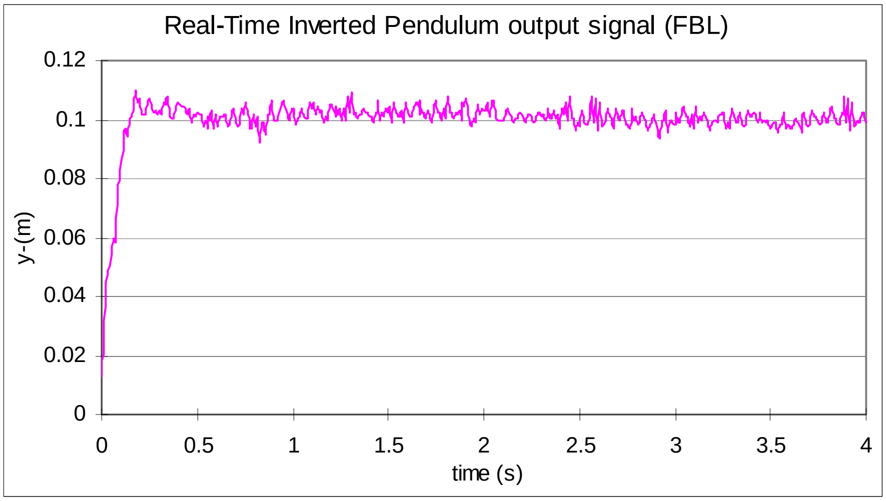


Figure 8.105: **Inverted pendulum Real-Time output signal when the set point is 0.1 m**

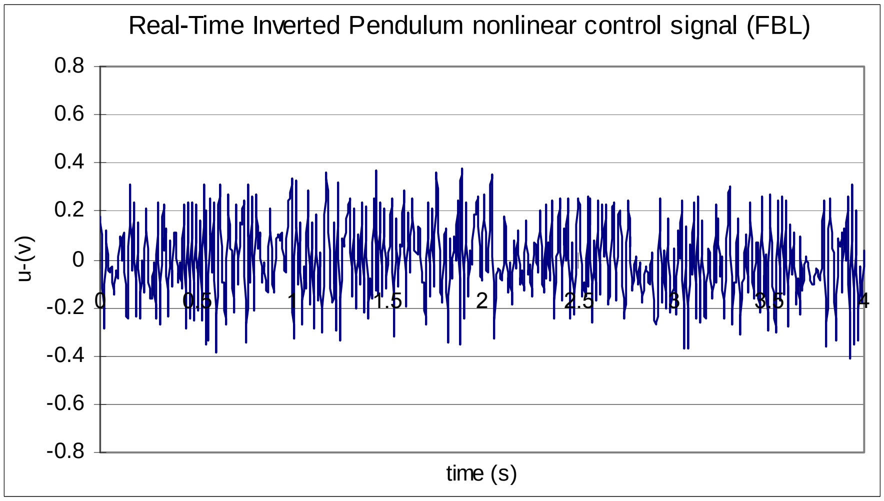


Figure 8.106: **Inverted pendulum Real-Time nonlinear control signal when the set point is 0.1 m**

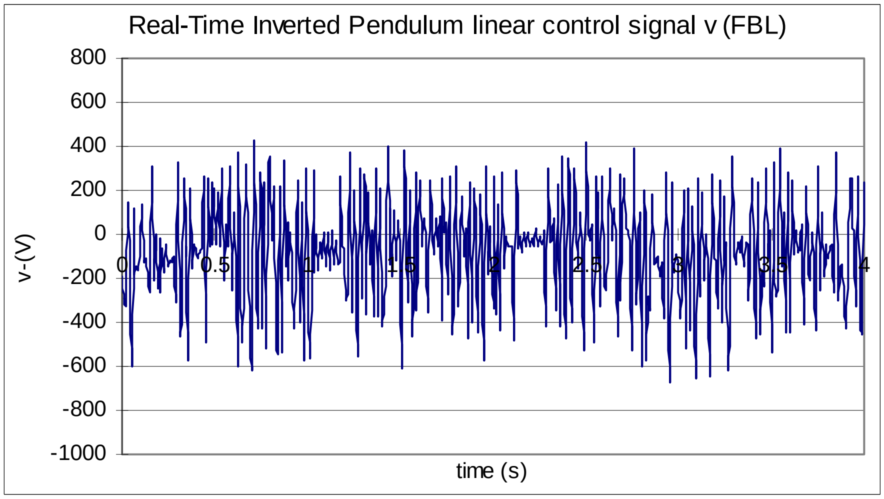


Figure 8.107: **Inverted pendulum Real-Time linear control signal when the set point is 0.1 m**

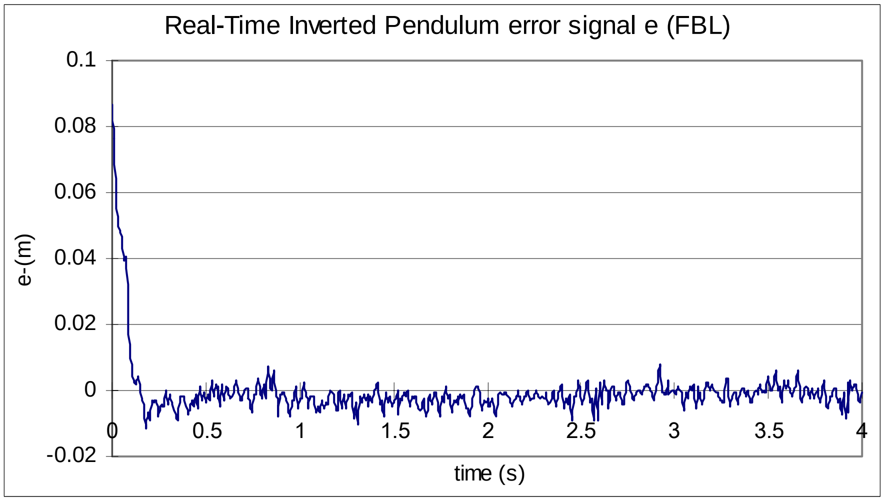


Figure 8.108: Inverted pendulum Real-Time error signal when the set point is 0.1 m

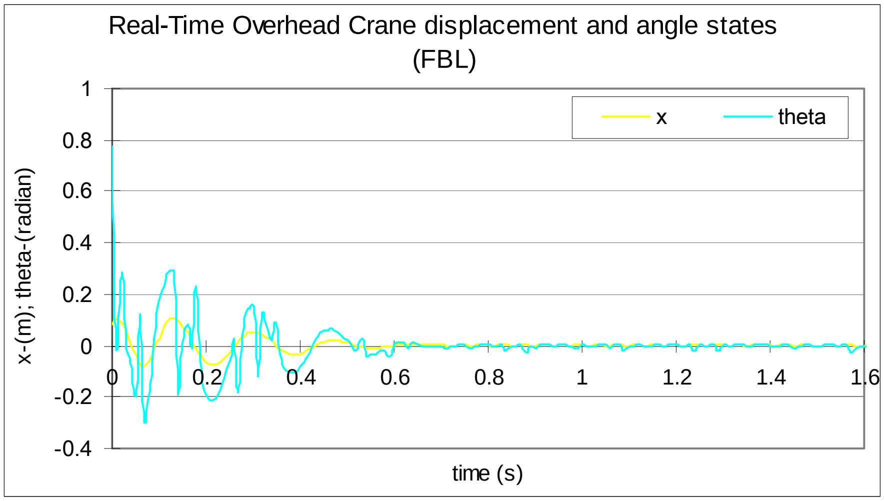


Figure 8.109: Overhead Crane Real-Time displacement and angle state signals when set point is 0 m

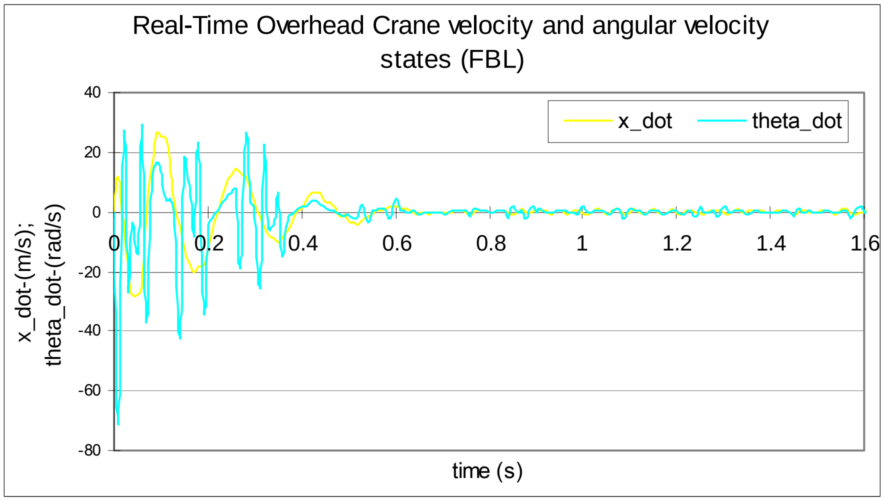


Figure 8.110: Overhead Crane Real-Time velocity and angular velocity state signals when the set point is 0 m

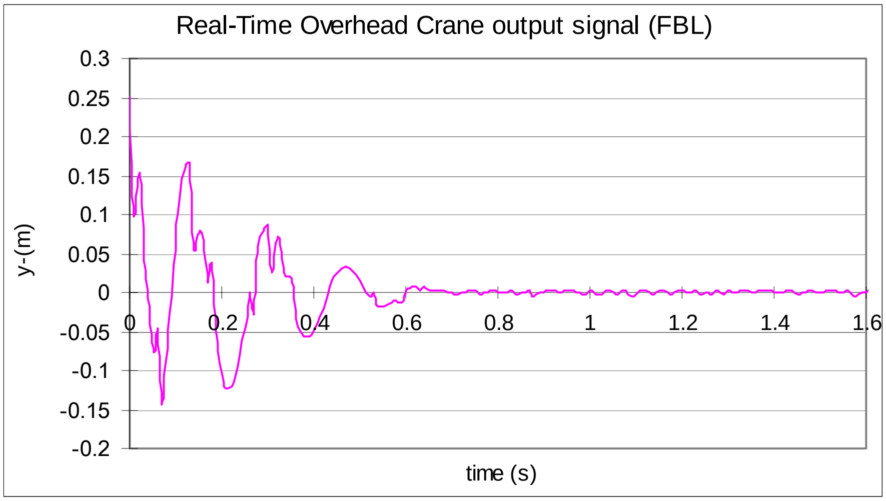


Figure 8.111: **Overhead Crane Real-Time output signal when the set point is 0 m**

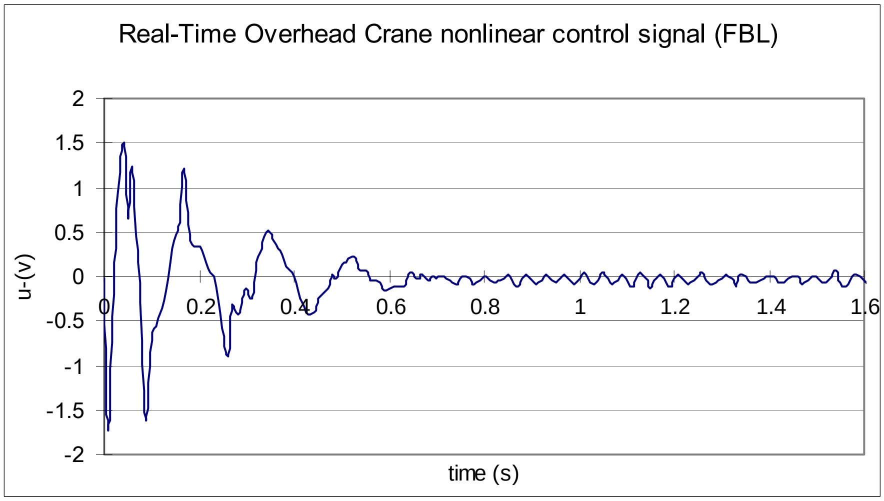


Figure 8.112: **Overhead Crane Real-Time nonlinear control signal when the set point is 0 m**

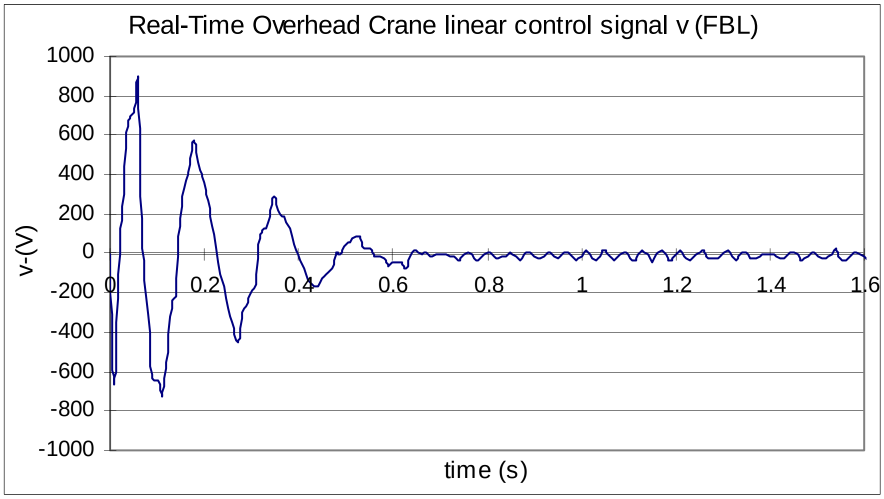


Figure 8.113: **Overhead Crane Real-Time linear control signal when the set point is 0 m**

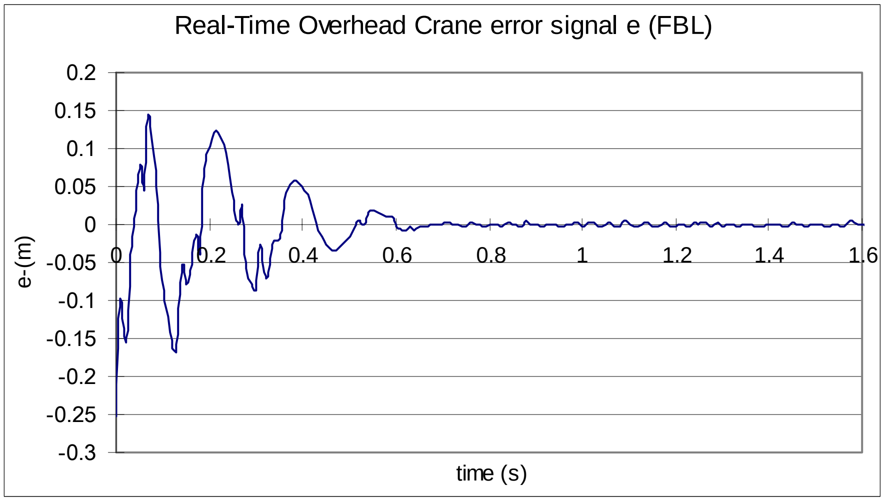


Figure 8.114: Overhead Crane Real-Time error signal when the set point is 0 m

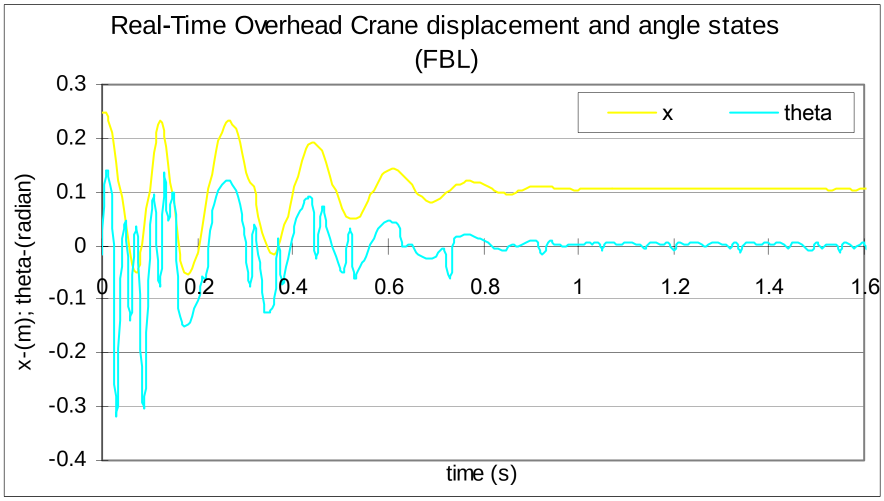


Figure 8.115: Overhead Crane Real-Time displacement and angle state signals when set point is 0.1 m

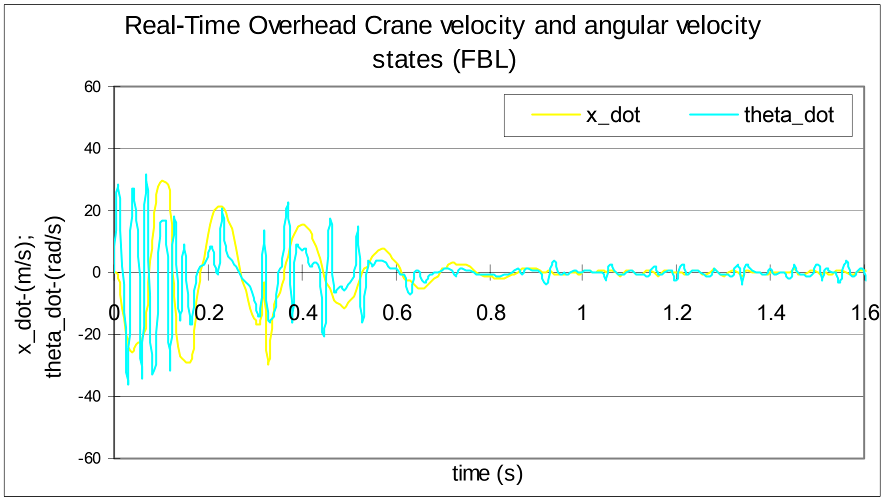


Figure 8.116: Overhead Crane Real-Time velocity and angular velocity state signals when the set point is 0.1 m

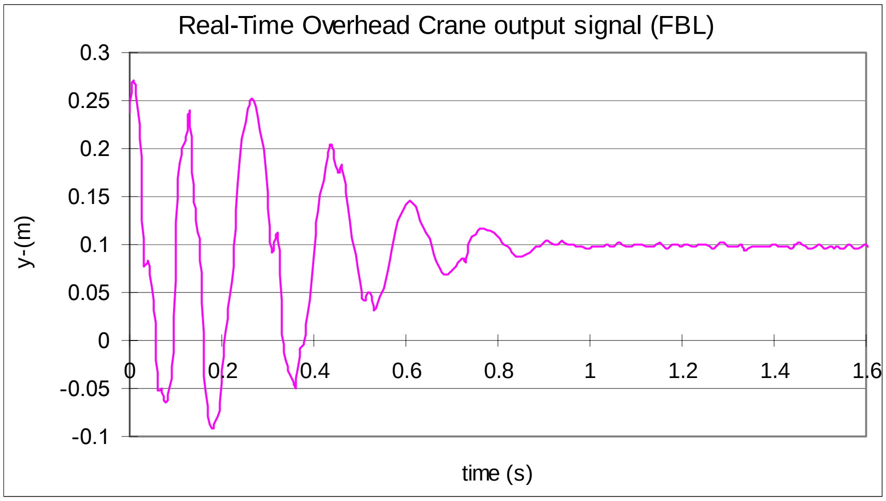


Figure 8.117: **Overhead Crane Real-Time output signal when the set point is 0.1 m**

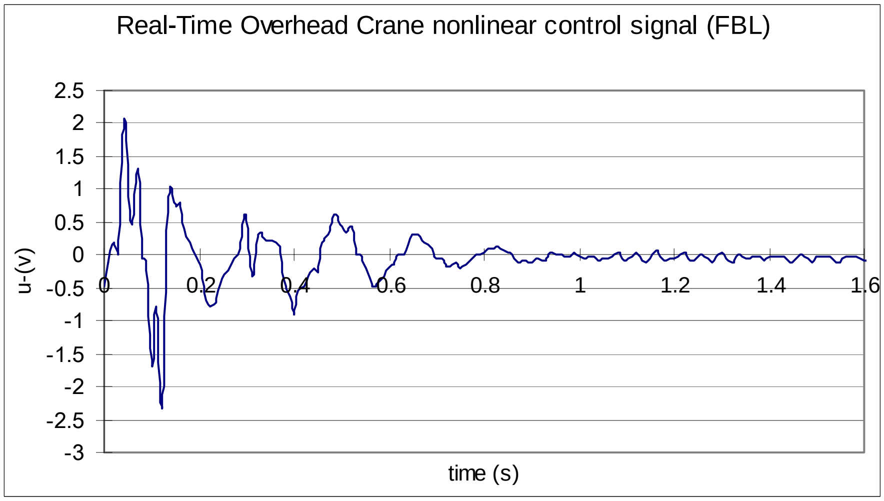


Figure 8.118: **Overhead Crane Real-Time nonlinear control signal when the set point is 0.1 m**

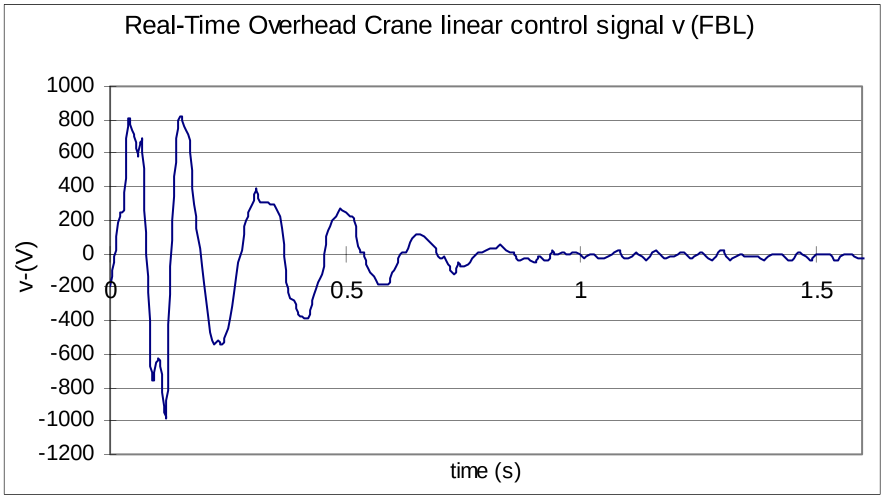


Figure 8.119: **Overhead Crane Real-Time linear control signal when the set point is 0.1 m**

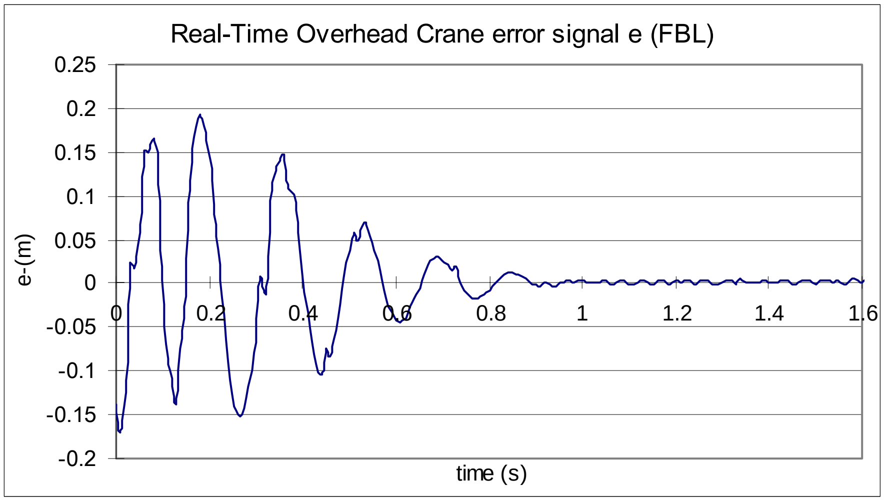


Figure 8.120: **Overhead Crane Real-Time error signal when the set point is 0.1 m**

On the NI CompactRIO Real-Time implementation platform, similar results are received as previously acquired data from the I/O DAQ device with signal conditioning and PC platform. The results acquired from the CompactRIO platform have much less noises. This is because of the structures of CompactRIO Real-Implementation platform. On the CompactRIO platform, the data are processed by the CompactRIO Real-Time processor according to the control algorithm which already is downloaded into the CompactRIO FPGA module. But for the I/O DAQ device with signal conditioning and PC platform, the signal acquired from the plant needs to be sent to the signal conditioning first, from the signal processing I/O units to the I/O DAQ device, then the data can be processed by the PC processor. Each time the data pass through the different cables and buses will absorb noises.

8.6 Conclusion

In this chapter, the Real-Time implementation platforms used in this project are introduced. There are two platforms including the I/O DAQ device with signal conditioning and PC platform and the NI CompactRIO platform. Referring to both Real-Time implementation platforms, both of the Lyapunov stability based MRC and the feedback linearization approach are implemented. The Real-Time implementation results are acquired, saved and plotted in MS-Excel.

Comparing the received results from these two Real-Time implementation platforms, they are similar. But from the system structure point of view, these two platforms are different from each other. In this project, the concept of the RMSs is presented. For the I/O DAQ device with signal conditioning and PC platform is mainly used for the prototyping in the laboratory. The platform is not suitable for the industrial environment. The CompactRIO platform is more reliable and it is built according to the industrial standards. Furthermore, the CompactRIO is a real reconfigurable plant in terms of rapidly design custom hardware; take advantage of customizable circuitry for IOCC and use off-the-shelf RIO measurement hardware for wide variety of signals. Till now, the project is successfully completed. In next chapter, the project is reviewed and concluded.

CHAPTER NINE
CONCLUSION AND FUTURE DIRECTION OF RESEARCH

9.1 Deliverables

The RMSs are become more and more important in the industry. They show the further development direction of the industry as a whole. For this project, a lab scale reconfigurable plant is used to represent the RMS. Based on the statement of the problem, the project aim and objectives are to develop the methodology for design and implementation of the reconfigurable nonlinear control systems in Real-Time based on embedded FPGA technology. According to the aim and objectives of the project, the following work has been done:

9.1.1 Mathematical model development and simulation

- The nonlinear mathematical model of the inverted pendulum is developed
- The nonlinear mathematical model of the inverted pendulum is simulated in Matlab/Simulink software environment
- The nonlinear mathematical model of the overhead crane is developed
- The nonlinear mathematical model of the overhead crane is simulated in Matlab/Simulink software environment
- The mathematical model of the trainer servo model is developed
- The complete mathematical model of the inverted pendulum incorporating the servo model is developed
- The complete mathematical model of the overhead crane incorporating the servo model is developed
- The linear mathematical model of the inverted pendulum is developed
- The linear mathematical model of the overhead crane is developed

9.1.2 Linear controller design and system simulation

- The linear controller for the inverted pendulum is designed based on the pole-placement method
- The linear controller for the overhead crane is designed based on the pole-placement method
- The closed-loop system which consists of the inverted pendulum and the pole-placement controller is simulated in Matlab/Simulink software environment for the output set points of 0 m, 0.05 m and 0.1 m
- The closed-loop system which consists of the overhead crane and the pole-placement controller is simulated in Matlab/Simulink software environment for the output set points of 0 m, 0.05 m and 0.1 m
- The linear controller for the inverted pendulum is designed based on the linear quadratic optimal control method
- The linear controller for the overhead crane is designed based on the linear quadratic optimal control method
- The closed-loop system which consists of the inverted pendulum and the linear quadratic controller is simulated in Matlab/Simulink software environment for the output set points of 0 m, 0.05 m and 0.1 m

- The closed-loop system which consists of the overhead crane and the linear quadratic controller is simulated in Matlab/Simulink software environment for set points of 0 m, 0.05 m and 0.1 m
- The simulation results are analyses and compared

9.1.3 Nonlinear controller design and closed-loop system simulation

9.1.3.1 Lyapunov stability based MRC controller design and system simulation

- The Lyapunov stability based MRC controller is designed for the inverted pendulum
- The closed-loop system which consists of the Lyapunov stability based MRC controller and the inverted pendulum is simulated in Matlab/Simulink for different output set points including 0 m, 0.05 m and 0.1 m
- The simulation results are acquired and plotted in MS-Excel for the inverted pendulum closed-loop behaviour for the output set points of 0 m, 0.05 m and 0.1 m
- The Lyapunov stability based MRC controller is designed for the overhead crane
- The closed-loop system which consists of the Lyapunov stability based MRC controller and the overhead crane is simulated in Matlab/Simulink for different output set points including 0 m, 0.05 m and 0.1 m
- The simulation data are acquired and plotted in MS-Excel for the overhead crane closed-loop behaviour for the output set points of 0 m, 0.05 m and 0.1 m
- The simulation results are analyses and compared

9.1.3.2 Feedback linearization method controller design and system simulation

- The feedback linearization method based controller is designed for the inverted pendulum
- The closed-loop system which consists of the feedback linearization method based controller and the inverted pendulum is simulated in Matlab/Simulink for different output set points including 0 m, 0.05 m and 0.1 m
- The simulation data are acquired and plotted in MS-Excel for the inverted pendulum system closed-loop behaviour
- The feedback linearization method based controller is designed for the overhead crane
- The closed-loop system which consists of the feedback linearization method based controller and the overhead crane is simulated in Matlab/Simulink for different output set points including 0 m, 0.05 m and 0.1 m
- The simulation data are acquired and plotted in MS-Excel for the overhead crane system closed-loop behaviour
- The simulation results are analyses and compared

9.1.4 Real-Time implementation

9.1.4.1 Real-Time implementation on I/O DAQ device, signal conditioning and PC platform

9.1.4.1.1 Real-Time Lyapunov stability based MRC control

- The Real-Time Simulink block diagram for the inverted pendulum is developed in Matlab/Simulink
- The Real-Time Simulink block diagram for the overhead crane is developed in Matlab/Simulink
- Real-Time implementation of the inverted pendulum control by using the Lyapunov stability based MRC controller of the set points of 0 m and 0.1 m is done
- The Real-Time implementation results are acquired and plotted in MS-Excel for the inverted pendulum system closed-loop behaviour

- Real-Time implementation for the overhead crane control by using the Lyapunov stability based MRC controller for the set points of 0 m and 0.1 m is done
- The Real-Time implementation results are acquired and plotted in MS-Excel for the overhead crane system closed-loop behaviour
- The Real-Time implementation results are analyses and compared

9.1.4.1.2 Real-Time feedback linearization based controller

- The Real-Time Simulink block diagram for the inverted pendulum is developed in Matlab/Simulink
- The Real-Time Simulink block diagram for the overhead crane is developed in Matlab/Simulink
- Real-Time implementation of the inverted pendulum control by using the feedback linearization based controller of the set points of 0 m and 0.1 m is done
- The Real-Time implementation results are acquired and plotted in MS-Excel for the inverted pendulum system closed-loop behaviour
- Real-Time implementation for the overhead crane control by using the feedback linearization based controller for the set points of 0 m and 0.1 m is done
- The Real-Time implementation results are acquired and plotted in MS-Excel for the overhead crane system closed-loop behaviour
- The Real-Time implementation results are analyses and compared

9.1.4.2 Real-Time implementation on the CompactRIO platform

9.1.4.2.1 Real-Time Lyapunov stability based MRC controller

- The Real-Time implementation host VI and FPGA VI are developed in LabVIEW by using the Lyapunov stability based MRC method
- Real-Time implementation for the inverted pendulum by using the Lyapunov stability based MRC controller for the output set points of 0 m and 0.1 m on the CompactRIO Real-Time implementation platform is done
- Real-Time implementation for the overhead crane by using the Lyapunov stability based MRC controller for the set points of 0 m and 0.1 m on the CompactRIO Real-Time implementation platform
- The Real-Time implementation results are acquired and plotted in MS-Excel for both of the inverted pendulum and the overhead crane

9.1.4.2.1 Real-Time feedback linearization based controller

- The Real-Time implementation host VI and FPGA VI is developed in LabVIEW by using the feedback linearization based method
- Real-Time implementation for the inverted pendulum by using the feedback linearization method based controller for the output set points of 0 m and 0.1 m on the CompactRIO Real-Time implementation platform is done
- Real-Time implementation for the overhead crane by using the feedback linearization method based controller for the output set points of 0 m and 0.1 m on the CompactRIO Real-Time implementation platform is done
- The Real-Time implementation results are acquired and plotted in MS-Excel for both of the inverted pendulum and the overhead crane
- The Real-Time implementation results on the CompactRIO platform are analyses and compared

9.1.5 Software programs for simulation and Real-Time Implementation

The methods for design described above are examined through simulation and Real-Time implementation using two software environments – Matlab and LabVIEW CompactRIO. Separate software programs are developed for every of the deliverables from 9.1.1 till 9.1.4. The developed software for simulation and Real-Time implementation are grouped in Table 9.1 and Table 9.2 respectively.

Table 9.1: **Table of programs used for simulation**

	Simulation									
	Model		**PP**		**OP**		**LDM**		**FBL**	
	m-file	**Simulink**	**m-file**	**Simulink**	**m-file**	**Simulink**	**m-file**	**Simulink**	**m-file**	**Simulink**
IP	model_test.m	model_test.mdl	PP_OP_IP.m	Linear_PP_OP.mdl	PP_OP_IP.m	Linear_PP_OP.mdl	LDM_IP.m	LDM_IP.mdl	Zero_dyn.m FBL_IP.m	FBL_IP.mdl
OC	model_test.m	model_test.mdl	PP_OP_OC.m	Linear_PP_OP.mdl	PP_OP_OC.m	Linear_PP_OP.mdl	LDM_OC.m	LDM_OC.mdl	Zero_dyn.m FBL_OC.m	FBL_IP.mdl

Table 9.2: **Table of programs used for Real-Time implementation**

	Real-Time implementation					
	LDM			**FBL**		
	m-file	**Simulink**	**LabVIEW**	**m-file**	**Simulink**	**LabVIEW**
IP	LDM_IP.m	RT_LDM_IP.mdl	LDM_IP_host.VI FPGA.VI	FBL_IP.m	RT_FBL_IP.mdl	FBL_IP_host.VI FPGA.VI
OC	LDM_OC.m	RT_LDM_OC.mdl	LDM_OC_host.VI FPGA.VI	FBL_OC.m	RT_FBL_OC.mdl	LDM_OC_host.VI FPGA.VI

Note:
PP – pole-placement method
LDM – Lyapunov Direct Method
IP – Inverted Pendulum
OP – Optimal control method
FBL – Feedback Linearization method
OC – Overhead Crane

For the entire project, based on the above deliverables a methodology for design and implement of the reconfigurable nonlinear control systems in Real-Time based on embedded FPGA technology is developed. The developed methodology can be used to guide the researcher and students into the field of the RMSs research and implementation.

For the detailed research outcomes, the complete linear and nonlinear mathematical models of the inverted pendulum and the overhead crane are derived. These derived mathematical models are complete and accurate. The linear and nonlinear controllers are derived based on different control strategies including the pole-placement method, linear quadratic method, Lyapunov stability based MRC method and the feedback linearization based method are developed. The simulation results show that the above mentioned controllers are designed successfully.

The Real-Time implementation results show that the project aim and objectives are achieved. On the DAQ with signal conditioning and PC platform, the software and hardware can be configured in the laboratory for prototype verification, but it is not suitable for the industry applications. On the CompactRIO platform, the different control algorithms can be deployed easily by simply download the new control algorithm to the CompactRIO. The CompactRIO is designed and manufactured according to the industry standards. It is perfect for the industry applications of RMSs.

9.2 Applications

The control of the inverted pendulum and the overhead crane are meaningful in the real world. The applications include in the military, outer space, industrial and education field.

9.2.1 Applications of the inverted pendulum

Military and outer space application: The inverted pendulum presents the same concept of how to control the rocket or the missile to follow the desired trajectories. The inverted pendulum is balanced by moving the cart, similarly by controlling the thrust angle at the bottom of the rocket, it does not fall over.

Education application: The control of the inverted pendulum is a classical problem in dynamics and control fields. It attracts the interests of the students towards the fields of control. The inverted pendulum is also a benchmark for testing control algorithms. The developed control algorithms in this project can be used for demonstration purpose or as examples for the students.

9.2.2 Applications of the overhead crane

The overhead cranes are well used in the workshops and harbours. Mainly they are used to move heavy loads. The essential idea is to suppress the oscillations while the goods are moving rapidly from one location to the other by using suitable controller.

9.3 Further work

For the further work, different control strategies are going to be implemented, for example: the fuzzy logic control, the sliding mode control, the backstepping method control and so on. Nonlinear control is one of the most important research areas. Based on the study of the existing control techniques, novel control strategy can be carried out.

9.4 Publications

1. Y. Han, R. Tzoneva and S. Behardien, 2007, “MATLAB, LabVIEW and FPGA Linear Control of an Inverted Pendulum”. Proceedings of Africon 2007; CD-ROM; Ref. No. 1-4244-0987-X/07/$25.00 ©2007 IEEE. IEEE Africon 2007 international conference, Windhoek, Namibia.
2. Han. Y., Tzoneva, R. 2008. “Reconfigurable Plant Control by Using Lyapunov Stability Theory based Model Reference Control Design”, submitted to SAIEE Research Journal.
3. Han. Y., Tzoneva, R. 2008. “Reconfigurable Plant Control by using Feedback Linearization Method with Optimal Control Method”, submitted to SAIEE Research Journal.

APPENDIX A

APPENDIX A

THE DEVELOPED MATLAB M-FILES

The m-files used in the thesis are presented in Appendix A. These m-files are associated with corresponding Simulink files. They are used to generate, calculate and provide the necessary parameter and coefficients for the Simulink files to run. The Simulink files include the pendulum mathematical models, the closed-loop system and the Real-Time implementation system.

A. 1: MATLAB script file – model_test.m

```
% M-file: model_test
%========================    M-FILE DESCRIPTION    =====================
% This m-file is used to provide the parameters for the Simulink file:
% model_test.
% The Simulink file: model_test contains the linear and nonlinear
% mathematical models of the inverted pendulum and the overhead crane
% Since the inverted pendulum and the overhead crane share the same
% parameters, this m-file is used for testing both systems.
% In this m-file, the necessary parameters are specified.
% For the linear model, A-the state matrix, B-the input matrix, C-the
% output matrix, D-the feedforward matrix are specified in the m-file.
% By using A, B, C, D the state space model of the inverted pendulum and
% the overhead crane are built
% For the nonlinear model x_dot=f(x)+g(x)u, the functions f(x) and g(x)
% are built in the Simulink file first as subsystems. The nonlinear
% model of the inverted pendulum and the overhead crane are built in
% Simulink by using these subsystems.

clear                   %Clear the workspace
clc                     %Clear the command window

J = 0.0013;             %Moment of inertia of the pendulum and rod
m = 0.285;              %Mass of the pendulum and the rod
M = 0.165;              %Mass of the cart
l = 0.22 ;              %Distance between the pivot and the pendulum and
                        %rod's centre of gravity
b = 0.1;                %Friction between the cart wheel and the surface
bt = 0.01;              %Friction at the pivot
g = 9.8;                %The gravity acceleration

% The inverted pendulum linearized model in the state-space form

sigma_l = (M+m)*(J+m*l^2)-m^2*l^2 % common denominator

% A_l, B_l, C_l and D_l represent the linear model



%
```

```
A_l_ip = [ 0           1                   0                   0;

        0    -(J+m*l^2)*b/sigma_l   -m^2*l^2*g/sigma_l    m*l*bt/sigma_l;

        0            0                     0                   1;

        0  (m*l*b)/sigma_l   ((M+m)*m*l*g)/sigma_l  (-(M+m)*bt)/sigma_l;

       ]

B_l_ip = [0; (J+m*l^2)/sigma_l; 0 ; -(l*m)/sigma_l]

C_l_ip = [1 0 l 0]

D_l_ip = 0

% Checking the controllability of the inverted pendulum
C_ip = ctrb(A_l_ip,B_l_ip)

% Checking the rank of the controllability matirx
R_ip = rank(C_ip)

% Linearized model of the overhead crane model in the state space form

% A_l_oc, B_l_oc, C_l_oc and D_l_oc represent the linear model of the
% overhead crane

A_l_oc = [ 0           1                      0                   0;

        0    -(J+m*l^2)*b/sigma_l    m^2*l^2*g/sigma_l    m*l*bt/sigma_l;

        0            0                     0                   1;

        0   (m*l*b)/sigma_l  -((M+m)*m*l*g)/sigma_l  (-(M+m)*bt)/sigma_l;

       ]

B_l_oc = [0; (J+m*l^2)/sigma_l; 0 ; -(l*m)/sigma_l]

C_l_oc = [1 0 l 0]

D_l_oc = 0

% Checking the controllability of the inverted pendulum
C_oc = ctrb(A_l_oc,B_l_oc)

% Checking the rank of the controllability matirx
R_oc = rank(C_oc)
```

A. 2: MATLAB script file – PP_OP_IP.m

```
% m-file: PP_OP_IP.m
%========================    M-FILE DESCRIPTION    =====================
% This m-file specifies and calculates all the parameters that are
% used for the linear control of the inverted pendulum (IP) by using
% the pole-placement method (PP) and the optimal control method (OP)
%
clc
clear
%=====================     PARAMETER DEFINITION    =====================
M = 0.165                % Mass of the cart; Unit is Kg.
m = 0.285                % Mass of the pendulum-bob and rod; Unit is Kg.
m1 = 0.22                % Mass of the bob; Unit is Kg.
m2 = 0.065               % Mass of the rod; Unit is Kg.
L = 0.22       % Length of the center gravity to the pivot; Unit is m.
r1 = 0.015               % Outside radius of the bob; Unit is m.
r2 = 0.003               % Internal radius of the bob; Unit is m.
J1 = (1/2)*m1*(r1^2+r2^2)% The bob moment of inertia; Unit is kg*m^2.
J2 = (1/3)*m2*L^2        % The rod moment of inertia; Unit is kg*m^2.
J = J1+J2                % Moment of inertia of the pendulum.
B = 0.1                  % Friction of the cart; Unit is N/m/sec.
BT = 0.01                % Friction of the rod;
g = 9.8
%======================================================================


%======================== Transfer Function ===========================
%Transfer function G1(s)=X(s)/F(s); G2(s)=theta(s)/F(s)
a0 = -m*L*g;
a1 = BT;
a2 = J+m*L^2;
b0 = 0;
b1 = B*m*L*g;
b2 = B*BT-(M+m)*m*L*g;
b3 = (M+m)*BT+(J+m*L^2)*B;
b4 = (J+m*L^2)*(M+m)-L^2*m^2;
c1 = -m*L;
G1 = tf([a2 a1 a0], [b4 b3 b2 b1 b0])   % X(s)/F(S)
G2 = tf([c1 0], [b4 b3 b2 b1])          % Theta(s)/F(s)
% Polynomial division (simplify the equation).
G11= deconv([b4 b3 b2 b1 b0], [a2 a1 a0])
G22= deconv([b4 b3 b2 b1 b0], [c1 0])
% Polynomail order reduce:
balred(G1, 3)
balred(G2, 2)

subplot(4,1,1)
rlocus (G1)
title('root locus plot x transfer function without load')

subplot(4,1,2)
rlocus (G2)
title('root locus plot theta transfer function without load')
%======================================================================
%=================== Linearized State-Space form ======================
% State vector identification: z=[x     x_dot   theta   theta_dot]'.
%                                 z1     z2      z3         z4
% For simplifying the state space representation, v1 & v2 are introduced:

v1=(M+m)/[J*(M+m)+M*m*L^2];
v2=(J+m*L^2)/[J*(M+m)+M*m*L^2];

A = [ 0        1                  0                         0;
      0      -B*v2    -[(m*L^2)*g*v2]/(J+L^2*m)   (m*L*BT*v2)/(J+L^2*m);
      0        0                   0                         1;
      0   (m*L*B*v1)/(m+M)     m*L*g*v1                 -BT*v1]

B = [ 0;
      v2;
      0;
-(L*m*v1)/(m+M)]

C = [1 0 0 0;
     0 0 1 0]

D =[0;
    0]
%=======================================================================



%========= Check the controllability matrix M of the system ============
M1 = [B A*B A^2*B A^3*B]
rank(M1)
%=======================================================================



%=====================Poles of the Transfer Function====================
pt=eig(A)
[num, den] = ss2tf(A, B, C, D)
h1 = tf([num(1:1, 1:4)], [den]);
h11=deconv([den], [num(1:1, 3:5)])
subplot(4,1,3)
rlocus (h1)
title('root locus plot x state space without load')
h2 = tf([num(2:2, 1:4)], [den]);
h22=deconv([den], [num(2:2, 3:3)])
subplot(4,1,4)
rlocus (h2)
title('root locus plot theta state space without load')
%=======================================================================



%======================== Pole Placement Method  =======================
% p is the desired poles
p = [-18+2.685j -18-2.685j -25 -26]
% computes a feedback gain matrix K that achieves the desired closed-
loop
% pole locations p
K = place(A,B,p)
%=======================================================================
phat = [-18+2.685j -18-2.685j -25 -26 -27]
```

```
Chat = [1 0 0 0]
Ahat = [A zeros(4,1); -Chat 0]
Bhat = [B;0]
Khat = place(Ahat, Bhat, phat)

%============================== Simulink ===============================
% Vector for initial conditions
z0=[0.3 0 0.2 0]
sim('poleplace_con_noload.mdl');
sim('poleplace_con_noload_ref.mdl');
%=======================================================================


%======================== Optimal Control Method =======================
% For no reference
Q = C'*C
Q1 = [50000     0     0     0;
          0   500     0     0;
          0     0   100     0;
          0     0     0    10]
R = 1
Ko = lqr(A, B, Q1, R)

% For reference
Q2 = [500000        0       0       0           0;
           0      500       0       0           0;
           0        0   10000       0           0;
           0        0       0     100           0;
           0        0       0       0      500000]
Kohat = lqr(Ahat, Bhat, Q2, R)
%=======================================================================


%=======================Optimal Control Simulation======================
sim('optimal_con_noload.mdl');
sim('optimal_con_noload_ref.mdl');
%=======================================================================
```

A. 3: MATLAB script file – PP_OP_OC.m

```
% m-file: PP_OP_OC.m
%========================    M-FILE DESCRIPTION    =====================

% This m-file specifies and calculates all the parameters that are
% used for the linear control of the overhead crane (OC) by using
% the pole-placement method (PP) and the optimal method (OP)
clc
clear
%=====================     PARAMETER DEFINE    ========================
M = 0.165               % Mass of the cart; Unit is Kg.
m = 0.285               % Mass of the pendulum-bob and rod; Unit is Kg.
m1 = 0.22               % Mass of the bob; Unit is Kg.
m2 = 0.065              % Mass of the rod; Unit is Kg.
L = 0.22        % Lenght of the center gravity to the pivot; Unit is m.
r1 = 0.015              % Outside radius of the bob; Unit is m.
r2 = 0.003              % Internal radius of the bob; Unit is m.
J1 = (1/2)*m1*(r1^2+r2^2)% The bob moment of inertia; Unit is kg*m^2.
J2 = (1/3)*m2*L^2       % The rod moment of inertia; Unit is kg*m^2.
J = J1+J2               % Moment of inertia of the pendulum.
B = 0.1                 % Friction of the cart; Unit is N/m/sec.
BT = 0.01               % Friction of the rod;
g = 9.8
%======================================================================



%======================== Transfer Function ===========================
%Transfer function G1(s)=X(s)/F(s); G2(s)=theta(s)/F(s)
a0 = -m*L*g;
a1 = BT;
a2 = J+m*L^2;
b0 = 0;
b1 = B*m*L*g;
b2 = B*BT-(M+m)*m*L*g;
b3 = (M+m)*BT+(J+m*L^2)*B;
b4 = (J+m*L^2)*(M+m)-L^2*m^2;
c1 = -m*L;
G1 = tf([a2 a1 a0], [b4 b3 b2 b1 b0])   % X(s)/F(S)
G2 = tf([c1 0], [b4 b3 b2 b1])          % Theta(s)/F(s)
% Polynomial division (simplify the equation).
G11= deconv([b4 b3 b2 b1 b0], [a2 a1 a0])
G22= deconv([b4 b3 b2 b1 b0], [c1 0])
% Polynomail order reduce:
balred(G1, 3)
balred(G2, 2)

subplot(4,1,1)
rlocus (G1)
title('root locus plot x transfer function without load')

subplot(4,1,2)
rlocus (G2)
title('root locus plot theta transfer function without load')
%======================================================================
%=================== Linearized State-Space form ======================
% State vector identification: z=[x     x_dot   theta   theta_dot]'.
%                                 z1     z2      z3          z4
% For simplifing the state space representation, v1 & v2 are introduced:

v1=(M+m)/[J*(M+m)+M*m*L^2];
v2=(J+m*L^2)/[J*(M+m)+M*m*L^2];

A = [ 0       1                   0                        0;
      0    -B*v2    +[(m*L^2)*g*v2]/(J+L^2*m)    (m*L*BT*v2)/(J+L^2*m);
      0       0                   0                        1;
      0  (m*L*B*v1)/(m+M)     -m*L*g*v1                 -BT*v1]

B = [ 0;
      v2;
      0;
-(L*m*v1)/(m+M)]

C = [1 0 0 0;
     0 0 1 0]

D =[0;
    0]
%=======================================================================


%========= Check the controllability matrix M of the system ============
M = [B A*B A^2*B A^3*B]
rank(M)
%=======================================================================


%=====================Poles of the Transfer Function====================
pt=eig(A)
[num, den] = ss2tf(A, B, C, D)
h1 = tf([num(1:1, 1:4)], [den]);
h11=deconv([den], [num(1:1, 3:5)])
subplot(4,1,3)
rlocus (h1)
title('root locus plot x state space without load')
h2 = tf([num(2:2, 1:4)], [den]);
h22=deconv([den], [num(2:2, 3:3)])
subplot(4,1,4)
rlocus (h2)
title('root locus plot theta state space without load')
%=======================================================================


%======================== Pole Placement Method  =======================
% p is the desired poles
p = [-18+2.685j -18-2.685j -25 -26]
% computes a feedback gain matrix K that achieves the desired closed-
loop
% pole locations p
K = place(A,B,p)
%=======================================================================
phat = [-18+2.685j -18-2.685j -25 -26 -27]
Chat = [1 0 0 0]
Ahat = [A zeros(4,1); -Chat 0]
Bhat = [B;0]
Khat = place(Ahat, Bhat, phat)

%============================== Simulink ===============================
% Vector for initial conditions
z0=[0.3 0 0.2 0]
sim('poleplace_con_noload.mdl');
sim('poleplace_con_noload_ref.mdl');
%=======================================================================


%======================== Optimal Control Method =======================
% For no reference
Q = C'*C
Q1 = [50000     0     0     0;
          0   500     0     0;
          0     0   100     0;
          0     0     0    10]
R = 1
Ko = lqr(A, B, Q1, R)

% For reference
Q2 = [500000        0       0       0           0;
           0      500       0       0           0;
           0        0   10000       0           0;
           0        0       0     100           0;
           0        0       0       0      500000]
Kohat = lqr(Ahat, Bhat, Q2, R)
%=======================================================================


%=======================Optimal Control Simulation======================
sim('optimal_con_noload.mdl');
sim('optimal_con_noload_ref.mdl');
%=======================================================================
```

A. 4: MATLAB script file – LDM_IP.m

```
% m-file: LDM_IP
%========================    M-FILE DESCRIPTION    =====================

% This m-file proved the parameters for Lyapunov direct method Simulink
% Block diagrams
% This is the m-file which is associated with the Simulink for the
% inverted pendulum simulation with Model reference control method based
% on the Lyapunov second method.
% In this file, only one output is considered, which is y=x+l*theta

%=======================================================================
clear
clc

% Parameters statement
n = 1.5;
J = 0.0013;
m = 0.285;
M = 0.165;
l = 0.22 ;
b = 0.1;
bt = 0.01;
g = 9.8;

M0 = m*l;          % coeff m*l
M1 = M+m;          % coeff M+m
M2 = M0*bt;        % coeff m*l*bt
M3 = M0*b;         % coeff m*l*b
M4 = M0^2;         % coeff m^2*L^2
M5 = M1*bt;        % coeff (M+m)*bt
M6 = M1*M0*g;      % coeff (M+m)*m*l*g
M7 = M4*g;         % coeff m^2*L^2*g

J1 = J + M0*l;     % coeff J+m*l^2
J2 = J1 * b;       % coeff (J+m*l^2)*b
J3 = J1 * M0;      % coeff (J+m*l^2)*m*l-

%=======================================================================

% The inverted pendulum linearized model in the state space form

sigma_l = (M+m)*(J+m*l^2)-m^2*l^2 % common denominator


% A_l, B_l, C_l and D_l represent the linear model
%
A_l = [ 0            1                      0                   0;

        0    -(J+m*l^2)*b/sigma_l   -m^2*l^2*g/sigma_l   m*l*bt/sigma_l;

        0            0                      0                   1;
        0    (m*l*b)/sigma_l  ((M+m)*m*l*g)/sigma_l  (-(M+m)*bt)/sigma_l;

       ]

B_l = [0; (J+m*l^2)/sigma_l; 0 ; -(l*m)/sigma_l]

C_l = [1 0 l 0]

D_l = 0

% form the linear model
sys_l = ss(A_l,B_l,C_l,D_l)

Cl=ctrb(A_l,B_l)

R_l = rank(Cl)

%=======================================================================

% Design the linear-quadratic (LQ) state-feedback regulator by using
% function lqr for the linear model in order to find the closed-loop
% poles Where Q is a positive-definite Hermitian or real symmetric
% matrix and R is a positive definite Hermitian or real symmetric matrix.

Q = [ 540    0     0    0   ;
       0     8     0    0   ;
       0     0    10    0   ;
       0     0     0    1   ;
     ]

R=1;

[K_l_hat, S_l_hat,E_l_hat]= lqr (A_l, B_l, Q, R)

%=======================================================================

% Form the desired model matrix A
% 4 by 4 matrix

A_d_n = A_l-B_l*K_l_hat

%=======================================================================

% Formation of a matrix A for closed-loop system
A_d_hat = [A_d_n zeros(4,1); -C_l 0]

% Calculation and formation of matrix B
V1 = M1/(J*M1+M*m*l^2);% M1 = M+m;   coeff M+m
V2 = J1/(J*M1+M*m*l^2);% J1 = J + M0*l;  coeff J+m*l^2
                       % M0 = m*l;   coeff m*l
b1s = V2
b2s = M0*V1/M1
B_d=[0; b1s; 0; -b2s]

B_d_hat = [B_d ; 0]
```

```
% Formation of a matrix C D
C_d = [1 0 l 0] % from C it can be seen that y = x + length*theta

D_d = 0

% Compute the controllability matrix
Co=ctrb(A_d_hat,B_d_hat)
% The system is controllable if Co has full rank n

% Check the rank of Co in order to find if the system is controllable
R_sys_d = rank(Co) % R_sys_d = 5 proves the system is controllable


% Form the desired system
sys_d = ss (A_d_n,B_d,C_d,D_d)

% Compute controllability and observability grammians
Wc = gram(sys_d,'c')
Wo = gram(sys_d,'o')

%=======================================================================

% Design linear-quadratic (LQ) state-feedback regulator for state-space
% systems Where Q is a positive-definite Hermitian or real symmetric
% matrix and R is a positive definite Hermitian or real symmetric matrix.

Q = [ 9000     0       0      0      0    ;
       0     100       0      0      0    ;
       0      0       1      0      0    ;
       0      0       0     10      0    ;
       0      0       0      0    700000 ]

R=1;

[Khat, Shat,Ehat]= lqr (A_d_hat, B_d_hat, Q, R)

%=======================================================================

% Lyapunov matrix

% set the positive definite matrix Q1

Q1 = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]

% By using Q to calculate P, where -Q1 = A'P+PA

syms P2

A_transp=A_d_n';

P2 = -Q1 /(A_transp + A_d_n)

% Check P2 whether positive definite
```

```
[R_P2,P_P2] = chol(P2)

[R_Q1,P_Q1] = chol(Q1)

% If X is not positive definite, an error message is printed

% Both P2 and Q1 are positive definite

%=======================================================================

% Call for the simulation

gain=1
sp=0
sim('Lyapunov_MRC_new_ip');
sim('Lyapunov_MRC_new_ip_new_V_dot');

%
%=======================================================================
%
% Convert the data from struct to cell array which is editable

% Ve_w ===> The linear control signal+++++++++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert; f for final

s_Ve_w = size(Ve_w); % Find the size of the signal [1 1 10001]
d_Ve_w = s_Ve_w(:,3:3); % Get the number of 10001

for n_Ve_w = 1:d_Ve_w;
Ve_w_c(n_Ve_w) = Ve_w(1, 1, n_Ve_w);
n_Ve_w = n_Ve_w+1;
end

Ve_w_f = (Ve_w_c)';

f_Ve_w = Ve_w_f;



%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% u_w ===> The nonlinear control signal+++++++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert; f for final

s_u_w = size(u_w); % Find the size of the signal [1 1 10001]
d_u_w = s_u_w(:,3:3); % Get the number of 10001

for n_u_w = 1:d_u_w;
u_w_c(n_u_w) = u_w(1, 1, n_u_w);
n_u_w = n_u_w+1;
end

u_w_f = (u_w_c)';

f_u_w = u_w_f;
```

```
%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% z_x ===> The state signal x
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_z_x_w = size(z_x_w); % Find the size of the signal [1 1 10001]
d_z_x_w = s_z_x_w(:,3:3); % Get the number of 10001 d for data

for n_z_x_w = 1:d_z_x_w;
z_x_w_c(n_z_x_w) = z_x_w(1, 1, n_z_x_w);
n_z_x_w = n_z_x_w+1;
end

z_x_w_f = (z_x_w_c)';

f_z_x_w = z_x_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% z_x_d ===> The state signal x dot
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_z_x_d_w = size(z_x_d_w); % Find the size of the signal [1 1 10001]
d_z_x_d_w = s_z_x_d_w(:,3:3); % Get the number of 10001 d for data

for n_z_x_d_w = 1:d_z_x_d_w;
z_x_d_w_c(n_z_x_d_w) = z_x_d_w(1, 1, n_z_x_d_w);
n_z_x_d_w = n_z_x_d_w+1;
end

z_x_d_w_f = (z_x_d_w_c)';

f_z_x_d_w = z_x_d_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% z_theta ===> The state signal theta
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_z_theta_w = size(z_theta_w); % Find the size of the signal [1 1 10001]
d_z_theta_w = s_z_theta_w(:,3:3); % Get the number of 10001 d for data

for n_z_theta_w = 1:d_z_theta_w;
z_theta_w_c(n_z_theta_w) = z_theta_w(1, 1, n_z_theta_w);
n_z_theta_w = n_z_theta_w+1;
end

z_theta_w_f = (z_theta_w_c)';

f_z_theta_w = z_theta_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% z_theta_d ===> The state signal theta dot
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_z_theta_d_w = size(z_theta_d_w); % Find the size of the signal [1 1
10001]
```

```
d_z_theta_d_w = s_z_theta_d_w(:,3:3); % Get the number of 10001 d for
data

for n_z_theta_d_w = 1:d_z_theta_d_w;
z_theta_d_w_c(n_z_theta_d_w) = z_theta_d_w(1, 1, n_z_theta_d_w);
n_z_theta_d_w = n_z_theta_d_w+1;
end

z_theta_d_w_f = (z_theta_d_w_c)';

f_z_theta_d_w = z_theta_d_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% y_n_w ===> The nonlinear output signal y_n
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_y_n_w = size(y_n_w); % Find the size of the signal [1 1 10001]
d_y_n_w = s_y_n_w(:,3:3); % Get the number of 10001 d for data

for n_y_n_w = 1:d_y_n_w;
y_n_w_c(n_y_n_w) = y_n_w(1, 1, n_y_n_w);
n_y_n_w = n_y_n_w+1;
end

y_n_w_f = (y_n_w_c)';

f_y_n_w = y_n_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% y_dsd_w ===> The linear output signal y_l
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_y_dsd_w = size(y_dsd_w); % Find the size of the signal [1 1 10001]
d_y_dsd_w = s_y_dsd_w(:,3:3); % Get the number of 10001 d for data

for n_y_dsd_w = 1:d_y_dsd_w;
y_dsd_w_c(n_y_dsd_w) = y_dsd_w(1, 1, n_y_dsd_w);
n_y_dsd_w = n_y_dsd_w+1;
end

y_dsd_w_f = (y_dsd_w_c)';

f_y_dsd_w = y_dsd_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% V_dot_w ===> The derivative of the Lyapunov function V dot
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_V_dot_w = size(V_dot_w); % Find the size of the signal [1 1 10001]
d_V_dot_w = s_V_dot_w(:,3:3); % Get the number of 10001 d for data

for n_V_dot_w = 1:d_V_dot_w;
V_dot_w_c(n_V_dot_w) = V_dot_w(1, 1, n_V_dot_w);
n_V_dot_w = n_V_dot_w+1;
end
```

```
V_dot_w_f = (V_dot_w_c)';

f_V_dot_w = V_dot_w_f;


%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% V_dot_p2_M_w ===> The 2nd part of the V dot signal M
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_V_dot_p2_M_w = size(V_dot_p2_M_w); % Find the size of the signal
d_V_dot_p2_M_w = s_V_dot_p2_M_w(:,3:3);

for n_V_dot_p2_M_w = 1:d_V_dot_p2_M_w;
V_dot_p2_M_w_c(n_V_dot_p2_M_w) = V_dot_p2_M_w(1, 1, n_V_dot_p2_M_w);
n_V_dot_p2_M_w = n_V_dot_p2_M_w+1;
end

V_dot_p2_M_w_f = (V_dot_p2_M_w_c)';

f_V_dot_p2_M_w = V_dot_p2_M_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% x_dsd_w ===> The desired states x
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_x_dsd_w = size(x_dsd_w); % Find the size of the signal
d_x_dsd_w = s_x_dsd_w(:,3:3);

for n_x_dsd_w = 1:d_x_dsd_w;
x_dsd_w_c(n_x_dsd_w) = x_dsd_w(1, 1, n_x_dsd_w);
n_x_dsd_w = n_x_dsd_w+1;
end

x_dsd_w_f = (x_dsd_w_c)';

f_x_dsd_w = x_dsd_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% x_dsd_dot ===> The desired state x dot
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_x_dsd_dot_w = size(x_dsd_dot_w); % Find the size of the signal
d_x_dsd_dot_w = s_x_dsd_dot_w(:,3:3);

for n_x_dsd_dot_w = 1:d_x_dsd_dot_w;
x_dsd_dot_w_c(n_x_dsd_dot_w) = x_dsd_dot_w(1, 1, n_x_dsd_dot_w);
n_x_dsd_dot_w = n_x_dsd_dot_w+1;
end

x_dsd_dot_w_f = (x_dsd_dot_w_c)';

f_x_dsd_dot_w = x_dsd_dot_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% theta_dsd_w ===> The desired state theta
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_theta_dsd_w = size(theta_dsd_w); % Find the size of the signal
d_theta_dsd_w = s_theta_dsd_w(:,3:3);

for n_theta_dsd_w = 1:d_theta_dsd_w;
theta_dsd_w_c(n_theta_dsd_w) = theta_dsd_w(1, 1, n_theta_dsd_w);
n_theta_dsd_w = n_theta_dsd_w+1;
end

theta_dsd_w_f = (theta_dsd_w_c)';

f_theta_dsd_w = theta_dsd_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% theta_dsd_dot_w ===> The desired state theta dot
% w for workspace;  s for size;  d for data;  c for convert; f for final
s_theta_dsd_dot_w = size(theta_dsd_dot_w); % Find the size of the signal
d_theta_dsd_dot_w = s_theta_dsd_dot_w(:,3:3);

for n_theta_dsd_dot_w = 1:d_theta_dsd_dot_w;
theta_dsd_dot_w_c(n_theta_dsd_dot_w) = ...
    theta_dsd_dot_w(1, 1, n_theta_dsd_dot_w);
n_theta_dsd_dot_w = n_theta_dsd_dot_w+1;
end

theta_dsd_dot_w_f = (theta_dsd_dot_w_c)';

f_theta_dsd_dot_w = theta_dsd_dot_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_y_w ===> The output y error signal +++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert; f for final

s_e_y_w = size(e_y_w); % Find the size of the signal [1 1 10001]
d_e_y_w = s_e_y_w(:,3:3); % Get the number of 10001

for n_e_y_w = 1:d_e_y_w;
e_y_w_c(n_e_y_w) = e_y_w(1, 1, n_e_y_w);
n_e_y_w = n_e_y_w+1;
end

e_y_w_f = (e_y_w_c)';

f_e_y_w = e_y_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_x_w ===> The state x error signal++++++++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert; f for final

s_e_x_w = size(e_x_w); % Find the size of the signal [1 1 10001]
d_e_x_w = s_e_x_w(:,3:3); % Get the number of 10001

for n_e_x_w = 1:d_e_x_w;
```

```
e_x_w_c(n_e_x_w) = e_x_w(1, 1, n_e_x_w);
n_e_x_w = n_e_x_w+1;
end

e_x_w_f = (e_x_w_c)';

f_e_x_w = e_x_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_x_d_w ===> The state x dot error signal+++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert; f for final

s_e_x_d_w = size(e_x_d_w); % Find the size of the signal [1 1 10001]
d_e_x_d_w = s_e_x_d_w(:,3:3); % Get the number of 10001

for n_e_x_d_w = 1:d_e_x_d_w;
e_x_d_w_c(n_e_x_d_w) = e_x_d_w(1, 1, n_e_x_d_w);
n_e_x_d_w = n_e_x_d_w+1;
end

e_x_d_w_f = (e_x_d_w_c)';

f_e_x_d_w = e_x_d_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_theta_w ===> The state theta error signal+++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert; f for final

s_e_theta_w = size(e_theta_w); % Find the size of the signal [1 1 10001]
d_e_theta_w = s_e_theta_w(:,3:3); % Get the number of 10001

for n_e_theta_w = 1:d_e_theta_w;
e_theta_w_c(n_e_theta_w) = e_theta_w(1, 1, n_e_theta_w);
n_e_theta_w = n_e_theta_w+1;
end

e_theta_w_f = (e_theta_w_c)';

f_e_theta_w = e_theta_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_theta_d_w ===> The state theta dot error signal+++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert; f for final

s_e_theta_d_w = size(e_theta_d_w); % Find the size of the signal [1 1
10001]
d_e_theta_d_w = s_e_theta_d_w(:,3:3); % Get the number of 10001

for n_e_theta_d_w = 1:d_e_theta_d_w;
e_theta_d_w_c(n_e_theta_d_w) = e_theta_d_w(1, 1, n_e_theta_d_w);
n_e_theta_d_w = n_e_theta_d_w+1;
end
```

```
e_theta_d_w_f = (e_theta_d_w_c)';

f_e_theta_d_w = e_theta_d_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


% The End
```

A. 5: MATLAB script file – LDM_OC.m

```
% m-file: LDM_OC.m
%========================    M-FILE DESCRIPTION    =====================

% This m-file proved the parameters for Lyapunov Direct Method (LDM)
% Simulink Block diagrams of Overhead Crane (OC)
% This is the m-file which associated with the Simulink for the inverted
% pendulum simulation with Model reference control method based on the
% Lyapunov second method.
% In this file, only one output is considered which is y=x+l*theta

%=======================================================================
clear
clc
% Parameters statement
n = 1.5;
J = 0.0013;
m = 0.285;
M = 0.165;
l = 0.22 ;
b = 0.1;
bt = 0.01;
g = 9.8;

M0 = m*l;          % coeff m*l
M1 = M+m;          % coeff M+m
M2 = M0*bt;        % coeff m*l*bt
M3 = M0*b;         % coeff m*l*b
M4 = M0^2;         % coeff m^2*L^2
M5 = M1*bt;        % coeff (M+m)*bt
M6 = M1*M0*g;      % coeff (M+m)*m*l*g
M7 = M4*g;         % coeff m^2*L^2*g

J1 = J + M0*l;     % coeff J+m*l^2
J2 = J1 * b;       % coeff (J+m*l^2)*b
J3 = J1 * M0;      % coeff (J+m*l^2)*m*l-

%=======================================================================

% The inverted pendulum Linearized model in the state space form

sigma_l = (M+m)*(J+m*l^2)-m^2*l^2 % common denominator

% A_l, B_l, C_l and D_l represent the linear model
%
A_l = [ 0            1                      0                   0;

        0    -(J+m*l^2)*b/sigma_l    m^2*l^2*g/sigma_l    m*l*bt/sigma_l;

        0            0                      0                   1;

        0   (m*l*b)/sigma_l  -((M+m)*m*l*g)/sigma_l  (-(M+m)*bt)/sigma_l;

       ]
```

```
B_l = [0; (J+m*l^2)/sigma_l; 0 ; -(l*m)/sigma_l]

C_l = [1 0 l 0]

D_l = 0

% form the linear model
sys_l = ss(A_l,B_l,C_l,D_l)

Cl=ctrb(A_l,B_l)
R_l = rank(Cl)

%========================================================================

% Design the linear-quadratic (LQ) state-feedback regulator by using
% function lqr for the linear model in order to find the closed-loop
% poles Where Q is a positive-definite Hermitian or real symmetric
% matrix and R is a positive definite Hermitian or real symmetric matrix.

Q = [ 10     0     0    0   ;
       0     10    0    0   ;
       0     0     1    0   ;
       0     0     0    1   ;
     ]
R=1;

[K_l_hat, S_l_hat,E_l_hat]= lqr (A_l, B_l, Q, R)

%========================================================================

% Form the desired model matrix A
% 4 by 4 matrix

A_d_n = A_l-B_l*K_l_hat

%========================================================================

% Formation of a matrix A for closed-loop system
A_d_hat = [A_d_n zeros(4,1); -C_l 0]

% Calculation and formation of matrix B
V1 = M1/(J*M1+M*m*l^2);% M1 = M+m;   coeff M+m
V2 = J1/(J*M1+M*m*l^2);% J1 = J + M0*l;  coeff J+m*l^2
                       % M0 = m*l;   coeff m*l
b1s = V2
b2s = M0*V1/M1
B_d=[0; b1s; 0; -b2s]

B_d_hat = [B_d ; 0]

% Formation of a matrix C D
C_d = [1 0 l 0] % from C it can be seen that y = x + length*theta

D_d = 0

% Compute the controllability matrix
```

```
Co=ctrb(A_d_hat,B_d_hat)
% The system is controllable if Co has full rank n

% Check the rank of Co in order to find if the syste mis controllable
R_sys_d = rank(Co) % R_sys_d = 5 proves the system is controllable


% Form the desired system
sys_d = ss (A_d_n,B_d,C_d,D_d)

% Compute controllability and observability grammians
Wc = gram(sys_d,'c')
Wo = gram(sys_d,'o')

%=======================================================================

% Design linear-quadratic (LQ) state-feedback regulator for state-space
% systems Where Q is a positive-definite Hermitian or real symmetric
% matrix and R is a positive definite Hermitian or real symmetric matrix.

Q = [ 45       0       0      0      0    ;
       0       1       0      0      0    ;
       0       0      120     0      0    ;
       0       0       0      1      0    ;
       0       0       0      0    3000000  ]
R=1;

[Khat, Shat,Ehat]= lqr (A_d_hat, B_d_hat, Q, R)

%=======================================================================

% Lyapunov matrix

% set the positive definite matrix Q1

Q1 = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]

% By using Q to calculate P, where -Q1 = A'P+PA

syms P2

A_transp=A_d_n';

P2 = -Q1 /(A_transp + A_d_n)

% Check P2 whether positive definite

[R_P2,P_P2] = chol(P2)

[R_Q1,P_Q1] = chol(Q1)

% If X is not positive definite, an error message is printed

% Both P2 and Q1 are positive definite
```

```
%=======================================================================

% Call for the simulation

gain=1
sp=0
sim('Lyapunov_MRC_new_oc_new_V_dot');

% %=====================================================================
%
% Convert the data from structure to cell array which is editable

% Ve_w ===> The linear control signal+++++++++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert;  f for
final

s_Ve_w = size(Ve_w); % Find the size of the signal [1 1 10001]
d_Ve_w = s_Ve_w(:,3:3); % Get the number of 10001

for n_Ve_w = 1:d_Ve_w;
Ve_w_c(n_Ve_w) = Ve_w(1, 1, n_Ve_w);
n_Ve_w = n_Ve_w+1;
end

Ve_w_f = (Ve_w_c)';

f_Ve_w = Ve_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% u_w ===> The nonlinear control signal+++++++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert;  f for
final

s_u_w = size(u_w); % Find the size of the signal [1 1 10001]
d_u_w = s_u_w(:,3:3); % Get the number of 10001

for n_u_w = 1:d_u_w;
u_w_c(n_u_w) = u_w(1, 1, n_u_w);
n_u_w = n_u_w+1;
end

u_w_f = (u_w_c)';

f_u_w = u_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% z_x ===> The state signal x
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_z_x_w = size(z_x_w); % Find the size of the signal [1 1 10001]
d_z_x_w = s_z_x_w(:,3:3); % Get the number of 10001 d for data

for n_z_x_w = 1:d_z_x_w;
z_x_w_c(n_z_x_w) = z_x_w(1, 1, n_z_x_w);
```

```
n_z_x_w = n_z_x_w+1;
end

z_x_w_f = (z_x_w_c)';

f_z_x_w = z_x_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% z_x_d ===> The state signal x dot
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_z_x_d_w = size(z_x_d_w); % Find the size of the signal [1 1 10001]
d_z_x_d_w = s_z_x_d_w(:,3:3); % Get the number of 10001 d for data

for n_z_x_d_w = 1:d_z_x_d_w;
z_x_d_w_c(n_z_x_d_w) = z_x_d_w(1, 1, n_z_x_d_w);
n_z_x_d_w = n_z_x_d_w+1;
end

z_x_d_w_f = (z_x_d_w_c)';

f_z_x_d_w = z_x_d_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% z_theta ===> The state signal theta
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_z_theta_w = size(z_theta_w); % Find the size of the signal [1 1 10001]
d_z_theta_w = s_z_theta_w(:,3:3); % Get the number of 10001 d for data

for n_z_theta_w = 1:d_z_theta_w;
z_theta_w_c(n_z_theta_w) = z_theta_w(1, 1, n_z_theta_w);
n_z_theta_w = n_z_theta_w+1;
end

z_theta_w_f = (z_theta_w_c)';

f_z_theta_w = z_theta_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% z_theta_d ===> The state signal theta dot
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_z_theta_d_w = size(z_theta_d_w); % Find the size of the signal [1 1
10001]
d_z_theta_d_w = s_z_theta_d_w(:,3:3); % Get the number of 10001 d for
data

for n_z_theta_d_w = 1:d_z_theta_d_w;
z_theta_d_w_c(n_z_theta_d_w) = z_theta_d_w(1, 1, n_z_theta_d_w);
n_z_theta_d_w = n_z_theta_d_w+1;
end
```

```
z_theta_d_w_f = (z_theta_d_w_c)';

f_z_theta_d_w = z_theta_d_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% y_n_w ===> The nonlinear output signal y_n
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_y_n_w = size(y_n_w); % Find the size of the signal [1 1 10001]
d_y_n_w = s_y_n_w(:,3:3); % Get the number of 10001 d for data

for n_y_n_w = 1:d_y_n_w;
y_n_w_c(n_y_n_w) = y_n_w(1, 1, n_y_n_w);
n_y_n_w = n_y_n_w+1;
end

y_n_w_f = (y_n_w_c)';

f_y_n_w = y_n_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% y_dsd_w ===> The linear output signal y_l
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_y_dsd_w = size(y_dsd_w); % Find the size of the signal [1 1 10001]
d_y_dsd_w = s_y_dsd_w(:,3:3); % Get the number of 10001 d for data

for n_y_dsd_w = 1:d_y_dsd_w;
y_dsd_w_c(n_y_dsd_w) = y_dsd_w(1, 1, n_y_dsd_w);
n_y_dsd_w = n_y_dsd_w+1;
end

y_dsd_w_f = (y_dsd_w_c)';

f_y_dsd_w = y_dsd_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% V_dot_w ===> The derivative of the Lyapunov function V dot
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_V_dot_w = size(V_dot_w); % Find the size of the signal [1 1 10001]
d_V_dot_w = s_V_dot_w(:,3:3); % Get the number of 10001 d for data

for n_V_dot_w = 1:d_V_dot_w;
V_dot_w_c(n_V_dot_w) = V_dot_w(1, 1, n_V_dot_w);
n_V_dot_w = n_V_dot_w+1;
end

V_dot_w_f = (V_dot_w_c)';

f_V_dot_w = V_dot_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% V_dot_p2_M_w ===> The 2nd part of the V dot signal M
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_V_dot_p2_M_w = size(V_dot_p2_M_w); % Find the size of the signal
d_V_dot_p2_M_w = s_V_dot_p2_M_w(:,3:3);

for n_V_dot_p2_M_w = 1:d_V_dot_p2_M_w;
V_dot_p2_M_w_c(n_V_dot_p2_M_w) = V_dot_p2_M_w(1, 1, n_V_dot_p2_M_w);
n_V_dot_p2_M_w = n_V_dot_p2_M_w+1;
end

V_dot_p2_M_w_f = (V_dot_p2_M_w_c)';

f_V_dot_p2_M_w = V_dot_p2_M_w_f;

%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% x_dsd_w ===> The desired states x
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_x_dsd_w = size(x_dsd_w); % Find the size of the signal
d_x_dsd_w = s_x_dsd_w(:,3:3);

for n_x_dsd_w = 1:d_x_dsd_w;
x_dsd_w_c(n_x_dsd_w) = x_dsd_w(1, 1, n_x_dsd_w);
n_x_dsd_w = n_x_dsd_w+1;
end

x_dsd_w_f = (x_dsd_w_c)';

f_x_dsd_w = x_dsd_w_f;

%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% x_dsd_dot ===> The desired state x dot
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_x_dsd_dot_w = size(x_dsd_dot_w); % Find the size of the signal
d_x_dsd_dot_w = s_x_dsd_dot_w(:,3:3);

for n_x_dsd_dot_w = 1:d_x_dsd_dot_w;
x_dsd_dot_w_c(n_x_dsd_dot_w) = x_dsd_dot_w(1, 1, n_x_dsd_dot_w);
n_x_dsd_dot_w = n_x_dsd_dot_w+1;
end

x_dsd_dot_w_f = (x_dsd_dot_w_c)';

f_x_dsd_dot_w = x_dsd_dot_w_f;

%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% theta_dsd_w ===> The desired state theta
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_theta_dsd_w = size(theta_dsd_w); % Find the size of the signal
```

```
d_theta_dsd_w = s_theta_dsd_w(:,3:3);

for n_theta_dsd_w = 1:d_theta_dsd_w;
theta_dsd_w_c(n_theta_dsd_w) = theta_dsd_w(1, 1, n_theta_dsd_w);
n_theta_dsd_w = n_theta_dsd_w+1;
end

theta_dsd_w_f = (theta_dsd_w_c)';

f_theta_dsd_w = theta_dsd_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% theta_dsd_dot_w ===> The desired state theta dot
% w for workspace;  s for size;  d for data;  c for convert;  f for
final
s_theta_dsd_dot_w = size(theta_dsd_dot_w); % Find the size of the signal
d_theta_dsd_dot_w = s_theta_dsd_dot_w(:,3:3);

for n_theta_dsd_dot_w = 1:d_theta_dsd_dot_w;
theta_dsd_dot_w_c(n_theta_dsd_dot_w) = ...
    theta_dsd_dot_w(1, 1, n_theta_dsd_dot_w);
n_theta_dsd_dot_w = n_theta_dsd_dot_w+1;
end

theta_dsd_dot_w_f = (theta_dsd_dot_w_c)';

f_theta_dsd_dot_w = theta_dsd_dot_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_y_w ===> The the output y error signal +++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert;  f for
final

s_e_y_w = size(e_y_w); % Find the size of the signal [1 1 10001]
d_e_y_w = s_e_y_w(:,3:3); % Get the number of 10001

for n_e_y_w = 1:d_e_y_w;
e_y_w_c(n_e_y_w) = e_y_w(1, 1, n_e_y_w);
n_e_y_w = n_e_y_w+1;
end

e_y_w_f = (e_y_w_c)';

f_e_y_w = e_y_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_x_w ===> The state x error signa++++++++++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert;  f for
final

s_e_x_w = size(e_x_w); % Find the size of the signal [1 1 10001]
d_e_x_w = s_e_x_w(:,3:3); % Get the number of 10001
```

```
for n_e_x_w = 1:d_e_x_w;
e_x_w_c(n_e_x_w) = e_x_w(1, 1, n_e_x_w);
n_e_x_w = n_e_x_w+1;
end

e_x_w_f = (e_x_w_c)';

f_e_x_w = e_x_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_x_d_w ===> The state x dot error signal+++++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert;  f for
final

s_e_x_d_w = size(e_x_d_w); % Find the size of the signal [1 1 10001]
d_e_x_d_w = s_e_x_d_w(:,3:3); % Get the number of 10001

for n_e_x_d_w = 1:d_e_x_d_w;
e_x_d_w_c(n_e_x_d_w) = e_x_d_w(1, 1, n_e_x_d_w);
n_e_x_d_w = n_e_x_d_w+1;
end

e_x_d_w_f = (e_x_d_w_c)';

f_e_x_d_w = e_x_d_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_theta_w ===> The state theta error signal+++++++++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert;  f for
final

s_e_theta_w = size(e_theta_w); % Find the size of the signal [1 1 10001]
d_e_theta_w = s_e_theta_w(:,3:3); % Get the number of 10001

for n_e_theta_w = 1:d_e_theta_w;
e_theta_w_c(n_e_theta_w) = e_theta_w(1, 1, n_e_theta_w);
n_e_theta_w = n_e_theta_w+1;
end

e_theta_w_f = (e_theta_w_c)';

f_e_theta_w = e_theta_w_f;

%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% e_theta_d_w ===> The state theta dot error signal+++++++++++++++++++++
% w for workspace;  s for size;  d for data;  c for convert;  f for
final

s_e_theta_d_w = size(e_theta_d_w); % Find the size of the signal [1 1
10001]
d_e_theta_d_w = s_e_theta_d_w(:,3:3); % Get the number of 10001

for n_e_theta_d_w = 1:d_e_theta_d_w;
```

```
e_theta_d_w_c(n_e_theta_d_w) = e_theta_d_w(1, 1, n_e_theta_d_w);
n_e_theta_d_w = n_e_theta_d_w+1;
end

e_theta_d_w_f = (e_theta_d_w_c)';

f_e_theta_d_w = e_theta_d_w_f;

%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

% The End
```

A. 6: MATLAB script file – zero_dyn.m

```
% m-file: zero_dyn.m
%=======================================================================
% The Zero dynamic checking for the linearized system.
% This m-file is used for specifing the parameters that are used in the
% Simulink block diagram which is checking the system stability
n = 1.5;
J = 0.0013;
m = 0.285;
M = 0.165;
l = 0.22 ;
b = 0.1;
bt = 0.01;
g = 9.8;
```

A. 7: MATLAB script file – FBL_IP.m

```
% m-file: FBL_IP
%========================    M-FILE DESCRIPTION    =====================

% This m-file provides all the parameters for the Simulink block diagram
% of simulating the closed-loop system based on the feedback
% linearization method for the inverted pendulum.
% In this file, only one output is considered which y=x+l*theta

%=======================================================================
clear
clc
% Parameters statement
n = 1.5;
J = 0.0013;
m = 0.285;
M = 0.165;
l = 0.22 ;
b = 0.1;
bt = 0.01;
g = 9.8;

M0 = m*l;          % coeff m*l
M1 = M+m;          % coeff M+m
M2 = M0*bt;        % coeff m*l*bt
M3 = M0*b;         % coeff m*l*b
M4 = M0^2;         % coeff m^2*L^2
M5 = M1*bt;        % coeff (M+m)*bt
M6 = M1*M0*g;      % coeff (M+m)*m*l*g
M7 = M4*g;         % coeff m^2*L^2*g

J1 = J + M0*l;     % coeff J+m*l^2
J2 = J1 * b;       % coeff (J+m*l^2)*b
J3 = J1 * M0;      % coeff (J+m*l^2)*m*l-

%=======================================================================

% Linearized model of the inverted pendulum model in the state space
form

sigma_l = (M+m)*(J+m*l^2)-m^2*l^2 % common denominator

% A_l, B_l, C_l and D_l represent the linear model
%
A_l = [

        0            1            0           0;

        0            0       -m*g*l^2/J    l*bt/J;

        0            0            0           1;

        0            0          m*g*l/J     -bt/J;

       ]
```

```
B_l = [

                0;

        (J+m*l^2)/sigma_l;

                0;

         -(l*m)/sigma_l

      ]

C_l = [1 0 l 0]

D_l = 0

% form the linear model
sys_l = ss(A_l,B_l,C_l,D_l)

%========================================================================

% Design the linear-quadratic (LQ) state-feedback regulator by using
% function lqr for the linear model in order to find the closed-loop
poles
% Where Q is a positive-definite Hermitian or real symmetric matrix and
R
% is a positive definite Hermitian or real symmetric matrix.

Q = [

      530    0     0    0   ;
       0     8     0    0   ;
       0     0     1    0   ;
       0     0     0    1   ;

     ]
R=1;

[K_l_hat, S_l_hat,E_l_hat]= lqr (A_l, B_l, Q, R)

%========================================================================

% Form the desired model matrix A

A_d_n = A_l-B_l*K_l_hat % 4 by 4 matrix

%========================================================================

% Formation of a matrix A for closed-loop system
A_d_hat = [A_d_n zeros(4,1); -C_l 0]

% Calculation and formation of matrix B
V1 = M1/(J*M1+M*m*l^2);% M1 = M+m;   coeff M+m
V2 = J1/(J*M1+M*m*l^2);% J1 = J + M0*l;  coeff J+m*l^2
```

```
                       % M0 = m*l;   coeff m*l
b1s = V2
b2s = M0*V1/M1
B_d=[0; b1s; 0; -b2s]

B_d_hat = [B_d ; 0]

% Formation of a matrix C D
C_d = [1 0 l 0] % from C it can be seen that y = x + length*theta

D_d = 0

% Compute the controllability matrix
Co=ctrb(A_d_hat,B_d_hat)
% The system is controllable if Co has full rank n

% Check the rank of Co in order to find if the syste mis controllable
R_sys_d = rank(Co)
% R_sys_d = 5 proves the system is controllable


% Form the desired system
sys_d = ss (A_d_n,B_d,C_d,D_d)

% Compute controllability and observability grammians
Wc = gram(sys_d,'c')
Wo = gram(sys_d,'o')

%========================================================================

% Design linear-quadratic (LQ) state-feedback regulator for state-space
% systems
% Where Q is a positive-definite Hermitian or real symmetric matrix and
R
% is a positive definite Hermitian or real symmetric matrix.

Q = [ 9000    0        0     0    0    ;
       0     130      0     0    0    ;
       0       0      10    0    0    ;
       0       0       0    3   0    ;
       0       0       0     0   220000  ]
R=1;

[Khat, Shat,Ehat]= lqr (A_d_hat, B_d_hat, Q, R)

%========================================================================

% Lyapunov matrix

% set the positive definite matrix Q1

Q1 = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]

% By using Q to calculate P, where -Q1 = A'P+PA

syms P2
```

```
A_transp=A_d_n';

P2 = -Q1 /(A_transp + A_d_n)

% Check P2 whether positive definite

[R_P2,P_P2] = chol(P2)

[R_Q1,P_Q1] = chol(Q1)

% If X is not positive definite, an error message is printed

% Both P2 and Q1 are positive definite

 %======================================================================
%The End
```

A. 8: MATLAB script file – FBL_OC.m

```
% m-file: FBL_OC
%========================    M-FILE DESCRIPTION    =====================

% This is the m-file which associated with the Simulink for the inverted
% pendulum simulation with Model reference control method based on the
% Lyapunov second method.
% In this file, only one output is considered which y=x+l*theta

%=======================================================================
clear
clc
% Parameters statement
n = 1.5;
J = 0.0013;
m = 0.285;
M = 0.165;
l = 0.22 ;
b = 0.1;
bt = 0.01;
g = 9.8;

M0 = m*l;          % coeff m*l
M1 = M+m;          % coeff M+m
M2 = M0*bt;        % coeff m*l*bt
M3 = M0*b;         % coeff m*l*b
M4 = M0^2;         % coeff m^2*L^2
M5 = M1*bt;        % coeff (M+m)*bt
M6 = M1*M0*g;      % coeff (M+m)*m*l*g
M7 = M4*g;         % coeff m^2*L^2*g

J1 = J + M0*l;     % coeff J+m*l^2
J2 = J1 * b;       % coeff (J+m*l^2)*b
J3 = J1 * M0;      % coeff (J+m*l^2)*m*l-

%=======================================================================

% Linearized model of the inverted pendulum model in the state space
form

sigma_l = (M+m)*(J+m*l^2)-m^2*l^2 % common denominator

% A_l, B_l, C_l and D_l represent the linear model
%
A_l = [

        0            1            0            0;

        0            0        m*g*l^2/J    l*bt/J;

        0            0            0            1;

        0            0        -m*g*l/J      -bt/J;

       ]
```

```
B_l = [

                0;

        (J+m*l^2)/sigma_l;

                0;

         -(l*m)/sigma_l

      ]

C_l = [1 0 l 0]

D_l = 0

% form the linear model
sys_l = ss(A_l,B_l,C_l,D_l)

%========================================================================

% Design the linear-quadratic (LQ) state-feedback regulator by using
% function lqr for the linear model in order to find the closed-loop
poles
% Where Q is a positive-definite Hermitian or real symmetric matrix and
R
% is a positive definite Hermitian or real symmetric matrix.

Q = [ 10    0     0    0   ;
       0   10     0    0   ;
       0    0     5    0   ;
       0    0     0    5   ;
     ]
R=1;

[K_l_hat, S_l_hat,E_l_hat]= lqr (A_l, B_l, Q, R)

%========================================================================

% Form the desired model matrix A

A_d_n = A_l-B_l*K_l_hat % 4 by 4 matrix

%========================================================================

% Formation of a matrix A for closed-loop system
A_d_hat = [A_d_n zeros(4,1); -C_l 0]

% Calculation and formation of matrix B
V1 = M1/(J*M1+M*m*l^2);% M1 = M+m;   coeff M+m
V2 = J1/(J*M1+M*m*l^2);% J1 = J + M0*l;  coeff J+m*l^2
                       % M0 = m*l;   coeff m*l
b1s = V2
b2s = M0*V1/M1
B_d=[0; b1s; 0; -b2s]
```

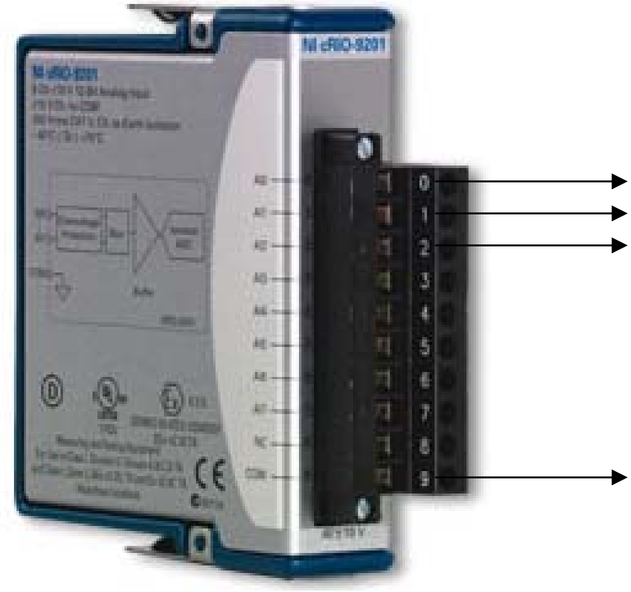
```
B_d_hat = [B_d ; 0]

% Formation of a matrix C D
C_d = [1 0 l 0] % from C it can be seen that y = x + length*theta

D_d = 0

% Compute the controllability matrix
Co=ctrb(A_d_hat,B_d_hat)
% The system is controllable if Co has full rank n

% Check the rank of Co in order to find if the system is controllable
R_sys_d = rank(Co) % R_sys_d = 5 proves the system is controllable


% Form the desired system
sys_d = ss (A_d_n,B_d,C_d,D_d)

% Compute controllability and observability grammians
Wc = gram(sys_d,'c')
Wo = gram(sys_d,'o')

%========================================================================

% Design linear-quadratic (LQ) state-feedback regulator for state-space
% systems
% Where Q is a positive-definite Hermitian or real symmetric matrix and
R
% is a positive definite Hermitian or real symmetric matrix.

Q = [ 150       0        0        0       0    ;
        0     150        0        0       0    ;
        0       0      100        0       0    ;
        0       0        0      100       0    ;
        0       0        0        0     3000   ]
R=1;

[Khat, Shat,Ehat]= lqr (A_d_hat, B_d_hat, Q, R)

%========================================================================

% Lyapunov matrix

% set the positive definite matrix Q1

Q1 = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1]

% By using Q to calculate P, where -Q1 = A'P+PA

syms P2

A_transp=A_d_n';

P2 = -Q1 /(A_transp + A_d_n)
```

```
% Check P2 whether positive definite

[R_P2,P_P2] = chol(P2)

[R_Q1,P_Q1] = chol(Q1)

% If X is not positive definite, an error message is printed

% Both P2 and Q1 are positive definite

%======================================================================
%The End
```

APPENDIX B

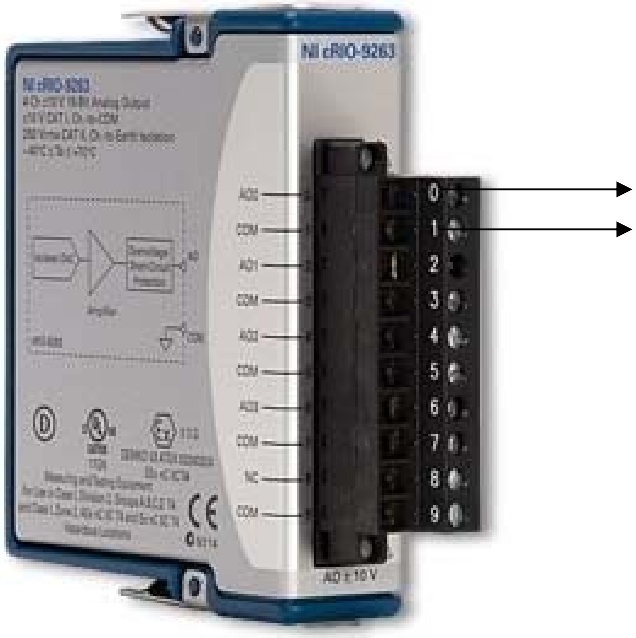
APPENDIX B

COMPACTRIO I/O WIRING



AI 0 (measure output signal y)
AI 1 (measure displacement signal x)
AI 2 (measure angle signal θ)
GND

AI-NI cRIO-9201



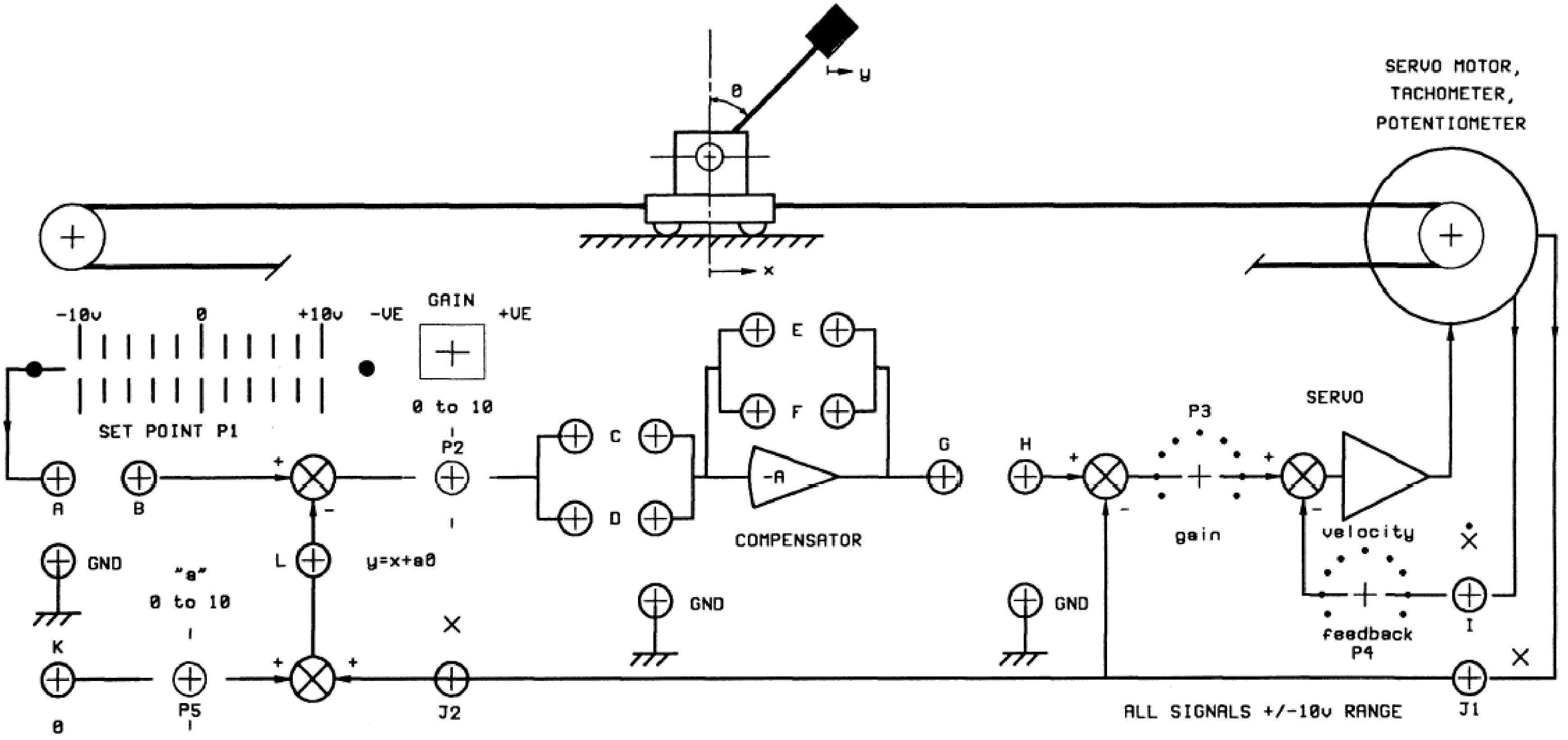
AO 0 (control signal u)
GND

AI-NI cRIO-9263

APPENDIX C

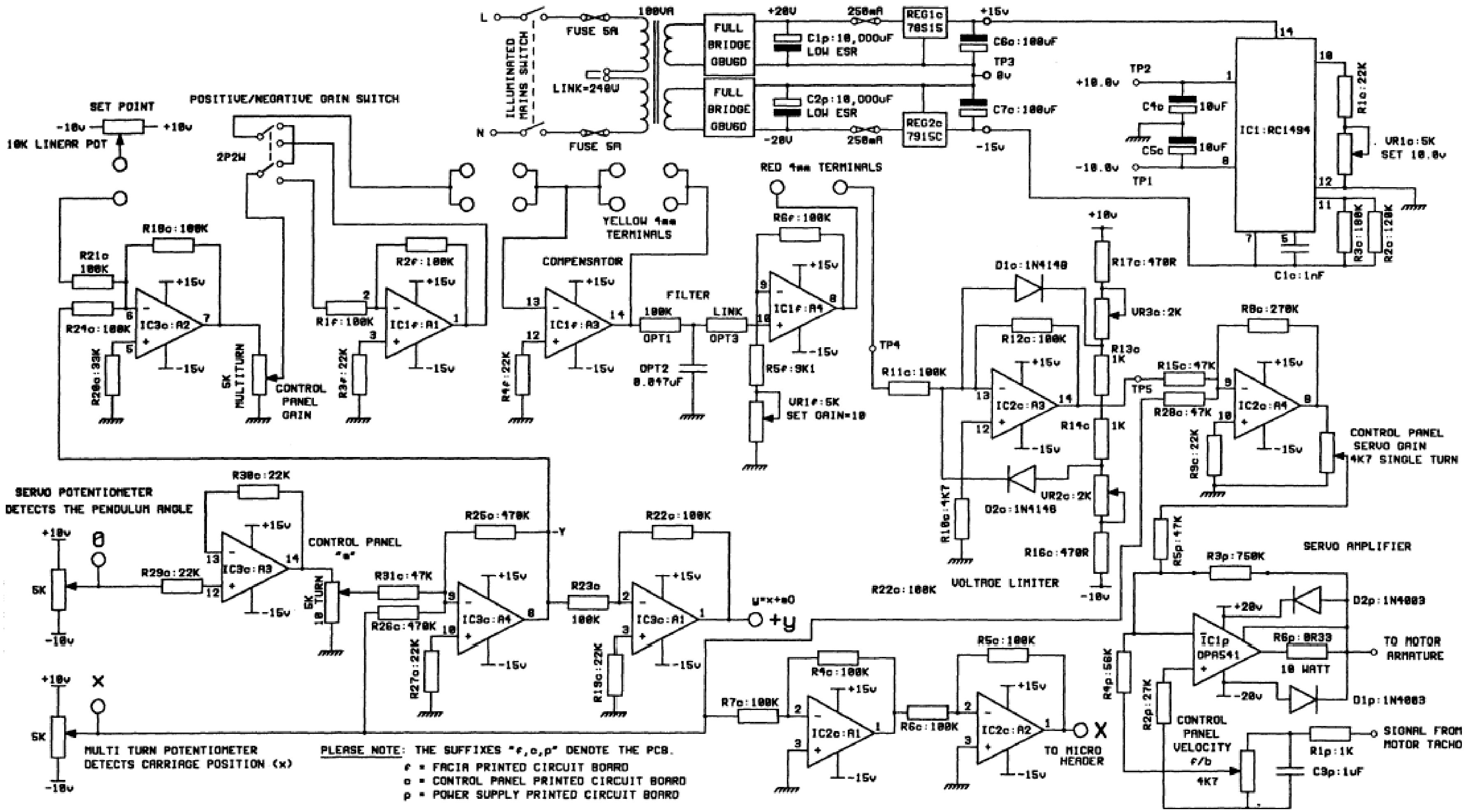
APPENDIX C. 1

THE PENDULUM UNIT CONTROL CONSOLE



APPENDIX C. 2

THE CIRCUIT DIAGRAM OF THE PENDULUM SYSTEM



REFERENCE

Abdelmalek, I., Goléa, N. and Hadjili, M. 2007. A New Fuzzy Lyapunov Approach to Non-Quadritic Stabilization of Takagi-Sugeno Fuzzy Models. Journal of Systems Engineering and Electronics. Volume 18, Issue 3, 2007, pp. 598-602.

Altinöz, Ö, T. 2007. Adaptive Integral Backstepping Motion Control for Inverted Pendulum. Proceedings of World Academy of Science, Engineering and Technology, Volume 23 August 2007 ISSN 1307-6884. http://www.waset.org/pwaset/v23/v23-51.pdf

Angeli, D. 2001. Almost Global Stabilization of the Inverted Pendulum via Continuous State Feedback. Automatica, Volume 37, 2001, pp. 1103-1108.
D'Azzo, J, J. and Houpis, C, H. 1988. Linear Control System Analysis and Design – Conventional and Modern, 3rd edition. McGraw-Hill Book Company. New York, United State of America.

Arreola, R, B. 2004. Output Feedback Nonlinear Control for a Linear Motor in Suspension Mode. Automatica, Volume 40, Issue 12, December 2004, pp. 2153-2160.

Bajodah, A, H. Stabilization of Under actuated Spacecraft Dynamics via Singularly Perturbed Feedback Linearization. JKAU: Eng. Sci., Volume 16, No. 2, pp: 35-53.

Batzel, J. 2001. Introduction to Linear Control Theory. Lecture 3 Math notes. University of Graz, Graz, Austria.

Bedrossian, N, S. 1992. Approximate Feedback Linearization: the Car-Pole example, Proceeding of the 1992 IEEE International Conference on Robotics and Automation, France, Nice, May, 1992, 1987-1992.

Blair, C, E. 2002. Lecturer 7: Lagrange's Equations. How, Deyst 2003 (Based on notes by Blair 2002). http://ocw.mit.edu/NR/rdonlyres/Aeronautics-and-Astronautics/16-61Aerospace-DynamicsSpring2003/644258CA-E401-4F96-A505-E4863A9BBE05/0/lecture7.pdf [24th March 2007]

Bugeja, M. 2003. Non-Linear Swing-Up and Stabilizing Control of an Inverted Pendulum System. In Proceeding of the IEEE Region 8 EUROCON 2003: Computer as a Tool, B. Zajc and M. Tkalčič, eds., Volume. 2, IEEE, pp. 437–441, Ljubljana, Slovenia

Burg, T., Dawson, D., Rahn, D. and Rhodes, W. 1996. Nonlinear Control Of An Overhead Crane Via The Saturating Control Approach Of Teel. Robotics and Automation, 1996. Proceedings, 1996 IEEE International Conference on Volume 4, Issue, 22-28, Aprial 1996, pp. 3155-3160 Volume. 4.

Bytronic international LTD. 2001. Documentation for the Bytronic pendulum control system manual. Version 3.0. West Midlands, England.

Cai, X, S., Wang, X, D. and Zhang H, R. 2007. Universal Construction of Control Lyapunov Functions for a Class of Nonlinear System. Journal of System Engineering and Electronics. Volume 18, Issue 3, 2007, pp. 598-602.

Chaturvedi, N, A., McClamroch, N, H. and Bernstein, D, S. 2008. Stabilization of a 3D axially symmetric pendulum. Chaturvedi, N.A., et al. Stabilization of a 3D axially symmetric pendulum. Automatica (2008), doi: 10.1016/j.automatica.01.013

Chen, C, C., Hsu, C, H., Chen, Y, J. and Lin, Y, F. Disturbance Attenuation of Nonlinear Control Systems Using an Observer-based Fuzzy Feedback Linearization Control. Chaos, Solitons and Fractals, Volume 33, Issue 3, August 2007, pp. 885-900.

Chen, Y, C. and Teng, C, C. 1995. A Model Reference Control Structure Using a Fuzzy neural Network. Fuzzy Sets and Systems, Volume. 73, Issue 3, 8 August 1995, pp. 291-312.

d'Andréa-Nove, B. and Boustany, F. 1991. Adaptive Control of a Class of Mechanical Systems using Linearization and Lyapunov methods. A comparative study on the overhead crane example. Decision and Control, 1991, Proceedings of the 30th IEEE Conference on Volume, Issue, 11-13, December 1991, pp120-125, vol.1, Brighton, England. Doi:10.1109/CDC.1991.261268.

d'Andréa-Novel, B. and Coron, J, M. 2000. Exponential Stabilization of an Overhead Crane with Flexible Cable via a Back-Stepping Approach. Automatica, Volume. 36 (2000), pp. 587-593.

Datta, A. and Ioannou, P, A. 1994. Performance Analysis and Improvement in Model Reference Adaptive Control. IEEE Transaction on Automat. Control, Volume 39, No. 12, pp. 2370-2387, December 1994.

Engleder, S. 2007. Time-Optimal Motion Planning and Control of Electohydralically actuated Toggle Mechanism. Mechatronics, Volume 17, Issue 8, October 2007, pp. 448-456.

Fang, Y., Dixon, W, E., Dawson, D, M. and Zergeroglu, E. 2001. Nonlinear Coupling Control Laws for a 3-DOF Overhead Crane System. Proceedings of the 40th IEEE Conference on Decision and Control Orlando, Florida USA, December 2001.

Formal'skii, A, M. 2006. An Inverted Pendulum on a Fixed and a Moving base. Journal of Applied Mathematics and Mechanics, Volume 70, 2006, pp. 56-64. doi:10.1016/j.jappmathmech.2006.03.010

Friedland, B. 1986. Control System Design: An Introduction to State-Space Methods. University of Michigan. McGraw-Hill, 1986. ISBN 0070224412, 9780070224414

ftp://ftp.ni.com/pub/devzone/pdf/tut_2856.pdf

Gajic, Z. http://www.ece.rutgers.edu/~gajic/psfiles/lqr.pdf. Class Notes, Rutgers University. New Jersey, United State of America.

Gawthrop, P and Ronco, E. 2002. Predictive Pole-Placement control with Linear Models. Automatica, Volume. 38, Issue 3, 2002, pp. 421-432.

Gawthrop, P. and Wang, L, P. 2006. Intermittent Preditive Control of an Inverted Pendulum. Control Engineering Practice, Volume 14, Issue 11, November 2006, pp. 1346-1356, doi:10.1016/j.conengprac.2005.09.002.

Hassanzadeh, I., Mobayen, S. and Harifi, A. 2008. Input-Output Feedback Linearization Cascade Controller Using Genetic Algorithm for Rotary Inverted Pendulum System. American Journal of Applied Science 5 (10): pp1322-1328, 2008. ISSN 1546-9239.

Hedrick, J, K. and Girard, A. 2005. Control of Nonlinear Dynamic Systems: Theory and Applications. Columbia University, Fall 2005.

Henders, M, G. and Soudack, A, C. 1996. Dynamics and Stability State-Space of a Controlled Inverted Pendulum. Int. J. Non-Linear Mechanics, Volume. 31, No. 2, pp. 215-227, 1996. Elsevier Science Ltd.

Henders, M, G. and Soudack, A, C. 1996. Dynamics and Stability State-Space of a Controlled Inverted Pendulum. Int. J. Non-Linear Mechnics, Volume. 31, No. 2, pp. 215-227, 1996.

http://library.wolfram.com/infocenter/MathSource/547/nl_stab.txt, April 18th 2008

http://netlib.org/linalg/html_templates/node20.html, February 20th, 2007

http://sine.ni.com/nips/cds/view/p/lang/en/nid/1191, April 12th 2008

http://sine.ni.com/nips/cds/view/p/lang/en/nid/11915 May 6th 2008

http://sine.ni.com/nips/cds/view/p/lang/en/nid/11915, May 10th 2008

http://sine.ni.com/nips/cds/view/p/lang/en/nid/14154 April 10th 2008

http://sine.ni.com/nips/cds/view/p/lang/en/nid/14147

http://sine.ni.com/nips/cds/view/p/lang/en/nid/14155 April 10th 2008

http://sine.ni.com/nips/cds/view/p/lang/en/nid/14170

http://sine.ni.com/nips/cds/view/p/lang/en/nid/14294

http://sine.ni.com/nips/cds/view/p/lang/en/nid/14588

http://sine.ni.com/nips/cds/view/p/lang/en/nid/1714 May 10th 2007

http://sine.ni.com/nips/cds/view/p/lang/en/nid/1717 May 10th 2007

http://www.bytronic.net/html/pcs.html January 29th, 2007

http://www.engin.umich.edu/group/ctm/examples/pend/digINVSS.html April 3rd 2007

http://www.engin.umich.edu/group/ctm/examples/pend/invPID.html April 3rd 2007

http://www.mathworks.de/products/daq/supportedio14005.html, April 10th 2008

http://www.ni.com/compactrio/whatis.htm July 29th 2008

http://www.ni.com/pdf/products/us/4daqsc253-265_194-196.pdf

http://www.transducertechniques.com/DAQ-SCC-2345.cfm May 11th 2008

Huang, J. and Lin, C, F. 1993. Applications of Sliding Mode Control to Bank-To-Turn Missile System. Aerospace Control System, 1993. Proceedings. The First IEEE Regional Conference on Volume, Issue, 25-27 May 1993, pp. 569-573.

Huilgol, R, R. Christie, J, R. and Panizza, M, P. 1994. The Motion of a Mass Hanging from at Overhead Crane. Chaos, Solution and Fractals, Volume 5, No. 9, pp. 1619-1631, 1995.

International Monetary Fund. Data and Statistics. Dissemination Standards Bulletin Board. Special Data Dissemination Standard. South Africa. Access National Summary Data Page. Economic and Financial Data for South Africa. South Africa Reserve Bank. http://www.reservebank.co.za/internet/Live%20SDDSAppl.nsf/sddsdata?openagent. Data of last update: 8 August 2008. [10 August 2008].

Ioannou, P, A. and Sun, J. 1995. Robust Adaptive Control. Prentice Hall, Inc. ISBN 0134391004, 9780134391007

Irfan, A., Ashraf, J., Ahmed, W., Ayub K, Suleman., Ali, Imran., Khalid, Z., Kumari, R. and Mahbob, Iram. 2000. Journal of AMSE France, Volume. 55, N^0 3, 4, December, 2000. http://www.ewh.ieee.org/sb/iiee/new/misc/pip.pdf

Jain, H., Kaul, V. and Ananthkrishnan, N. 2005. Parameter Estimation of unstable, Limit Cycling Systems Using Adaptive Feedback Linearization: example of Delta Wing Roll Dynamics. Journal of Sound and Vibration, Volume 287, Issues 4-5, 4 November 2005, pp. 939-960. doi:10.1016/j.jsv.2004.12.013

Khalil, H, K. 1996. Nonlinear Systems, 2nd Ed. Upper Saddle River, New Jersey: Prentice-Hall.

Koren, Y., Heisel, U., Jovan, F., Moriwaki, T., Pritschow, G., Ulsoy, G. & Van Brussel. H. 1999. “Reconfigurable Manufacturing Systems”. Annals of the CIRP Volume. 48/2/1999 page(s): 527-540.

Kuo, A, D. 2007. The Six Determinations of Gait and the Inverted Pendulum Analogy: A Dynamic Walking Perspective. Human Movement Science, Volume. 26 (2007), pp. 617-656, ScienceDirect.

Lam, J. Control of an Inverted Pendulum. http://anadrasis.web.auth.gr/pendulum/Control%20of%20inverted%20pendulum.pdf [10th February 2007]

Landau, I, D. 1979. Adaptive control: the model reference approach. Marcel Dekker. New York.

Lee, S. & Tilbury, D. 2003. “An Application of Supervisory Control Methods for a Serial/Parallel Multi-Part Flow Line: Modelling and Deadlock Analysis”, Systems, Man and Cybernetics, 2003. IEEE International Conference on Volume 4, Issue, 5-8 Oct. 2003 pp. 3402-3407.

Lin, Z, L., Saberi, A., Gutmann, M. and Shamash, Y. 1996. Linear Controller for an Inverted Pendulum Having Restricted Travel: A High-and-Low Gain Approach. Automatica, Volume. 32, No. 6. pp. 933-937, 1996.

Liu, D, L., Yi, J, Q., Zhao, D, B. and Wang, W. 2005. Adaptive Sliding Mode Fuzzy Control for a Two-Dimensional Overhead Crane. Mechatronics Volume. 15 (2005) pp. 505-522, ScienceDirect, doi: 10.1016/j.mechatronics. 2004.11.004.

Lozano, R., Fantoni, I. and Block, D, J. 2000. Stabilization of the Inverted-Pendulum around its Homoclinic Orbit. Systems & Control Letter, Volume 40, Issue 3, 5, July 2000, pp. 197-204.

Lu, Z., Shieh, L, S., Chen, G, R. and Coleman, N, P. 2006. Adaptive Feedback Linearization Control of Chaotic System via Recurrent High-Order Neural Networks. Information Sciences, Volume 176, Issue 16, 22 August 2006, pp. 2337-2354. doi: 10.1016/j.ins.2005.08.002
Lundberg, K, H. and Roberge, J, K. 2003. Classical Dual-Inverted-Pendulum Control.

Presented at the 2003 IEEE Conference on Decision and Control. http://web.mit.edu/klund/www/papers/CDC03_dip.pdf

Mehrabi, M. G., Ulsoy, A. G. and Koren, Y. 1998. Reconfigurable Manufacturing Systems: Key to Future Manufacturing.

Mohammad, J., Mahmoud, T. and Bahram, S. 1992. Computer-aided Analysis and Design of Linear Control Systems. Prentice Hall, 1992. ISBN 013161472x, 9780131614727

Moyne, J. R., Tilbury, D. M., Sukerkar, K & Wijaya, H. “An Integrated Distributed Software System for Reconfigurable Manufacturing”.

Moyne, J., Korsakas, J. & Tilbury, D. 2004. “Reconfigurable Factory Testbed (RFT): A Distributed Testbed for Reconfigurable Manufacturing System”. Proceedings of 2004 JUSFA 2004 Japan – USA Symposium on Flexible Automation Denver, Colorado, July 19-21, 2004.

Muškinja, Nenad. and Tovornik, Boris. 2006. Swinging Up and Stabilization of a Real Inverted Pendulum. IEEE Transactions on Industrial Electronics, Volume 53, No. 2, April 2006. doi: 10.1109/TIE.2006.870667

Nayyar, D. 2008. “Rapid growth in large emerging economies is changing the balance of world power”. World Institute for Development Economics Research. United Nations University. Helsinki, 23 June 2008.

Niemann, H. and Poulsen, J, K. 2005. Design and analysis of controllers for a double inverted pendulum. ISA Transactions, Volume. 44, Issue 1, January 2005, pp. 145-163.

Ogata, K. 2002. Modern Control Engineering. Fourth Edition. Upper Saddle River, New Jersey, Prentice Hall.

Oh, S, N., Kim, K. and Lim, S. 2003. Motion Control of Biped Robots Using a Single-Chip Drive. Robotics and Automation, 2003. Proceedings, ICRA apos; 03. IEEE International Conference on Volume 2, Issue, 14-19 September 2003, pp. 2461-2465 Volume 2.

Park, H., Chwa, D. and Hong, K. 2007. A feedback linearization control of container cranes: varying rope length. International Journal of Control, Automation, and Systems, vol. 5, no. 4. pp. 379-387, August 2007.

Park, K., Chung, H. and Lee, J, G. 2000. Point Stabilization of Mobile Robots via State-Space Exact Feedback Linearization. Robotics and Computer-Integrated Manufacturing, Volume 16, Issue 5, October 2000, pp. 353-363.

Paz, R, A. 2002. The Model Reference Design Technique. Klipsch School of Electrical and Computer Engineering. Spring 2002.

Phan, P, A. and Gale, T, J. 2008. Direct Adaptive Fuzzy Control with a Self-Structuring Algorithm. Fuzzy Sets and System, Volume. 159 (2008), pp. 871-899. doi: 10.1016/j.fss.2007.09.012

Ruan, X, G., Ding, M, X., Gong, D, X. and Qiao, J, F. 2007. On-Line Adaptive Control for Inverted Pendulum Balancing Based on Feedback-Error-Learning. Neurocomputing, Volume. 70, Issues 4-6, January 2007, pp. 770-776. doi:10.1016/j.neucom.2006.10.012
Sharma, B, B. and Kar, I, N. 2007. Parametric Convergence and Control of Chaotic System

Using Adaptive Feedback Linearization. Chaos, Solitons and Fractals (2007). doi: 10.1016/j.chaos.2007.09.060

Slotine, J, J, E. and Li, W, P. 1991. Applied Nonlinear Control. Prentice Hall. Englewood Cliffs. New Jersey. ISBN: 0-13-040890-5.

South Africa Government Information. About South Africa: Economy – Manufacturing. http://www.info.gov.za/aboutsa/economy.htm#manufacturing [10 May 2007].

Stimac, A, K. 1999. Standup and Stabilization of the Inverted Pendulum. Standup and Stabilization of the Inverted Pendulum. Unpublished Bsc Thesis, Massachusetts Institute of Technology, United State of America. http://ocw.mit.edu/NR/rdonlyres/Mechanical-Engineering/2-003Spring-2005/B37B7209-7C66-4E27-BB2A-2B07A7DC1E26/0/andrew.pdf

Su, H. and Woodham, C, A. 2003. On the uncontrollable damped triple inverted pendulum. Journal of Computational and Applied Mathematics, Volume. 151 (2003), pp. 425-443.

Tan, C, P. and Habib, M, K. 2007. Robust Sensor Fault Reconstruction applied in Real-Time to an Inverted Pendulum. Mechatronics 17 (2007) pp 368-380, ScienceDirect.

Tanaka, K. and Sugeno, M. 1992. Stability Analysis and Design of Fuzzy Control System. Fuzzy Sets and Systems, Volume 45, Issue 2, 24 January 1992, pp. 135-156.

Tanaka, K., Hori, T. and Wang, H, O. 2003. A Multiple Lyapunov Function Approach to Stabilization of Fuzzy Control Systems. IEEE Trans. Fuzzzy Systems, Volume 11, No. 4, pp. 582-589.

Uçar, A. 2007. Model-reference control of chaotic systems. Chaos, Solitons and Fractals, Volume 31, Issue 3, February 2007, pp. 712-717.

Wu, Q., Sepehri, N. and He, S. 2002. Neural inverse modelling and control of a base-excited inverted pendulum. Engineering Applications of Artificial Intelligence, Volume. 15, Issues 3-4, June-August 2002, pp. 261-272.

Yavin, Y. 1999. Control of a Rotary Inverted Pendulum. Applied Mathematical Letters, Volume. 12 (1999), pp. 131-134.

Yi, J, Q., Yubazaki, N. and Hirota, Kaoru. 2002. A New Fuzzy Controller for Stabilization of Parallel-Type Double Inverted Pendulum System. Fuzzy Sets and Systems, Volume. 126 (2002), pp. 105-119.

Yurkovich, S. and Widjaja, M. 1996. Fuzzy Controller Synthesis for an Inverted Pendulum System. Control Engineering Practice, Volume. 4, No. 4. pp. 455-469, 1996.

Zhang L.K. (2005). *CPLE/FPGA application development technology and engineering practice*. Posts & telecom press, pp 10, 12, 15-17, 146-200. Beijing, China.