



Cape Peninsula  
University of Technology

**DEVELOPMENT OF NONLINEAR CONTROL ALGORITHMS FOR  
IMPLEMENTATION IN DISTRIBUTED SYSTEMS**

**by**

**YOHAN DARCY MFOUMBOULOU**

**Thesis submitted in fulfilment of the requirements for the degree**

**Master of Technology: Electrical Engineering**

**in the Faculty of Engineering**

**at the Cape Peninsula University of Technology**

**Supervisor:** Prof R. Tzoneva

**Co-Supervisor:** Mr. C. Kriger

**Bellville Campus**

June 2014

## DECLARATION

I, Yohan Darcy Mfoumboulou, declare that the contents of this dissertation/thesis represent my own unaided work, and that the 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

In the past decade, the need for flexibility and reconfigurability in automation has contributed to the rise of the distributed concept in control systems engineering. The IEC 61499 standard is used to define a distributed model for dividing various components of an industrial application in automation process and complicated control of machinery into function blocks. Such function blocks have the flexibility to be distributed and interconnected across a number of controllers. However, this new standard for automation faces two main challenges: the complexity in designs of distributed systems and the lack of utilization of the standard in industry. Most applications of controllers based on functional block programming are for linear systems. As most of industrial processes are nonlinear there is a need to extend the functional block approach for implementation of nonlinear controllers.

Design complexity involves the exact modeling of the system in function blocks to obtain its accurate behaviour and the lack of utilization of the standard is understandable because new technologies are not easily accepted in industry due to their high prices and risks of compromising the performance at the production level.

The thesis describes a methodology for design and implementation of nonlinear controllers for nonlinear plants in IEC 61499 standard compliant real-time environment of TwinCAT 3 and Beckhoff Programmable Logic Controller (PLC). The first step is to design the nonlinear controllers and simulate the closed-loop system in MATLAB/SIMULINK software. Then the new engineering based concepts to transform the obtained closed-loop system model to an IEC 61499 Function Block Model. This is accomplished by applying one method which involves a complete model transformation between two block-diagram languages: Simulink and TwinCAT 3. The development tools that support the transformation algorithm in the thesis sets the foundation stone of the verification and validation structure for IEC 61499 function blocks approach. The transformed model of the closed-loop system is downloaded to the Beckhoff PLC and is simulated in real-time.

The obtained results demonstrate that the developed methodology allows complex nonlinear controllers to be successfully transformed to IEC 61499 standard compliant environment and to be applied for real-time PLC control of complex plants.

## **ACKNOWLEDGEMENTS**

Firstly, I would like to thank God for giving me the strength to finish this thesis despite all the difficulties encountered during these past years of research. There are times, I doubted that I could complete this degree but he provided me with the spiritual support to give me the strength to never give up.

Secondly, I would like to thank my supervisor, Professor Tzoneva for all the guidance and support she had given me throughout my years of research. Her assistance was massive toward the completion of this thesis. I started my research with little knowledge of how nonlinear control systems are analyzed, built and implemented. Her insightful suggestions in the domain helped me to improve my knowledge of control systems.

Furthermore, I would like to thank my colleagues at the Centre for Substation Automation and Energy Management Systems for their constant words of encouragement. The atmosphere they bring in the centre makes it a great place to be. Special words of thank to Phaphama for helping all the students of the centre with administrative issues. Without her, our studies would have been much more complicated and stressful.

Finally, I would like to extend my heartfelt gratitude to my friends: Clency Ngary, John Retonda, and Akim Aminou for their indescribable support and encouragement during these years of research.

The financial assistance of the National Research Foundation towards this research is acknowledged. Opinions expressed in this thesis and the conclusions arrived at, are those of the author, and are not necessarily to be attributed to the National Research Foundation.

## **DEDICATION**

I dedicate this thesis to my family. An unutterable feeling of gratitude to my loving parents: my father, Anatole Mfoumboulou, and my mother, Philippine Magandji Bouka. Their words of encouragement, financial support and push for tenacity ring in my ears. My father taught me that success can only be achieved through hard work. My mother taught us to be united in the family and support each other.

My sisters: Charmelle, Lindsey, and Raissa have always supported me and are very special in my life. My brothers Darren, Jonas, and Laurince have always seen me as a source of inspiration and give more strength in my studies. My nephew Clause-Reece and my niece Alvinne-Tricia who brought joy and blessings in our family.

Finally, I also dedicate this thesis to my late elder sister, Patricia, and my uncle, Ozone that I lost when I was still in high school. They took care of me when I was young, without them, I would have not been there today. There are no words that can describe these amazing people.

## TABLE OF CONTENTS

DECLARATION .....	ii
ABSTRACT .....	iii
ACKNOWLEDGEMENTS .....	iv
TABLE OF CONTENTS .....	vi

### CHAPTER ONE: PROBLEM FORMULATION, AIMS, OBJECTIVES

1.1 Introduction.....	18
1.2 Motivation for research and problem definition .....	18
1.3 Problem statement .....	18
1.3.1 System modeling sub-problem.....	20
1.3.1.1 Sub-problem 1: Nonlinear mathematical model development .....	20
1.3.1.2 Sub-problem 2: Linear mathematical model development .....	20
1.3.2 Sub-problem 3: Linear and nonlinear magnetic levitation controllers design and development.....	20
1.3.3 System simulation and emulation sub-problems.....	20
1.3.3.1 Sub-problem 4: Mathematical models simulation .....	21
1.3.3.2 Sub-problem 5: Closed-loop system simulation .....	21
1.3.4 Real-time implementation .....	21
1.3.5 Results analysis .....	21
1.3.5.1 Sub-problem 6: Simulation results and analysis .....	21
1.3.5.2 Sub-problem 7: Real-time implementation results.....	21
1.4 Research aim and objectives .....	21
1.4.1 Research aim .....	22
1.4.2 Research objectives .....	22
1.5 Hypothesis .....	22
1.6 Delimitation of research .....	23
1.7 Assumption.....	23
1.8 Research methods.....	23
1.8.1 Literature review method .....	23
1.8.2 Design methods.....	24
1.8.3 Simulation and experimental methods .....	24
1.8.4 Description method .....	24
1.9 Thesis contribution.....	24
1.10 Outline of the thesis .....	24

### CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction.....	27
2.2 Magnetic levitation .....	28
2.2.1 Description.....	28
2.2.2 Electro-magnetic suspension (EMS).....	28
2.2.3 Electro-dynamic suspension (EDS).....	28
2.2.4 Magnetic ball levitation .....	29
2.3 Control of the magnetic levitation system .....	30
2.4 Concept of distributed control system in IEC 61499.....	35
2.4.1 Intelligent control.....	37
2.4.2 Multi-Agent systems (MAS) .....	38
2.4.2.1 Overview .....	38
2.4.2.2 Challenges in Multi-Agents Systems.....	40
2.4.3 Drawbacks in distributed implementation .....	41
2.5 Creation of the IEC 61499 Standard .....	41
2.5.1 Validation of IEC 61499.....	41
2.6 Comparison between the IEC 61131-PLC and IEC 61499 standards.....	44
2.6.1 IEC 61131-PLC .....	44

2.6.2 IEC 61499 standard-based Function Blocks.....	45
2.6.2.1 Basic Function Blocks .....	46
2.6.2.2 Network of Basic and Composite Function Blocks .....	46
2.6.2.3 Service Interface Function Blocks .....	47
2.6.3 Advantages of IEC 61499 standard over IEC 61131-PLC standard .....	48
2.6.4 Disadvantages of IEC 61499 standard over IEC 61131-PLC standard .....	48
2.6.5 IEC 61499 standard-based Function Blocks Development Status .....	49
2.6.6 Other International Standards Related to the IEC 61499 standard.....	50
2.6.6.1 IEC 61850 standard for substation automation .....	50
2.6.6.1.1 IEC 61850 Generic Object Oriented Substation Event (GOOSE) Messaging.....	51
2.6.6.2 IEC 61400-25 .....	52
2.6.6.3 IEC 61970-301 and IEC 61968-11 .....	52
2.6.6.4 IEC 61508 and IEC 61511 .....	53
2.7 Modeling and Design Environments .....	55
2.7.1 Matlab.....	55
2.7.2 Simulink.....	55
2.7.3 LabVIEW.....	57
2.8 Existing Distributed Control Techniques in IEC 61499.....	58
2.8.1 Model Transformation Approach.....	58
2.8.2 Co-Modeling Approach.....	59
2.9 Comparative literature review.....	67
2.9.1 Migration techniques from classical PLC controller to IEC 61499 standard.....	67
2.9.2 Structure design methods to model, verify, and implement function blocks following the IEC 61499 standard principles.....	68
2.10 Conclusion.....	68

## **CHAPTER THREE: MATHEMATICAL MODEL OF THE MAGNETIC LEVITATION SYSTEM**

3.1 Introduction.....	70
3.2 Modeling of physical systems .....	71
3.2.1 Modeling of Electrical systems.....	71
3.2.2 Modeling of Mechanical systems .....	71
3.3 Mathematical model derivation of magnetic levitation systems .....	72
3.3.1 Electrical sub-system.....	73
3.3.2 Mechanical sub-system .....	73
3.3.3 Full model of the system.....	75
3.4 Derivation of the state space model.....	75
3.4.1 Linearization of the magnetic levitation system at equilibrium point.....	77
3.5 Simulation and analysis of the dynamic responses of the magnetic levitation system.....	81
3.6 Analysis of results.....	83
3.7 Conclusion.....	84

## **CHAPTER FOUR: CONTROL THEORY OVERVIEW**

4.1 Introduction.....	85
4.2 Linear systems.....	86
4.2.1 Stability of linear control systems.....	87
4.2.1.1 Zero-input and asymptotic stability of continuous data systems.....	87
4.2.1.2 Rough-Hurwit criterion.....	88
4.2.1.3 Nyquist criterion.....	88
4.2.1.4 Bode diagram.....	88
4.2.2 Controllability.....	88
4.2.3 Observability.....	89

4.3 Techniques of linear controller design.....	89
4.3.1 Root Locus Technique.....	89
4.3.2 Pole placement design using feedback.....	90
4.3.3 Robustness.....	92
4.3.4 Optimal formulation for design of regulators.....	93
4.4 Behaviour of nonlinear systems.....	95
4.4.1 Nonlinearities.....	95
4.4.2 Fundamentals of Lyapunov theory.....	95
4.4.2.1 Linearization and local stability.....	96
4.4.2.1.1 Direct method of Lyapunov.....	96
4.4.2.1.2 Positive definite functions theorem.....	97
4.4.2.1.3 Lyapunov functions theorem.....	97
4.4.2.1.4 Lyapunov analysis of Linear Time Invariant systems (LTI).....	97
4.4.2.2 Krasovski's method.....	98
4.5 Nonlinear control systems design.....	98
4.5.1 Feedback linearization.....	99
4.5.1.1 Input-state linearization.....	100
4.5.1.2 Input-output linearization.....	101
4.6 Conclusion.....	102

## CHAPTER FIVE: LINEAR CONTROL DESIGN FOR THE MAGNETIC BALL

### LEVITATION SYSTEM

5.1 Introduction.....	103
5.2 Linear control design methods.....	103
5.3 Pole placement methods.....	104
5.3.1 Selection of the desired place of the poles of the closed loop system.....	105
5.3.1.1 Desired transfer function.....	105
5.3.1.1.1 Butterworth filter.....	105
5.3.1.1.2 Transfer functions of a closed-loop system.....	106
5.3.1.1.3 Closed-loop desired transfer function with no overshoot.....	106
5.3.1.2 Feedback matrix coefficients calculation.....	107
5.3.2 State feedback control design.....	108
5.3.2.1 Controllability of the system.....	109
5.3.2.2 Poles selection.....	110
5.3.2.3 Calculation of the controller parameters.....	111
5.3.2.4 Simulation of the closed-loop system.....	112
5.3.3 State feedback with integral control.....	122
5.3.3.1 Integration of the integral control into the closed-loop system.....	122
5.3.3.2 Calculation of the controller parameters.....	124
5.3.3.3 Simulation of the closed loop system.....	124
5.4 Linear quadratic regulator LQR design.....	135
5.4.1 Determination and solution of the LQR problem parameters.....	136
5.4.2 Closed-loop system behaviour simulation.....	136
5.4.3 Results analysis.....	147
5.5 Conclusion.....	147

## CHAPTER SIX: DESIGN OF A NONLINEAR LINEARIZING CONTROLLER ON THE BASIS OF INPUT-STATE/INPUT-OUTPUT LINEARIZATION

6.1 Introduction.....	149
6.2 Feedback linearization and the canonical form of the process model.....	150
6.3 Mathematical tools for feedback linearization.....	152
6.3.1 Lie derivatives and Lie brackets.....	152
6.3.2 Diffeomorphisms and state transformation.....	153
6.3.3 Frobenius theorem.....	154
6.3.4 Relative degree of a system.....	155



6.3.5 Linearization techniques.....	155
6.3.5.1 Input-state linearization.....	155
6.3.5.1.1 Input-state linearization procedure.....	156
6.3.5.2 Input output linearization.....	157
6.3.5.2.1 Control design procedure for the input-output linearization.....	157
6.4 Design of a feedback linearizing controller for magnetic levitation.....	158
6.4.1 Relative degree of the system.....	158
6.4.2 Construction of the vector fields for input-state linearization.....	159
6.4.3 Check for involutivity of the system.....	162
6.4.4 Coordinate transformation.....	163
6.4.5 Linear quadratic regulator design for the linearized model.....	164
6.5 Simulation.....	166
6.5.1 Simulation results.....	168
6.6 Discussion of the results.....	178
6.7 Conclusion.....	180

## **CHAPTER SEVEN: DESIGN OF A NONLINEAR LINEARIZING CONTROLLER ON THE BASIS OF MODEL REFERENCE CONTROL AND LYAPUNOV THEORY**

7.1 Introduction.....	181
7.2 Design of nonlinear control based on Lyapunov direct method.....	182
7.2.1 Plant model.....	182
7.2.2 Design of the reference model. Error between the states of the reference and plant models.....	182
7.2.3 Design of the nonlinear linearizing controller.....	184
7.2.3.1 Construction of a Lyapunov function for the system and derivation of its first derivative.....	184
7.2.3.2 Calculation of the nonlinear linearizing control.....	185
7.3 Design of a Lyapunov-based and MRC based nonlinear linearizing controller for the magnetic levitation system.....	186
7.3.1 The nonlinear model of the magnetic levitation system.....	186
7.3.2 Model of the desired linear system (reference model).....	187
7.3.2.1 Determination of the error between the reference model and the plant states.....	188
7.3.2.2 Design of the nonlinear controller.....	188
7.3.2.2.1 Construction of the Lyapunov function.....	188
7.3.2.2.2 Calculation of the first derivative of the Lyapunov function.....	188
7.3.2.2.3 Calculation of the nonlinear linearizing controller $u$ .....	189
7.3.2.3 Representation of the diagram of the closed loop system.....	189
7.3.3 Design of a linear control $v$ for the linearized closed loop system.....	190
7.3.3.1 Specification of the closed loop system with the reference model and for the linearized closed loop system.....	191
7.3.3.2 Design of the linear quadratic controller.....	191
7.3.3.3 Solution of the linear quadratic regulator problem.....	192
7.3.3.4 Application of the linear integral controller to the closed-loop system with the nonlinear controller system and the reference model.....	193
7.4 Simulation.....	194
7.4.1 Simulation results.....	196
7.5 Discussion of results.....	207
7.6 Conclusion.....	208

## **CHAPTER EIGHT: IMPLEMENTATION OF THE DEVELOPED CLOSED-LOOP SYSTEM MODELS IN A PROGRAMMABLE LOGIC CONTROLLER (PLC) BASED ON MODEL TRANSFORMATION APPROACH**

8.1 Introduction.....	209
8.2 Programmable Logic Controller.....	210

8.3 Programmable Logic Controller Hardware.....	211
8.3.1 Programming Devices.....	211
8.3.2 Central Processing Unit.....	211
8.3.3 Power Supply.....	212
8.3.4 Input and Output Modules.....	212
8.4 Programming languages.....	212
8.5 Total Windows Control and Automation Technology (TwinCAT) development software.....	213
8.5.1 eXtended Automation Engineering (XAE).....	214
8.5.1.1 TwinCAT 3 standard.....	214
8.5.1.2 TwinCAT 3 integrated.....	215
8.5.2 eXtended Automation Runtime (XAR).....	217
8.6 Beckhoff CX5020 PLC.....	219
8.7 Algorithm for real-time communication.....	221
8.8 Real-Time Implementation of the developed closed-loop control systems.....	222
8.8.1 Real-Time Implementation on the CX5020 and PC real time platform.....	222
8.8.1.1 Real-Time Implementation of the closed-loop system model based on feedback linearization system.....	223
8.8.1.1.1 Code generation process for the closed-loop system based on feedback linearization control.....	223
8.8.1.1.2 Requirements for proper model transformation and simulation.....	224
8.8.1.1.3 Real-time simulation of the closed-loop system.....	226
8.8.1.2 Real-time implementation of the closed-loop system based on Lyapunov stability and MRC.....	230
8.8.1.3 Real-Time simulation of the closed-loop system.....	230
8.8.2 Results analysis.....	234
8.9 Real-time implementation and simulation of a distributed system.....	234
8.10 Conclusion.....	238

## **CHAPTER NINE: CONCLUSION: THESIS DELIVERABLES, APPLICATIONS AND FUTURE RESEARCH**

9.1 Introduction.....	239
9.2 Thesis deliverables.....	239
9.2.1 Mathematical model development and simulation.....	239
9.2.2 Linear controller design and closed loop simulation.....	240
9.2.3 Nonlinear controllers design and closed-loop system simulation.....	240
9.2.3.1 Feedback linearization controller design and closed-loop system simulation.....	240
9.2.3.2 Lyapunov stability based and Model Reference (MR) based linearizing controller design and closed loop system simulation.....	240
9.2.4 Real-time implementation on TwinCAT 3 software and deployment on Beckhoff CX 5020 PLC.....	241
9.2.4.1 Real-time implementation and simulation of the closed-loop system based on nonlinear input-state feedback linearizing controller.....	241
9.2.4.2 Real-time implementation and simulation of the closed-loop system based on the Lyapunov stability MR nonlinear linearizing controller.....	241
9.2.4.3 Real-time implementation and simulation of a distributed nonlinear system.....	242
9.2.5 Software development.....	242
9.3 Application.....	243
9.4 Future research.....	244
9.5 Publications.....	244

REFERENCES.....	271
-----------------	-----

## LIST OF FIGURES

Figure 2.1: Publication rate of nonlinear control of nonlinear plants.....	31
Figure 2.2: Architecture for intelligent control (Saridis, 1990).....	38
Figure 2.3: Multiple controllers individually control a device (Aldea, et al., 2004).....	40
Figure 2.4: Publication rate for IEC 61499 standard-based papers from 2000 to 2014.....	42
Figure 2.5: Publication rate for IEC 61499 standard by country.....	43
Figure 2.6: Publication rate for IEC 61499 standard by authors.....	43
Figure 2.7: PLC implementation of a conveyor control system (Vyatkin, 2011).....	45
Figure 2.8: EC 61499 FB implementation of Conveyor Control System (Vyatkin, 2011).....	47
Figure 2.9: Sample simulation results from MATLAB and SIMULINK (The MathWorks, 2010).....	57
Figure 2.10: The transformed Function Block System (Yang, 2010).....	59
Figure 2.11: Closed-loop simulation between MATLAB and FB models (Yang, 2010).....	60
Figure 3.1: Diagram of the magnetic levitation system (Ahmad, Javaid, 2010).....	73
Figure 3.2: Poles-zeros map of the magnetic levitation system.....	81
Figure 3.3: Magnetic levitation system nonlinear state space model.....	82
Figure 3.4: Magnetic levitation system state space linear model.....	82
Figure 3.5: Open loop response of the nonlinear system when the initial position is at 0.012[m].....	83
Figure 3.6: Open loop response of the linearized system when the initial position is at 0.012[m].....	83
Figure 5.1: Closed loop system representation.....	105
Figure 5.2: Poles of a Butterworth filter.....	106
Figure 5.3: Simulink diagram of the closed-loop magnetic levitation system.....	112
Figure 5.4: Linear controller algorithm.....	113
Figure 5.5: Position response with initials conditions of $[0.09 \ 0 \ 2.56]'$ .....	113
Figure 5.6: Velocity response with initials conditions of $[0.09 \ 0 \ 2.56]'$ .....	114
Figure 5.7: Current response with initials conditions of $[0.09 \ 0 \ 2.56]'$ .....	114
Figure 5.8: Control signal response with initial conditions of $[0.09 \ 0 \ 2.56]'$ .....	115
Figure 5.9: Position response with initials conditions of $[0.035 \ 0 \ 0.89]'$ .....	115
Figure 5.10: Velocity response with initials conditions of $[0.035 \ 0 \ 0.89]'$ .....	116
Figure 5.11: Current response with initials conditions of $[0.035 \ 0 \ 0.89]'$ .....	116
Figure 5.12: Control signal response with initial conditions of $[0.035 \ 0 \ 0.89]'$ .....	117
Figure 5.13: Position response with initials conditions of $[0.09 \ 0 \ 2.56]'$ .....	117
Figure 5.14: Velocity response with initials conditions of $[0.09 \ 0 \ 2.56]'$ .....	118
Figure 5.15: Current response with initials conditions of $[0.09 \ 0 \ 2.56]'$ .....	118
Figure 5.16: Control signal response with initial conditions of $[0.09 \ 0 \ 2.56]'$ .....	119
Figure 5.17: Position response with initials conditions of $[0.035 \ 0 \ 0.89]'$ .....	119
Figure 5.18: Velocity response with initials conditions of $[0.035 \ 0 \ 0.89]'$ .....	120
Figure 5.19: Current response with initials conditions of $[0.035 \ 0 \ 0.89]'$ .....	120
Figure 5.20: Control signal response with initial conditions of $[0.035 \ 0 \ 0.89]'$ .....	121
Figure 5.21: Full state feedback with integral control.....	122
Figure 5.22: Position response with integral control with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	125
Figure 5.23: Velocity response with integral control with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	125
Figure 5.24: Current response with integral control with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	126
Figure 5.25: Control signal response with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	126
Figure 5.26: Error signal response with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	127

Figure 5.27: Position response with integral control with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	127
Figure 5.28: Velocity response with integral control with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	128
Figure 5.29: Current response with integral control with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	128
Figure 5.30: Control signal response with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	129
Figure 5.31: Error signal response with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	129
Figure 5.32: Position response with integral control with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	130
Figure 5.33: Velocity response with integral control with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	130
Figure 5.34: Current response with integral control with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	131
Figure 5.35: Control signal response with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	131
Figure 5.36: Error signal response with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	132
Figure 5.37: Position response with integral control with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	132
Figure 5.38: Velocity response with integral control with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	133
Figure 5.39: Current response with integral control with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	133
Figure 5.40: Control signal response with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	134
Figure 5.41: Error signal response with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	134
Figure 5.42: Position response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	137
Figure 5.43: Velocity response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	137
Figure 5.44: Current response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	138
Figure 5.45: Control signal response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	138
Figure 5.46: Error signal response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	139
Figure 5.47: Position response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	139
Figure 5.48: Velocity response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	140
Figure 5.49: Current response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	140
Figure 5.50: Control signal response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	141
Figure 5.51: Error signal response with LQR control with $r=0.15$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	141
Figure 5.52: Position response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	142
Figure 5.53: Velocity response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	142
Figure 5.54: Current response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	143
Figure 5.55: Control signal response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	143

Figure 5.56: Error signal response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$ .....	144
Figure 5.57: Position response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	144
Figure 5.58: Velocity response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	145
Figure 5.59: Current response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	145
Figure 5.60: Control signal response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	146
Figure 5.61: Error signal response with LQR control with $r=0.2$ and $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$ .....	146
Figure 6.1: Feedback linearizing system with external linear controller.....	150
Figure 6.2: Input-state linearization control system.....	156
Figure 6.3: Structure of the input-state input-output linearization of the magnetic levitation system.....	166
Figure 6.4: Simulink diagram of the input-state linearization method.....	167
Figure 6.5: Simulink block diagram of the linear controller.....	167
Figure 6.6: Nonlinear linearizing controller structure.....	168
Figure 6.7: Input-state transformation function.....	168
Figure 6.8: Linear control signal when the set point is at 0.55[m].....	169
Figure 6.9: Error signal between the set point and the nonlinear plant output when the set point is 0.55[m].....	169
Figure 6.10: Velocity of the ball when the set point is 0.55[m].....	170
Figure 6.11: Nonlinear linearizing control signal when the set point is 0.55[m].....	170
Figure 6.12: Nonlinear system response under linear and linearizing controls when the set-point is 0.55[m].....	171
Figure 6.13: Linear control signal when the set point is 0.6[m].....	171
Figure 6.14: Error signal between the set point and the nonlinear plant output when the set point is 0.6[m].....	172
Figure 6.15: Velocity of the ball when the set point is 0.6[m].....	172
Figure 6.16: Nonlinear linearizing control signal when the set point is 0.6[m].....	173
Figure 6.17: Nonlinear system response under linear and linearizing controls when the set-point is 0.6[m].....	173
Figure 6.18: Linear control signal when the set point is 0.75[m].....	174
Figure 6.19: Error signal between the set point and the nonlinear plant output when the set point is 0.75[m].....	174
Figure 6.20: Velocity of the ball when the set point is 0.75[m].....	175
Figure 6.21: Nonlinear linearizing control signal when the set point is 0.75[m].....	175
Figure 6.22: Nonlinear system response under linear and linearizing controls when the set-point is 0.75[m].....	176
Figure 6.23: Linear control signal when the set point is at 0.85[m].....	176
Figure 6.24: Error signal between the set point and the nonlinear plant output when the set point is 0.85[m].....	177
Figure 6.25: Velocity of the ball when the set point is 0.85[m].....	177
Figure 6.26: Nonlinear linearizing control signal when the set point is at 0.85[m].....	178
Figure 6.27: Nonlinear system response under linear and linearizing controls when the set-point is 0.85[m].....	178
Figure 7.1: Model-reference control system.....	183
Figure 7.2: Open loop response of the nonlinear reduced-order model when the step input is at 0.3[volts].....	190
Figure 7.3: Block diagram of the closed loop system.....	190
Figure 7.4: Block diagram of the Lyapunov stability based on model reference control system.....	193
Figure 7.5: Simulink diagram of the Lyapunov direct method based on MRC.....	194

Figure 7.6: Simulink block diagram of the linear reference model and its controller.....	195
Figure 7.7: Simulink block diagram of the linear integral controller.....	195
Figure 7.8: Simulink block diagram of the nonlinear linearizing controller on the basis of Lyapunov second method.....	196
Figure 7.9: Linear control signal when the set point is $y^{sp} = 0[m]$ .....	197
Figure 7.10: Error signal between the set point and the nonlinear plant output when the set point is $y^{sp} = 0[m]$ .....	197
Figure 7.11: Errors between the reference model and the plant states when the set point is $y^{sp} = 0[m]$ .....	198
Figure 7.12: Nonlinear linearizing controller signal when the set point is $y^{sp} = 0[m]$ .....	198
Figure 7.13: Position of the ball when the set point is $y^{sp} = 0[m]$ .....	199
Figure 7.14: Linear control signal when the set point is $y^{sp} = 0.01[m]$ .....	199
Figure 7.15: Error signal between the set point and the nonlinear plant output when the set point is $y^{sp} = 0.01[m]$ .....	200
Figure 7.16: Errors between the reference model and the plant states when the set point is $y^{sp} = 0.01[m]$ .....	200
Figure 7.17: Nonlinear linearizing controller signal when the set point is $y^{sp} = 0.01[m]$ .....	201
Figure 7.18: Position of the ball when the set point is $y^{sp} = 0.01[m]$ .....	201
Figure 7.19: Linear control signal when the set point is $y^{sp} = 0.09[m]$ .....	202
Figure 7.20: Error signal between the set point and the nonlinear plant output when the set point is $y^{sp} = 0.09[m]$ .....	202
Figure 7.21: Errors between the reference model and the plant states when the set point is $y^{sp} = 0.09[m]$ .....	203
Figure 7.22: Nonlinear linearizing controller signal when the set point is $y^{sp} = 0.09[m]$ .....	203
Figure 7.23: Position of the ball when the set point is $y^{sp} = 0.09[m]$ .....	204
Figure 7.24: Linear control signal when the set point is $y^{sp} = 0.15[m]$ .....	204
Figure 7.25: Error signal between the set point and the nonlinear plant output when the set point is $y^{sp} = 0.15[m]$ .....	205
Figure 7.26: Errors between the reference model and plant states when the set point is $y^{sp} = 0.15[m]$ .....	205
Figure 7.27: Nonlinear linearizing controller signal when the set point is $y^{sp} = 0.15[m]$ .....	206
Figure 7.28: Position of the ball when the set point is $y^{sp} = 0.15[m]$ .....	206
Figure 8.1: Block diagram of the Standard components of a PLC (NIIT, 2005).....	212
Figure 8.2: TwinCAT 3 extended architecture (TwinCAT, 2013).....	213
Figure 8.3: TwinCAT 3 standard (TwinCAT, 2012).....	214
Figure 8.4: Integrated system manager directory tree (TwinCAT, 2012).....	216
Figure 8.5: TwinCAT 3 integrated (TwinCAT, 2012).....	217
Figure 8.6: TwinCAT 3 runtime modular system (TwinCAT, 2014).....	217
Figure 8.7: TwinCAT 3 multicore support (TwinCAT 3, 2014).....	218
Figure 8.8: TwinCAT 3 multicore assignment (TwinCAT 3, 2013).....	218
Figure 8.9: Order of different CX5000 devices (Beckhoff, 2013).....	220
Figure 8.10: Diagram of the CX5020.....	221
Figure 8.11: Closed-loop algorithm for real-time communication.....	221
Figure 8.12: Code Generation process.....	223
Figure 8.13: Connectivity of a Simulink TwinCAT 3 module.....	225
Figure 8.14: Closed-loop feedback linearization-based system Model Transformation from Simulink to TwinCAT 3 PLC Function Blocks.....	226

Figure 8.15: Magnetic levitation real-time position when the set point is $y^{sp1} = 0.55[m]$ .....	227
Figure 8.16: Magnetic levitation real-time position when the set point is $y^{sp2} = 0.6[m]$ .....	227
Figure 8.17: Magnetic levitation real-time position when the set point is $y^{sp3} = 0.75[m]$ .....	228
Figure 8.18: Magnetic levitation real-time position when the set point is $y^{sp4} = 0.85[m]$ .....	228
Figure 8.19: Transformation of the model of the closed-loop system based on the Lyapunov and MRC from Simulink to TwinCAT.....	230
Figure 8.20: Position of the magnetic levitation system when the set-point is $y^{sp1} = 0[m]$ .....	231
Figure 8.21: Position of the magnetic levitation system when the set-point is $y^{sp2} = 0.01[m]$ .....	232
Figure 8.22: Position of the magnetic levitation system when the set-point is $y^{sp3} = 0.09[m]$ .....	232
Figure 8.23: Position of the magnetic levitation system when the set-point is $y^{sp4} = 0.15[m]$ .....	233
Figure 8.24: Block diagram of the distributed solution.....	235
Figure 8.25: Block diagram algorithm developed for the distributed implementation.....	236
Figure 8.26: Simulation of the distributed system when the set-points are: 1.1[m] and 1.24[m].....	236
Figure 8.27: Simulation of the distributed system when the set-points are: 1.15[m] and 1.45 [m].....	237
Figure 8.28: Simulation of the distributed system when the set-points are: $y_{Maglev1}^{sp3} = 1.2[m]$ and $y_{Maglev2}^{sp3} = 1.4[m]$ .....	237

## LIST OF TABLES

Table 2.1 Control techniques applied to control magnetic levitation systems.....	32
Table 2.2: Description of some IEC standards (IEC standards, 2013).....	54
Table 2.3: Comparison between SIMULINK and Function Block environment (Yang, 2010).....	56
Table 2.4: IEC 61499 related projects.....	61
Table 5.1: Characteristics of the position signal dynamic behaviour for the case of state feedback control.....	123
Table 5.2: Characteristics of the position signal dynamic behaviour for the case of state feedback with integral action.....	135
Table 5.3: Characteristics of the position transition behaviour for the case of a LQR with integral action controller.....	147
Table 6.1: Parameters of simulation.....	166
Table 6.2: Characteristics of the dynamic behaviour of the closed-loop system output for the case of input-state/input output and linear controls.....	179
Table 7.1: Parameters obtained for the LQR.....	192
Table 7.2: Simulation results comparison.....	207
Table 8.1: CX5020 technical data (CX5020, 2013).....	219
Table 8.2: Primitive data mapping between Simulink and TwinCAT3.....	224
Table 8.3: Discretization and Sampling Times.....	225
Table 8.4: Simulink versus real-time results comparison.....	229
Table 8.5: Simulink versus TwinCAT 3 real-time simulation results.....	233
Table 8.6: Parameters of simulation for the distributed system.....	235
Table 9.1: Programs used for pole-placement, LQR, feedback linearization and Lyapunov controller designs for Matlab simulation of the closed-loop systems behaviours.....	242
Table 9.2: Programs for linear and nonlinear controllers implementation	

in the TwinCAT 3 development of Beckhoff CX5020 PLC.....	242
<b>APPENDICES.....</b>	<b>245</b>
<b>APPENDIX A.....</b>	<b>245</b>
<b>APPENDIX B.....</b>	<b>264</b>
<b>APPENDIX C.....</b>	<b>265</b>
<b>APPENDIX D.....</b>	<b>266</b>
<b>GLOSSARY</b>	

<b>Terms/Acronyms/Abbreviations</b>	<b>Definition/Explanation</b>
AIM	Analog Input Module
AOM	Analog Output Module
CPU	Central Processing Unit
D/A	Digital to Analog Converter
DIM	Discrete Input Module
DOM	Discrete Output Module
E.M.F	Electro Magnetic Force
FB	Feedback
IEC	International Electrotechnical Commission
I/O	Input/Output
IOB	Input Output Block
IL	Instruction List
ILQR	Integral Linear Quadratic Regulator
LD	Ladder Diagram
LQR	Linear Quadratic Regulator
LS	Least Squares
LTI	Linear Time Invariant
LTV	Linear Time Variant
MagLev	Magnetic Levitation
MATLAB	Matrix Laboratory
MRC	Model Reference Control
O	Output
OP	Optimal Control Method
PC	Personal Computer
PI	Proportional Integral
PID	Proportional Integral Derivative
PLC	Programmable Logic Controller
PP	Pole Placement
PS	Power Supply
RISC	Reduced Instruction Set Computer
RT	Real-Time
SCADA	Supervisory Control and Data Acquisition
SIMULINK	Simulation and Link
SISO	Single-input Single-output
SP	Set-Point
ST	Structure Text
TwinCAT	Total Windows Control and Automation Technology

## MATHEMATICAL NOTATION

<b>Symbols/Letters</b>	<b>Definition/Explanation</b>
$\theta$	Phase angle
$\omega_n$	Natural frequency
a	State vector



$l$	Winding current
$\bar{g}$	Acceleration of the ball
$h$	Height of the ball
$k$	Proportional constant
$L$	Winding inductance
$m$	Masse of the ball
$R$	Winding Resistance
$r$	Reference Signal
$V$	Voltage
$\phi(x)$	State transformation matrix
$x$	State vector
$u$	Control vector
$f(x)$	Nonlinear function
$g(x)$	Nonlinear function
$w$	Disturbance vector
$u(k)$	Airflow rate
$x \in \mathbb{R}^n$	State vector
$u \in \mathbb{R}^m$	Control vector
$y \in \mathbb{R}^l$	Output vector
$A \in \mathbb{R}^{n \times n}$	Input state matrix
$B \in \mathbb{R}^{n \times m}$	Control output state matrix
$C \in \mathbb{R}^{l \times m}$	Output control matrix
$D \in \mathbb{R}^{l \times m}$	Output control matrix
$x_0$	Initial state space vector
$f \in \mathbb{R}^n$	Vector of the nonlinear function of the state
$h \in \mathbb{R}^l$	Vector of the nonlinear output function of the state
$H = [h_1 \quad h_2 \quad h_3]$	Full state feedback control gain
$\bar{C} = [B \quad AB \quad A^2 B]$	Controllability Matrix
$\bar{O} = [C^T \quad C^T A \quad C^T A^2]$	Observability check
$P_1 \quad P_2 \quad P_3$	Desired poles
$\varepsilon = \frac{\ln\left(\frac{\%OS}{100}\right)}{\pi^2 \ln^2\left(\frac{\%OS}{100}\right)}$	Damping ratio
$u = \frac{-L_f^3 h(x) + v}{L_g L_f^2 h(x)}$	Feedback linearizing controller
$z = \phi(x) = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix}$	Input/state coordinates transformation
$J = \int_0^\infty \left( \bar{x}^T \bar{Q} \bar{x} + u^T \bar{R} u \right) dt$	Performance index
$e = y^{sp} - y$	Error signal
$V = e^T P e$	Lyapunov function

## **CHAPTER ONE: PROBLEM FORMULATION, AIMS, OBJECTIVES**

### **1.1 Introduction**

In the past decades, the production plants in the industrial sector became much more difficult due to the constant changes in market specifications, which required manufacturing of small proportions of many customized products rather than mass production of a single product (Tsuchiya, 1999) and (Vyatkin, 2008). Several recent surveys have indicated that the always increasing demand for adaptability, flexibility, reconfigurability or robustness in automation and control systems domain imposes the need for new paradigms to address these requirements. For these requirements, manufacturing systems as well as control software applications result in an increase of the complexity of the task for design of process control to a point where the methodologies will fail. Development of new methods for design of control and control software are always the main elements in industrial control and automation systems to provide correct and safe operation of the corresponding process. Therefore, the development of new control design methods and software is essential. The IEC standard for control implementation in Programmable Logic Controllers (PLCs) and distributed control, IEC 61499 and the new software programming language based on Function Blocks (FBs) become one solution towards fulfilling these upcoming demands (Yang, 2010).

This chapter is divided in ten (10) parts. Section 1.1 is the introduction, it gives an overview of the challenges faced in industry today. The motivation for the research and problem definition is explained in Section 1.2. Section 1.3 focuses on the problem statement, and Section 1.4 gives the research aims and objectives of the thesis. Section 1.5 and Section 1.6 give respectively the hypothesis and the delimitation of research. The assumptions considered in the thesis are given in Section 1.7. The research methods applied are explained in Section 1.8. In Section 1.9 the thesis contribution is presented, and finally in Section 1.10 the outline of the thesis is given.

### **1.2 Motivation for research and problem definition**

Development of the research in the thesis faces several challenges. First of all, 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 to the world's one. Function block programming based on IEC 61499

standard and technology provides possibilities for the manufacturing companies to produce the suitable product to the market in short period of time with low cost and flexibility.

Secondly, from the country scientific competitiveness point of view, the research on real time control of nonlinear system with new advanced PLC is one of the latest, most complex and across research area in the world. Many companies, universities, institutes and even governments are investing in this new area of research to improve control solutions in industry. But, still there has not a real involvement from the South African's side. As in our knowledge the Cape Peninsula University of Technology is the only institution investing in this challenging field.

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

- Most of the industrial processes are nonlinear and are represented by nonlinear dynamic models of the system. These models are difficult to be used for design of controllers and their real-time implementation in PLCs because of the following nonlinear models difficulties:
  - a) They do not comply with the principle of superposition (linearity and homogeneity).
  - b) They may show properties such as limit-cycle, bifurcation, chaos.
  - c) They may have several isolated equilibrium points (linear systems can have only one).
  - d) Finite escape time: The state of a none stable nonlinear system can go to infinity in finite time.
- Difficulties for applying nonlinear control design theories:

The analysis of today's status of the existing methods for design of nonlinear controllers for nonlinear plants and of the possibilities to program these controllers' nonlinear expressions in the software environment of the existing programmable logic controllers shows the following:

- 1) The methods for design of nonlinear controllers are process dependent, not universal one.
  - 2) The software for programming of the existing IEC 61131 standard-based PLC is very difficult to be applied to program the nonlinear expressions of the nonlinear controllers.
  - 3) The new IEC 61499 standard-based software is convenient for programming of the nonlinear controllers but is very new one and it is mainly used for programming of linear controller in the universities research laboratories.
- Difficulties in the real-time implementations of the designed closed-loop system on IEC 61499 standard-based PLC environment. Many unsolved problems exist:

- a) Model transformation from Matlab/Simulink to PLC environment is very difficult to achieve.
- b) Understanding of function block programming based on IEC 61499.
- c) Correct sampling time for real-time simulation has to be evaluated.
- d) Many variables of the nonlinear plants are difficult to be measured.
- e) Many parameters are difficult to be determined, especially in the nonlinear processes.

### **1.3 Problem statement**

The research work in the thesis developed a methodology for design and implementation of nonlinear controllers in the frameworks of the IEC 61499 standard-based software and hardware environment for the case of a magnetic levitation system. The following problem is formulated:

To develop a methodology consisting of:

- Methods for design of nonlinear linearizing controllers for nonlinear plants.
- Software allowing use of nonlinear functions for simulation of the designed closed-loop systems.
- Transformation of the developed software to IEC 61499 standard-based software environment.
- Implementation of the designed closed-loop system in the PLC and real-time simulation.

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

#### **1.3.1 System modeling sub-problem**

##### **1.3.1.1 Sub-problem 1: Nonlinear mathematical model development**

The complete nonlinear mathematical model of the magnetic levitation is derived. The nonlinear mathematical model derivation is one of the most important tasks of the project.

##### **1.3.1.2 Sub-problem 2: Linear mathematical model development**

The complete linear mathematical model of the magnetic levitation is obtained by linearizing the magnetic levitation nonlinear mathematical model.

##### **1.3.2 Sub-problem 3: Linear and nonlinear magnetic levitation controllers design and development**

The linear controllers and nonlinear controllers design methods are applied for designing and developing the magnetic levitation controllers. Different modern control methods are applied,

such as: pole-placement method; linear quadratic method; model reference control method; Feedback linearization method and Lyapunov direct method, to achieve the desired system response.

### **1.3.3 System simulation and emulation sub-problems**

#### **1.3.3.1 Sub-problem 4: Mathematical models simulation**

In order to verify, confirm and finalize the system mathematical models, the models which are mentioned in section 1.3.1.1 and 1.3.1.2 are simulated in Matlab/Simulink software environment.

#### **1.3.3.2 Sub-problem 5: Closed-loop system simulation**

The closed-loop systems, which consist of the magnetic levitation 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 real-time implementation platform.

### **1.3.4 Real-time implementation**

Matlab/Simulink software is used first as a model verification platform before model transformation. The closed-loop system is transformed from Matlab/Simulink to TwinCAT 3 IEC 61499 standard-based function block programming environment for real-time implementation. The method is used to reduce the programming development time. The real-time program is uploaded into the CX 5020 PLC for simulation.

### **1.3.5 Results analysis**

#### **1.3.5.1 Sub-problem 6: Simulation results and analysis**

The systems performances are studied by analyzing 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 comparison, the effects of each control method on the magnetic levitation plant behaviour are studied.

#### **1.3.5.2 Sub-problem 7: Real-time implementation results**

The real-time implementation platform is selected according to the requirement of the research project. The obtained real-time implementation results from the selected software and hardware platforms and the different control strategies are compared.

## **1.4 Research aim and objectives**

### **1.4.1 Research aim**

The aim of this project is to develop a methodology for nonlinear control designed and implementation in Beckhoff CX5020 Programmable Logic Controller (PLC) using function block programming approach.

### **1.4.2 Research objectives**

The methodology is developed based on the following objectives:

- Develop nonlinear model of the magnetic levitation plant.
- Develop linearized model of the magnetic levitation plant.
- Design of a linear controller for the linearized model of the magnetic levitation plant.
- Design of a nonlinear linearizing controller based on the input state feedback linearization theory.
- Design of nonlinear linearizing controller based on the second method of the Lyapunov theory and a reference model based controllers.
- Simulation of the closed-loop systems in Matlab/Simulink.
- Model transformation of the closed-loop systems from Matlab/Simulink to the IEC 61499 function blocks using TwinCAT 3 software environment.
- Implementation of the closed-loop systems in real-time using Beckhoff CX5020 PLC.
- Experimentation and analysis of the results.

## **1.5 Hypothesis**

The hypothesis is connected with the possibilities of the developed software for the IEC 61499 function blocks programming environment to implement nonlinear controllers in PLC.

- 1) The thesis proves that the nonlinear controllers of the magnetic levitation plant give better performance of the closed-loop system behavior than that given by the linear controllers. The parameters of the nonlinear controller could be changed in real-time from the TwinCAT 3 system and the set-point of the nonlinear controller could also be changed in real-time from the TwinCAT 3 system.
- 2) The real-time implementation and simulation of the designed closed-loop system in IEC 61499 standard-based environment produces the same results as these obtained from Matlab/Simulink simulations.

## **1.6 Delimitation of research**

The research is mainly based on designing and implementing nonlinear controllers in the Programmable logic controller (PLC) because the linear controller do not give good performance of the nonlinear magnetic levitation system in real-time. The real data is collected by using TwinCAT 3 on the basis of personal computer (PC) and Beckhoff CX5020 PLC. The problem for design of nonlinear control is solved using the Benchmark models of magnetic levitation plants. Linear and nonlinear control theories are used for design of the controllers in state space domain. The classic control theory based on Laplace domain representations is used for design of the linear controller.

The real-time simulations are performed for the designed closed-loop systems.

## **1.7 Assumption**

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

- The magnetic levitation plant is working to the prescribed instructions.
- The mathematical models can be derived according to the existing physical laws.
- The linearization of the nonlinear model can lead to linear models capable of being used for design of linear controllers in neighborhood of the selected equilibrium points.
- The linear and nonlinear control methods and theories are powerful enough to support the design of linearizing controllers.
- The Matlab/Simulink simulation results provide good approximation of the real-time implementation results.
- The hardware implementation can provide high performance of the real-time simulation, based on the high speed Gigabit Ethernet communication between the computer and the PLC.

## **1.8 Research methods**

### **1.8.1 Literature review method**

Literature review introduces magnetic levitation, different control methods used to bring the system to stability and function blocks programming based on IEC 61499 standards for implementation in programmable logic controllers.

### **1.8.2 Design methods**

- Feedback linearization control theory is used for design of a nonlinear linearizing controller on the basis of the linear reference model.
- Method of Lyapunov theory for stability is used for design of a nonlinear linearizing controller on the basis of a linear reference model.

### **1.8.3 Simulation and experimental methods**

- The closed-loop systems based on the designed two nonlinear controllers are simulated in Matlab/Simulink environment.
- Implementation of the closed-loop system in PLC and TwinCAT 3 software environment.
- Real-time simulation of the closed-loop system PLC/TwinCAT 3 environment.

### **1.8.4 Description method**

The developed methods, Matlab simulations and PLC real-time simulations are described with all their specifications, characteristics, functions. The obtained results are described in the thesis too.

### **1.9 Thesis contribution**

The contributions of this thesis are the following:

- The research shows that it is possible to achieve semi-automatic model transformation from Matlab/Simulink models to IEC 61499 standard-based function blocks models for implementation in PLCs.
- The closed-loop model transformation approach can be successfully used to control distributed systems in the field of nonlinear control.
- IEC 61499 function block methodology is implemented with demonstrated examples to boost the acceptance of the standard in industry.

### **1.10 Outline of the thesis**

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 consists of the background of the project. The challenges and problems faced by the manufacturing industry in South Africa are discussed. The project statement include: the research aim and objectives, statement of the problem, hypothesis, research delimitation, research motivations, assumptions and research methods.

Chapter 2 presents the literature review of the existing works that are related to this project. It introduces magnetic levitation system, and its existing control techniques to bring the system to stability. The IEC 61499 standard-based function blocks programming for real-time implementation is briefly discussed.

Chapter 3 presents the magnetic levitation detailed mathematical nonlinear and linearized models derivation. The magnetic levitation plant is described first. Based on the understanding of the magnetic levitation physical structure, the nonlinear and linearized mathematical models are derived. The linearized model of the plant is simulated in Matlab.

Chapter 4 provides the background study of both linear and nonlinear control theory by referring to the control of the magnetic levitation process. The typical linear control strategies include the root locus method, the pole-placement, and the quadratic optimal control method. The typical nonlinear control strategies include the model reference control method, Lyapunov stability theory, and feedback linearization method.

Chapter 5 describes the linear controller design based on the pole-placement method and the LQR method for linearized model of the plant and the corresponding closed-loop simulation results. Comparison of the results from simulation is done and the LQR method for design of an integral linear controller is selected to be used further for improving the performance of the feedback linearized closed-loop system.

Chapter 6 presents the nonlinear controller design based on the feedback linearization method and describes the closed-loop simulation results. The linearized model states are selected through input-state linearization. And the mathematical expression of the linear linearizing controller is obtained. By using the integral LQR method, a linear controller is designed for the linearized closed-loop system. The closed-loop is simulated in Matlab/Simulink environment and the simulation results are shown and discussed.

Chapter 7 describes a nonlinear linearizing controller design based on the Lyapunov Direct Method (LDM) and a reference model. The feedback linearized closed-loop system behaviour is improved by design of an integral LQ controller. The closed-loop system is simulated and the simulation results are presented.

Chapter 8 describes the real-time implementation of the designed closed-loop systems based nonlinear linearizing controller on IEC 61499 standard-based PLC. The model transformation is performed to move the closed-loop algorithms from Matlab/Simulink environment to TwinCAT 3

function blocks on the basis of IEC 61499 standard. Real-time simulations are performed and the results are compared with the results obtained from the simulations in Matlab/Simulink.

Chapter 9 describes the deliverables of the thesis and outlines the future research in the field of nonlinear control in programmable logic controllers.

## **CHAPTER TWO: LITERATURE REVIEW**

### **2.1 Introduction**

The research work of this project focuses on the development of an algorithm for nonlinear control implementation in an IEC 61499 standard compatible Programmable Logic Controller (PLC). The ever increasing need for flexibility, robustness and re-configurability in automation and control systems requires the need for new platforms to address these issues. The concept of distributed systems has grown in interest in past decades because of the growing demand of a computer based control system used to control the production lines in industry. In distributed control system the entire system of controllers is connected by network for communication and monitoring. To materialize these features the International Electro-technical Commission (IEC) developed an open architecture for the next generation of distributed control and automation: IEC61499 standard. To demonstrate the functionality of IEC61499 a magnetic levitation system (MagLev) is selected as a plant to be controlled because it is a highly nonlinear system that suits well with the basic IEC61499 strategies. The closed loop control of the MagLev system is done in MATLAB/SIMULINK; the system closed loop system is then transformed in basic function blocks based on a full block to block transformation method in IEC61499 designed on Beckhoff TwinCAT 3 software. If the simulation shows the expected results then the Function Blocks closed loop system can be deployed to the physical PLC. Details on the Magnetic Levitation System are given in Chapter 3 which focuses mainly on its modeling.

The introduction of IEC 61499 standard was officially done in 1999 by Vyatkin and Hanisch which is internationally recognized as the first real approach to deal with distributed control systems and IEC 61499 in a formalized way. That was more than a decade ago, and some authors of this important contribution who introduced the research work in 1999 had the impression that almost nobody in the audience had any idea of what the work was all about.

Fortunately for the evolution of the standard, the situation has positively improved since that time. Unfortunately, a huge area of unsolved problems and questions remains.

Therefore, the main focus of this chapter is to review the works done up to now in both magnetic levitation and distributed control systems to make the acceptance of the standard possible and to address the upcoming issues in future.

This chapter is divided in ten (10) parts focusing on the main parts of the research topic. Section 2.1 is the introduction of the research project and an overview of the existing tools to achieve distributed control. Section 2.2 explains magnetic levitation. Section 2.3 gives an overview of the control techniques applied to control the highly unstable magnetic levitation system.

Section 2.4 explains the concept of distributed control in details. In Section 2.5 the motives behind the creation of the IEC 61499 standard are given. Section 2.6 focuses on the comparison between the IEC 61131 and IEC 61499 standards and an idea of other related standards is described. In Section 2.7 the description of the modeling and design platform that plays a major role in the acceptance of IEC 61499 in industry is given. Section 2.8 depicts the two existing modeling techniques for implementation of IEC 61499. In Section 2.9 a comparative literature review is presented and in section 2.10 the conclusion of the chapter is given.

## **2.2 Magnetic levitation**

### **2.2.1 Description**

Magnetic levitation is a fairylike phenomenon that has always captivated the attention of people. It is an interesting highly nonlinear and open loop unstable system suitable for control systems applications. Nowadays levitation has the potential to improve numerous sectors such as levitation vehicles, magnetic bearings, aerodynamics and noise mitigation, and fiber reinforced plastics for vehicles and structural concretes Wong (1986), Blanchid *et al.* (1994) and Suster *et al.* (2012).

Magnetic actuation is very important in harsh environments where conventional mechanical or hydraulic actuators might not survive. At an ecological level, it presents massive benefits to the protection of the environment and to strengthen land cleaning. It would reduce the destruction of green vegetation that covers railways, the location and the barrier against animals, the loss of wild animals and natural resources.

This phenomenon is typically carried out by using actively controlled electromagnets. The two principal means of levitation are: electro-magnetic suspension (EMS) and electro-dynamic suspension (EDS).

### **2.2.2 Electro-magnetic suspension (EMS)**

This phenomenon is widely used in advanced trains in which electronically controlled electromagnets attract a magnetically conductive track. The train levitates above the railways while electro-magnets attached to it are orientated toward the rail from below. Magnetic attraction changes inversely with the cube of distance, so slight changes in gap between the magnets and the rail produce significantly varying forces. Unlike electro-dynamic systems which only operate at a minimum speed of 30 km/h, the major advantage of magnetically suspended systems is that they work at full speed. This eradicates the necessity for a separate low-speed

suspension system, and can make the track layout simple as a result. On the downside, the dynamic uncertainty of the system requires high tolerances of the track, which can offset, or remove this advantage (Jayawant, 1981), (Xu, 2011) and (Aly, 2012).

### **2.2.3 Electro-dynamic suspension (EDS)**

This concept utilizes superconducting electromagnets or strong permanent magnets which generate a magnetic field that induces currents in neighbouring metallic conductors when there is a motion which pushes and pulls the train towards the designed levitation position on the railway. The main drawback of electro-dynamic suspension is that it induces a current in its coils at low speeds and the resultant flux is not big enough to carry the weight of the train. To avoid such potentially chaotic scenario, the train must have wheels or a reliable form of landing gear to support the train until it gets to a speed at which it can sustain levitation. Such system is quite problematic because the train may stop at any position due to faulty equipments and to overcome this potential issue, the entire track must be able to handle both low and high speed operations and is based on the work of Wilt (1997), Scribd (2010) and Long (2011).

### **2.2.4 Magnetic ball levitation**

The magnetic levitation experiment consists of a magnetic ball suspension system which has for aim to levitate a steel ball in the air by applying an electro-magnetic force produced by an electro-magnet. The whole set up consists of an electro-magnet, a steel ball, a ball at rest and a ball position sensor.

The objective of the experiment is to levitate the ball from its initial position and track its specified trajectory. The magnetic levitation is composed of two sub-systems:

- Electrical
- Mechanical

The ball position in the mechanical sub-system can be controlled by an electro-magnetic force produced by the inductor in the electrical sub-system which depends on the intensity of the current through the inductor whereas the current through the inductor is controlled by applying a voltage across the inductor terminals. Therefore, the voltage across the inductor terminals provides an indirect control of the ball position.

The design and control of magnetic levitation has caught the attention of many researchers. Wong, (1986) stated that there are generally two methods to design magnetic levitation system. One way is by using the magnetic repulsive force and the other way is by using electro-magnetic attractive force. In his design, he applied the second approach because it is more efficient that

the first one in terms of energy consumption. Ahmad and Javaid, (2010) developed a non linear dynamic model for Magnetic levitation system by calculating the inductivity of the system.

### **2.3 Control of the magnetic levitation system**

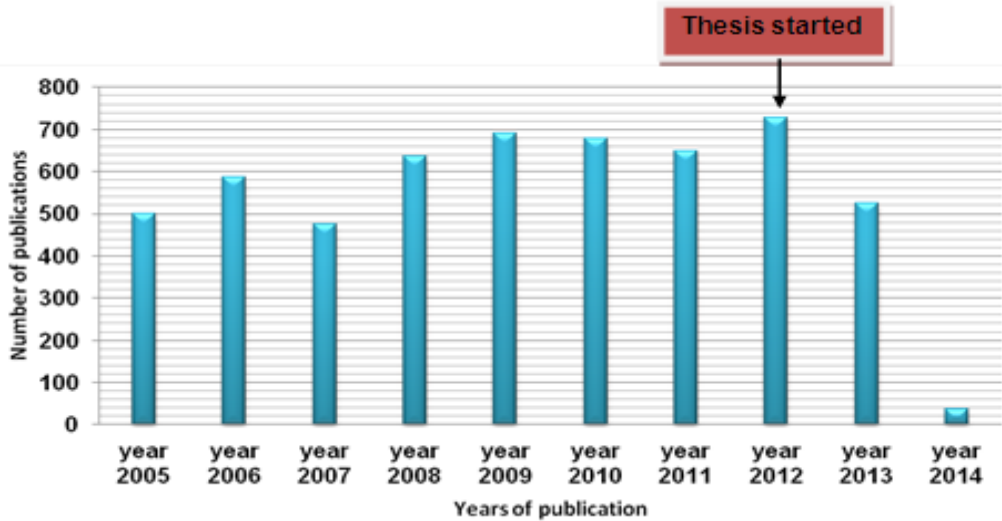
The control of a magnetic levitation system is a very demanding task as it is an unstable system and its inherent nonlinearities make the modeling and the control problem very challenging. One of the most important parts in the control systems is to develop an accurate model of the system under control to be able to obtain satisfactory results because it is impossible to control a system whose behaviour is not accurately defined. Researchers have shown that different methods can be applied to control the magnetic levitation system. Wong, (1986) developed an approximate linear model based on linearized equations describing the variations from the operating point by only using the linear terms of Taylor series. A linear phase-lead compensator was used to bring the system to stability for step responses of 1.5 mm around the operating point and satisfactory results of the linear control on nonlinear systems were obtained.

Cho *et al.*, (1993) designed a Sliding Mode Controller (SMC) to correct disturbances and model uncertainties of the magnetic levitation system and obtained superior performance of the SMC over classical control. Blanchid *et al.*, (1994) conceived a robust control based on state feedback technique and developed computer generated Lyapunov functions to stabilize magnetic levitation. They found that the system was extremely accurate, stable and did not cause sampling ripple effects. Barie and Chiasson, (1996) designed linear and non-linear state space controllers to compare both control strategies; they obtained good tracking of high frequencies trajectories. Yang and Tateishi, (1999) demonstrated that adaptive robust non-linear controller achieves better control performance than the classical PI control to track the ball position in magnetic levitation. Hosseini *et al.*, (2002) combined fuzzy-sliding-mode and a state-feedback controllers and observed fast response and low steady error in magnetic levitation. Moghaddam and Ganji, (2011) proposed a methodology to control a magnetic levitation system with uncertain parameters in the dynamics and the measurements. They mixed sliding mode control and hybrid extended Kalman Filter techniques and obtained an efficient and robust controller. Šuster and Jadlovská, (2012) designed a control algorithm for magnetic levitation systems based on input–output feedback linearization and pole placement methods, the system was stable and tracked accurately the reference input.

Table 2.1 summarizes the different control methods developed by researchers and their outcomes, and Figure 2.1 shows the evolution of the number of paper published from 2005 to

January 2014 by the IEEE (Institute for Electrical and Electronic Engineers) with regard to the field of nonlinear control of nonlinear plants.

In this thesis feedback linearization control is developed because it is a more flexible method and can provide an accurate linear dynamic behaviour of the closed-loop nonlinear unstable system. Then it is possible to design a linear controller for the obtained linearized system. The next section focuses on the distributed control techniques based on IEC 61499 standard.



**Figure 2.1:** Publication rate of nonlinear control of nonlinear plants

From the year 2005 to January 2014, thousands of papers have been published in the field of interest, namely nonlinear control of nonlinear plants. The graph shows clearly that there is peak in the number of publications for the year 2012, which is the year at which the current Master’s thesis commenced. This histogram serves to illustrate the importance of the research field.

**Table 2.1: Control techniques applied to control magnetic levitation systems**

Paper	Aim of the paper	Method of control	Structure of the system	Plant under control	Simulation	Implementation	Advantages/ Drawbacks	Achievements
(Wong, 1986)	Design of magnetic levitation system	Phase lead compensator	Single control solution	Magnetic Levitation System	MATLAB	Amplifiers RC circuit	Fast response, Not robust	Satisfactory results of linear control on non-linear systems
(Cho et al., 1993)	Correct the disturbances and model uncertainties	Sliding mode controller (SMC)	Single control solution	Magnetic Levitation System	MATLAB	BM PSI2 Model 80 with an Analog Devices RTI205 multifunction board	Very Stable. Excellent tracking.	Superior performance of the SMC over classical control
(Blanchid et al., 1994)	Computer generated Lyapunov functions to stabilize Magnetic Levitation	Robust control based on state feedback	Single control solution	Magnetic Levitation System	Not mentioned	486 personal computer	Extremely accurate, stable and did not cause sampling ripple effects	At initial conditions no saturation occurs during the transient time
(Barie and Chiason, 1996)	Compare two control strategies	Linear and non-linear state space controllers	Single control solution	Magnetic Levitation System	MATLAB	Motorola DSP 56001, Aerotech 4020 linear amplifier	Tracking of high frequencies trajectories.	Application of non-linear control in magnetic bearing systems
(Charara et al., 1996)	Correction of inherent non-linearities of suspension systems	Feedback linearizing controller	Single control solution	Active magnetic bearing (wheel suspended against gravity)	MATLAB	Digital signal processor (DSP)	Excellent performance, requires accurate modeling	Sampling rates for discrete-time implementations have been shown to be important, especially at small air gaps.
(Trumper, et al., 1997)	Correction of inherent non-linearities of suspension systems	Feedback linearizing controller	Single control solution	Basic steel ball levitation system	MATLAB	90MHz Pentium based digital computer	Excellent performance, requires accurate modeling	Sampling rates for discrete-time implementations have been shown to be important, especially at small air gaps.
(Kaplan and Sarafian, 1997)	Stabilize the complicated motion of a	Chaos control technique	Single control solution n	Basic lab scale MagLev	Not specified	LCR tuned circuit	Fast and stable	Implementation of chaotic control system for practical



	suspended object							applications
(Yang and Tateishi, 1999)	Achieve position tracking of MagLev	Robust non-linear control	Single control solution	Lab MagLev	Not specified	Current feedback power amplifier	Stable, good transient performance	Adaptive robust nonlinear controller achieves better control performance than classical PI control
(Hassan, and Mohamed, 2001)	Robust stabilization and disturbance rejection	Variable structure control (VSC)	Single control solution	One degree of freedom lab MagLev	MATLAB	Personal computer	Robust, stable and Reject disturbances	Robust stability against parameter perturbations is achieved.
(Hosseini et al., 2002)	Application of fuzzy sliding state feedback	Fuzzy-sliding-mode and state-feedback control	Single control solution	Magnetic Levitation System	MATLAB	LCR circuit	Fast response and low steady state error	A combination of the fuzzy-sliding-mode and the state-feedback control law was achieved.
(Shameli et al., 2007)	Nonlinear controller design for sub-micron positioning accuracy.	Nonlinear control based on Lyapunov stability criteria	Single control solution	Electromagnets, disc pole and an iron yoke	Not mentioned	Current control amplifier and laser sensor via a 16-bit A/D converter and a 16-bit D/A converter	Stable, precise, good tracking	Position control of a magnetically levitated permanent Magnet based on Lyapunov stability second law.
(Torres et al., 2010)	Combination of two techniques to control a nonlinear system	Exact linearization with state Feedback and Adaptive control	Single control solution	ECP MagLev	MATLAB/SIMULINK	N/A	Tracks reference input, small error, fast transitory and no oscillation	Adaptive controller is able to control satisfactorily the magnetic disc position, in spite of the presence of model uncertainties and non-modelled dynamics.
(Ahmad and Javaid, 2010)	Development of a nonlinear dynamic model for MagLev System	Linear and nonlinear state space controllers	Single control solution	Ferromagnetic ball suspended in a voltage controlled magnetic field.	MATLAB	N/A	Stability is valid in a large region, controller cannot bring the system to equilibrium	Dynamic model of the levitation system for each mode of calculating the inductivity
(Moghaddam and Ganji, 2011)	Proposes an approach to control a magnetic levitation system with uncertainty	Combination of sliding mode control and Hybrid Extended Kalman Filter	Single control solution	Experimental Set-up of Magnetic Levitation System	MATLAB	Kalman filter	Efficient and robust	Correction of the tracking performance of the sliding mode controller due to uncertainties

	in the dynamics and the measurements.							
(Gazdo et al.,2011)	Proposes a methodology for robust control of unstable systems.	Robust control based sensitivity functions and their $H^\infty$ norms.	Single control solution n	CE 152 magnetic levitation apparatus	MATLAB/Simulink	Computer and CE152 operating as a feedback loop	Stable, reject disturbances	The experimental results on the magnetic levitation system show applicability of the approach to safer control of unstable processes.
(Šuster and Jadlovská, 2012)	Design of control algorithm for the Magnetic levitation system	Input –output feedback linearization method and a pole placement method	Single control solution	Magnetic levitation CE 152 of Humusoft	MATLAB/Simulink	Control PC linked to a lab card and a CE 152	Tracking of reference input, stable	Utilization of Magnetic levitation to support teaching in the Optimal and Nonlinear system subject.
(Jitha, 2013)	Control of a magnetic levitation suspension system for reaction wheel	Linearization and phase lead compensation techniques.	Single control solution	Magnetic levitation suspension system	MATLAB/Simulink	Amplifiers	Excellent performance, accurate plant model	Feedback linearization gave optimum performance from magnetic bearings.
(Kaldmae and Kotta, 2013)	Input-output linearization controller design for discrete time nonlinear control systems.	Input-output linearization by dynamic output feedback	Single control solution	Set of input-output equations	N/A	N/A	N/A	Developed an algorithm to find the set of functions of a nonlinear MIMO system.
(Chang and Hung, 2014)	Design of an input-output linearizing controller for nonlinear control of proton exchange membrane fuel cell.	Input/output state feedback linearization	Single control solution	Proton exchange membrane fuel cells (PEMFCs)	MATLAB/Simulink	N/A	Better transient and steady state performance.	The PEMFC with input/output state linearizing controller has better transient and steady-state performance compared to conventional linear techniques.
(Enev, 2014)	Design of an linearizing controller for decoupled torque and stator flux control.	Discrete-Time Modeling and Input-Output Linearization	Single control solution	Induction motor	MATLAB/Simulink	N/A	Stability, no coupling exists between the outputs	Application of input-output linearization technique on induction motors.

## **2.4 Concept of distributed control system in IEC 61499**

As cited in Yang, (2010), the concept of the real distributed systems comes from the implementation of control in manufacturing industry. This involves physical components such as switches, pumps and valves which are directly linked to their personalized control unit. The idea of distributed systems is to substitute a centralized control unit with distributed controllers that control a specific part of a system. This implies that every element or subsystem inside a system can possibly have its personalized control unit. Distributed control possesses diverse implementations in various fields like Electrical Power Generation, Manufacturing, Pharmaceutical Manufacturing, Oil Refining etc.

Chokshi and McFarlane, (1963) proposed that the re-configurability of the system is the main concept to be taken in consideration as the target of distributed control. Currently, no genuine software tools and design architecture have yet been fully developed for the implementation of distributed process control in industrial practice.

Wei, (2001) successfully investigated the implementability, reliability and performance of IEC61499 standard on process control. A system development tool was developed using JAVA. The apparatus enables users to download and visualize function blocks, remotely set up the system, and then the function blocks are downloaded to field devices to remotely enable the devices on the network. His findings showed that design time is greatly reduced, cost at the engineering side is also reduced and system will be more flexible and maintainable.

The concept of distributed power system automation was developed by Karlheinz, *et al.*, (2008) who discussed the means to achieve flexible power system automation. His concept highlights the use of IEC61499 as integration, extension and verification apparatus for IEC61850 based system. The results of this research work enhanced the implementation on IEC61499 compliant platform in devices such as bay controllers, protective relays and substation controllers.

Strasser, *et al.*, (2008) worked on the programming of a distributed control of a six (6) degrees of freedom modular robot which is made of four remote devices. One device is remotely implemented on the visualisation PC; the three other are implemented on the PC/104 embedded hardware furnished with FORT, a small portable implementation of an IEC 61499 runtime environment which targets compact embedded control devices (16/32 Bit). This research showed that large algorithms of distributed automation and control programs are operated advantageously with IEC61499 function blocks.

Zoitl and Vyatkin, (2009) demonstrated that IEC61499 describes several means which can help to enhance software quality and lessen the development effort of distributed control systems. However, it also pinpointed some open matters to be solved. How fast these concerns can be solved will have an effect on the success of IEC61499.

Schimmel and Zoitl, (2010) showed how to utilize switched Ethernet for real time application modelled in IEC61499. This method works based on the closed system approach, in which the emerging communication is essentially predefined by the application. The results that it is possible for distributed real-time automation systems to use switched Ethernet by certifying the real-time requirement of the application with the less advantageous communication delay.

William and Vyatkin, (2010) discussed the issue of moving from the PLC control based methodology to the event-driven and component based architecture of IEC61499 function blocks. They demonstrated the concept with the use of a conveyor belt to illustrate the three migrations methods which are: object oriented built in basic function blocks, object oriented built in composite function blocks and class oriented built in basic function blocks. The results of the research showed that class oriented architecture permit the re-utilization of the whole PLC algorithm which reduces the migration time. There is no mandatory data to be streamed around the function blocks to minimize the execution time of the program. Such requirements are more optimal for distributed control systems.

Despite the importance of the full integration of IEC61499 in industry, some major issues have been raised by researchers. Yoong and Partha, (2010) showed that the standard lacks of semantic strictness essential for automated verification of function blocks programs. Several solutions to address this issue have been proposed, but these have so far concentrated on verification of control properties by extracting data from the program. This research work resulted to the validation of a tool for interpreting function blocks to Esterel synchronous programming language in order to check safety properties of function blocks algorithms using reliable tools for Esterel. The ability to achieve such functionality is significant, as mature tool for function blocks verification are virtually non-existent. At present, ongoing researches are done to develop a way to express counter-examples directly in terms of function blocks syntax.

However, at the beginning, an issue with this distributed process control depends on the feasibility of the implementation in industry, where currently there are no conventional design methods and software tools fully developed. Further examination show that the tools for human interaction are also crucial to define human implication in distributed control. Therefore several researches have been conducted by following this distributed concept, including the field of:

- Intelligent Control
- Multi-agent Systems
- Function Block concept of IEC61499
- Drawbacks in Distributed Implementation

#### 2.4.1 Intelligent control

Intelligent control is a combination of high-level decision making: using computers, and advanced mathematical modelling and synthesis methodologies of systems theory.

Gray and Cardwell, (2012) stated that intelligent control systems integrate several kind of data, including task specification, and the task state from sensory data integration. An intelligent control system must deal with the information about its own state and the state of the environment too, and is being capable of reasoning under uncertainties.

(He, *et al*, 2001) and (Aldea, *et al*, 2004) proposed a research field in manufacturing which deals with intelligent control methodologies working on biological concepts. Intelligent control and Model Predictive Control (MPC) are favoured in the ongoing process related research. The integration of these control methods with distributed control has the means to become an interesting research field in the future.

Saridis, (1990) stated that intelligent machines require the use of 'generalised' control methodologies to achieve intelligent functions such as:

- The simultaneous utilisation of memory.
- Learning or multi-level decision-making to respond to 'fuzzy' or qualitative commands.

Figure 2.2 shows a diagram developed by Saridis to highlight the three major sub-systems for machine intelligence: sensor, actuator and control.

Astrom and McAvoy, (1992) asserted that the term “Intelligent control” is commonly applied to the ability of controllers to understand and learn about processes and operating conditions, and Hangos, *et al.*, (2001) explained that it has come out as an interdisciplinary field of computerized control systems and artificial intelligence (AI) computing approaches, such as fuzzy logic, neural networks and machine learning, etc. Also an intelligent system is capable to upgrade its performance automatically with time. One benefit of intelligent control is to keep the upgrading to more recent system, which is based on the data and knowledge of the system. A knowledge-based system is mainly used in a distributed system for supervisory purposes. It functions as an assistant operator. In real-time, it adjusts and monitors control loops, on the basis of the real-time information collected. It is also extremely vital in alarm examination and process diagnosis.

Yao, *et al.*, (2005) presented the intelligent knowledge-based supervisory control (IKBSC) system to implement in a hot rolling mill. His research shows the advantage of a knowledge-based system as IKBSC supports in programming for an accurate, timely and safe rolling.

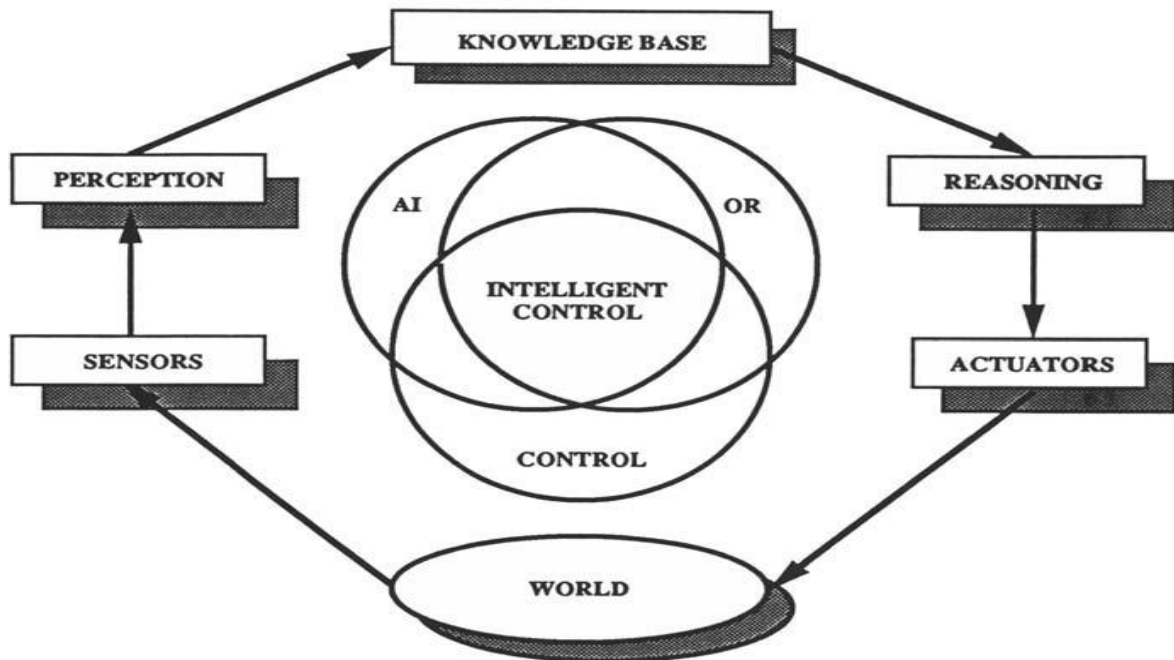


Figure 2.2: Architecture for intelligent control (Saridis, 1990)

## 2.4.2 Multi-Agent systems (MAS)

### 2.4.2.1 Overview

Luck, *et al.*, (2005) presented in the 1990s the multi-agent systems as one of the most attractive fields of research development to flourish in information technology (IT). They defined an agent as a computerized system that possesses the ability to adjust the actions of autonomous systems in dynamic, unpredictable, typically multi-agent areas. The concept of multi-agent systems is found in sub-disciplines of IT such as:

- Computer Networks
- Software Engineering
- Artificial Intelligence
- Human-Computer Interaction
- Distributed and Concurrent Systems
- Mobile Systems
- Telematics

- Computer-Supported Cooperative Work
- Control Systems
- Decision Support
- Information Retrieval and Management
- Electronic Commerce

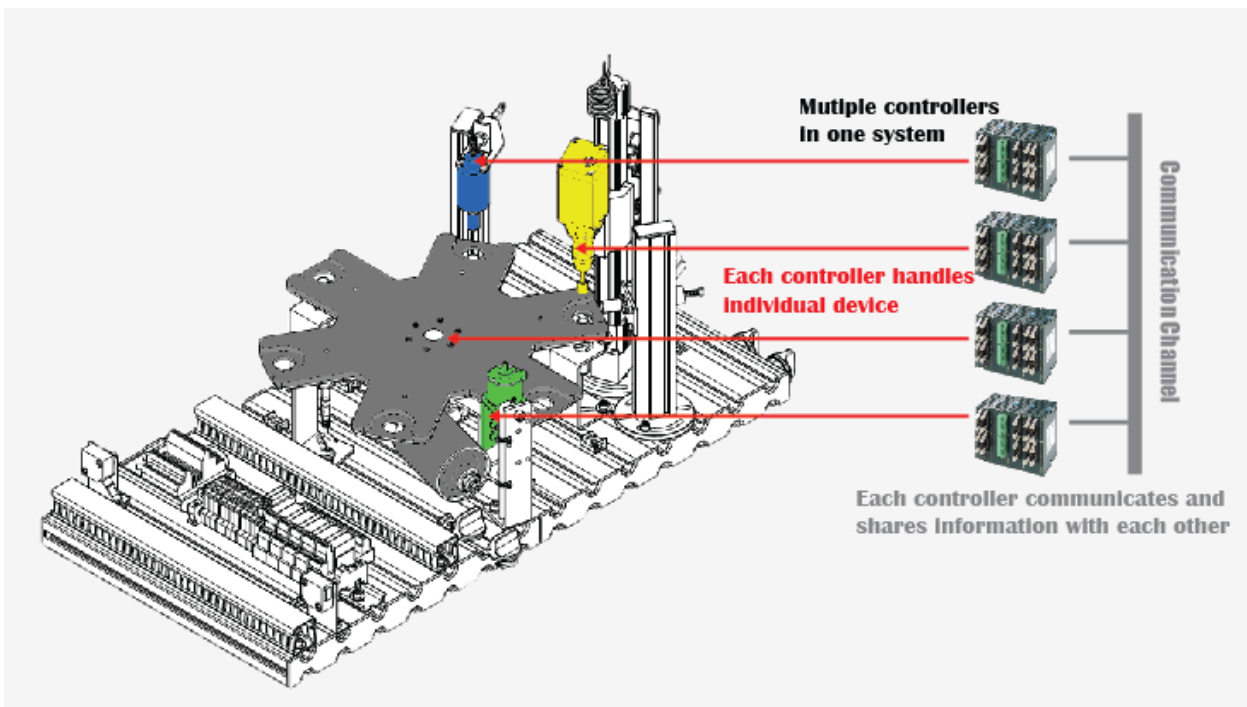
Tatara, *et al.*, (2007) stated that because of the complex form of distributed systems, agent-based control becomes a popular method and a powerful tool. Hall, *et al.*, (2007) stressed that primary implementations of system distribution involved dividing the control program into distinct components while keeping the structure of the code identical. While this method showed the idea of physical distribution to a certain degree, the functionality of the system had no concept of logical separation. Bader and Pennington, (2001) explained that the code structure would have the same performance as a single process ran by a computer cluster tightly coupled.

Russell and Norvig, (2003) investigated more contemporary techniques, describing multi-agent system on the basis of intelligent, goal-oriented agent and autonomous notions. The concept of an agent is understood to represent the combination of software and hardware autonomously combined to interact with the environment. The definition of a multi-agent system is given as being a system made of a cluster of agents designed to work and communicate together to make sure that a common goal is achieved. Further details of multi-agent system can be seen in Figure 2.3.

Aldea, *et al.*, (2004) explained that dynamic distributed systems are made very robust and flexible by the agent-based architecture methodology. Due to its modular form, it is also suitable to develop modular systems without a single centralized controller.

Hangos, (2001) cited that the concept of agent-based system merges both local and global controller agents that organize all the information gathered from the whole system. For example in the industrial process sector, this approach is specifically helpful when working with a class of networks to interconnect continuous stirred tank reactors (CSTRs).

Aldea, *et al.*, (2004) presented some examples of MAS applications in industry, which includes an intelligent search system to create a platform of knowledge management and a system to supply concurrent process design to facilitate information sharing between system engineers.



**Figure 2.3: Multiple controllers individually control a device (Aldea, et al., 2004)**

Polaków and Metzger, (2010) presented a research architecture supporting calculations task in control theory. They demonstrated that theory-based calculations with real-world control systems can consolidate actuators and hardware sensors methods. A new protocol based on networking was developed precisely for this application, using producer-distributor-consumer data distribution algorithm over Ethernet network. The results of this research confirmed that the agent based framework allows time-synchronized thread to execute and support interconnectivity between real world instrumentation and analytical computations.

#### **2.4.2.2 Challenges in Multi-Agent Systems**

The concept of multi-agent systems presents a number of challenges. Hayden, *et al.*, (2008) stated that one of the basic barriers to the acceptance of agent technology in industry is the current absence of fully developed software methodologies to develop agent-based systems. The main task in hand to achieve agent based computing systems is to improve the already available solutions in the domains of scalability, security, transaction management, etc., to suit the multiple requirements of the new paradigm while learning from reliable methods.

Luck, *et al.*, (2005) showed that specific technical challenges keep on changing as the research area of agent-based computing improves and gets more mature; the broad areas are as follow:



- Agent systems developers must create supporting methodologies, tools and techniques.
- Make the requirement to develop and manage agent systems automatic.
- Establishing suitable compromise between adaptability and predictability.
- Integrating components and features.
- Establishment of proper connection with fields such as economics, sociology and biology, and other domains of computer science.

### **2.4.3 Drawbacks in distributed implementation**

Despite the development of numerous applications based on the concept of distributed control in various sectors in industry, there remain drawbacks to the adoption of large-scale distributed control technologies. The following weaknesses have been cited by Yang, (2010):

- The risks coming with every new technology that is yet to be proven in industry.
- Reliable design and development tools are lacking for development in industry.
- There are no methodologies that can make the integration of the new technology simple.

## **2.5 Creation of the IEC 61499 Standard**

The need for decentralized control systems is ever increasing in industry due to several reasons such as flexibility and reliability. The IEC 61131-1 standard for programmable logic controller (PLC) is a standard adopted worldwide by automation technologies' users and vendors. Nevertheless, limitations in achieving flexibility, portability and re-configurability in automation systems have proven to be a very interesting challenge in industry today. To overcome these issues, the International Electro-technical Commission (IEC) put in place a new standard: IEC 61499 to achieve portability, reconfigurability and interoperability of automation systems. IEC 61499 is event-driven and works on the idea of function blocks programming.

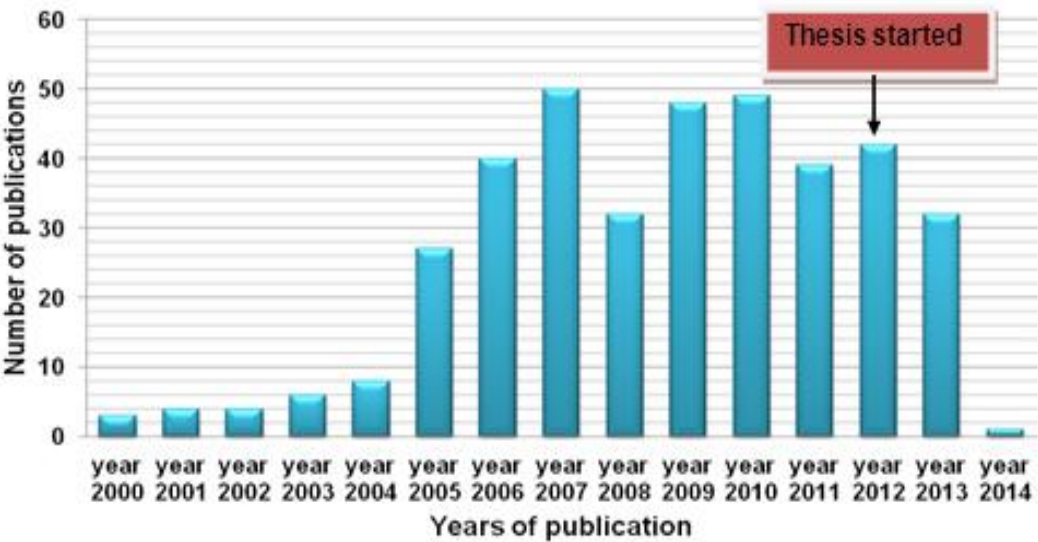
### **2.5.1 Validation of IEC 61499**

In 2005, the International Electro technical Commission (IEC) unveiled new international standards, such as IEC61499 to truly implement complex distributed control design. In a very complicated distributed system, they improved interoperability and re-usability, and aimed to augment re-configurability and flexibility in terms of both hardware and software of the control systems.

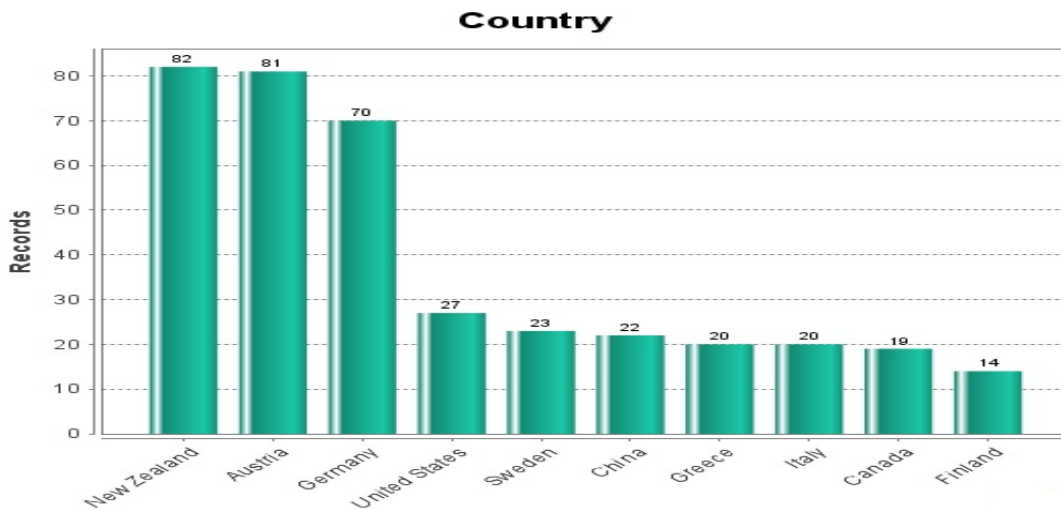
Vyatkin, (2007) introduced the notion of modules driven by events, known as Function Blocks (FB), to discuss the growing need of flexibility, reusability, re-configurability and applications of

distributed control. Software-reusability and simple reconfiguration can be improved through the object-oriented nature of the Function Block approach. IEC 61499 is attractive to developers because of the easier approach in terms of specification and language's abstraction advantages. Event-driven function blocks (FB), take place only when an event occurs to one of the event inputs. Simultaneously some stated data inputs will be updated. For the rest of the operation time the FB stays idle.

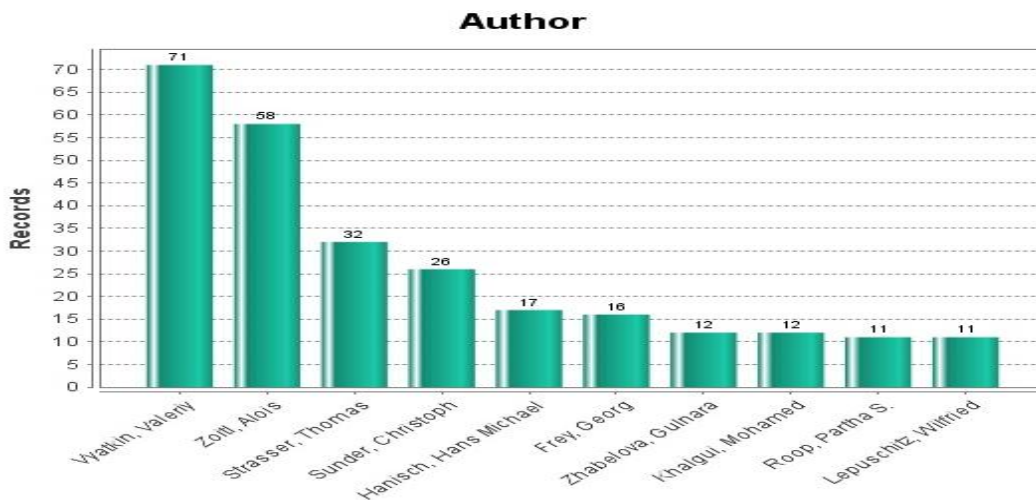
Dain and Vyatkin, (2011) confirmed that during idle time, the efficiency of the system improves and it reduces computing power utilization and communication bandwidth. In the same research they stated that small companies will have the possibility to create their own intellectual properties in IEC 61499 FBs into a library of components and allow a future re-use. Figures 2.4, 2.5 and 2.6 show the publication rate for the application of IEC 61499 standard in terms of: number of publications, country from which the publications are released and name of the publishers.



**Figure 2.4:** Publication rate for IEC 61499 standard-based papers from 2000 to March 2014



**Figure 2.5: Publication rate for IEC 61499 standard by country**



**Figure 2.6: Publication rate for IEC 61499 standard by authors**

From 2000 to March 2014, hundreds of papers have been published in the field of IEC 61499 standard. The graphs show clearly the professor Vyatkin of New Zealand has published the highest number of papers in the field with 71 publications. These numbers show that IEC 61499 standard becomes a very interesting research field in tertiary education and in industry. At the moment, there is no African university that has published a paper in this research area, and this serves to illustrate the importance of this research work in the thesis and also how relevant the current research is for the development of an IEC 61499 standard-based project at the Cape Peninsula University of Technology.

## **2.6 Comparison between the IEC 61131-PLC and IEC 61499 standards**

### **2.6.1 IEC 61131-PLC**

According to Hugh, (2008) the evolution of Control engineering over time has been tremendous. In the young age of control systems, humans were the only means of control. More recently with the evolution of technology, electricity has been utilized for control and primary electrical control was developed on the basis of relays. Switching power on and off was made possible by these relays without the usage of mechanical switch. Relays are usually used to make simple logical decisions. The introduction of cheap computer in industry has come with the latest revolution, the Programmable Logic Controller (PLC). PLC made its arrival in the 1970s, and has become the most popular and reliable solution for manufacturing controls.

As explained in IEC 61131-3, the PLC has five standard programming languages: Structured Text (ST), Ladder Diagram (LD), Instruction List (IL), Sequential Function Chart (SFC) and Function Block Diagram (FBD). PLCs offer certain advantages:

- Cost productive to achieve the control complicated systems.
- Adaptable and has the flexibility to be reapplied to control several systems quickly and with relative ease.
- Computational capacities allowing more advanced control.
- Trouble shooting makes programming simpler and reduces downtime.
- Solid components make these likely to function for years before a possible breakdown.

Programmable Logic Controller (PLC) traders implement PLC programming languages based on the IEC 61131-3 standard; however, they are not limited to it. Such approach creates serious problems with compatibility of PLCs. For example, despite their similar form, a ladder diagram from one manufacturer is impossible to be imported to a different ladder diagram developed by another manufacturer. This incompatibility invokes the problem of portability of the programs in IEC 61131-3.

Vyatkin, (2011) stated that the IEC 61131-3 standard based on function block approach is a subroutine with frameworks and local data. Nevertheless, the syntax of specific implementations may incorporate a number of specifications of vendor characteristics. In a function block, it is not always possible to obtain all the standard IEC 61131-3 programming languages. It must be noted that the scope of admission to controller memory changes between manufactures, some PLCs are only compatible with (memory/variable/tags) affiliated within a specific block (local/tag/memory); however others can have a full access of the global variables.

Figure 2.7 illustrates the concept behind IEC 61131-3 programming for the PLC implementation of a conveyor control system (Vyatkin, 2011).

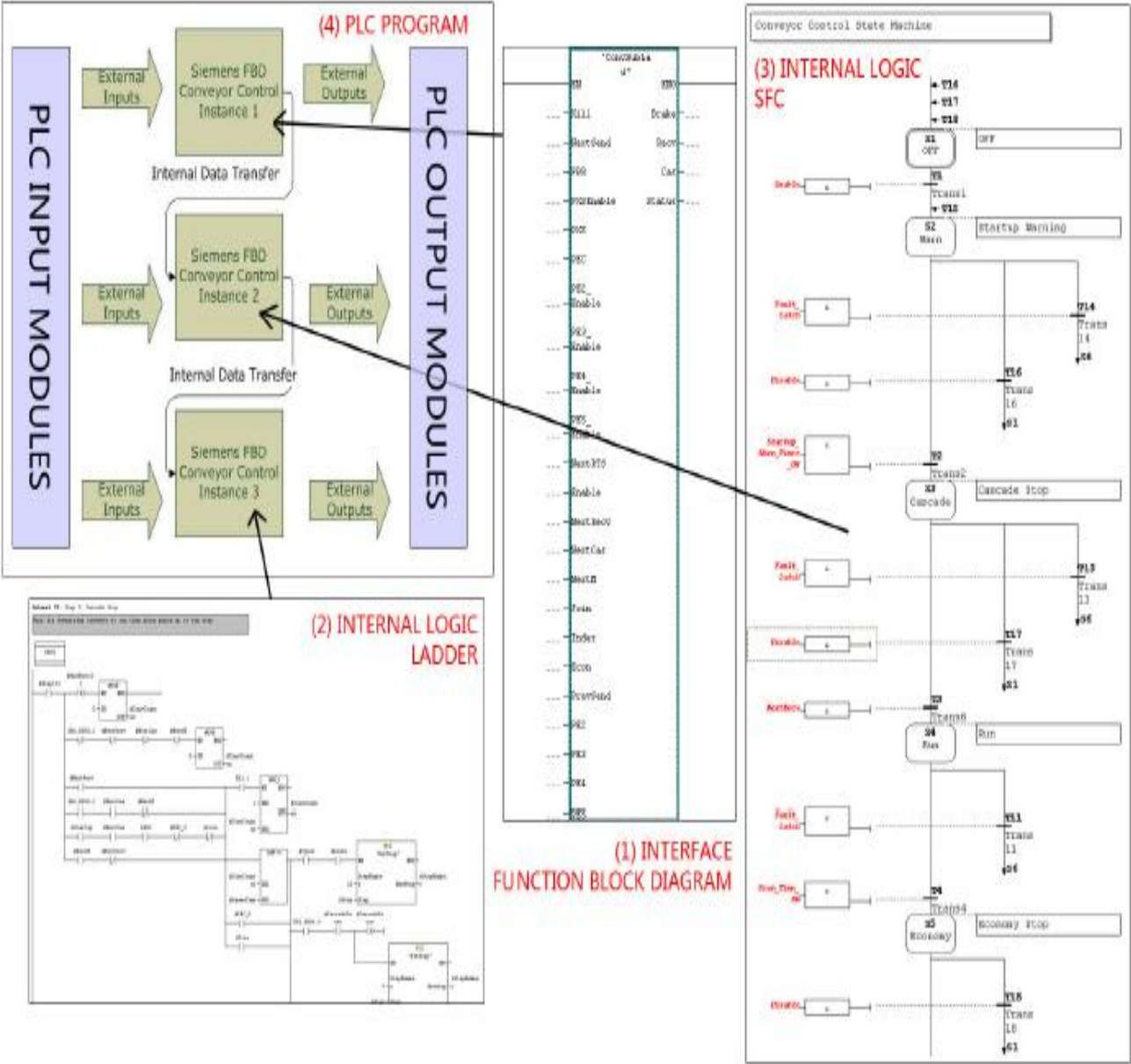


Figure 2.7: PLC implementation of a conveyor control system (Vyatkin, 2011)

2.6.2 IEC 61499 standard-based Function Blocks

IEC 61499 standard, established by IEC in 2005, is viewed as the next stage to improve PLC system engineering. IEC 61499 standard makes projects easily imported to others machines. Encapsulation of functions into function blocks augments reusability. The direct consequence for standard component library of function blocks is to provide an efficient reduction of project re-

work time. IEC61499 stretches the existing standard IEC61131-1 to ease the development of distributed control systems based on function blocks programming. Goran, *et al.*, (2006) developed FUBER, a run-time for IEC61499 standard-based applications to lessen the time to mature the concept of distributed control systems. The results of this publication presented some initial results of the possible generation of a standard model of an IEC61499 application running inside FUBER.

Function blocks are classified in three (3) categories (Hugh, 2008):

- Basic
- Network Basic and Composite
- Service Interface

#### **2.6.2.1 Basic Function Blocks**

Basic function block always contain an Execution Control Chart (ECC), which is a state machine that possesses conditional subdivisions and equivalent states to execute developed algorithms. There are two types of inputs and outputs in the IEC 61499 FBs standard-based: events and data. The sole condition to trigger a Function Block is when an input event is triggered. In addition, data inputs and outputs combined with that specific input will be brought up to date. As described in IEC 61499, the algorithms developed internally based on the function block methodology can be programmed in various languages, such as basic IEC 61131 programming languages, and higher programming languages like Java or C.

#### **2.6.2.2 Network of Basic and Composite Function Blocks**

According to literature, a composite function block is made of a network of basic and composite function blocks. In this way, to increase reusability of the program, a hierarchical framework can be built. Similarly to basic FBs approach, the IEC 61499 standard-based composite function block possesses its unique interface. In the composite function block approach, the inputs and outputs can be linked straight to the inputs and outputs of the component FBs.

Finally, an application is the composition of basic and composite FBs. Configuration of the system mixes the device topology with the application logic, abstracts the meaning of communication networks and precise function blocks mapping to the devices.

### 2.6.2.3 Service Interface Function Blocks

The service interface function block (SIFB) is a type of atomic function block. It interfaces low-level services provided by the hardware embedded device or the operation system, such as:

- Graphical User Interface (GUI) represents elements such as sliders or knobs.
- Communication services such as the communication of a “client” for a remote “server”.
- Interfaces to hardware such as a control valve or a temperature sensor.

SIFBs can be utilized to implement diverse interfaces to databases, communication protocols or human-machine interfaces (HMI).

As an example, Figure 2.8 illustrates the concept behind the IEC 61499 function block implementation of a conveyor control system (Vyatkin, 2011).

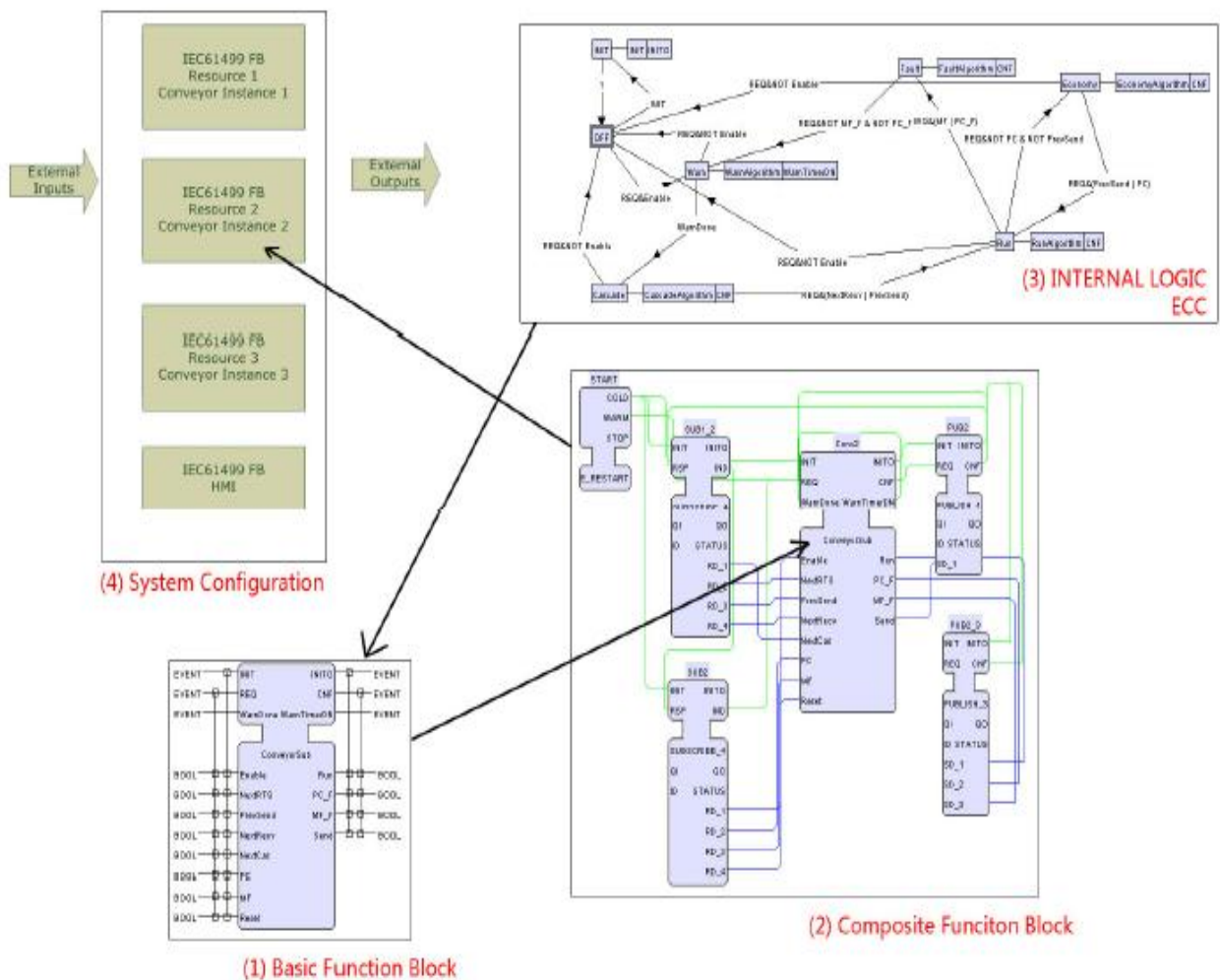


Figure 2.8: IEC 61499 FB implementation of Conveyor Control System (Vyatkin, 2011)

### **2.6.3 Advantages of IEC 61499 standard over IEC 61131-PLC standard**

Yang, (2010) in his research found several advantages of IEC 61499 over IEC 61131-PLC:

- IEC 61499 saves the CPU's usage since it is event driven compared to IEC 61131-PLC which runs 100% of CPU. It shows that IEC 61499 is very effective in terms of power saving.
- Performance of the system can benefit of massive improvement if tasks are dynamically allocated to varying controllers. It means that if one PLC fails then the task that the PLC run can be handled by another PLC.
- Function Blocks help to achieve the design in a distributed way. Which means that distributed designs might be more reliable in terms of node failure.
- Code mapping to different devices is done automatically rather than handshaking done explicitly by the programmer.
- Platform dependence is achieved because different controllers can be selected at the end of development to deploy to.
- Maintainability of IEC 61499 standard may be easier as it is represented in a graphical form with a visual interface.

The generation of the Human Machine Interface is simple especially with NXT Control, although it is not universally true for all IEC61499 based Intelligent Electronic Devices (IEDs).

### **2.6.4 Disadvantages of IEC 61499 standard over IEC 61131-PLC standard**

The migration from IEC61131 to IEC61499 standard also presents some disadvantages, Yang, (2010) highlighted the following drawbacks:

- Possible non-deterministic execution of the function blocks algorithms.
- Performance of the system can be affected because implementation of an event driven scheduler may bring overhead that slow down the execution of the application.

Yang, (2010) proposed methods to overcome the drawbacks highlighted above to help to diminish these potential issues in IEC61499 standard applications in the following ways:

- The Co-modeling method developed to migrate from IEC61131 to IEC61499 standard has a very powerful timing application that improves the performance related to the event driven scheduling.



- Environments such as Verification and Validation reveal potential non-deterministic behaviour of the controller.

### **2.6.5 IEC 61499 standard-based Function Blocks Development Status**

This section describes some design platforms for the implementation of distributed control based on the IEC61499 standard. These tools are the following:

- Function Block Development Kit(FBDK)
- ISAGRAF Control Tool
- NxtStudio from NxtControl
- 4Diac
- TwinCAT 3

Vyatkin, (2008) described FBDK as the earliest developed platform for distributed control systems created by Rockwell Automation which played a pivotal role to evaluate IEC61499. FBDK consists of a run-time platform (FBRT) and a development editor (FBEditor). FBDK uses Java programming language and compiles function blocks to Java classes, hence Java is a fundamental pre-requisite for working with FBRT on specific embedded targets. Nevertheless, the usage of Java although advantageous for portability is not suggested in the standard itself.

ICS Triplex defines ISAGRAF as the source and de facto standard in automation software technology. Proposed in 1990, it was primarily designed to join microcomputer systems and PLCs. It has evolved to become the most influential software technology for open automation, applicable to embedded control, conventional automation and soft logic markets. The applications developed on ISAGRAF provide all of the internationally standard IEC61131-1 control languages. An important feature of ISAGRAF is that it possesses series of toolkits which give to the user the ability to write its personal I/O drivers, add market special function blocks, connect to more advanced systems, or ideally brand label the artefact. This becomes a sole packaging or user's personal value added intellectual property.

NxtControl is software for integrated automation of buildings, machines and processes. It enables up to date solution and allows more efficient automation engineering. NxtControl designs complete systems and plants, not only individual controls. Control Systems and visualization are further more designed using a single engineering tool. This is very beneficial because it eliminates many interfaces and thus sources of errors. Control programs are developed to meet all the requirements for systems with one or many controllers according to the IEC61499 standard. The following automation tasks are solved with NxtControl:

- Programming of the control system for an entire plant
- Multi-Client visualization of an entire plant
- Linking the process level to the application
- Documentation of the entire plant
- Testing and simulation of the control and visualization application
- Commissioning of the control hardware via Visu-Clients

The main focus of the 4DIAC enterprise is to supply a free, open, IEC61499 standard that complies with automation and control environment on the basis of targets such as interoperability, portability and configurability. FORTE is one of the most important small portable implementation of IEC61499 Run-time environment of 4DIAC to target small embedded control devices (16/32 Bit). FORTE has the following features:

- Basic, Composite and Service interface function blocks, and Adapters
- Data-types for Basic IEC 61131-3 programming.
- Event and data connections
- Communication FB's based on Service/Client topology and Publisher/Subscriber topology for Ethernet.

TwinCAT 3 is software developed by Beckhoff systems that turns almost any personal computer (PC) based into real-time controller with multiple runtime systems. It is the software selected to develop the control system in this thesis, and its functionalities are explained in the next chapters.

## **2.6.6 Other International Standards Related to the IEC 61499 standard**

### **2.6.6.1 IEC 61850 standard for substation automation**

The IEC 61850 standard describes a group of architectural artefacts meant to classify intelligence monitoring, control, protection, and functions of automation. These functions generate and use signals that are customarily linked by countless wires between the primary equipments and the automation devices.

Vyatkin, *et al.*, (2008) introduced Substation Automation System (SAS) based on the concept of IEC 61850 standard for wide range elements of related power system automation architecture. The IEC 61850 standard is widely used in power systems. But one area was deliberately left empty in this standard: IEC 61850 does not normalize the illustration of rule-based, combinatorial, sequential control in power system and logic in automation.

The chief advantages of IEC 61850 standard are the following (IEC 61850-1, 2003), (IEC 61850-2, 2004) and (IEC 61850-6, 2006):

- Easy substation structure: interface difficulties are resolved. With IEC 61850 standard application, the protocol variety and integration issues are no longer existent.
- The entire concept is elementary: From engineering to implementation, and from operation to service. It reduces time consumption, costs on configuration and the commissioning.
- Minimizing of costs: IEC 61850 substitutes wiring between feeders, control switches, and signalling devices.
- Improves reliability: Only one communication channel is used for all data in real time, and the synchronization is done via Ethernet.

Generic Object Oriented Substation Events (GOOSE) messaging is based on the IEC 61850 most reliable communication protocol and a brief description is given in the next section.

#### **2.6.6.1.1 IEC 61850 Generic Object Oriented Substation Events (GOOSE) Messaging**

Generic Object Oriented Substation Events (GOOSE) protocol is an Ethernet-based technology within the IEC 61850 standard that allows complete integration of all required system data, given the physical limitations of the communications architecture (Ali and Thomas, 2011). GOOSE represents data sets and is transmitted within a time period of four milliseconds. The following methods are used to ensure fast transmission and reliability of GOOSE messages:

- GOOSE information is embedded directly into Ethernet data packets and operates on publisher-subscriber principle.
- GOOSE uses Virtual Local Area Network and priority tagging to obtain distinct virtual network within a physical network and puts suitable signal level of priority.
- Improves retransmission procedures: The same GOOSE message is reconveyed with changing and growing re-transmission periods.
- GOOSE messages are developed to be brand independent. Some manufacturers possess intelligent electronic devices (IED) that are fully compliant to the IEC61850 standard for a truly interoperable method within the substation network with no necessity for the vendor to specify cables or algorithms.

IEC 61850 standard is used in conjunction with IEC 61499 standard in order to enable the development of complete automated devices and system solutions. Both standards once combined have the potential to personalize logic of automation and control to improve the

adaptability and flexibility of automation systems, step up the progress toward the implementation of the concept of Smart Grid (Vyatkin *et al.*, 2008).

#### **2.6.6.2 IEC 61400-25**

The IEC 61400-25 standard is a methodology for making easier the functions of SCADA systems and wind turbine. The standard defines the most important part of the communication stacks and information methods in wind power plant industry. The IEC committee constructed a model in which procurement requirements and contracts could be easier to refer to. The standard specifies five mapping to communication protocol stacks. To reach interoperability, all data in the information model require a reliable definition based on syntax and semantics. The semantics of the data is mostly provided by names given to logical nodes and data they contain. Interoperability is easier when the data are defined as mandatory.

As the smart grid is fast developing all over the world during these years, Mu Wei, (2011) showed that the communication technologies of the IEC 61400-25 standard can provide reliable connection between the wind farm and the main grid. IEC 61400-25 overcomes the difficulty to control the wind energy, this problem arises because the connection to the wind farm makes the grid more vulnerable and a suitable control solution had to be provided.

IEC 61400-25 standard can be used as visualization (SCADA) interface for IEC 61499 standard-based application that controls a wind power plants, logical nodes or real circuit breaker.

#### **2.6.6.3 IEC 61970-301 and IEC 61968-11**

McMorran, (2007) defined the IEC 61970 as a semantic model that defines the elements of power systems at an electrical level and the relationship linking each component. The IEC 61968-11 broaden this model to incorporate the other features of the power systems software. These two standards are jointly called: Common Information Model (CIM) for power systems. Their primary uses are the following:

- To ease the exchange of power systems network data between companies.
- Facilitate data sharing between applications in the same company.

IEC 61970-301 and IEC 61968-11 standards can be used to improve the data exchanges within companies that utilize IEC 61499 standard to control distributed power system components.

#### **2.6.6.4 IEC 61508 and IEC 61511**

Gall, (2008) described the IEC 61508 and IEC 61511 as standards for functional safety. IEC 61508 is defined as the standard used to classify applications such as functional safety of electrical, electronic and programmable electronic systems, on the other hand IEC 61511 is for functional safety of instrumentation safety systems for the industrial sector. Both standards are accepted worldwide and are becoming day to day practice in many industries.

According to Endress+Hauser company, the relationships between both standards are the following:

- Process sector safety of instrumentation based system standards
- Suppliers and manufacturers of devices
- Safety instrumented system integrators, designers and users
- Development of new hardware with prior use

IEC 61508 and IEC 61511 standards guarantee the safety of IEC 61499 standard based applications in the highly demanding process industry sector.

Table 2.2 describes some of the relevant IEC standards related to electrical engineering (IEC standards, 2013). These standards are used to improve the performance of IEC 61499 standard-based applications in specific fields of electrical engineering.

**Table 2.2: Description of some IEC standards (IEC standards, 2013)**

<b>Standards</b>	<b>Areas of application</b>
IEC 60034	Rotating electrical machinery.
IEC 60044	Instrument transformers.
IEC 60076	Power transformers.
IEC 60228	Conductors of insulated cables.
IEC 60255	Electrical Relays.
IEC 61131	Programmable Logic Controllers.
IEC 61346	Industrial systems, installations and equipment and industrial products.
IEC 61400	Wind turbine and SCADA systems.
IEC 61439	Low-Voltage switchgear and control gear assemblies.
IEC 61499	Function blocks in distributed industrial-process measurement and control systems.
IEC 61508	Functional safety of electrical/electronic/programmable electronic safety-related systems.
IEC 61511	Functional safety – safety instrumented systems for the process industry sector.
IEC 61558	Safety of power transformers, power supplies, reactors and similar products
IEC 61804	Function blocks for distributed control systems
IEC 61850	Communication Networks and Systems in Substations
IEC 61968	Application integration at electric utilities – System interfaces for distribution management.
IEC 61970	Application integration at electric utilities – Energy management system application program interface.
IEC 62056	Communication protocol for reading utility meters
IEC 62264	Enterprise-control system integration
IEC 62304	Medical Device Software – Software Life Cycle Processes
IEC 62351	Power System Control and Associated Communications – Data and Communication Security
IEC 62379	Common control interface for networked digital audio and video products
IEC 62455	Internet protocol (IP) and transport stream (TS) based service access

## **2.7 Modeling and Design environments**

This section discusses the possibility to use Matlab, Simulink or LabView as integrated modeling, design, and simulation environments to test the function block principles before implementation of IEC 61499 standard-based development platforms.

### **2.7.1 Matlab**

The research problem addressed in this thesis is the design and implementation of non-linear control algorithms in distributed systems on a programmable software environment. MATLAB is used as a software environment for simulation purposes because most of the engineers and developers know well its applications in industry.

The MathWorks describes Matlab as an advance language and interactive development platform for mathematical computation, programming, and visualization (The MathWorks, 2013). Matlab enables the user to study data, develop algorithms, and build applications and models. The language, tool, and built-in math functions investigate numerous techniques and find a solution quicker than with spreadsheets or primary programming languages. Matlab can be utilized for various applications, such as image and video processing, signal processing and communications.

Silva and Krogh, (2000) stated that MATLAB provides a very strong environment developed to validate and verify designed models. For example, Matlab designed CHECKMATE to investigate nonlinear continuous dynamics systems, under Simulink environment.

### **2.7.2 Simulink**

Simulink is a block diagram environment developed by the MathWorks for graphical simulation and Model-Based Design of embedded and dynamic systems. It contains automatic code generation, design at system-level, simulation, and continuous test and verification of embedded systems. It possesses a graphical editor, block libraries that are modifiable, and solvers to model and simulate dynamic systems. Simulink is merged with Matlab, allowing the user to include Matlab developed programs into models and export the results of the simulation to Matlab for deeper understanding. It has the ability to support linear and nonlinear systems, modelled in any simulation time.

State flow is a Simulink package which provides blocks that can be customized to describe finite state machine (FSM). Simulink State Flow package is crucial in the recognition of IEC61499

standard in industry because it works like the basic function block developed on the basis the IEC61499 standard.

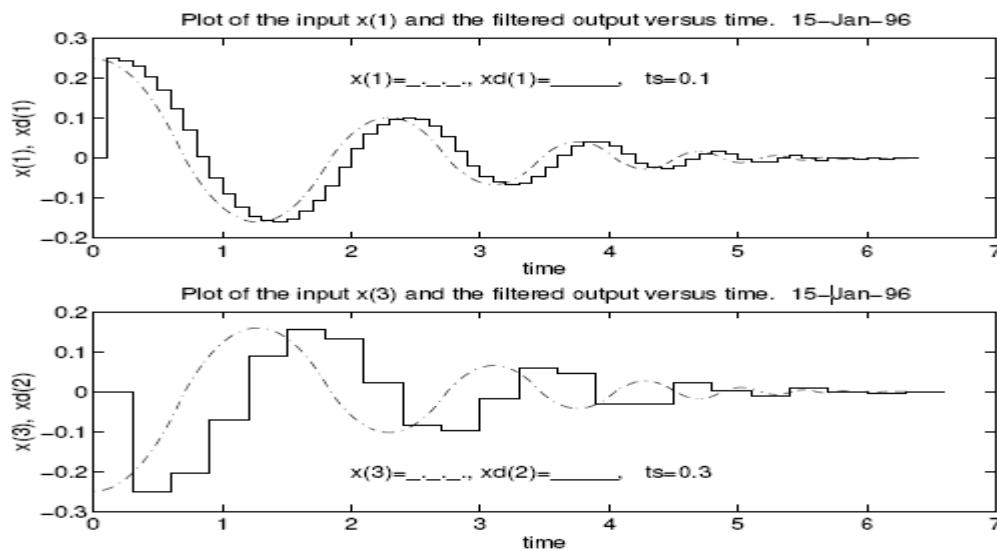
Yang, (2010) developed a comparison between Simulink and function block modeling environments as shown in Table 2.3.

**Table 2.3: Comparison between SIMULINK and Function Block environment (Yang, 2010)**

Advantage of FB	Disadvantage of SIMULINK
Structured layout for distributed purpose	Harder to manage multi-layered subsystems
FB instances update all the network	Harder in modification of duplicated blocks
Distributed modeling	Not designed for distributed modeling
Direct Deployment	Focus on Simulation and Analysis

This comparison shows that the IEC 61499 standard-based applications are more suitable for the distributed implementation on the basis of the function block concept compared to Simulink.

Taylor, (1994), and Kebede, (1995) described the Hybrid System Modeling Language (HSML) created precisely to model hybrid systems. HSML is capable of constructing state events and discrete time modules in Matlab and Simulink. It is widely known HSML is very useful especially to handle time and state-event (see Figure 2.9). This is a very important concept of IEC 61499 standard because it reduces power consumption.



**Figure 2.9: Sample simulation results from MATLAB and SIMULINK (The MathWorks, 2010)**



The results of the sample simulations on both Matlab and Simulink environments show that their real-time capabilities are reliable, and can be used as verification platforms to confirm the results obtained on IEC 61499 standard-based applications.

### **2.7.2 LabVIEW**

National Instrument describes LabVIEW as a graphical programming language that makes use of icons rather than lines of text to build applications. Contrary to programming languages that make use of text, where specifications determine the execution of the program, LabVIEW is based on dataflow programming, where program's execution is determine by the data flow (LabVIEW, 2013).

In LabVIEW, programmers use a set of objects and tools to build a user interface.

The development user interface is called the front panel. The programmer then adds code by making use of graphical representations of functions to control the objects generated on front panel.

The block diagram encapsulates the code. In some applications, the block diagram looks like a flowchart. It is possible to buy numerous add-on software toolsets for the creation of special applications. Any of the toolsets can be integrated without any problem in LabVIEW. In LabVIEW the Virtual Instruments (VIs) can communicate with different processes, which includes the ones that run on different applications or on remote computers. LabVIEW has the following networking attributes:

- Can build VIs that convey information with other applications.
- Emails data from VIs.
- Publishes the front panel images and VI documentation on the Web.
- Shares live data with alternative VIs running on a network.
- Interface to .NET

Catani, (2003) stated that LabVIEW is compatible with other standards and widely used protocols. The results of his research work showed that only TCP/IP and UDP libraries provide enough flexibility and the possibility to non-LabVIEW systems to achieve distributed control systems.

Matlab, Simulink, and LabVIEW can be used as simulation environments to allow a better understanding of the IEC 61499 standard. These simulation environments are using the principle of function block programming which is the main idea of the IEC 61499 standard. However no

applications have been developed with LabVIEW as verification environment till today, it is the reason why LabVIEW is not selected in this thesis.

## **2.8 Existing Distributed Control Techniques in IEC 61499**

Vyatkin and Yang, (2010) developed two new modeling based techniques for verification and validation structure of IEC 61499 Function Blocks on the basis of closed-loop model of the plant. These frameworks strengthen the integration of the distributed control systems concept in industry because they are based on Matlab and Simulink which are the most widely used softwares by the control engineers. These existing modeling techniques are the following:

- Model Transformation Approach
- Co-Modeling Approach

### **2.8.1 Model Transformation Approach**

Yang, (2010) described the model transformation as a block to block transformation technique. The idea behind the concept is to transform a Simulink model to basic function blocks that have precisely the same inputs, outputs, internal variables and behaviours as its equivalent Simulink model. Since block to block mapping is the focal point therefore some mapping techniques have to be considered:

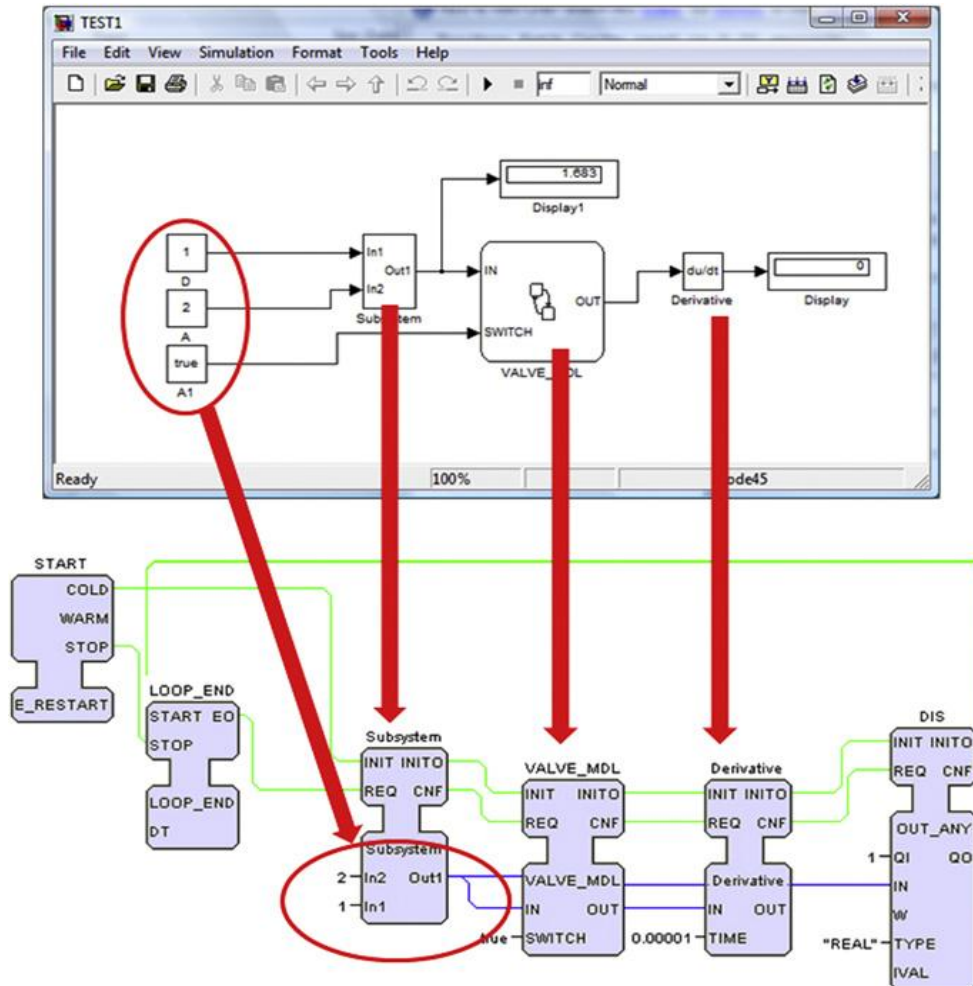
- A singular Simulink block is converted to a basic Function Block
- A Simulink subsystem block to a composite Function Block
- A Simulink system to a Function Block application

Vyatkin and Yang, (2012) stated that the transformation of State flow diagram from Simulink to the Execution Control Chart of the basic Function Blocks is almost effortless especially when the classic State flow semantics is taken in consideration, that only considers one transition for a specific simulation time step. A Simulink subsystem can be put in a composite Function Block. This permits hierarchical construction of the model. However, hierarchical layout requires that the programmer considers the flow of execution carefully. An example of model transformation of a two tank system developed by Yang, (2010) is shown in Figure 2.10.

The advantages of the model transformation are the following:

- The model transformation is done automatically to decrease time and effort taken for model development and help considerably to validate the designs derived on IEC61499 Function Blocks.
- The aptitudes of the developer are broden by embedded models to implement model-predictive controls in the IEC 61499 platform.

The model transformation has been successfully used in many industrial cases such as Baggage Handling Systems, Process Control and Food. This thesis focuses on the model transformation approach which is selected as a method to achieve real-time control of the Magnetic Levitation system.

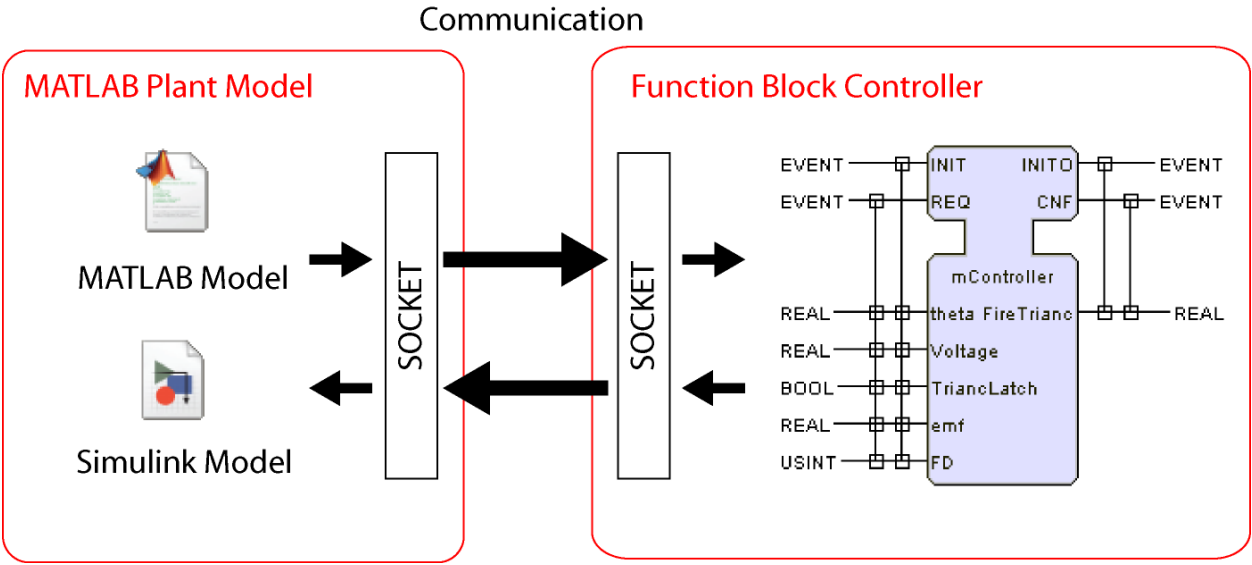


**Figure 2.10:** The transformed Function Block System (Yang, 2010)

### 2.8.2 Co-Modeling Approach

The co-modeling approach is a communication technique using socket communication protocol to bridge the link between the Matlab/Simulink and the Function Blocks software. This method allows closed-loop simulation between the two software environments. In a distributed control systems scenario the ideal concept would be to design the plant in the Simulink environment and the controller in the Function Blocks development software such as ISAGRAF, NxtControl or TwinCAT 3 toolset. If a closed-loop plant-controller model has already been done in SIMULINK therefore it becomes easy to establish an immediate communication channel with the function

block controller. The controller model developed in Function Blocks can then be deployed in the physical controller provided by manufacturers such as Beckhoff (CX1020) or Siemens (MicroBox). The testing of the physical controller can be done through socket communication with Simulink before it is deployed in the real control application. Synchronization of both programming environments is key to correct closed loop simulation. The production of data of the controller can for example be more frequent than the one of the plant due to the fact that both environments have different time execution. One should keep in mind that the execution time in Simulink is based on simulation model time scenario and it can be different from the real-time. This assertion is also valid for the controller because its execution time on ISAGRAF, NxtControl or TwinCAT 3 can differ from the physical controller in Beckhoff or Siemens. Sample time synchronization is regarded as the adequate method to ensure time maintenance of both the MATLAB and the Function Block model. To achieve time synchronization, it is vital to create a handshaking protocol between the sockets. Several communication sockets can be put in to perform an execution order between the controller and the plant models, even if it is done only on one of the controllers. Figure 2.11 shows how the closed loop simulation can be achieved (Yang, 2010).



**Figure 2.11: Closed-loop simulation between MATLAB and FB models (Yang, 2010)**

Compared to the transformation method, the Co-Modeling is time and cost-saving which allows an easy set-up to the control engineer. The Co-Modeling approach has the advantage that no modification of the existing SIMULINK model is required while the controller can be tested and validated through closed-loop simulation with the plant.

Table 2.4 resumes the projects done on distributed control systems so far to permit the acceptance of the IEC 61499 standard in industry.

**Table 2.4: IEC 61499 related projects**

Paper	Aim of the paper	Method of control	Structure of the system	Plant under control	Software development and deployment environments		Advantages/ Drawbacks	Achievements
					Simulation	Implementation		
(Vyatkin and Hanisch, 1999)	Summary of the development of formal modeling and verification of function Blocks following the IEC 61499.	Execution control of function block	Distributed	N/A	Function Block Development Kit, 4DIAC, ISAGRAF	N/A	Reliability and re-configurability of systems. Difficult to accept the standard in industry because it is still not clear.	Introduction and recommendation of IEC 61499 in future distributed control applications
(Vyatkin and Hanisch, 2001)	Discuss the issues related to the correctness of agile manufacturing systems with distributed architectures.	Network control	Distributed	Processing (drilling) station with a carrier bringing work pieces.	VEDA, Function block development kit	N/A	Challenging, robust to disturbances, adaptable and flexible to rapid changes on the factory floor.	The nominative algorithm of input event processing in the basic function block may lead to missing some events when they occur and arrive rapidly.
(Wei, 2001)	Implementation of IEC61499 Distributed Function Block Architecture for Industrial Measurement and Control Systems.	PID, Network control and function block development for real-time control applications	Distributed	Coupled tank with analog inputs and outputs	Function block development kit	3 x PC Pentium 350 MHz, 64MB RAM with Win 98 OS	Hard to make AI and AO function blocks. Good implementation, reliable and efficient.	Reduction of design time, cost at the engineering side is also reduced and system will be more flexible and maintainable.
(Hussain and Frey, 2005)	Migration of a PLC Controller to an IEC 61499 Compliant Distributed Control System	Network enabled controllers	Distributed	Didactic Modular production system (MPS) from Festo.	Function Block Development Kit (FBDK)	4 x NETMASTER controller devices	Control algorithm and validation were partly circumvented	Migrate a PLC controlled centralized laboratory application into an IEC61499 compliant

								distributed control application.
(Thramboulidis, 2005)	To Highlight the inefficiencies of IEC 61499 to support the whole development process of distributed control applications at software level.	PID	Distributed	Tank	Function Block Development Kit (FBDK), CORFU-FBDK	CORBA component Model, Real-time Linux, Real-time Java.	Lack of methodology to develop the control application, no guidance to compose the control application, reliable reference implementation.	IEC 61499 represents an important step for the exploitation of current software engineering practices in factory automation and mainly in low level control.
(Goran et al., 2006)	Presentation of a new runtime environment, Fuber, along with a formal execution model	Embedded controller	Distributed	Automatic carriage	Fuber	Java	Implemented in Java which limits its utilization and composite data types for variables are not handled. Saves time.	Experimentation with multiple execution models Fuber, a runtime for IEC 61499 applications, has been developed.
(Ferrarini, 2006)	Development of an environment able to convert IEC 61499 into a generic programming language.	Embedded controller	Distributed	Shuttle device	FBDK	Siemens PLC	Reduced the amount of code and avoided large cycle-time, improvement of reusability	Addresses the definition of a set of rules which can be used to convert an IEC 61499 application into AWL control code executable on Siemens PLC
(Molina et al., 2007)	Review aspects relevant to industrial standards related to PLC programming	Embedded controller	Distributed	Industrial safety machinery	Ladder and Function Block	Soft PLC	Limitations of IEC 61131 for distributed control applications, reliability and availability of IEC 61499.	Introduce new standards and new programming methodology to overcome IEC 61131 limitations in distributed control.

(Strasser et al., 2008)	Cover the usage of IEC 61499 sub-applications for introducing a simpler and more effective, engineering approach to structure large scale distributed control	High-Gain Observer, the PD controller	Distributed	Modular Robot Arm	FORTE, 4DIAC-IDE	PC/104 embedded controller hardware	Flexibility, scalability	Large scale distributed automation and control programs are handled advantageously with IEC61499 function blocks.
(Higgins et al., 2008)	Presents new approaches to power systems automation, based on distributed intelligence rather than centralized control.	On/off	Distributed	Switches, breakers	SCADA, ISAGRAF	Microprocessor	Ability to customize protection, monitoring, control and automation functions.	Benefits of distributed automation with IEC 61850 and IEC 61499 to reach powerful performance in embedded computing platforms
(Yang and Vyatkin, 2008)	Overview of the works on design and validation of distributed control in process industry.	Intelligent control algorithms	Distributed	2-tank system, Batch process reactor, beet sugar factory	MATLAB, Simulink, Function Block Development Kit (FBDK) and FBench	N/A	Flexibility, re-configurability And software re-usability.	Perspective of system validation challenged by the re-configurability demand achieved via decentralised control approach.
(Dai and Vyatkin, 2009)	To develop a guide to migrate from IEC 61131 PLC technologies to IEC 61499 function blocks	On/Off control	Distributed	Conveyor system	FBDK, ISaGRAF, and FBench	Siemens S7-300 PLC is selected as a PLC and a TCS-NZ MO'intelligence embedded controller is used to deploy IEC 61499	Distribution and reconfigurable abilities are not benefiting small single processor systems. Reconfigurable, interoperable and portable.	Simulation of IEC 61499 was accomplished and the transformation from IEC 61131 to IEC 61499 is not 100% equivalent.
(Goh and	Conceptual design of a new						Limitation of the standard,	Development of a flexible embedded

Tjahjono, 2009)	embedded system code generator framework which is based on the IEC61499 Function Block to address the code generation issues.	On/Off	Distributed	Light switches	OOONEIDA	FPGA gates	compatibility issues between different tools, function block is user friendly the price of FPGA is affordable.	software environment that could be used together with new or existing tools within the tool chain.
(Vyatkin, 2009)	Discuss IEC61499 standard and its semantics, outline ideas of some solutions currently being developed.	On/Off function block controller	Distributed	Pneumatic cylinder, safety light curtain	FB Development Kit (FBDK), ISAGRAF	N/A	Ambiguities in the execution semantics descriptions, standard still incomplete.	Demonstrated that it is impossible at this stage to come up with a single execution model of FB networks.
(Wenger et al., 2009)	Possibility for a semantic correct transformation of existing IEC 61131-3 into newer IEC 61499 standard	N/A	Distributed	3 cycle generation, 8bit counter	4DIAC-IDE	STReader	Portability, interoperability and re-configurability	The transformation of the IEC 61131-3 execution order caused some overhead due to the event concept of IEC61499
(Schimmel and Zoitl, 2010)	Real-Time Communication for IEC 61499 in Switched Ethernet Networks	N/A	Distributed	Network of nodes	FORTE	Switched Ethernet with UDP and TCP protocols.	Reduction of delay, cheap hardware solution, efficiency of communication	Utilization of switched Ethernet for distributed real time automation systems
(Hegny et al., 2010)	To propose the use of an IEC 61499 industrial automation runtime environments to simulate plants behaviour.	Embedded controller	Distributed	Sorting Machine	FORTE, 4DIAC-IDE	N/A	Reduction of the implementation time.	Efficient control development for production systems based on IEC 61499.
(Gerber et al,	Proposes different design	Task controller function block					Vendor independency,	Benefits of real object oriented and



2010)	approaches for IEC 61499 control applications.		Distributed	conveyor	FBDK	Wago IPC 750-860	reconfiguration, re-usability, flexibility.	distributed controller design.
(Yang and Vyatkin, 2010)	Design and Validation Environment for IEC61499 Function Blocks	PID control	Distributed	Motor and valve	MATLAB Simulink and FBDK	N/A	Complexity of distributed system design, lower recognition of standard.	Utilization of two engineering methods with IEC61499
(Vyatkin, 2011)	Reviews research results related to the design of distributed automation systems with IEC 61499.	Embedded controller	Distributed	Airport baggage handling, intelligent power distribution networks and Smart Grid.	ISaGRAF and NxtStudio	PLCs: Beckhoff, WAGO, SIEMENS and Advantech	Portability and interoperability	The wider adoption of IEC61499 will help the industry to benefit from the promise of holonic and intelligent automation research results.
(Zhabelova and Vyatkin, 2011)	Propose a pathway for industrial deployment of multi-agent Information Communication Technologies (ICT) architectures.	N/A	Distributed	Power distribution utility and location of the fault	MATLAB (SimPowerSystem), Function Blocks Development Kit (FBDK), ISaGRAF or NxtControl.	Intelligent Electronic devices (IEDs), microprocessor-based relays, remote terminal units (RTUs) and PLC.	Inter-operability and configurability, and enables efficient simulation of distributed automation systems.	Feasibility of distributed intelligent Smart Grid automation based on two international standards: IEC 61499 and IEC 61850.
(Yan and Vyatkin, 2011)	Prove the feasibility of a fully distributed automation design of baggage handling systems automation.	Embedded controller	Distributed	Conveyor of group of conveyors	ISAGRAF, FBDK	Netburner device	Portability, interoperability, re-configurability and distribution of control applications.	The ISAGRAF implementation of the IEC 61499 standard allows FBs to run on a targeted hardware with simple user application.
(Yang and Vyatkin, 2012)	Transformation of Simulink models to IEC 61499 Function Blocks	PID control	Distributed	Motor	MATLAB Simulink and	N/A	Time and efforts	Design of complex distributed systems allows to take advantage of

	for verification of distributed control systems.				function Blocks (in FBDK)		are reduced	the simulation and analysis capability of MATLAB Simulink.
(Sorouri et al., 2012)	Propose a systematic approach toward controller design of mechatronic components	Master-Slave Controllers, Peer-to-Peer controllers and Independent and distributed controllers.	Distributed	A pick and place robot	NxtStudio 1.5	N/A	Versatile and reusable.	Successful implementation of all control logics in NxtStudio V1.5
(Suender et al., 2013)	Formal approach to validation of on-the-fly modification of control software in automation systems	Down timeless system evolution	Distributed	N/A	eCEDAC project	N/A	Correctness of the system to be implemented	Utilization of the down timeless system evolution with IEC 61499 saves time.
(Vyatkin, 2013)	Gives a perspective on recent development related to software engineering in industrial automation	PLC	Distributed	N/A	IsaGRAF	PLC compliant with IEC 61499	Improvement of the production in industry	Bridging the gap between automation and software engineering worlds.
(Wenbin, et al., 2014)	Proposes new control methodology to migrate from IEC 61131 to IEC 61499 function blocks.	Embedded PLC controller	Distributed	Single airport baggage handling system	Rockwell ControlLogix PLC	Rockwell ControlLogix PLC	Flexible, less resource consuming	Ontological-based migration methodology can be applied to IEC 61131-3 compliant PLC source code to transform it to IEC 61499 platforms.

## 2.9 Comparative literature review

From the diverse groups of papers presented in section 2.8.2, points that emerge from the literature review are:

- Basic concept of IEC 61499 standard (already considered in section 2.6.2).
- Migration techniques from classical PLC controller to IEC 61499 standard.
- Structure design methods to model, verify, and implement function blocks on the basis of the IEC 61499 standard principles.
- Distributed applications, and techniques based on IEC 61499 standard (already discussed in section 2.8).

The sub-sections below elaborate further on the points mentioned above.

### 2.9.1 Migration techniques from classical PLC controller to IEC 61499 standard

The work of Hussain and Frey (2005) based on the network enabled controllers technique migrates a PLC application to an IEC 61499 standard compliant one for distributed control system. They achieved the control of a didactic modular production system with four netmaster controller devices. Thramboulidis (2005) has highlighted the inefficiencies of IEC 61499 to support the whole process of distributed control application at software level.

Ferrarini (2006) has developed a very interesting software environment to convert IEC 61499 function blocks into a generic programming language. Molina *et al.* (2007) reviewed all features relevant to industrial standards related to PLC programming. They showed the limitations of PLC controllers for distributed control applications, and proved that IEC 61499 standard gives more reliability and availability.

Dai and Vyatkin (2009) developed a guide to migrate from standard PLC to IEC 61499 function blocks. They concluded that the simulation of IEC 61499 was successful, and the transformation from PLC to IEC 61499 standard was not 100% equivalent.

To partially conclude, there are many aspects that can be considered to migrate from PLC control to IEC 61499 standard such as:

- Reliability of the migration technique to be considered.
- Software development platform needed to migrate from PLC to IEC 61499 standard.
- The complexity of the process under control influences the migration techniques development.

## **2.9.2 Structure design methods to model, verify, and implement function blocks following the IEC 61499 standard principles**

After the introduction of the IEC 61499 standard, the number of design methods, verification environments and implementation platforms were gradually developed. The work of Vyatkin and Hanisch (1999) summarised the development of conventional modeling and verification of function blocks following the IEC 61499 standard concept. They demonstrated the reliability and reconfigurability of distributed system, but they emphasized that the standard has difficulties to be accepted in industry because it is still not clear. Wei (2001) implemented distributed function block architecture for process control based on IEC 61499 standard. He achieved reduction of design time, cost at engineering side, and the system was more flexible and maintainable.

The work of Hegny *et al.* (2010) proposed the use of an IEC 61499 industrial runtime environment to simulate plant behaviour. This research showed significant reduction of the implementation time. Yang and Vyatkin (2010) proved the feasibility of a full distributed automation design of baggage handling system automation. They used an embedded controller to control a group of conveyors, and they achieved portability, interoperability, reconfiguration and distribution of control applications.

Sorouri *et al.* (2012) proposed a systematic approach toward controller design of mechatronic components to synchronize the movements of a pick and place robot. They successfully implemented all control logics in NxtStudio software platform.

## **2.10 Conclusion**

In this chapter, some of the existing techniques of real time control design based on the IEC 61499 standard are studied. An overview of distributed control systems is given based on Intelligent Control and Multi-Agent Systems which represent two of the most widely used methods to develop distributed applications. A broad overview of the IEC 61499 standard, its advantages and disadvantages compared to its IEC 61131 counterpart is given. Other existing standards such as IEC 61508, 61850 and 61970 are discussed to show that the International Electro-technical Commission (IEC) possesses a wide range of standards that fit in the design of certain specific control applications. Existing modeling techniques in IEC 61499 such as the Model Transformation and Co-Modeling are overviewed. In this thesis Model Transformation technique is chosen to implement the concept of IEC 61499 standard-based function block programming.

The next chapter focuses on the mathematical modeling of the Magnetic Levitation System which is a highly nonlinear system.

## CHAPTER THREE: MATHEMATICAL MODEL OF THE MAGNETIC LEVITATION SYSTEM

### 3.1 Introduction

Mathematical modeling of a system is one of the most crucial tasks in the analysis and design of control systems. The first task of the control engineer when given a control problem is to undertake the development of the mathematical model of the plant to be controlled in order to have a perfect understanding of its behaviour. In practice, there are only few ways to obtain the model of a system. The control engineer can use the first principles of physics to obtain the dynamic of the system and write down the model. Another technique is the perform system identification of the physical plant to generate a model of the system. In certain cases both methods are used to obtain a set of differential equations that represent the behaviour of the system. In these forms, the frequency response and the system transient response are easily analysed. It is of massive importance to understand the meaning of the word system. Tewari, (2002) defined the word system in the control domain as a set of self contained processes under study. A control system by definition consists of the system to be controlled called the plant and the system which exercises control over the plant, called the controller.

Emmons, (2011) stated that the challenge in mathematical modelling is not to develop the most comprehensive descriptive model but to develop the simplest possible model that includes the main features of the phenomenon of interest.

The mathematical model can be represented in transfer function or state space. The transfer function method is mostly used to describe linear time invariant systems, whereas the state space representation can be applied for both linear and nonlinear systems. Although the analysis and design of linear control have been developed with precision by numerous researchers, their nonlinear counterparts are usually quite complex, and not so well developed.

In modelling nonlinear systems, the control systems engineer often has the task of determining not only how to produce accurately the mathematical description of the system, but, more importantly how to make accurate assumptions and approximations, at any necessary time, so that the system may be realistically characterized by a linear mathematical model (Kuo, 2003:77). It is important to understand that the analytical and computer simulation of any systems are as good as the model used to describe it. It should also be stressed that the

modern control engineer should place special emphasis on the mathematical modelling so that analysis and design problems can be conveniently solved by computers.

This chapter is organized in the following way that: an introduction to the modeling of physical systems is given in section 3.2. The model of the magnetic levitation is developed according to Kirchoff, Faraday and Newton's laws in section 3.3. In section 3.4 the nonlinear and linear mathematical models are constructed. In section 3.5 shows the results of the simulations in Matlab/Simulink environment of the plant, and section 3.6 presents the analysis of the results. The conclusion is done in section 3.7.

### **3.2 Modeling of physical systems**

Kuo and Golnararaghi, (2003) stated that the two most reliable methods of modelling linear systems are the transfer function method and the state-variable method. The transfer function is applied only to linear time-invariant systems, while the state equations can be applied in both linear and non-linear systems.

Despite the fact that the analysis and design of linear time invariant systems have evolved considerably in the past decades, their counterparts for non-linear systems are usually quite complex. Thus, the control systems engineer has often the task of determining not only the accuracy of the system described mathematically, but, more precisely, how to make realistic assumptions and approximations, at any necessary time, so that the system may be properly characterized by a linear representation of the mathematical model. It is clear that the analytical and computer simulation of a mathematical system is only as good as the model describes it. Modern control engineer must pay a special attention on the mathematical modelling of systems so that design and analysis of real life problems can be conveniently solved by computers.

#### **3.2.1 Modeling of Electrical systems**

The magnetic ball levitation system is made of two sub-systems: electrical and mechanical. According to Kuo and Golnararaghi, (2003) the classical way of writing equations of electrical networks is based on the loop method or the node method, both of which are formulated from the two Kirchoff's laws. The first law explains that the sum of voltages around a closed path is equal to zero in a circuit. This law answer the primary statement of the definition of potential in an electrical circuit. Since every point in the circuit has a specific value of potential, therefore travelling around an electrical circuit, through any path must bring one back to the potential. The

second law states that the sum of currents going through a node is always equal to the sum of currents exiting the node. This law shows that there is a current conservation in the electrical network.

### **3.2.2 Modeling of Mechanical systems**

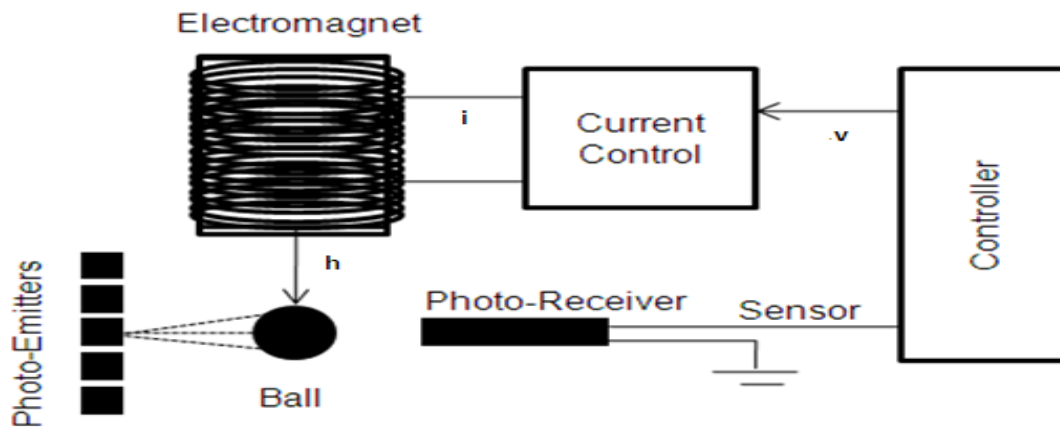
In real life control problem, most systems contain electrical as well as mechanical components; some systems even have hydraulic and pneumatic elements. From a mathematical point of view both electrical and mechanical components are analogous. In most of the control systems books, it is demonstrated that an electrical device is usually a mathematical analogous to its mechanical counterpart, and vice-versa. Describing the motion of mechanical elements can be done in various dimensions such as: translational, rotational or a combination of both movements. Newton's three laws of motion govern the equations of motion of mechanical systems which are often directly or indirectly formulated by them. The first law is based essential on Galileo's concept of inertia, this law asserts that every object in a state of constant movement tends to stay in that state of motion except if there is the application of an external force to it. The second law which represents the most powerful of all the three laws asserts that the relationship between the mass  $m$  of an object, its acceleration  $a$ , and a force  $F$  applied is  $F = ma$ . Acceleration and force are vectors; the force vector in this law has a direction exactly identical to the acceleration vector's direction. There is a fundamental difference between this law and the law of dynamics of Aristotle which declared that there is a velocity only if there is a force, but in accordance with Newton, an object with a certain velocity keeps that velocity except if a force is applied on it to generate acceleration or a variation in velocity. Aristotle law is more in accord to common sense but his analysis failed to value the role played by frictional forces. The third law affirms that for every action there is an equal and counter reaction.

### **3.3 Mathematical model derivation of magnetic levitation systems**

In this thesis a magnetic ball suspension systems is selected because it is a nonlinear inherently unstable system. The goal of the system is to control the position of the steel ball by controlling the current in the electromagnet through the input voltage. Figure 3.1 shows the schematic diagram's representation of the magnetic levitation system developed by (Ahmad and Javaid, 2010). The magnetic levitation system has two main parts: an electrical sub-system and a



mechanical sub-system. The dynamic of the system is derived on the basis of the first principles using the basic electrical and mechanical laws.



**Figure 3.1:** Diagram of the magnetic levitation system (Ahmad, Javaid, 2010)

### 3.3.1 Electrical sub-system

The dynamics of the coil are usually represented as an  $R-L$  equivalent circuit. The electrical sub-system can be developed according to Kirchoff's first law described in section 3.2.1. This sub-system works on the basis of the voltage applied to the coil, it creates an electromagnetic force that attracts the ball to the coil. The dynamics of the electrical sub-system is described as follow:

$$v = Ri + L \frac{di}{dt} \quad (3.1)$$

Where:

$v$  = Input voltage

$i$  = Winding current

$L$  = Winding inductance

$R$  = Winding resistance

The current through the winding creates an electromagnetic force  $f(h, i)$ .

### 3.3.2 Mechanical sub-system

According to Newton's third law which states that for every action there is an equal counter reaction, the force acting on the ball can be described as:

$$F = F(g) - f(h,i) \quad (3.2)$$

With:

$$F = ma$$

$$F(\bar{g}) = m\bar{g}$$

Where:

$F$  = Resultant force moving the ball

$F(g)$  = Gravitational force

$m$  = Mass of the ball

$\bar{g}$  = Gravitational acceleration

$h$  = Distance between the ball and the winding (ball position)

The force created by the winding defined as  $f(h,i)$  is a function of the air gap or distance between the winding and the ball,  $h$ , and the current supplied to the winding,  $i$ . It is found by the direct application of both Ampere's and Faraday's laws (Wong, 1986):

$$f(h,i) = -\frac{i^2 dL(h)}{2dh} \quad (3.3)$$

The total inductance is a nonlinear function of the position of the ball:

$$L = L_C + \frac{L_e h_e}{h} \quad (3.4)$$

$L_C$  = Constant

$L_e$  = Additional inductance in the equilibrium point

$h_e$  = Equilibrium point of the ball

By replacing Equation (3.4) in Equation (3.3), we obtain:

$$f(h,i) = -\frac{i^2 d}{2dh} \left[ L_C + \frac{L_e h_e}{h} \right] = \frac{L_e h_e i^2}{2 h^2}$$

$$f(h,i) = \frac{L_e h_e i^2}{2 h^2}$$

This can be written as:

$$\frac{L_e h_e}{2} \left( \frac{i^2}{h^2} \right) = k \left[ \frac{i^2}{h^2} \right] \quad (3.5)$$

Where:

$k$  = Proportional constant

$$k = \frac{L_e h_e}{2} \quad (3.6)$$

Then the electromagnetic force attracting the ball can be described as:

$$f(h, i) = k \frac{i^2}{h^2} \quad (3.7)$$

This force plays a role in the mechanical sub-system. It has to be bigger than the gravitational force of the ball in order to be possible that the ball is attracted to the coil.

The acceleration moving the ball is derived as second derivative of the position:

$$a = \frac{d^2 h}{dt^2} \quad (3.8)$$

Then on the basis of Equations (3.7) and (3.8) the dynamics of the ball are expressed as:

$$m \frac{d^2 h}{dt^2} = m \bar{g} - k \left( \frac{i^2}{h^2} \right)^2 \quad (3.9)$$

Equation (3.9) can also be expressed as:

$$\ddot{h} = \bar{g} - \frac{k}{m} \left( \frac{i^2}{h^2} \right) \quad (3.10)$$

### 3.3.3 Full model of the system

Equation (3.10) describes the mechanical subsystem and the Equation (3.2) describes the electrical subsystem. These two subsystems are connected through the current  $i$ , which can be described as:

$$\frac{di}{dt} = -\frac{R}{L} i + \frac{1}{L} v \quad (3.11)$$

### 3.4 Derivation of the state space model

Equations (3.9) and (3.10) describe the dynamic behaviour of the electrical and mechanical sub-systems. They represent the dynamic model of the levitation system expressed by nonlinear first order differential equations. The system is of third order.

It is necessary to derive the state space equation of the system to be used for the purpose of controller design. The following state space variables are introduced:

$x_1 = h$  = Position of the ball

$$x_2 = \frac{dh}{dt} = \text{Velocity of the ball} \quad (3.12)$$

$x_3 = i = \text{Inductor current}$

Substitution of Equations (3.12) into Equations (3.10) and (3.11) gives:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= g - \frac{kx_3^2}{mx_1^2} \end{aligned} \quad (3.13)$$

$$\dot{x}_3 = -\frac{R}{L}x_3 - \frac{1}{L}v$$

With initial conditions:  $x_1 = x_{10}$ ,  $x_2 = x_{20}$  and  $x_3 = x_{30}$ .

The vector format of the dynamics of the system is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ g - \frac{k}{m} \frac{x_3^2}{x_1^2} \\ -\frac{R}{L}x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} v, x(0) = x_0 \quad (3.14)$$

$$y = C[x_1 \quad x_2 \quad x_3]^T = [1 \quad 0 \quad 0][x_1 \quad x_2 \quad x_3]^T \quad (3.15)$$

In shorter, the Equations (3.14) and (3.15) can be expressed by the standard nonlinear model:

$$\dot{x} = f(x) + g(x)u \quad (3.16)$$

$$y = \bar{h}(x) \quad (3.17)$$

Where:  $f(x)$ ,  $g(x)$  and  $\bar{h}(x)$  are nonlinear functions of the state vector.

At equilibrium, the ball rate must strictly be equal to zero,  $\dot{h} = 0$ . The state satisfying this condition is:

$$x_e = [x_{e1} \quad 0 \quad x_{e3}]^T$$

$x_{e1} = \text{Ball position at equilibrium}$

$x_{e2} = \text{Ball velocity at equilibrium} = 0$

$x_{e3} = \text{Inductor current at equilibrium}$

The inductor current at equilibrium can be calculated from the equation:

$$\dot{x}_2 = -g - \frac{k x_3^2}{m x_1^2} \quad (3.18)$$

Then:

$$-g - \frac{k x_3^2}{m x_1^2} \quad (3.19)$$

And from here:

$$x_3 = x_1 \sqrt{\frac{gm}{k}} \quad (3.20)$$

The state model (3.16), (3.17) is used further in the thesis for design of nonlinear control.

### 3.4.1 Linearization of the magnetic levitation system at equilibrium

The linearization of the nonlinear model of the magnetic levitation system is done around its equilibrium point; it is achieved by using only the linear terms from the Taylor series expansion of the nonlinear model (3.16), (3.17).

At equilibrium:

$$y_e = x_{e1} = \text{Constant}$$

$$x_{e2} = \frac{dx_{e1}}{dt} = 0$$

$$x_{e3} = x_{e1} \sqrt{\frac{gm}{h}} \quad (3.21)$$

The vector matrix of the system can be expressed as follows (Kuo, 2003):

$$\Delta x = A\Delta x + B\Delta r \quad (3.22)$$

$A$  and  $B$  can be expressed analytically as follow (Kuo, 2003):

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} \end{bmatrix}; \quad \begin{bmatrix} \frac{\partial f_1}{\partial v_1} \\ \frac{\partial f_2}{\partial v_1} \\ \frac{\partial f_3}{\partial v_1} \end{bmatrix} \quad (3.23)$$

The linearization according to Taylor series is performed using the functions  $f_1, f_2, f_3$  described below:

$$f_1 = x_2$$

$$f_2 = g - \frac{k}{m} \frac{x_3^2}{x_1^2} \quad (3.24)$$

$$f_3 = -\frac{R}{L} x_3 + \frac{1}{L} v$$

The derivatives of the three functions based on the three states of the system are:

- Derivatives of the first function  $f_1$ :

First derivative:

$$\frac{\partial f_1}{\partial x_1} = \frac{\partial}{\partial x_1}(x_2) = 0 \quad (3.25)$$

Second derivative:

$$\frac{\partial f_1}{\partial x_2} = \frac{\partial}{\partial x_2}(x_2) = 1 \quad (3.26)$$

Third derivative:

$$\frac{\partial f_1}{\partial x_3} = \frac{\partial}{\partial x_3}(x_2) = 0 \quad (3.27)$$

- Derivatives of the second function  $f_2$ :

First derivative:

$$\frac{\partial f_2}{\partial x_1} = \frac{\partial}{\partial x_1} \left[ g - \frac{k}{m} \frac{x_3^2}{x_1^2} \right] = -\frac{2k}{m} \frac{x_3^2}{x_1^3} \quad (3.28)$$

Second derivative:

$$\frac{\partial f_2}{\partial x_2} = \frac{\partial}{\partial x_2} \left[ g - \frac{k}{m} \frac{x_3^2}{x_1^2} \right] = 0 \quad (3.29)$$

Third derivative:

$$\frac{\partial f_2}{\partial x_3} = \frac{\partial}{\partial x_3} \left[ g - \frac{k}{m} \frac{x_3^2}{x_1^2} \right] = -\frac{2k}{m} \frac{x_3}{x_1^2} \quad (3.30)$$

- Derivatives of the third function  $f_3$ :

First derivative:

$$\frac{\partial f_3}{\partial x_1} = \frac{\partial}{\partial x_1} \left[ -\frac{R}{L} x_3 + \frac{1}{L} v \right] = 0 \quad (3.31)$$

Second derivative:

$$\frac{\partial f_3}{\partial x_2} = \frac{\partial}{\partial x_2} \left[ -\frac{R}{L} x_3 + \frac{1}{L} v \right] = \frac{2k}{L} \left( \frac{x_3}{x_1^2} \right) \quad (3.32)$$

Third derivative

$$\frac{\partial f_3}{\partial x_3} = \frac{\partial}{\partial x_3} \left[ -\frac{R}{L} x_3 + \frac{1}{L} v \right] = -\frac{R}{L}$$

(3.33)

The terms of the control matrix  $B$ , can also be found with the same method:

- Derivative of  $f_1$  according to the voltage input:

$$\frac{\partial f_1}{\partial v} = \frac{\partial}{\partial v} [x_2] = 0 \quad (3.34)$$

- Derivative of  $f_2$  according to the voltage input:

$$\frac{\partial f_2}{\partial v} = \frac{\partial}{\partial v} \left[ g - \frac{k}{m} \left( \frac{x_3}{x_1^2} \right) \right] = 0 \quad (3.35)$$

- Derivative of  $f_3$  according to the voltage input:

$$\frac{\partial f_3}{\partial v} = \frac{\partial}{\partial v} \left[ -\frac{R}{L} x_3 + \frac{1}{L} v \right] = \frac{1}{L} \quad (3.36)$$

After substitution of Equations (3.25)-(3.36) in the Equation (3.23), the linearised state space representation of the system is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2k}{m} \left( \frac{x_3}{x_1^2} \right) & 0 & -\frac{2k}{m} \left( \frac{x_3}{x_1^2} \right) \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} v, x(0) = x_0 \quad (3.37)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.38)$$

Where the coefficients of the matrices  $A$  and  $B$  are calculated for the equilibrium values of the state variables given by Equation (3.21).

In the Humusoft design (Humusoft, 2002) of the magnetic levitation system, the following values of the system parameters are given:

$$m = 8.27 \times 10^{-3} \text{ Kg}$$

$$\bar{g} = 9.81 N \times Kg^{-1}$$

$$R = 1[\Omega]$$

$$k = 0.0001$$

$$L = 0.01[H]$$

$$i = 0.84[A]$$

$$x_{e1} = 0.012[m]$$

The state space representation of the system becomes:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 9875 & 0 & -141.1 \\ 0 & 0 & -100 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix} v, x(0) = x_0 \quad (3.39)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.40)$$

The transformation of the representation of the system to transfer function is:

$$G(s) = \frac{b(s)}{a(s)} = C(sI - A)^{-1} B + D$$

(3.41)

Where  $D=0$  in the considered case.  $b(s)$  and  $a(s)$  are the polynoms in the numerator and denominator of transfer function, and:

$$sI = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix}$$

$$G(s) = \frac{-1.411 \times 10^4}{(s - 99.37)(s + 99.37)(s + 100)} \quad (3.42)$$

The system is of third order so we have three poles and no zeros. Two poles are located on the left hand side and one is located on the right hand side of the complex plane which means that the system is highly unstable.

$$P_1 = -100$$

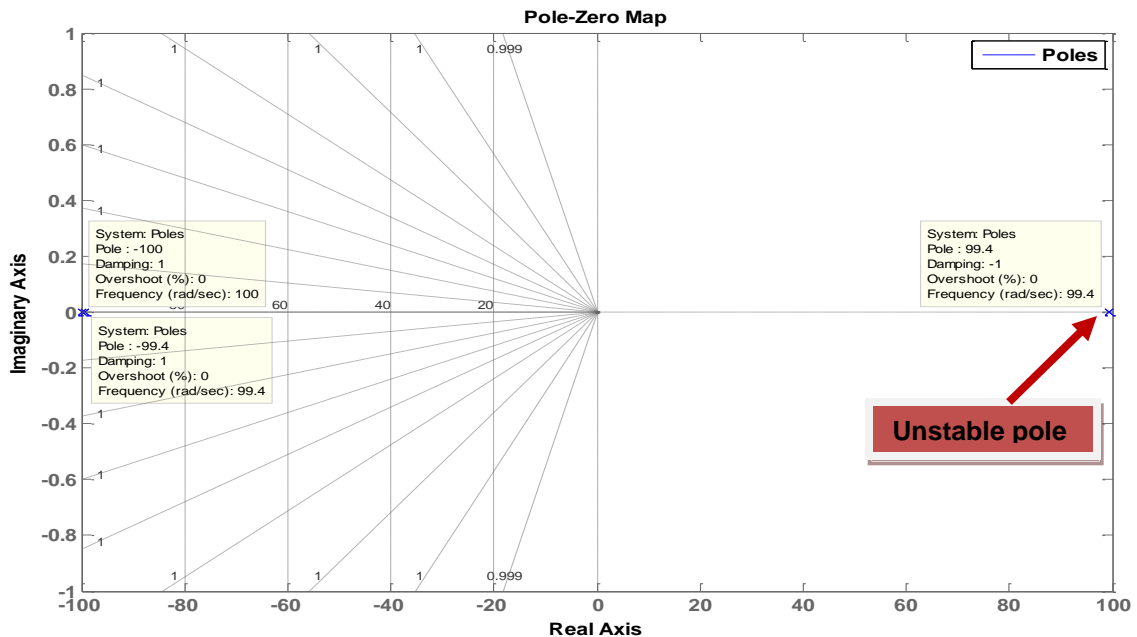
$$P_2 = -99.37$$

$$P_3 = 99.37 \text{ (Pole making the system unstable)}$$



The simulation showed that the third pole is difficult to be controlled. A poles-zeros map of the obtained third order transfer function is presented in Figure 3.2

The next section focuses on the simulation and result analysis of the dynamic responses of the magnetic ball levitation system.



**Figure 3.2: Poles-zeros map of the magnetic levitation system**

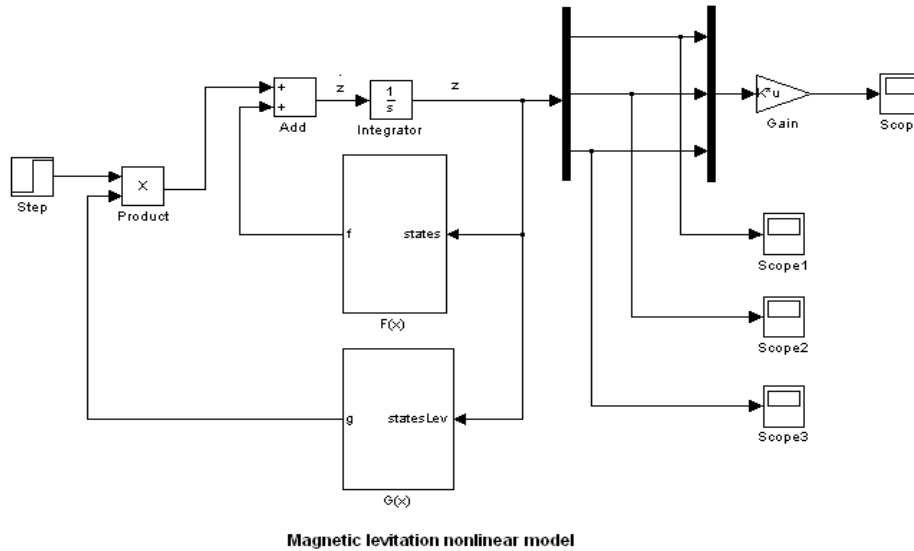
### 3.5 Simulation and analysis of the responses of the magnetic levitation system

Simulation is one of the most important tasks in control system because it allows the control systems engineer to clearly understand the behaviour of the system and therefore decide on the most appropriate control technique to apply to stabilize the system. This process is done through computer programming and in our case; MATLAB is used as programming platform loaded on a normal desktop computer. Diverse simulations were done to have a clear understanding of the performance of the system when there is no compensator to stabilize the magnetic levitation system:

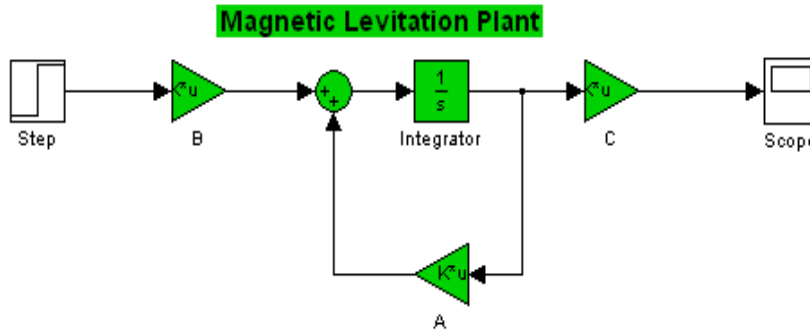
The nonlinear and linear models of the magnetic levitation have the same parameters. The parameters necessary for the calculation are introduced in Matlab workspace by the same program which is called "*levitationparameters.m*". The corresponding Simulink files for nonlinear and linear models called "*levitationnonlinear.mdl*" and "*levitationlinear.mdl*". The m-file is given

in Appendix A.1. Figure 3.3 and 3.4 show the Simulink diagrams of the nonlinear system, and the linear system.

- Simulations of the nonlinear and the linearized one are done according to the same initial conditions. The initial position of the ball is set at the following:  $x_{e1} = 0.012[m]$ .



**Figure 3.3: Magnetic levitation system nonlinear state space model in Simulink environment**



**Figure 3.4: Magnetic levitation system linear model**

- Simulations of the nonlinear and the linearized one are done according to the same initial conditions. The initial position of the ball is set at the following:  $x_{e1} = 0.012[m]$ .

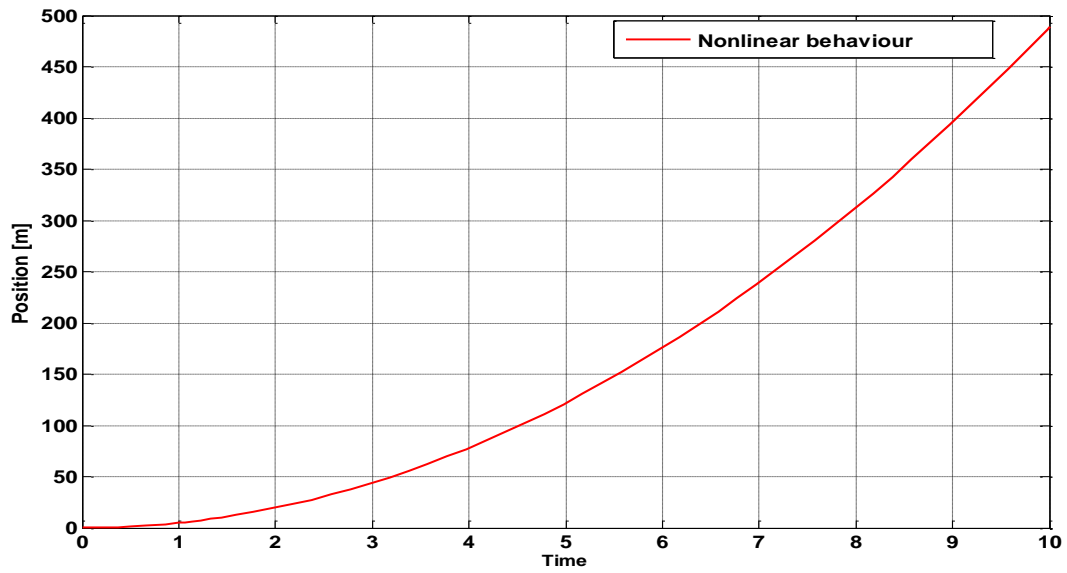


Figure 3.5: Open loop response of the nonlinear system when the initial position is 0.012[m]

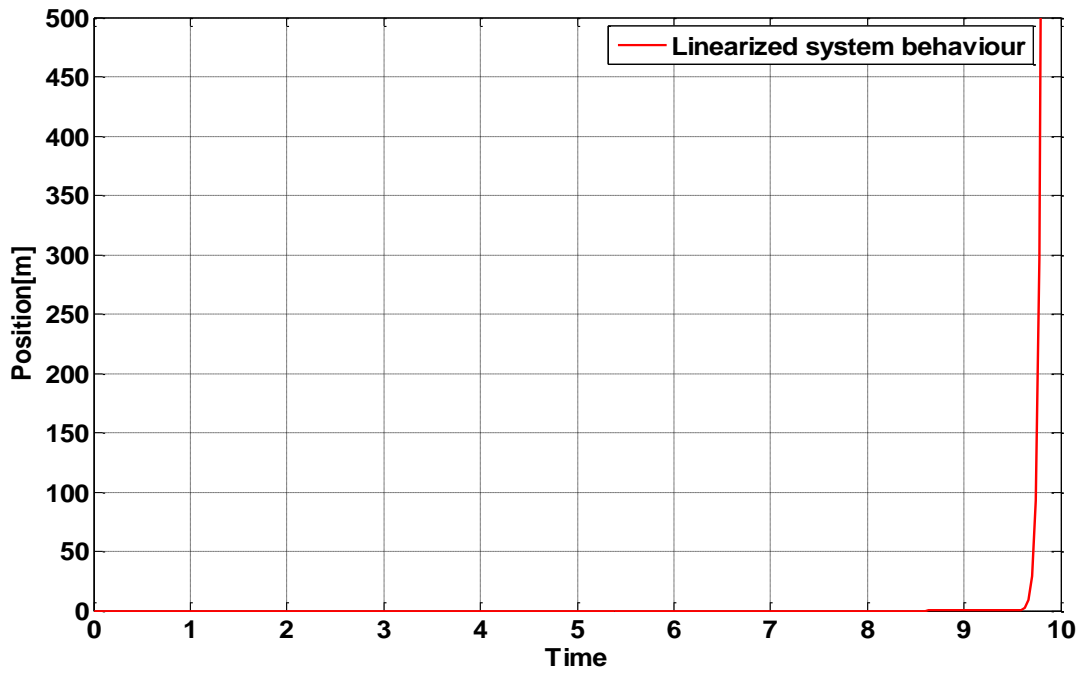


Figure 3.6: Open loop response of the linearized system when the initial position is 0.012[m]

### 3.6 Analysis of results

The behaviours of the magnetic levitation system are shown from Figure 3.5 to 3.6. Referring to the physical magnetic levitation system, the above obtained results are discussed. The linear model is a good approximation of the plant behaviour only around the none stable equilibrium point  $x_{e1} = 0.012[m]$ . Ad the plant is not under control and set continuous force is applied, the magnetic levitation system is moving far from the point  $x_{e1} = 0.012[m]$ .

The position state response of the magnetic levitation nonlinear model shows that under step continuous force, the ball position moves toward infinity.

This analysis confirms that that the magnetic ball levitation is a nonlinear open loop unstable system that needs to be controlled effectively.

### 3.7 Conclusion

In this chapter, the modelling of the magnetic levitation system is done, and its closed loop behaviour is analyzed. Simulations in Matlab/Simulink environment are done to assess the trajectories of the system. These results show open loop instability of the system. Based on the mathematical model, the linear controllers and nonlinear controllers for magnetic levitation, the stability of the closed-loop system is achieved in the following chapters of the thesis.

Chapter 4 gives short description of the control theory which explains the different control system techniques that can be utilized to stabilize the magnetic ball levitation system.

## CHAPTER FOUR: CONTROL THEORY OVERVIEW

### 4.1 Introduction

Slotine and Li, 1991 stated that linear control is a fully grown subject with diversity of strong techniques and a long history of successful projects. Consequently, it is normal for one to ponder why researchers and designers around the world, from such vast research areas as process control, have lately showed a strong interest in developing methodologies of nonlinear control applications. Diverse reasons can be cited for this interest (Isidori, 1985), (Isidori, 1994) and (Atherton, 1996):

- Improvement of existing control systems: Slotine and Li, 1991 explained that linear control procedures rely on the consideration that there is a small operating region that validates the linear model considered. When the operation region is much larger, a linear controller is expected to show a very poor performance or an unstable behavior, because the nonlinearities in the system are not properly handled. On the other hand, nonlinear controllers may directly sustain the nonlinearities in bigger operating region.
- Analysis of hard nonlinearities: One more consideration of linear control is that the model of the system is definitely linearizable (Hendrik and Girard, 2010). Nevertheless, in control systems designs there are nonlinearities whose discontinuous behavior cannot be linearly approximated. These “hard nonlinearities” known as backlash, saturation, Coulomb friction, dead-zones, and hysteresis are mostly found in control engineering. Linear techniques cannot give a proper approximation of their effects, therefore nonlinear analysis methods must be derived to forecast a system’s functioning in the presence of these intrinsic nonlinearities.
- Dealing with model uncertainties: Astom, 2000 explained that in designing linear controllers, it is normally necessary to suppose that the model of the system has fairly well known parameters. Nevertheless, many control issues include uncertainties in the parameters of the model. The reason for this could be due to a slow time change of parameters, or to an unexpected variation in parameters. A linear controller designed with inaccurate or outdated parameters of the model may show dramatic performance deterioration or even instability. Nonlinearities can be deliberately added into the control system especially at the controller part to make sure that the uncertainties of the model can be accepted. There are two types of nonlinear controllers for this aspiration: adaptive controllers and robust controllers.

- Design simplicity: Accurate designs of nonlinear controllers may be easier and more instinctive than their linear counterparts (Slotine and Li, 1991). This is a priori paradoxical result that emanates from the fact that the designs of nonlinear controllers are often greatly rooted into the physical representation of the plant.

The utilization of nonlinear control techniques may be related or unrelated to other reasons such as cost and performance optimality. Linear control can involve high quality actuators and sensors to produce linear behaviour in a defined operating range, while nonlinear control can permit the use of less expensive components with nonlinear characteristics.

The field of nonlinear control systems is of great importance in automatic control. It occupies a growing prominent position in control engineering, as reflected by the ever-increasing number of papers and reports on nonlinear control research and applications.

Therefore to discuss the concepts of linear and nonlinear control theory, this chapter is structured as follow: section 4.1 gives a broad introduction of the reasons behind the increasing interest in applications of nonlinear control methodologies. Linear systems are presented in section 4.2. Techniques of linear controller design are discussed in section 4.3. Section 4.4 deals with the behaviour of nonlinear systems. Further nonlinear control systems design is presented in section 4.5 and finally the conclusion is done in section 4.6.

## 4.2 Linear systems

Linear control theory has been predominant in the study of linear time invariant (LTI) systems of the form:  $\dot{x} = Ax$ , where  $x$  is the states' vector and  $A$  is the matrix of the system. The following properties are related to linear time invariant systems (Rowell, 2002), (Nise, 2008) and (Lygeros and Ramponi, 2013):

- A linear system has its own point of equilibrium if  $A$  is non singular;
- The stability of the point of equilibrium is achieved if all eigenvalues of  $A$  have negative real parts, despite the initial conditions;
- The transient behavior of a linear system is made of the natural modes of the system, and it is possible to solve analytically the general solution;
- In the presence of an external input  $u(t)$ , with:  $\dot{x} = Ax + Bu$  the response of the system has many interesting properties. Firstly the principle of superposition is fulfilled and

secondly, its asymptotic stability means bounded-input bounded-output stability when  $u$  is present.

#### 4.2.1 Stability of linear control systems

In control systems, when all types of systems are considered: linear, nonlinear, time-variant and time-varying systems, stability can be defined in various forms. In linear control systems, stability is classified in different forms such as: absolute stability and relative stability. Slotine and Li, 1991 stated that absolute stability makes reference to the condition whether the system is stable or unstable; it is a yes or no answer. Once the stability of the system is proven, the next step is to find how stable it is, and the degree of stability is a measure of the relative stability.

Time invariant systems can be categorized by two types of responses:

- Zero-state response: it is due to the input of the system only; at this moment all initial conditions of the system are zero.
- Zero-input response: it is due to the initial conditions of the system only; all inputs are zero.

In some cases of linear control systems, the system may be subjected to both inputs and initial conditions, in such cases the total response is expressed as:

$$\text{Total response} = \text{zero state response} + \text{zero input response} \quad (4.1)$$

##### 4.2.1.1 Zero-input and asymptotic stability of continuous data systems

Kuo and Golnaraghi, 2003 defined zero-input stability as a condition where the input is zero and the system is driven only by its initial conditions. It also depends on the roots of the characteristic equation.

To illustrate this concept: If the input of  $n$ th-order system be zero, and the output due to the initial conditions be  $y$ . Then,  $y$  can be expressed as:

$$y = \sum_{k=0}^{n-1} g_k y^{(k)}(t_0) \quad (4.2)$$

Where:

$t_0$  is the initial time.

$$y^{(k)}(t_0) = \frac{\partial^k y}{\partial t^k}; \text{ With } t = t_0 \quad (4.3)$$

And  $g_k$  denotes the zero-input due to  $y^{(k)}(t_0)$ . The definition of zero input stability is: if the zero-input response  $y$ , subject to the finite initial conditions,  $y^{(k)}(t_0)$ , goes to zero as  $t$  gets to infinity, therefore system is considered to be zero-input stable, or stable; else, the system is unstable.

Mathematically a linear time-invariant system is said to be zero-input stable if for any set of finite  $y^{(k)}(t_0)$ , there is the existence of a positive number  $M$ , which depends on  $y^{(k)}(t_0)$ , such that (Albertini, 1996):

- $|y| \leq M < \infty$  for all  $t \geq t_0$  (4.4)
- $\lim_{t \rightarrow \infty} |y| = 0$

In the last equation, it is required that the magnitude  $y$  gets to zero as time goes to infinity, the zero-input stability is commonly known as asymptotic stability.

#### 4.2.1.2 Rough-Hurwitz criterion

The Rough-Hurwitz criterion is a mathematic property that gives information on the absolute stability of a linear time-invariant system that possesses a characteristic equation with continuous coefficients. The criterion examines if the characteristic equation has any roots in the right half s-plane. The criterion shows also the roots that lie on the  $j\omega$ -axis and in the left-half plane without solving for the zeros (Clark, 1996) and (Nise, 2003).

#### 4.2.1.3 Nyquist criterion

The Nyquist criterion is a semi-graphical method that gives information on the residue between the number of poles and the number of zeros of the closed loop transfer function that are in the right-half s-plane looking at the behaviour of the transfer function of the Nyquist plot (Berman, 1961), (Haris and Valencia, 1978) and (Huang, 1993).

#### 4.2.1.4 Bode diagram

Kuo and Golnaraghi, 2003 described the bode plot as a diagram showing the closed-loop transfer function magnitude  $G(j\omega)H(j\omega)$  in dB and the phase of  $G(j\omega)H(j\omega)$  in degrees, all



versus the frequency  $\omega$ . The stability of the closed-loop system can be found by looking at the behaviour of the plots (York, 2009).

#### 4.2.2 Controllability

To define the concept of controllability, a linear-time invariant system of the following dynamic equations can be defined:

$$\frac{\partial x}{\partial t} = Ax + Bu \quad (4.5)$$

$$y = Cx + Du \quad (4.6)$$

Where  $x$  is the  $n \times 1$  state vector,  $u$  is the  $r \times 1$  input vector, and  $y$  is the  $p \times 1$  output vector.  $A, B, C$  and  $D$  are coefficients of the suitable dimensions.

The input state  $x$  is said to be controllable at  $t = t_0$  if there is the existence of a piecewise constant input  $u$  that will drive the state to any final state  $x(t_f)$  for a finite time  $(t_f - t_0) \geq 0$ . If every state  $x(t_0)$  of the system is controllable in a finite interval, the system is regarded as completely controllable or, simply, controllable.

#### 4.2.3 Observability

In control systems, a system is considered as totally observable if every initial state  $x(t_0)$  can be precisely determined from the measurement of the output  $y$  over the finite interval of time  $t_0 \leq t \leq t_f$ . This definition means that every state of  $x$  influences the output  $y$ . For a linear time invariant system, if the rank of the observability matrix  $O = \begin{bmatrix} C & CA & CA^2 & \dots & \dots & CA^{n-1} \end{bmatrix}$  has the same value as the number of states, then the system is observable (D'Azzo, 1988). A linear time variant system is observable if and only if the columns of the matrix  $C\Phi(t_0)$  are linearly independent in the interval  $[t_0, t_1]$ .

### 4.3 Techniques of linear controller design

The linear system controller design techniques are well studied in both classical and modern control fields. Many methods have been developed for both complex and time domain. Techniques such as root locus, pole placement using feedback, robustness and optimal control are discussed in the next sections.

### 4.3.1 Root locus technique

The root locus is a graphical method applied to represent the closed-loop poles of a linear time variant system. The root locus gives the graphical representation of a system transient response and its stability. It is also used to show qualitatively the operation of a system which has various changes in parameters.

Ogata, 2002 defined the root locus as a fundamental method to add poles and zeros to the open loop transfer function of a system and to force the root loci to pass through the desired closed loop poles in the s-plane.

The main properties of the root locus technique are the magnitude and angle requirements (Shinners, 1998), (Kuo and Golnaraghi, 203) and (Gosh, 2007):

- The magnitude of an open loop transfer function is:

$$|KG(s)H(s)| = 1 \text{ or } K = \frac{1}{|G(s)H(s)|} \quad (4.7)$$

- The angle of an open-loop transfer function is an odd multiple of  $180^\circ$ :

$$\begin{aligned} \angle(KG(s)H(s)) &= n180^\circ \text{ for } n = \pm 1, \pm 3, \pm 5, \dots \text{ or} \\ \angle(G(s)H(s)) &= n180^\circ \text{ for } n = \pm 1, \pm 3, \pm 5, \dots \end{aligned} \quad (4.8)$$

### 4.3.2 Pole placement design using feedback

In control systems it can be required to change the characteristics of a plant by using a closed-loop controller model in which a controller is designed to place the desired poles at desired locations. Such a design technique is referred as the pole-placement approach. Tewari, 2002 demonstrated that in classical control design approach using a controller transfer function with a few parameters is insufficient to place all the closed-loop poles at desired locations. The state space approach using full-state feedback provides sufficient number of controller design parameters to move all the closed-loop poles independently of each other. The full state-feedback is regarded as a controller which generates the input vector,  $u$ , according to the following control law:  $u = K[x_d - x] - K_d x_d - K_n x_n$  (4.9)

Each term of the control law has a specific meaning,  $x$  represents the state-vector of the plant,  $x_d$  is the desired state vector,  $x_n$  is the noise state vector and  $K, K_d$  and  $K_n$  are the controller gain matrices. The desired state vector,  $x_d$  and the noise state vector,  $x_n$  are generated by

external processes, and act as inputs of the control system. The challenge of the control system designer is to design a controller to achieve the desired state-vector in the steady state, while counteracting the effect of noise. The input vector  $u$ , or the control law is applied to the plant described by a set of state and output equations:

$$\dot{x} = Ax + Bu + Fx_n \quad (4.10)$$

$$y = Cx + Du + Ex_n \quad (4.11)$$

$F$  and  $E$  are the noise coefficient matrices in the state and output equations, respectively.

There are different controllers based on the pole placement technique. For example, when  $x_d = 0$ , the controller designed based on this assumption is called a regulator. In some simple control system approaches, it is assumed that all the measurements are perfect, and the modeling of the plant is error free which means that all undesirable inputs of the system in form of noise are non existent  $x_n = 0$ . The control law is then reduced to  $u = -Kx$ . The closed loop system is then represented as:

$$\dot{x} = Ax + Bu \quad (4.12)$$

$$y = Cx + Du \quad (4.13)$$

$$u = -Kx \quad (4.14)$$

If the value of  $u$  is substituted in Equations (4.12) and (4.13), the closed-loop state and output equations of the regulator as follows:

$$\dot{x} = (A - BK)x \quad (4.15)$$

$$y = (C - DK)x \quad (4.16)$$

If the plant is completely controllable the matrix  $K$  gives the possibility to place the poles of the characteristic equation to some desired places.

$$\Delta(s) = \det[sI - (A - BK)] = 0 \quad (4.17)$$

If the system is controllable then it can be represented in controllable canonical form (CCF):

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (4.18)$$

The feedback gain matrix is:

$$K = [k_1 \quad k_2 \quad \dots \quad k_n]; \text{ with: } k_1, k_2, \dots, k_n \text{ real constants.} \quad (4.19)$$

The full system is represented as:

$$A - BK = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 - k_1 & -a_1 - k_2 & -a_2 - k_3 & \dots & -a_{n-1} - k_n \end{bmatrix} \quad (4.20)$$

The eigenvalues of the system can be found from the characteristic equation:

$$|sI - (A - BK)| = s^n + (a_{n-1} + k_n)s^{n-1} + (a_{n-2} + k_{n-1})s^{n-2} + \dots + (a_0 + k_1) \quad (4.21)$$

This characteristic equation is applicable at desired places under the restriction that the complex poles have to be complex conjugates. The pole placement procedure is based on this capability of the feedback and has the following steps (Walter, 1944), (Rohs *et al.*, 1993), (Gojic and Lelic, 1996) and (Panos and Anthony, 1997):

- Choose the desired places of the closed loop poles.
- Determine the feedback matrix
- Build the transient response, as the system behavior depends also on zeros and the fulfillment of the desired behavior has to be checked.

### 4.3.3 Robustness

Tewari, 2002 defined the robustness of a control system as its sensitivity to unmodeled dynamics. It is the part of the behavior of a selected control system which is not part of the mathematical model of the control system (governing differential equations, transfer function, etc.). In control system, it is practically impossible to model all physical processes mathematically. It is crucial to find out if the control system based on the model derived mathematically really works. Performance and stability objectives in the presence of noises in a

control system must be met in order to deduce that the system is robust. Thus, robustness is a ideal property that shows whether a control system is not vulnerable to uncertainties in its mathematical model. To be specific, robustness can be subdivided in two parts: stability robustness and performance robustness, relying on whether the control system designer is looking at the robustness of the stability of a system (location of system's poles), or that of its performance objectives (peak overshoot, settling time, etc.) (Guo, 1997).

#### 4.3.4 Optimal control formulation for regulators

Optimal control provides an alternative design strategy by which all control design parameters can be determined even for multi-inputs, multi-outputs systems (Tewari, 2002). Optimal control permits to directly formulate the performance objectives of a control system. More importantly, optimal control produces the best feasible control system to meet the objectives of a set of performance. The objective of optimal control problem must be the time integral function of the sum of transient energy and control energy indicated as functions in time (Ogata, 2002).

To illustrate the concept of optimal control, a linear model of a plant is represented in state equation:

$$\dot{x} = Ax + Bu \quad (4.22)$$

The system is time varying because the formulation of optimal control problem is generally made for time-varying systems. If we design a full state-feedback regulator for the linear plant in such a way that the control law input is given by:

$$u = -Kux \quad (4.23)$$

The control law is linear and since the plant is also linear, therefore the closed-loop control system would also be linear. The expression of the control energy is  $u^T Ru$ , where  $R$  a square, symmetric matrix is called the control cost matrix. The expression of the control energy is known as quadratic form, because the scalar function,  $u^T Ru$ , possesses quadratic functions elements of  $u$ . In the same fashion, it is possible for the transient energy to be derived in a quadratic form as  $x^T Qx$ , where  $Q$  the state weighting matrix is a square, symmetric matrix. The objective function is expressed as:

$$J(t, t_f) = \int_t^{t_f} [x^T(\tau)Q(\tau)x(\tau) + u^T(\tau)R(\tau)u(\tau)]d\tau$$

Where:  $t$  = initial time and  $t_f$  = final time (4.24)

The control is exercised in  $t$  and  $t_f$  more specifically, the control starts at  $\tau = t$  and stops at  $\tau = t_f$ , where  $\tau$  is the variable of integration. The objective of the optimal control problem is to solve the feedback gain matrix  $K$ , in such a way that the scalar objective function  $J(t, t_f)$  is minimized.

By substituting the control law equation into the state-equation, a closed-loop state equation representation of the system is then obtained:

$$\dot{x} = [A - BK]x = A_{CL}x \quad (4.25)$$

The closed-loop state dynamics matrix is:

$$A_{CL} = [A - BK] \quad (4.26)$$

The solution of the closed-loop state equation is:

$$x(t) = \phi_{CL}(t, t_0)x(t_0) \quad (4.27)$$

Where:  $\phi(t, t_0)$  = state transition matrix

The objective function can be expressed in terms of  $x$  and the following expression is found:

$$J(t, t_f) = \int_t^{t_f} x^T(\tau) \phi_{CL}^T(\tau, t) [Q(\tau) + K^T(\tau)R(\tau)K(\tau)] \phi_{CL}(\tau, t) x_0(t) d\tau \quad (4.28)$$

Or, taking the initial state-vector,  $x$ , outside the integral sign, the expression becomes:

$$J(t, t_f) = x^T M(t, t_f) x_0 \quad (4.29)$$

$$\text{Where: } M(t, t_f) = \int_t^{t_f} \phi_{CL}^T(\tau, t) [Q(\tau) + K^T(\tau)R(\tau)K(\tau)] \phi_{CL}(\tau, t) d\tau \quad (4.30)$$

The Equation 4.29 shows that objective function is a quadratic function of the initial state  $x(t)$ . Hence the linear optimal regulator problem posed in Equations (4.22) and (4.23) is also called the linear quadratic regulator (LQR) problem. Since  $Q$  and  $R$  are both symmetric then  $J(t, t_f)$  can be expressed by substituting Equation (4.27) into Equation (4.28):

$$J(t, t_f) = \int_t^{t_f} x^T(\tau) [Q(\tau) + K^T(\tau)R(\tau)K(\tau)] x(\tau) d\tau \quad (4.31)$$

According to Leibniz rule, Equation 4.31 can be partially differentiated with respect to the lower limit:

$$\frac{\partial J(t, t_f)}{\partial t} = -x^T [Q + K^T R K] x \quad (4.32)$$

Similarly, the partial differentiation of Equation (4.29) can be partially differentiated:

$$\frac{\partial J(t, t_f)}{\partial t} = \begin{bmatrix} \dot{x} \\ \cdot \end{bmatrix}^T M(t, t_f) x + x^T \left[ \frac{\partial J(t, t_f)}{\partial t} \right] x + x^T M(t, t_f) \dot{x} \quad (4.33)$$

Since  $\dot{x} = A_{CL}x$ , Equation (4.33) becomes:

$$\frac{\partial J(t, t_0)}{\partial t} = x^T \left[ A_{CL}^T M(t, t_f) + \frac{\partial M(t, t_f)}{\partial t} + M(t, t_f) A_{CL} \right] x \quad (4.34)$$

By equating Equations (4.32) and (4.34), it is obtain:

$$-\frac{\partial M(t, t_f)}{\partial t} = A_{CL}^T M(t, t_f) + M(t, t_f) A_{CL} + [Q + K^T R K] \quad (4.35)$$

The linear optimal problem consists of finding the gain matrix  $K$  such that the solution  $M(t, t_f)$  to Equation (4.35) is minimized, subject to initial condition  $M(t, t_f) = 0$ . The choice of matrices  $Q$  and  $R$  is left to the control system designer.

#### 4.4 Behaviour of nonlinear systems

Physical systems are intrinsically nonlinear, therefore all control systems are considered to be nonlinear up to a certain point. The description of nonlinear systems can be obtained by developing nonlinear differential equations. In some control systems applications, the operating range of the control system can be quite small, and if they contain smooth nonlinearities, then the control system may be fairly estimated by a linearized system, whose dynamics is represented in terms of a set of linear differential equations (Isidori, 1994).

##### 4.4.1 Nonlinearities

Slotine and Li, 1991 classified nonlinearities as inherent (natural) and intentional (artificial). Nonlinearities considered as inherent normally come with system's hardware and motion. Such nonlinearities are found in applications that involve centripetal forces in rotational motion, and Coulomb friction between touching surfaces. On the other hand, intentional nonlinearities are artificially included by the designer of the control system. Typical examples of such nonlinearities are found in adaptive control laws and bang-bang optimal control laws.

Nonlinearities can also be categorized in terms of their mathematical characteristics, such as continuous and discontinuous. Discontinuous systems are described as "hard" nonlinearities because linear functions cannot locally approximated them. They are generally found in control systems, both in small and large operating range. The magnitude of the hard nonlinearities

determine if a system is whether linear or non-linear, they also affect the performance of the system.

#### **4.4.2 Fundamentals of Lyapunov theory**

The most beneficial and common approach for analyzing nonlinear control systems' stability is the theory proposed by the Russian mathematician Alexandr Mikhailovich Lyapunov in the late 19<sup>th</sup> century (Slotine, 1991). The work of Lyapunov, "The general Problem of Motion Stability", deals with two methods to analyze the stability of a system: Linearization method and direct method, it was issued in 1892. Lyapunov demonstrated that the linearization method concludes if a linear system is locally stable around its point of equilibrium from the stability methodologies of its linear approximation (Kawski, 2009). On the other hand, the direct method is not limited to local motion, and finds the properties of stability of a nonlinear system by deriving a "scalar energy like" function for the system and investigating the function's time change.

Nowadays, Lyapunov's linearization method has come to demonstrate the theoretical importance of the control of linear systems, while Lyapunov's direct method has proven to be the most crucial tool for nonlinear systems analysis and design. Together both methods constitute the Lyapunov stability theory (Lyapunov, 1892).

##### **4.4.2.1 Linearization and local stability**

The linearization method of Lyapunov deals with the local stability of a nonlinear system (Yang, 2013). It is a formal representation of the assumption that a nonlinear system should behave identically to its linearized estimation for small operating ranges. Due to the fact that all physical systems are inherently nonlinear, the linearization methodology of Lyapunov serves as the fundamental ground of making use of linear control methods in practice.

To satisfy Lyapunov's linearization method the following theorems have to be applied (Brill, 1987):

- If the linearized system is globally stable, then the point of equilibrium is asymptotically stable.
- If the linearized system is unstable, then the system is unstable at equilibrium.
- If the linearized system shows marginal stability, then no conclusion can be made from the linear estimation.



#### 4.4.2.1.1 Direct method of Lyapunov

The fundamental approach of Lyapunov's direct method is to expand mathematically the basic physical observation (Slotine and Li, 1991): if in a mechanical or electrical system, the global energy is constantly lost, then the system, regardless of being linear or nonlinear, must eventually go to rest at a point of equilibrium. Consequently, a conclusion about the stability of the system can be given by investigating the variation registered in a single scalar function.

In the direct method of Lyapunov, the first property is formalized by the notion of positive definite functions, and the second is formalized by Lyapunov functions (Li, 2010).

#### 4.4.2.1.2 Positive definite functions theorem

Lyapunov's theorem for positive definite functions states that:

A scalar continuous function  $V(x)$  is considered to be locally definite if  $V(0) = 0$  and, in a ball  $B_{R_0}$   $x \neq 0 \Rightarrow V(x) > 0$ .

If  $V(0) = 0$  and the previous theorem holds over the whole state space, then the function  $V(x)$  is globally positive definite (Han, 2009).

#### 4.4.2.1.3 Lyapunov functions theorem

This theorem explains that:

If, in a ball  $B_{R_0}$ ,  $V(x)$  is a positive definite function and possesses constant partial derivatives, and if its derivative according to time along any trajectory of the state is negative semi-definite, i.e.,  $\dot{V}(x) \leq 0$  then: this means that  $V(x)$  is regarded as a Lyapunov function of the system (Bandyopadhyay, 2003).

#### 4.4.2.1.4 Lyapunov analysis of Linear Time Invariant systems (LTI)

The stability of a LTI system:  $\dot{x} = Ax$  according to Lyapunov theorem is satisfied if:

$$V(x) > 0, V(0) = 0 \tag{4.36}$$

$$\dot{V}(x) = \frac{\partial V}{\partial t} = \frac{\partial V}{\partial t} \frac{dx}{dt} \leq 0 \tag{4.37}$$

A LTI system is stable, if a scalar function  $V(x)$  associated with the system and satisfying both conditions (4.36), (4.37) exists. The system is asymptotically stable if  $\dot{V}$  is negative definite. A quadratic Lyapunov function can be expressed as:

$$V = x^T P x \quad (4.38)$$

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

$$\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x} = x^T (A^T P + P A) x = -x^T Q x$$

Where  $Q \in R^{n \times m}$  is a positive definite matrix ( $Q = Q^T > 0$ ).

The equation  $A^T P + P A = -Q$  is called the Lyapunov algebraic equation.

In summary, Lyapunov stability theorem for LTI system states that: an essential and sufficient condition for a LTI  $\dot{x} = Ax$  to be strictly stable is that, for any symmetric positive definite matrix  $Q$ , a unique matrix  $R$ , the solution of the Lyapunov algebraic equation is symmetric and positive definite (Marquez, 2003).

In this thesis, Lyapunov stability theory based on the direct method is applied to control the magnetic levitation. This part of the work is presented in Chapter 7.

#### 4.4.2.2 Krasovski's method

Gopal, 2005 defined Krasovski's method as a simple representation of Lyapunov function candidate for nonlinear autonomous systems of the form:  $V = f^T f$ . The fundamental concept of the method is mainly to check whether this specific choice leads to a Lyapunov function.

This theorem can be formulated as follows (Fridman, 2007):

The autonomous system defined by  $\dot{V} = f^T f$  is considered, with the specific point of equilibrium being the origin. Let  $A(x)$  represent the Jacobian matrix of the system  $D(x) = \frac{\partial f}{\partial x}$ .

- If the matrix  $F = D + D^T$  is negative definite in a neighborhood  $\Omega$ , hence the point of equilibrium at the origin is said to achieve asymptotical stability. A Lyapunov function of the system is expressed as:  $V(x) = f^T(x)f(x)$ .

- If  $\Omega$  is the complete state space and, in addition,  $V(x) \rightarrow \infty$  as  $\|x\| \rightarrow \infty$ , then the point of equilibrium is globally asymptotically stable.

## 4.5 Nonlinear control systems design

Slotine and Li, 1991 defined the objective of control system design as: given a physical system to be controlled and the requirements of its optimal behavior, build a control law based on the feedback method to make the closed-loop system achieve the desirable behavior. In control systems applications that involve high speed motion, nonlinear will be of considerable effects in the dynamic and nonlinear control may be crucial for the desired performance to be achieved.

Generally the tasks of control systems can be classified into two parts: stabilization and tracking. The problems of stabilization require the design of a control system called a regulator. The regulator is designed to achieve the stability of the closed loop system around a point of equilibrium. In the other hand, tracking problems have to objectives the construction of a control system called a tracker so that the system output tracks a given time-varying trajectory (Papea, 2006). Feedback linearization technique is in the next section discussed to achieve nonlinear control.

### 4.5.1 Feedback linearization

Feedback linearization is one of the most attractive nonlinear control design method which is used in most of the researches related to the field of nonlinear control (Chiasson, 1998) and (Krener, 1999). The concept of feedback linearization is to mathematically perform the transformation of the dynamics of a nonlinear system into a fully or partly linear one, so that it is possible to apply linear control methods (Calvet, 1988).

The idea of feedback linearization is to get rid of the nonlinearities of a system and impose a desired linear dynamics that can be directly applied to a category of nonlinear systems described in companion, or controllability canonical form. The expression of a system considered to be in companion form is: (Slotine, 1991):

$$\dot{x}^{(n)} = f(x) + \bar{g}(x)u \quad (4.39)$$

The state space representation of the system is represented:

$$\frac{\partial}{\partial t} \begin{bmatrix} x_1 \\ \dots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} x_2 \\ \dots \\ x_n \\ f(x) + \bar{g}(x)u \end{bmatrix} \quad (4.40)$$

$u$  = scalar control input

$x$  = scalar output of interest

$$x = \left[ x, \dot{x}, \dots, x^{(n-1)} \right]^T = \text{state vector}$$

$f(x)$  = nonlinear function of the states

For system that can be expressed in controllable canonical form, with the use of the control input (if  $b$  is assumed to be non-zero), we have:

$$u = \frac{1}{g}(v - f) \quad (4.41)$$

Where  $v$  is a control function for the linearised system.

The cancellation of nonlinearities to obtain input-output linearized relation shows that:

$$x^{(n)} = v \quad (4.42)$$

Therefore the control law is:

$$v = -k_1 x - k_2 \dot{x} - \dots - k_{n-1} x^{(n-1)} \quad (4.43)$$

The selection of the  $k_i$ 's terms is made so that  $p^n + k_{n-1}p^{n-1} + \dots + k_1$  is a stable polynomial, which leads to the exponentially stable dynamics:

$$x^{(n)} + k_{n-1}x^{(n-1)} + \dots + k_1 x \quad (4.44)$$

With:  $x \rightarrow 0$

For tasks that involve tracking of a desired output  $x_d$ , the control law is:

$$u = x_d^{(n)} - k_1 e - k_2 \dot{e} - \dots - k_{n-1} e^{(n-1)} \quad (4.45)$$

Where:  $e = x - x_d$

The feedback linearization and the canonical form are very useful to solve non-linear control problems because of their simplicity and flexibility. Many applications are described in (Arkun and Calvet, 1988), (Henson and Seborg, 1990), (Spong, 1994) and (Ravi, 2013).

#### 4.5.1.1 Input-state linearization

To introduce the concept of input state linearization of single-input nonlinear system, we can consider the following equation (Ravi, 2013):

$$\dot{x} = f(x) + g(x)u \quad (4.46)$$

With  $f$  and  $g$  being smooth vectors fields. The system is said to be in linear in control or affine. Conte, 2007 defined input-state linearization as a system in the form of Equation 4.46), with  $f(x)$  and  $g(x)$  being smooth vector fields on  $\mathfrak{R}^n$ , is considered to be input linearizable if there is an existing region  $\Omega$  in  $\mathfrak{R}^n$ , a diffeomorphism  $T : \Omega \rightarrow \mathfrak{R}^n$ , and a nonlinear feedback control law:

$$u = a(x) + \beta(x)v \quad (4.47)$$

In such a way that the new state variables  $z = T(x)$  and the new input  $v$  fulfil a linear time invariant relation:  $\dot{z} = Az + bv$  (4.48)

To fulfil the conditions for input-state linearization, Slotine and Li, 1991 stated that the nonlinear system described by Equation (4.46), with  $f(x)$  and  $g(x)$  being smooth vector fields is input-state linearizable if, and only if, there is the existence of a region  $\Omega$  with the following conditions:

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

To perform input-state linearization of a nonlinear system, the control system designer must go through the following steps:

- Build the vector fields  $\{g, ad_f g, \dots, ad_f^{n-1} g\}$  for the selected system.
- Investigate the fulfillment of the controllability and involutivity conditions.
- If both are fulfilled, then find the first state  $T_1$  (the function of the output that leads to a relative degree  $n$  of the input-output linearization) from the equations:

$$\nabla T_1 ad_f^i g = 0, \quad i = 1, \dots, (n-1)$$

$$\nabla T_1 ad_f^{n-1} g \neq 0$$

- Computation of the state transformation  $z = \phi(x) = (T_1 \quad L_f T_1 \quad \dots \quad L_f^{n-1} T_1)^T$  and the input transformation of Equation (4.47), with:

$$\alpha(x) = -\frac{L_f^n T_1}{L_g L_f^{n-1} T_1}$$

$$\beta(x) = \frac{1}{L_g L_f^{n-1} T_1}$$

### 4.5.1.2 Input-output linearization

The representation of a single-input single-output nonlinear system is expressed as (Lee et al., 2000):

$$\dot{x} = f(x) + g(x)u \quad (4.49)$$

$$y = \bar{h}(x) \quad (4.50)$$

In Equation (4.50),  $y$  is the output of the system taken in consideration.

To generate input-output linearization relation, Khalil and Sastry, 1991 demonstrated that from Equations (4.49) and (4.50), it can be seen that the output  $y$  is not directly 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 the continuous differentiation of the output function  $y$  until the input  $u$  appears. After that, design  $u$  to nullify the nonlinearity.

The design of a controller based on input-output linearization is based on three steps:

- Perform the differentiation of the output  $y$  until the input  $u$  is found.
- Select  $u$  to nullify the nonlinearities and achieve convergence of tracking.
- Analyze the internal dynamics stability.

Slotine and Li, 1991 explained that the internal dynamics stability can be studied through the zero-dynamics study.

In Chapter 6, feedback linearization is studied in a deeper manner and the feedback linearization is applied to the magnetic levitation system.

## 4.6 Conclusion

In this chapter, control theory has been briefly discussed. Linear and nonlinear control techniques are introduced. Both fields have been under development for thousands of years and have become more useful and powerful than ever because of their positive impact. Control system theory provides the theoretical basis which is an important tool for engineers and researchers to solve the control problems in different fields. Chapter 5 presents the design of linear controllers based on pole-placement and optimal quadratic regulation methods to provide stability to the linearized model of the magnetic levitation system to achieve its stability and optimal behaviour.

## CHAPTER FIVE: LINEAR CONTROL DESIGN FOR THE MAGNETIC BALL LEVITATION SYSTEM

### 5.1 Introduction

In this thesis, the magnetic levitation system's mathematical model is derived in Chapter 3. The aim of this project is to develop an algorithm for nonlinear control implementation in a programmable logic controller (PLC). For the purpose of best performance of the closed-loop system consisting of the nonlinear plant and the nonlinear controller, it is necessary to design a linear controller that will produce a control signal to bring the states of the nonlinear plant to their desired values. Nonlinear systems are very sensitive to parameters changes, and it is crucial to design a strong linear controller to ensure that the performance of the closed loop system is satisfactory. In this section, two linear control methods are selected to achieve the desired characteristics of the magnetic levitation system. The closed-loop system behaviour under controllers designed by application of the pole placement and the linear quadratic methods are compared, and the method with better results is utilized in the next chapters to achieve feedback linearization and Lyapunov stability of the magnetic levitation system.

The organization of this chapter is as follows: in section 5.1 an introduction of the chapter is given. Linear control methods are discussed in section 5.2. Pole placement and linear quadratic regulation are presented in section 5.3. The design of the linear quadratic regulator is done in section 5.4, and the section 5.5 is the conclusion.

### 5.2 Linear control design methods

The idea is to find the coefficient of the state feedback matrix  $H$  such that some requirements of the closed loop system are satisfied. This thesis focuses on two particular methods:

- Controller design by pole placement: It means that a matrix  $H$  is determined such that the closed loop system has desired place of its poles.
- Controller design by using quadratic criterion quality: In this method, the matrix  $H$  is found such that some integral estimation of the system is minimized. The closed loop system is referred as optimal because it gives the best value of the criterion.

In this chapter, both methods are used and the results of their application are compared to select the best one that can be used further in combination with the nonlinear controller design.

### 5.3 Pole placement method

The closed loop system is described by the equations of the plant and of the controller.

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \quad , \quad x(0) = x_0 \\ u &= -Hx + Fr \end{aligned} \tag{5.1}$$

Where  $A \in \mathfrak{R}^{n \times m}$  is the state matrix,  $B \in \mathfrak{R}^{n \times m}$  is the control matrix,  $C \in \mathfrak{R}^{1 \times n}$  is the output matrix.  $H \in \mathfrak{R}^{m \times n}$  is the feedback matrix,  $F \in \mathfrak{R}^{m \times l}$  is the set-point matrix, and  $x_0$  is the initial state.

In the first equation  $u$  is substituted by its expression. The closed loop system is described by

$$\begin{aligned} \dot{x} &= (A - BH)x + BFr = A_{cl}x + BFr \quad , \quad x(0) = x_0 \\ y &= Cx \end{aligned} \tag{5.2}$$

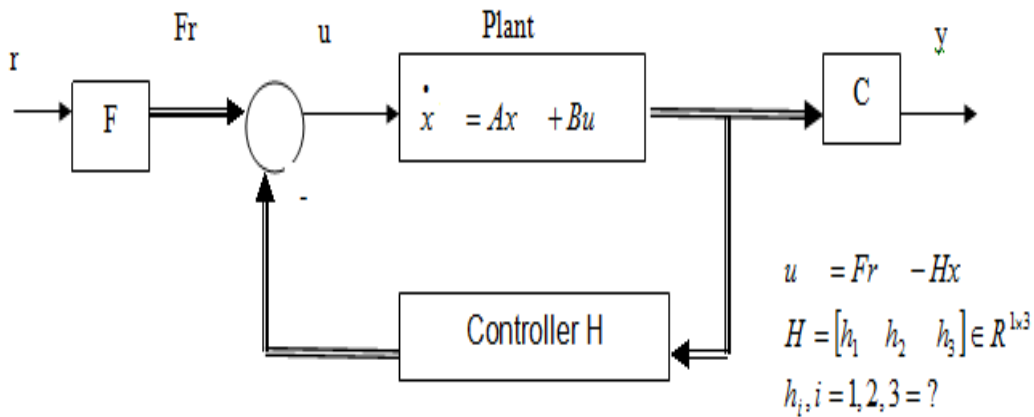
The matrix  $H$  changes the state matrix of the closed loop system. If the plant is completely controllable the matrix  $H$  gives the possibility to place the poles of the characteristic equation  $\Delta(s) = \det[sI - (A - BH)] = 0$  at desired places under the restriction that the complex poles have to be complex conjugates.

The controller's design procedure is based on this capability of the feedback and has the following steps:

- Choose the desired places of the closed loop poles.
- Determine the controller matrix.
- Build the transient response, as the system behavior depends also on zeros and the fulfillment of the desired behavior has to be checked.

Figure 5.1 shows the closed loop system representation with the state space controller.





**Figure 5.1: Closed loop system representation**

### 5.3.1 Selection of the desired place of the poles of the closed loop system

There are different ways to determine the desired performance of the closed loop system such as (Nise, 2003):

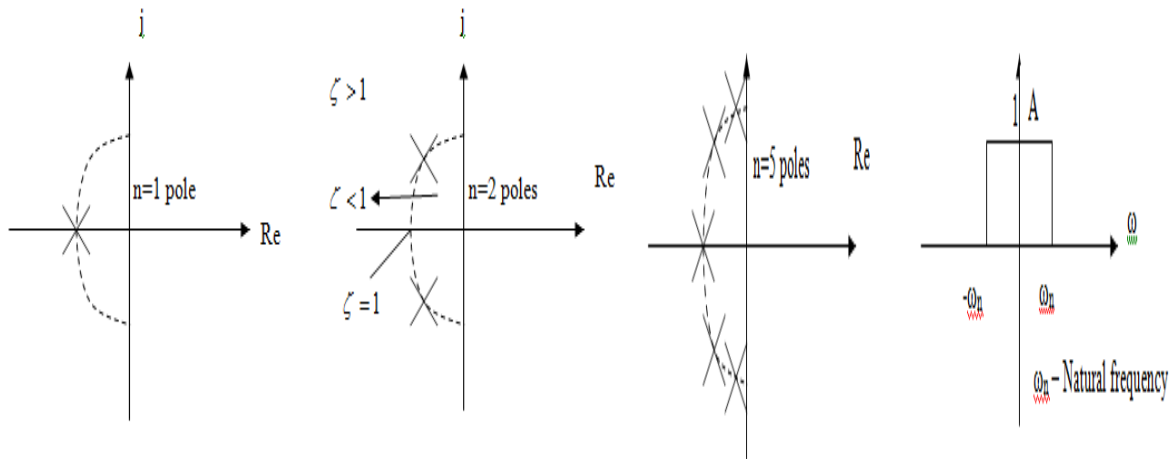
- Desired transfer functions
- Performance specifications
- Location of poles in the complex plane.

#### 5.3.1.1 Desired transfer function

In this method, some standard pole placement or standard transfer function of the closed loop are proposed in the literature. Then the closed loop system is designed such that it equates to some standard one. The standard desired transfer functions can be of the following forms: Butterworth filter, transfer function of closed loop system and closed loop desired function with no overshoot.

##### 5.3.1.1.1 Butterworth filter

For the Butterworth filter, the poles are put on the semi-circle in the left complex plane as shown in the following figures:



**Figure 5.2: Poles of a Butterworth filter**

### 5.3.1.1.2 Transfer functions of a closed loop system

This method is applied to transfer functions of the following form:

$$W(s) = \frac{a_0}{s^n + a_1 s^{n-1} + \dots + a_0} \quad (5.3)$$

The Equation (5.3) is a desired transfer function that will lead to the design of a closed loop system with small overshoot, big damping ratio and zero steady state error. Depending on the order of the system, for various powers of  $n$  the characteristic polynomial in the denominator of the desired transfer function are:

$$n = 1 \quad s + \omega_n$$

$$n = 2 \quad s^2 + 1.4\omega_n s + \omega_n^2$$

$$n = 3 \quad s^3 + 1.75\omega_n s^2 + \omega_n^3$$

$$n = 4 \quad s^4 + 2.1\omega_n s^3 + 3.4\omega_n^2 s^2 + 2.7\omega_n^3 + \omega_n^4$$

### 5.3.1.1.3 Closed loop desired transfer function with no overshoot

If a system has  $n$  negative real poles:  $p_i = -a$ ,  $i = \overline{1, n}$  it is necessary to select the characteristic polynomial (denominator) in such a way that:  $D(s) = (s + a)^n$ .

The time of the transient response is at:  $t_s \approx \frac{n+2}{a}$ .

### 5.3.1.2 Feedback matrix coefficients calculation

To determine the feedback matrix coefficients, it is supposed that a single input-single output plant is fully controllable. Then it can be described as the following equation:

$$\dot{x} = Ax + Bu \quad (5.4)$$

The matrices  $A$  and  $B$  are in companion form:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix}$$

and where  $a_i, i = \overline{0, n-1}$  are the coefficients of the characteristic polynomial

$$\Delta_{plant}(s) = \det(sI - A) = s^n + \alpha_{n-1}s^{n-1} + \alpha_{n-2}s^{n-2} + \dots + a_1s + a_0 \quad (5.5)$$

The equation of the regulator is:

$$u = -Hx + Fr, \text{ where } H = [h_1 \ h_2 \ \dots \ h_n] \quad H \in R^{1 \times n} \quad (5.6)$$

The closed loop system is

$$\dot{x} = (A - BH)x + BFr \quad (5.7)$$

The state matrix  $(A - BH)$  of the closed loop system is:

$$(A - BH) = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \\ -a_0 - h_1 & -a_1 - h_2 & -a_2 - h_3 & \dots & -a_{n-1} - h_n \end{bmatrix} \quad \text{as } BH = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & & 0 \\ h_1 & h_2 & & h_n \end{bmatrix} \quad (5.8)$$

This matrix is also in a companion form and its characteristic polynomial is,

$$\Delta_{closedloop}(s) = s^n + (a_{n-1} + h_n)s^{n-1} + \dots + (a_1 + h_2)s + (a_0 + h_1) \quad (5.9)$$

There is a full freedom in selecting the coefficients  $h_i, i = \overline{1, n}$  in the matrix  $H$  and it is possible to realize arbitrary characteristic polynomial of the desired closed loop system

$$\Delta_{desired}(s) = s^n + d_{n-1}s^{n-1} + \dots + d_1s + d_0; \quad (5.10)$$

and to place the poles to arbitrary desired places in s-plane. The limitation is the poles to be couple by couple complex conjugate. Then the coefficients of the matrix  $H$  can be calculated from comparison of the desired with the characteristic polynomial of the closed loop system with unknown coefficients of the controller

$$s^n + (a_{n-1} + h_n)s^{n-1} + \dots + (a_1 + h_2)s + (a_0 + h_1) = s^n + d_{n-1}s^{n-1} + \dots + d_1s + d_0 \quad (5.11)$$

$$\Delta_{closedloop}(s) = \Delta_{desired}(s),$$

Calculation is done on the basis of comparison of the coefficients in front of the equal powers of the  $s$  variable:

$$\begin{aligned} a_0 + n_1 &= d_0 & h_1 &= d_0 - a_0 & \rightarrow s^0 \\ a_1 + n_2 &= d_1 & h_2 &= d_1 - a_1 & \rightarrow s^1 \\ \dots & & & & \\ a_{n-1} + h_n &= d_{n-1} & h_n &= d_{n-1} - a_{n-1} & \rightarrow s^{n-1} \end{aligned} \quad (5.12)$$

### 5.3.2 State feedback controller

In chapter three, the state space representation of the magnetic levitation system is:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2k}{m} \left( \frac{x_3^2}{x_1^3} \right) & 0 & -\frac{2k}{m} \left( \frac{x_3}{x_1^2} \right) \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} v \quad (5.13)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

In a benchmark model of magnetic levitation system, the following values are given as:

$$m = 8.27 \times 10^{-3} \text{ Kg}$$

$$\bar{g} = 9.81 \text{ N} \times \text{Kg}^{-1}$$

$$R = 1[\Omega]$$

$$k = 0.0001$$

$$L = 0.01[H]$$

$$i_e = 0.84[A]$$

$$x_{e1} = 0.012[m]$$

The state space representation of the system becomes:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 9875 & 0 & -141.1 \\ 0 & 0 & -100 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix} v, x(0) = x_0 \quad (5.14)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

The transfer function is found on MATLAB using the following function:  $[b, a] = ss2tf(A, B, C, D)$ .

The explicit representation of the transfer function is:

$$G(s) = \frac{-1.411 \times 10^4}{(s - 99.37)(s + 99.37)(s + 100)} \quad (5.15)$$

The magnetic levitation system is of third order with poles located at the following positions:

$$P_1 = -100$$

$$P_2 = -99.37$$

$$P_3 = 99.37$$

The system presents a pole on the right side of the complex plane which means that it is open loop unstable.

To design a feedback controller based on pole placement technique means that we are looking for a control scheme as given in Equation (5.1):

### 5.3.2.1 Controllability of the system

Let's check if the system is completely state controllable, according to theory: The plant controllability matrix is  $\bar{C} = [B \quad AB \quad A^2B]$  (5.16)

The calculations of the matrices  $AB$  and  $A^2B$  are:

$$AB = \begin{bmatrix} 0 & 1 & 0 \\ 9875 & 0 & -141.1 \\ 0 & 0 & -100 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix} = \begin{bmatrix} 0 \\ -14110 \\ -10000 \end{bmatrix}$$

$$A^2B = \begin{bmatrix} 9875 & 0 & -141.1 \\ 0 & 9875 & 14110 \\ 0 & 0 & 10^4 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix} = \begin{bmatrix} -14110 \\ 14.11 \times 10^5 \\ 10^6 \end{bmatrix}$$

Then the controllability matrix is:

$$\bar{C} = \begin{bmatrix} 0 & 0 & -14110 \\ 0 & -14110 & 14.11 \times 10^5 \\ 100 & -10000 & 10^6 \end{bmatrix}$$

The check of controllability of the system is done on MATLAB with the function  $ctrd(A, B)$ . The rank of the matrix  $M$  is 3 therefore it is controllable.

### 5.3.2.2 Poles selection

Before the design of the controller based on the pole placement method, it is necessary to decide where the desired closed-loop poles have to be selected. This is based on the specifications for the behaviour of the closed-loop system. They are selected to be:

- Settling time < 0.67 seconds
- Overshoot < 1.5%.

From the specifications, the values of the poles can be obtained. The idea is to determine two dominant poles at desired positions.

For a percentage of overshoot of 1.5%, the damping ratio of the system is:

$$\varepsilon = \frac{\ln\left(\frac{\%OS}{100}\right)}{\sqrt{\pi^2 \ln^2\left(\frac{\%OS}{100}\right)}} = \frac{\ln\left(\frac{1.5}{100}\right)}{\sqrt{\pi^2 \ln^2\left(\frac{1.5}{100}\right)}} = 0.8 \quad (5.17)$$

The phase angle of the pole is:

$$\theta = \cos^{-1} \varepsilon = \cos^{-1} 0.8 = 36.8^\circ$$

The intersection point between the settling time and the phase angle gives the desired regions for the dominant poles of the second order approximation.

The real and imaginary values of the poles are:

$$\operatorname{Re} al(s) = -\varepsilon\omega_n = -\frac{4}{T_s} = -\frac{4}{0.67} = -5.97$$

$$\operatorname{Im} ag(s) = \operatorname{Re} al(s) \tan(\theta) = -5.97 \tan 36.8^\circ = 4.47 \quad (5.18)$$

$$P_1 = \operatorname{Re} al(s) + \operatorname{Im} ag(s) = -5.97 + 4.47i$$

$P_2$  is selected as the complex conjugate of  $P_1$ :

$$P_2 = -5.97 - 4.47i$$

The obtained polynomial based on the dominant poles is:

$$P(s) = (s + P_1)(s + P_2) = (s + 5.97 - 4.47i)(s + 5.97 + 4.47i) = s^2 + 11.94s + 55.62 \quad (5.19)$$

Then, the natural frequency  $\omega_n$  can be calculated:

$$\omega_n = \sqrt{55.62} = 7.46 \text{ rad/s}$$

The third pole is selected in such a way that the two dominant poles at  $P_1$  and  $P_2$  can reduce its effect so that the system is controlled with more effectiveness. After investigations, its optimal value is found to be:  $P_3 = 50$ .

### 5.3.2.3 Calculation of the controller parameters

In this scenario, the damping ratio and the speed of the system requirements will be satisfied.

The closed loop characteristic equation is obtained as follows:

$$\dot{x} = (A - BH)x + BF_r$$

$$H = [h_1 \quad h_2 \quad h_3]$$

$$\Delta_{closed}(s) = |sI - A + BH|$$

$$\Delta_{closed}(s) = \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ 9875 & 0 & -141.1 \\ 0 & 0 & -100 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix} [h_1 \quad h_2 \quad h_3]$$

$$\Delta_{closed}(s) = \begin{bmatrix} s & -1 & 0 \\ -9875 & s & 141.1 \\ 0 & 0 & (s+100) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 100h_1 & 100h_2 & 100h_3 \end{bmatrix} \quad (5.20)$$

$$\Delta_{closed}(s) = \begin{bmatrix} s & -1 & 0 \\ -9875 & s & 141.1 \\ 100h_1 & 100h_2 & (s+100+100h_3) \end{bmatrix}$$

$$\Delta_{closed}(s) = s^3 + (100 + 100h_3)s^2 - (9875 + 141.1 \times 10^2 h_2)s - (987500 + 987500 h_3 + 141.1 \times 10^2 h_1)$$

$$\Delta_{Desired}(s) = (s + 5.97 - 4.47i)(s + 5.97 + 4.47i)(s + 50) = s^3 + 61.94s^2 + 652.62s + 2781$$

$h_1$ ,  $h_2$  and  $h_3$  can be found by identification:

$$61.94 = 100 + 100h_3 \Rightarrow h_3 = -0.381$$

$$652.62 = -(9875 + 141.1 \times 10^2 h_2) \Rightarrow h_2 = 0.75$$

$$2781 = -(987500 + 987500 h_3 + 141.1 \times 10^2 h_1) \Rightarrow h_1 = 43.52$$

The full state feedback control gain based on pole placement is:

$$H = [43.52 \quad 0.75 \quad -0.381], \text{ the coefficient } F \text{ is accepted to be zero.}$$

### 5.3.2.4 Simulation of the closed loop system

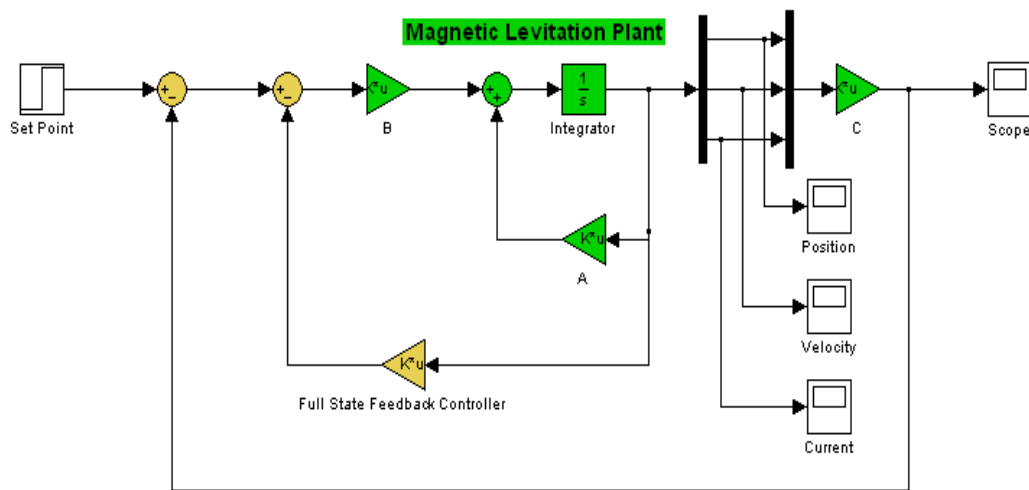
The simulation of the closed-loop system is done in Simulink environment. Figure 5.3 shows the block diagram “MagLev\_P.mdl”. The Simulink diagram is associated with the m-file

“MagLev\_PF.m”. The m-file is given in Appendix A.2. The algorithm of the controller is shown in

Figure 5.4. Simulations of the closed-loop systems are done for two set points 0.15[m], 0.2[m]

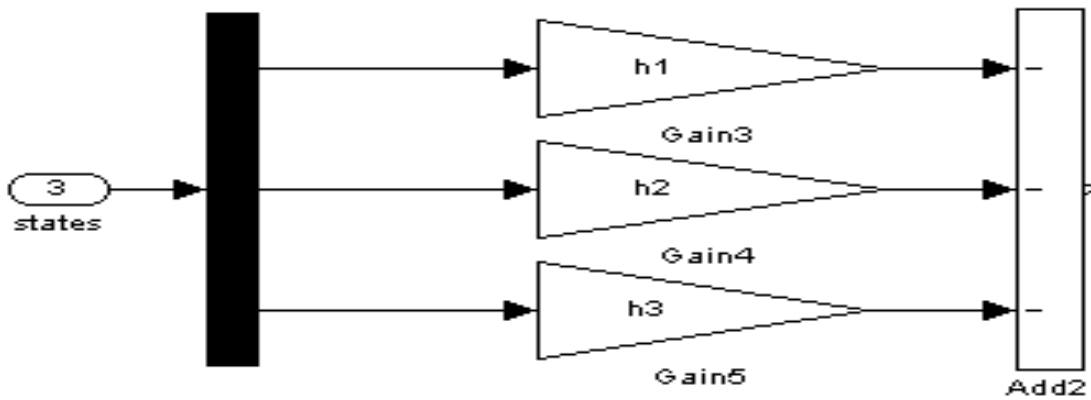
and for two cases of initial conditions  $[0.09 \quad 0 \quad 2.56]^T$  and  $[0.035 \quad 0 \quad 0.89]^T$ .

Figures 5.5 to 5.8 present the closed-loop transition behaviour of the position, velocity, current and control signal for the first initial conditions and Figures 5.9 to 5.12 respectively for the second ones at the first set point. Figures 5.13 to 5.20 show the same responses for the second set-point.



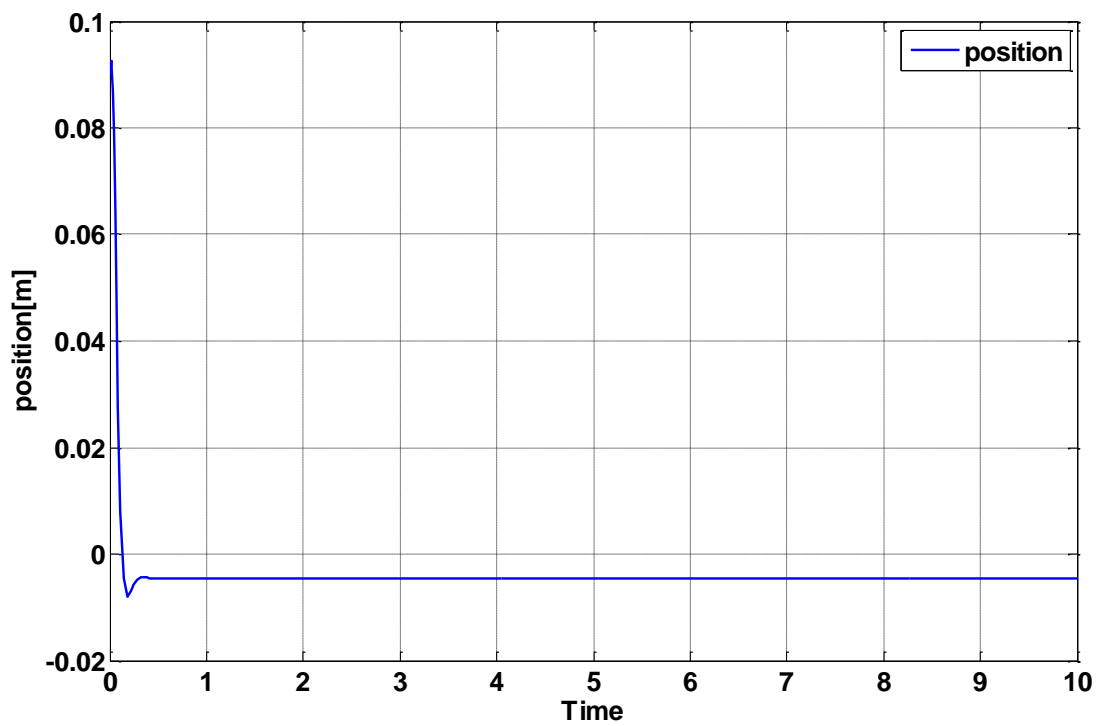
**Figure 5.3:** Simulink diagram of the closed-loop magnetic levitation system





**Figure 5.4: Linear controller algorithm**

- Responses of the system when the set-point is 0.15[m] and the initial conditions [0.09 0 2.56]':



**Figure 5.5: Position response with initials conditions of [0.09 0 2.56]'**

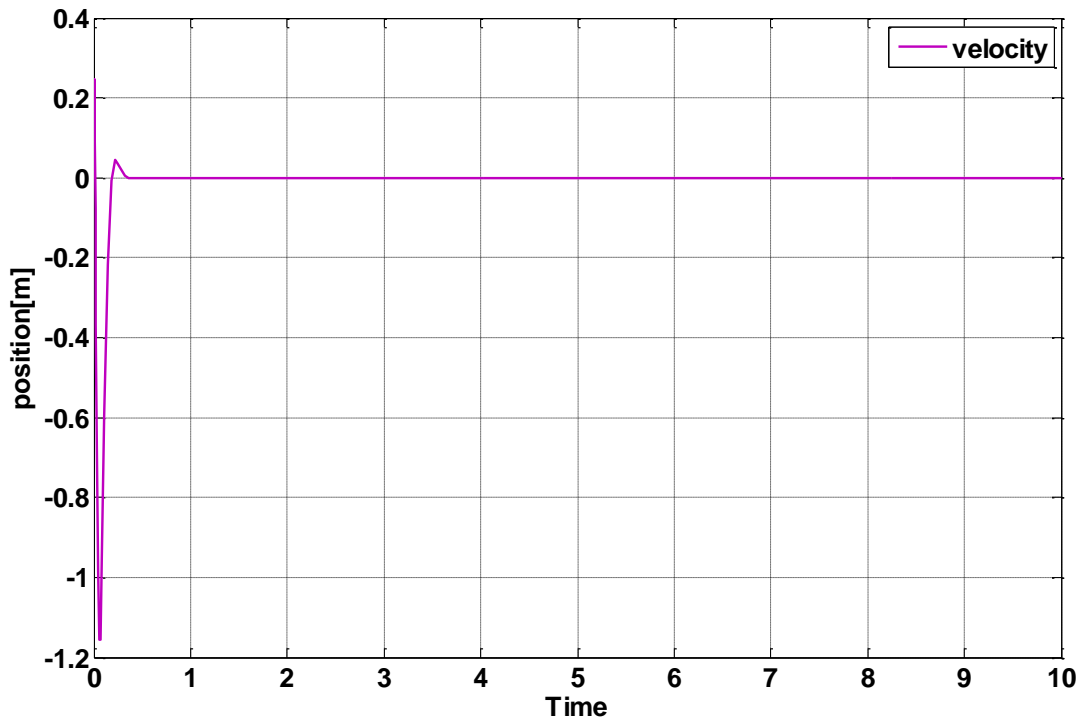


Figure 5.6: Velocity response with initials conditions of  $[0.09 \ 0 \ 2.56]'$

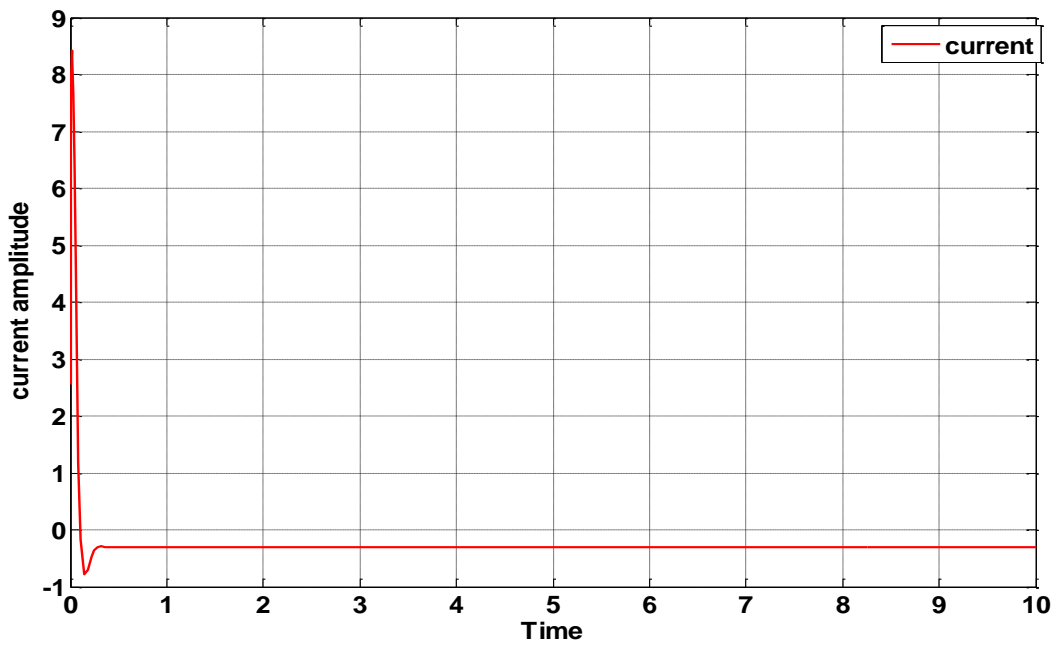


Figure 5.7: Current response with initials conditions of  $[0.09 \ 0 \ 2.56]'$

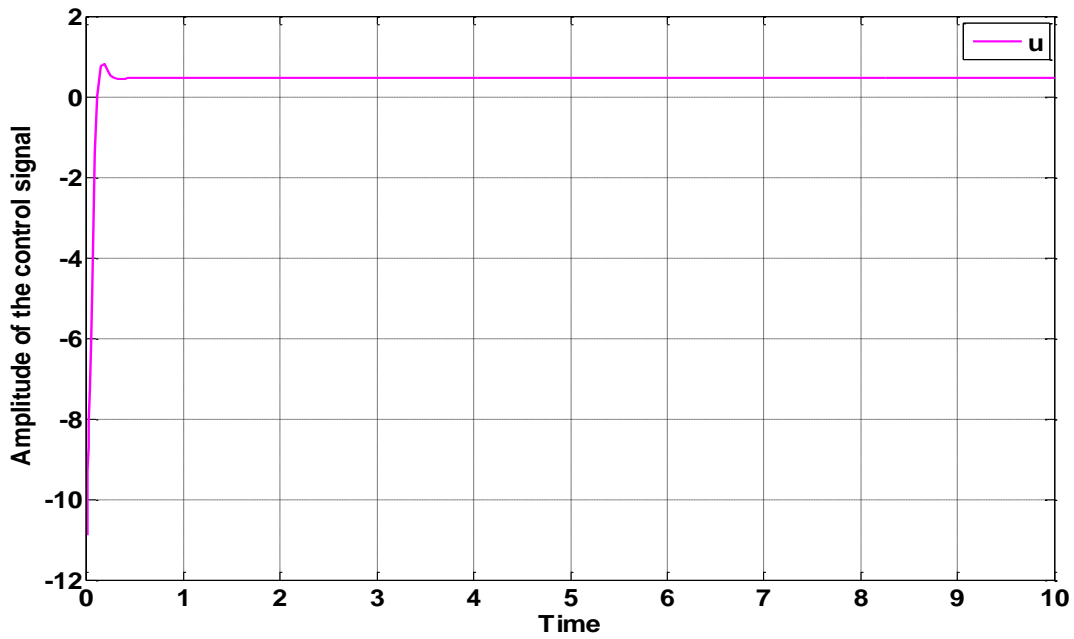


Figure 5.8: Control signal response with initial conditions of  $[0.09 \ 0 \ 2.56]'$

- Responses of the system when the set-point is 0.15[m] and the initial conditions  $[0.035 \ 0 \ 0.89]'$ :

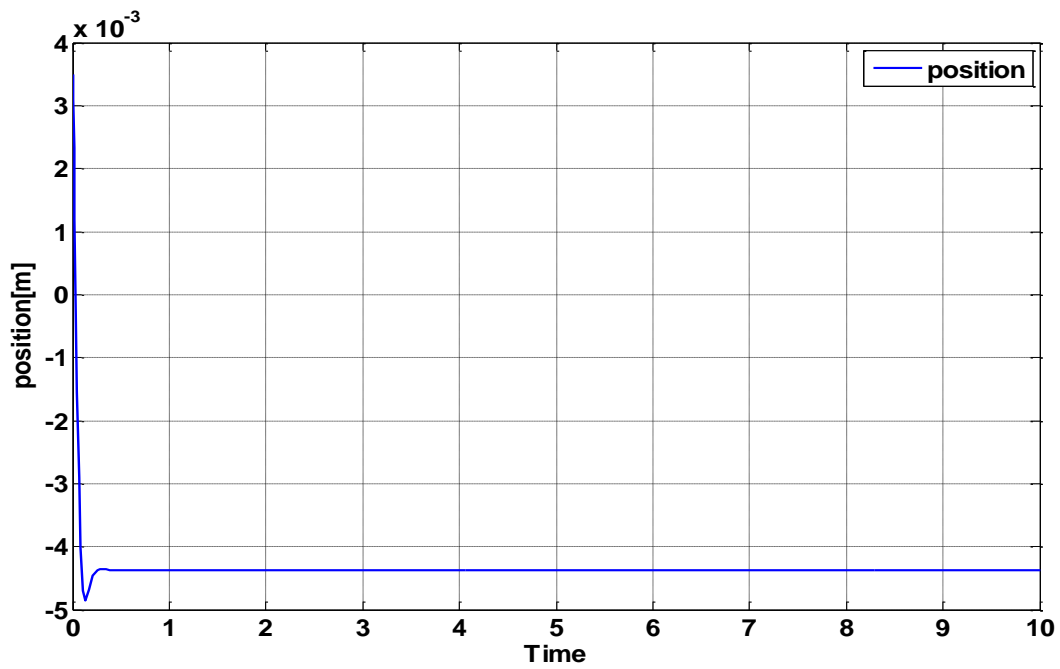


Figure 5.9: Position response with initials conditions of  $[0.035 \ 0 \ 0.89]'$

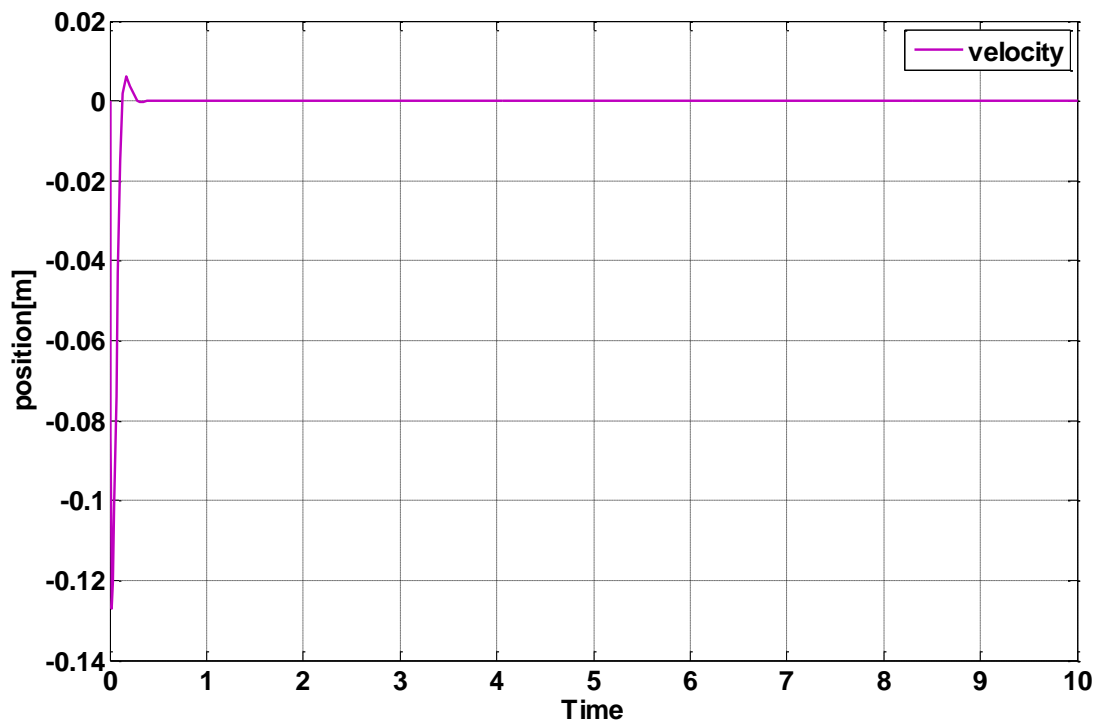


Figure 5.10: Velocity response with initials conditions of [0.035 0 0.89]'

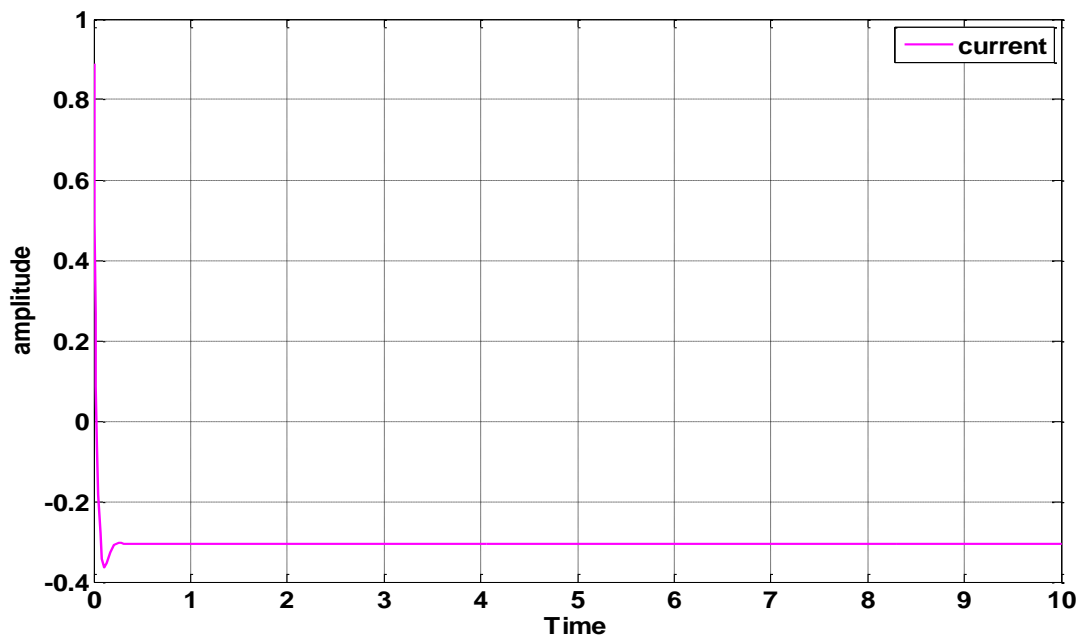


Figure 5.11: Current response with initials conditions of [0.035 0 0.89]'

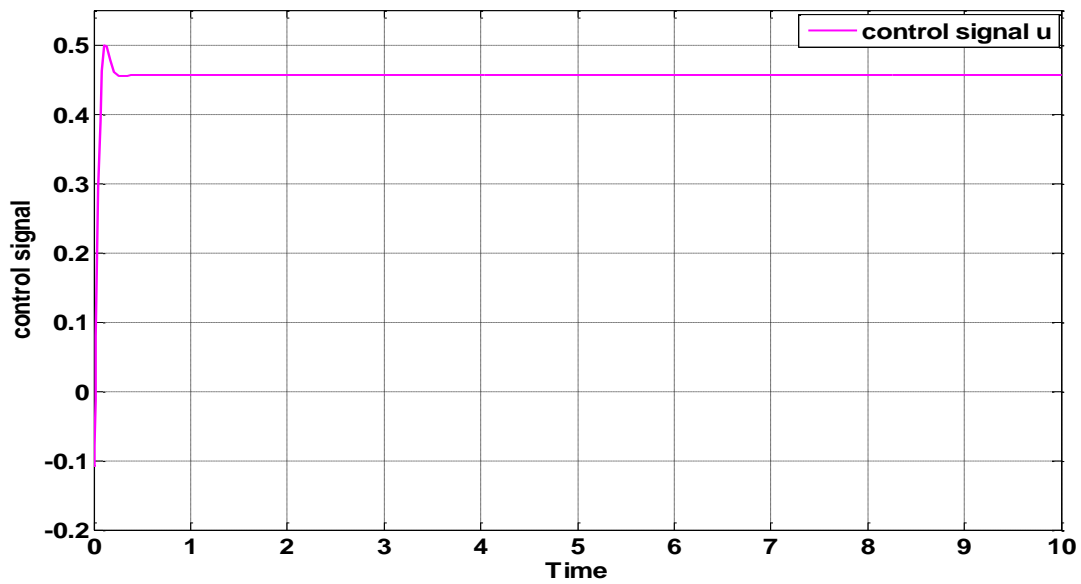


Figure 5.12: Control signal response with initial conditions of  $[0.035 \ 0 \ 0.89]'$

- Responses of the system when the set-point is 0.2[m] and the initial conditions  $[0.09 \ 0 \ 2.56]'$ :

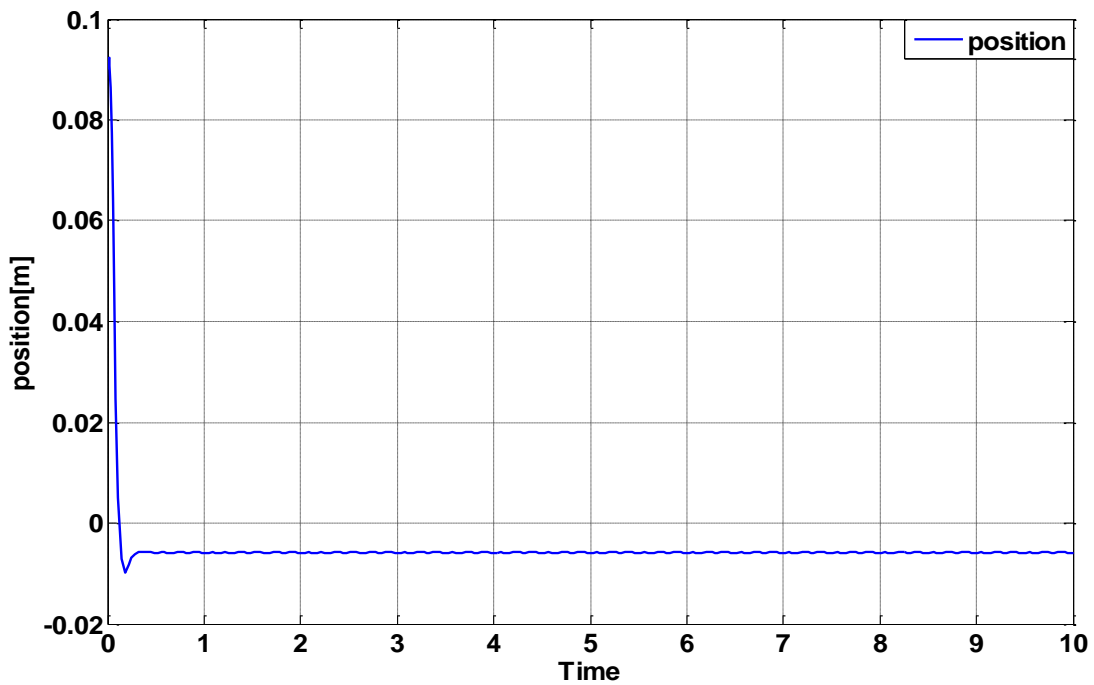


Figure 5.13: Position response with initials conditions of  $[0.09 \ 0 \ 2.56]'$

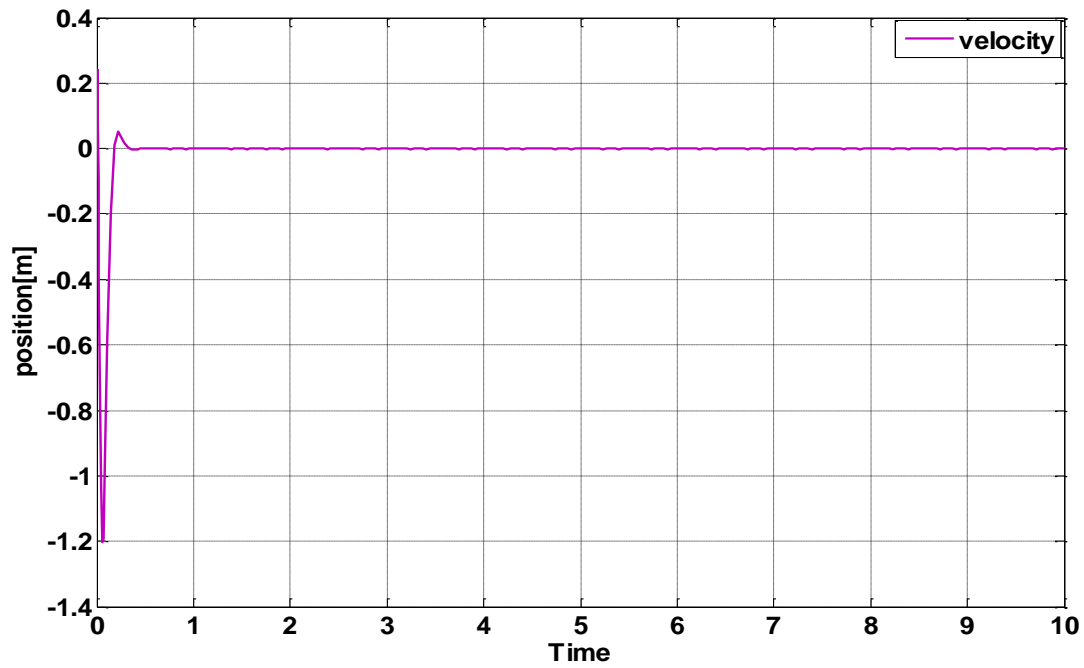


Figure 5.14: Velocity response with initials conditions of [0.09 0 2.56]'

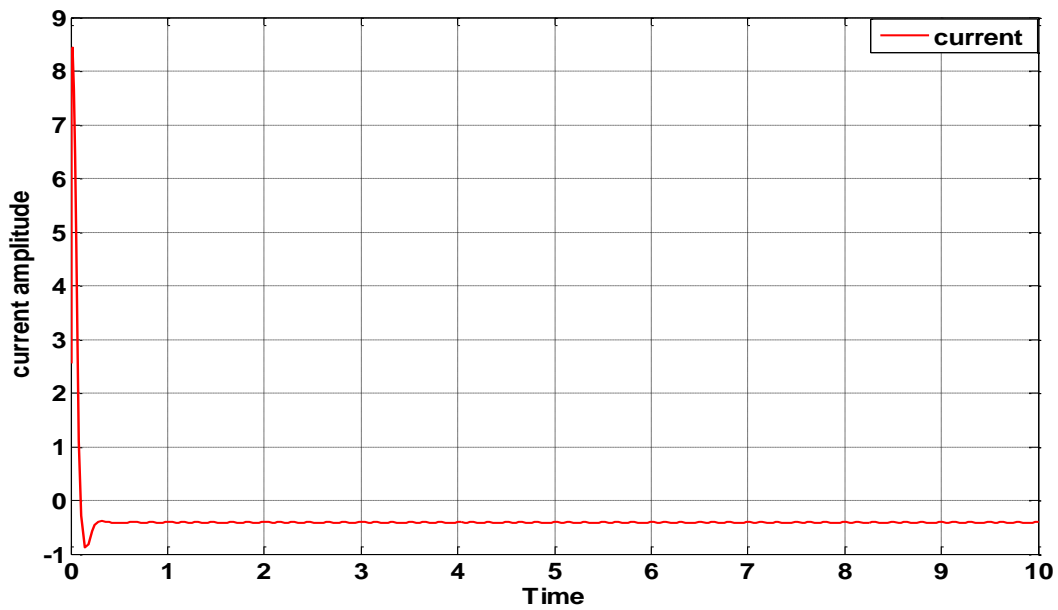


Figure 5.15: Current response with initials conditions of [0.09 0 2.56]'

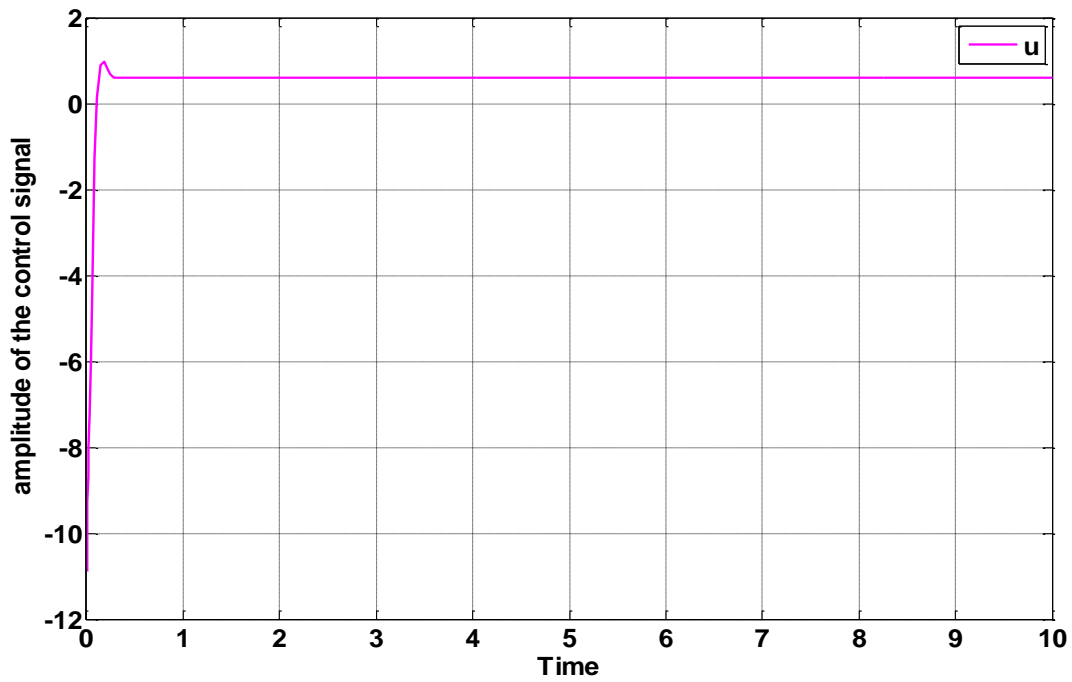


Figure 5.16: Control signal response with initial conditions of  $[0.09 \ 0 \ 2.56]'$

- Responses of the system when the set-point is  $0.2[m]$  and the initial conditions  $[0.035 \ 0 \ 0.89]$ :

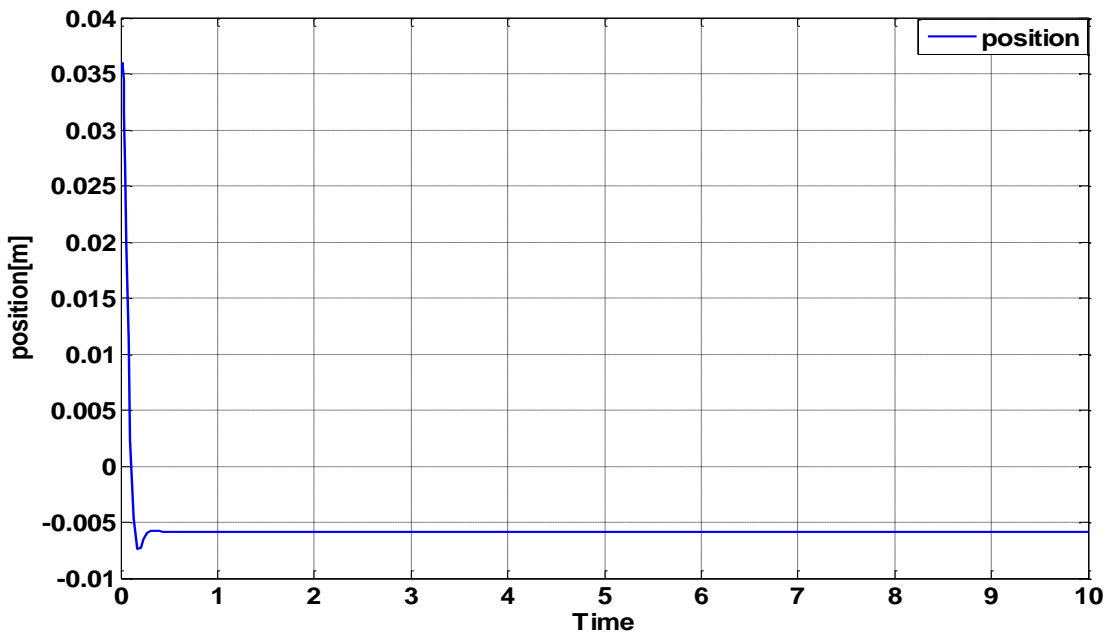


Figure 5.17: Position response with initial conditions of  $[0.035 \ 0 \ 0.89]'$

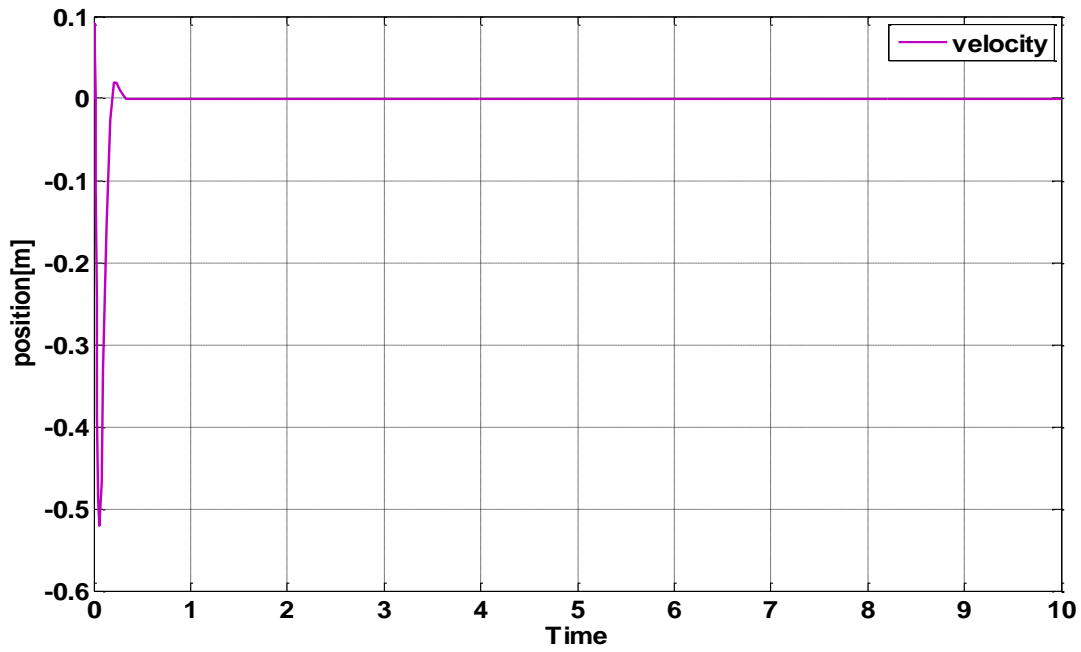


Figure 5.18: Velocity response with initials conditions of [0.035 0 0.89]'

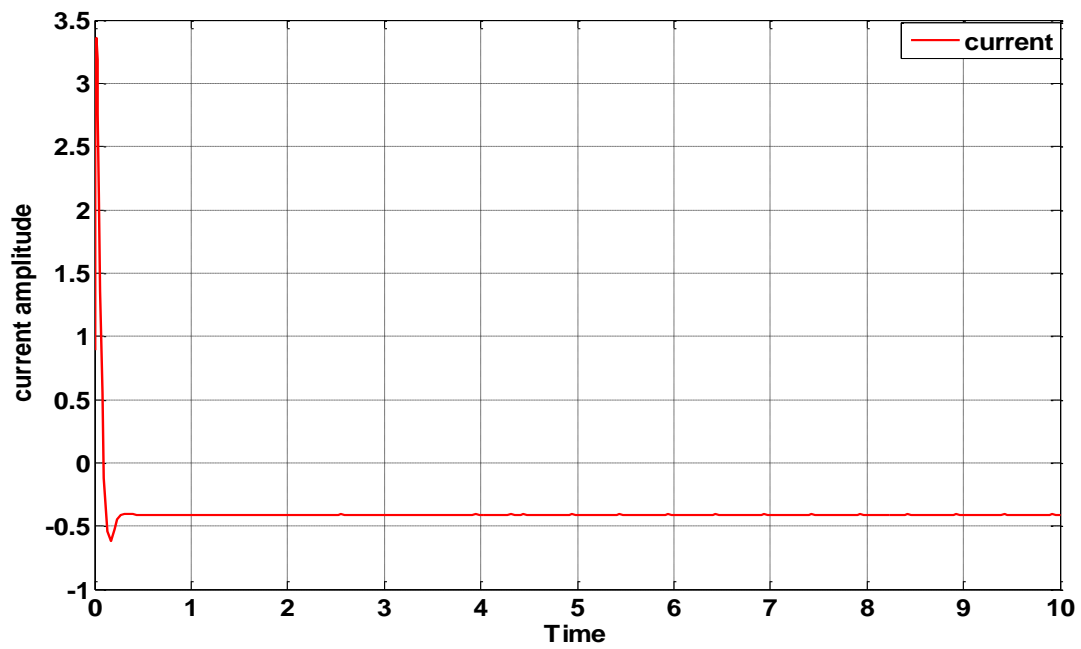
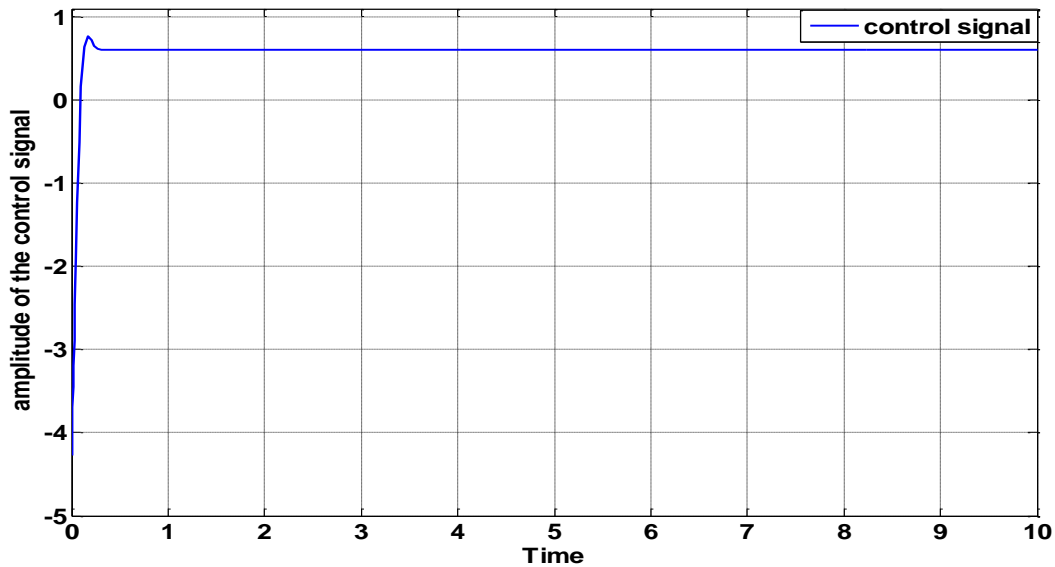


Figure 5.19: Current response with initials conditions of [0.035 0 0.89]'





**Figure 5.20:** Control signal response with initial conditions of  $[0.035 \ 0 \ 0.89]'$

The characteristics of the dynamic output behaviour according to the obtained results are given in Table 5.1.

**Table 5.1:** Characteristics of the position signal dynamic behaviour for the case of state feedback control

Set point	Characteristics	Magnetic levitation initial conditions $[0.09 \ 0 \ 2.56]'$	Magnetic levitation initial conditions $[0.035 \ 0 \ 0.89]'$
0.15	Time Delay	0	0
	Overshoot	200%	180%
	Rising Time	0.8s	0.6s
	Steady State Error	0.9	0.85
	Settling Time	3.7s	3.6s
0.2	Time Delay	0	0
	Overshoot	200%	180%
	Rising Time	0.8s	0.65s
	Steady State Error	0.7	0.76
	Settling Time	3.7s	3.6s

The system response shows that our requirements are not met because of the inability of the system to track the reference input. Therefore, it is necessary to add an integral action in order to reduce the steady state response of the closed loop system.

### 5.3.3 State feedback with integral control

The state feedback structure used in the previous section has one inconvenience in that it does not improve the transition behaviour of the system. This inconvenience means that a state feedback controller with constant gain feedback is mainly used for regulator systems for which the system does not achieve tracking of the reference input signal. It is always the case especially if all roots of the characteristic equation are to be placed at the desired position.

Generally, the purpose of a controller in a system is to track a particular reference input. To solve this problem, most of the control theory recommends the addition of an integral control action, just a classic integral (I) controller, combined with a constant feedback gain. In Figure 5.21, the Simulink block diagram “*Linear\_MagLevPP.mdl*” of the magnetic levitation system is shown. The Simulink block diagram is associated with the m-file “*MagLev\_PPI.m*”, and the m-file is given in Appendix A.3.

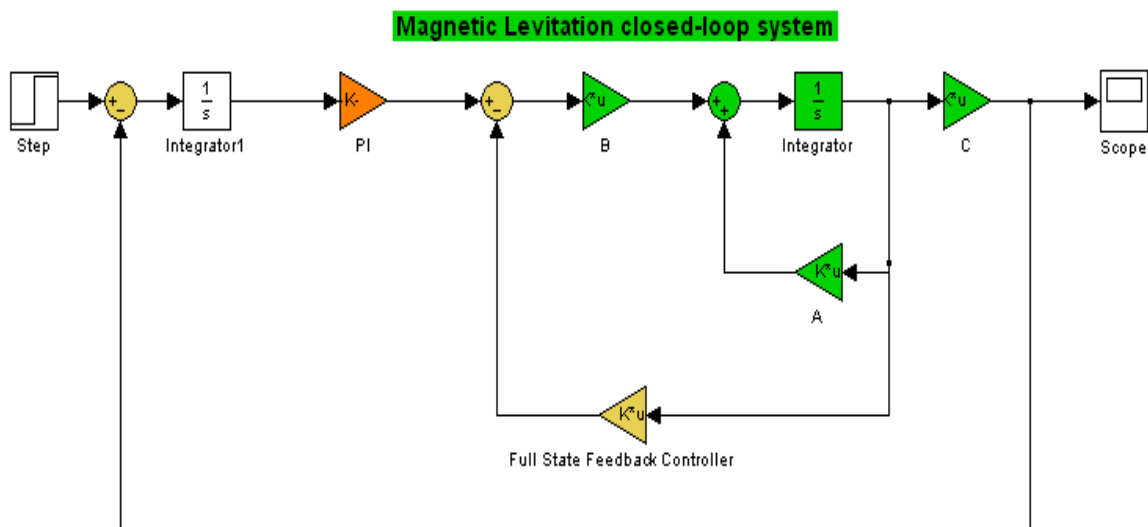


Figure 5.21: Full state feedback with integral control

#### 5.3.3.1 Integration of the integral control into the closed-loop system

In theory, the dynamic equations of the system are written as:

$$\dot{x} = Ax + Bu + En \quad (5.21)$$

$$\dot{x}_{n+1} = r - y = \dot{\xi} \quad (5.22)$$

$$y = Cx + Du \quad (5.23)$$

Where:  $x$  is the  $n \times 1$  state vector,  $u$  and  $y$  are scalar control signal and output, respectively;  $r$  is the scalar reference input, and  $n$  is the scalar disturbance input.

The matrices  $A, B, C, D$  and  $E$  must be of appropriate dimensions. The control signal  $u$  is related to state variables through constant state and integral feedback:

$$u = -Hx + h_{n+1}x_{n+1}(t) \quad (5.24)$$

Where:

$$H = [h_1 \quad h_2 \quad h_3 \quad \dots \quad h_n] \quad (5.25)$$

With constant real gain elements, and  $h_{n+1}$  is the scalar integral-feedback gain.

If Equation (5.24) is substituted in Equation (5.21), and combined with Equation (5.22), the  $n + 1$  state equation of the overall system with constant gain and integral feedback is written:

$$\overline{\dot{x}} = (\overline{A} - \overline{B}\overline{H})\overline{x} + \begin{bmatrix} 0_{n \times 1} \\ 1 \end{bmatrix} r + \overline{E}n \quad (5.26)$$

Where:

$$\overline{\dot{x}} = \begin{bmatrix} \dot{x} \\ x_{n+1} \end{bmatrix} \in \mathfrak{R}^{(n+1) \times 1}, \overline{x}(0) = \overline{x}_0 \quad (5.27)$$

$$\overline{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \in \mathfrak{R}^{(n+1) \times (n+1)}, \quad \overline{B} = \begin{bmatrix} B \\ 0 \end{bmatrix} \in \mathfrak{R}^{(n+1) \times 1} \quad (5.28)$$

$$\overline{H} = [H \quad h_{n+1}] = [h_1 \quad h_2 \quad \dots \quad h_n \quad h_{n+1}] \in \mathfrak{R}^{1 \times (n+1)} \quad (5.29)$$

$$\overline{E} = \begin{bmatrix} E \\ 0 \end{bmatrix} \in \mathfrak{R}^{(n+1) \times 1} \quad (5.30)$$

Substituting equation (5.24) into (5.23), the equation of the output of the overall system becomes:

$$y = \overline{C}\overline{x} \quad (5.31)$$

Where:

$$\overline{C} = [(C - DH) \quad DH] \in \mathfrak{R}^{1 \times (n+1)} \quad (5.32)$$

### 5.3.3.2 Calculation of the control parameters

The design objectives of this control strategy are the following:

- The steady state value of the output  $y$  follows a reference input with zero error; that means:  $e_{ss} = \lim_{t \rightarrow \infty} e(t) = 0$
- The  $(n+1)$  eigenvalues of  $(\bar{A} - \bar{B}\bar{H})$  are placed at desired locations. For the last condition to be possible, the pair  $(\bar{A}, \bar{B})$  must be completely controllable.

According to the presented theory, the state feedback with integral control of the levitation system is done as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 9875 & 0 & -141.1 & 0 \\ 0 & 0 & -100 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \varepsilon \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 100 \\ 0 \end{bmatrix} v$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \varepsilon \end{bmatrix}$$

The design of the state feedback with integral control is done according to the poles placed at the following positions:

$$P_1 = -5.97 + 4.47i$$

$$P_2 = -5.97 - 4.47i$$

$$P_3 = -50$$

$$P_4 = -10$$

### 5.3.3.3 Simulations of the closed-loop system

The initial conditions of the simulations are:  $[0.09 \ 0 \ 2.56 \ 0]'$  and  $[0.035 \ 0 \ 0.89 \ 0]'$  with set point at 0.15[m] and 0.2[m].

- Responses of the system when the set point is at 0.15[m]

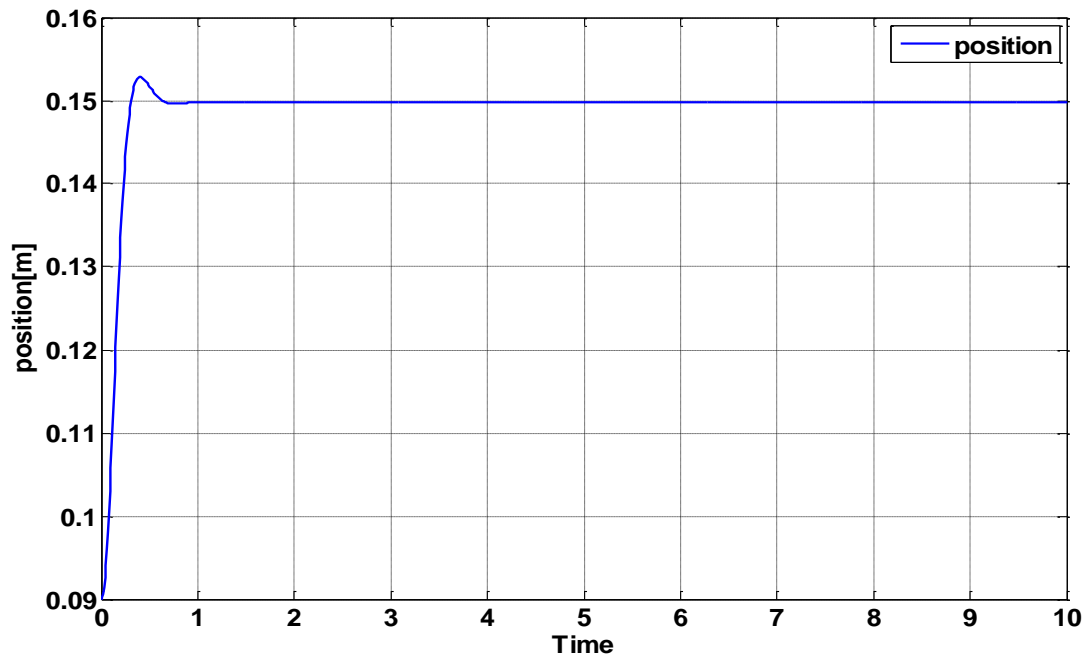


Figure 5.22: Position response with integral control with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

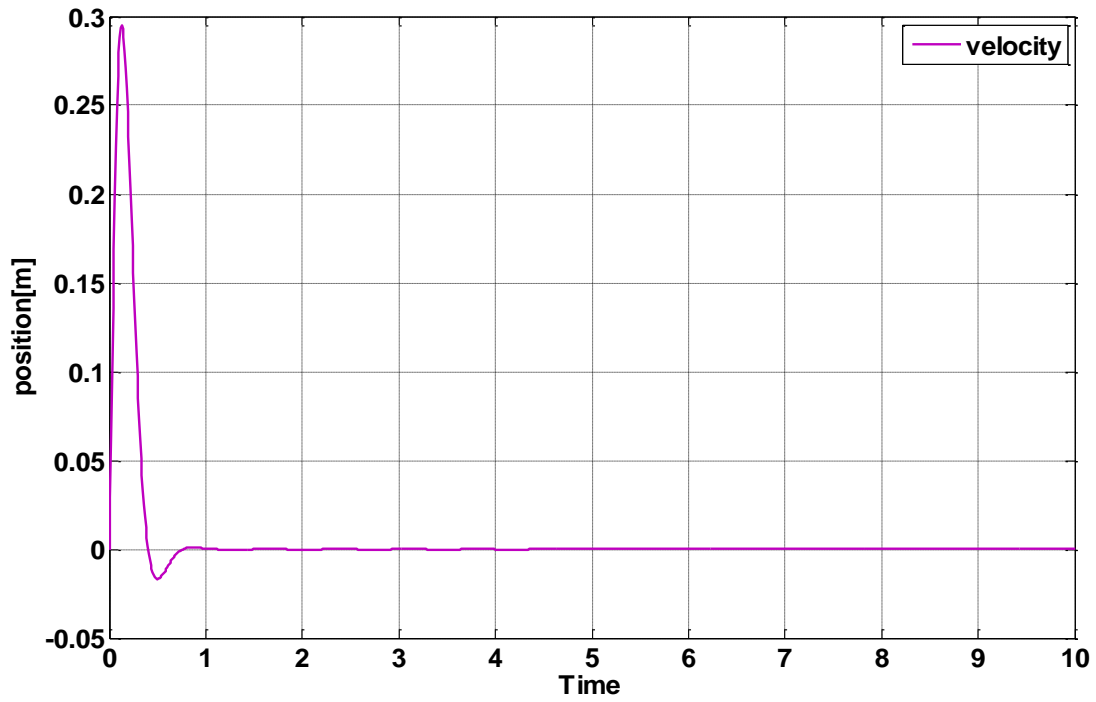


Figure 5.23: Velocity response with integral control with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

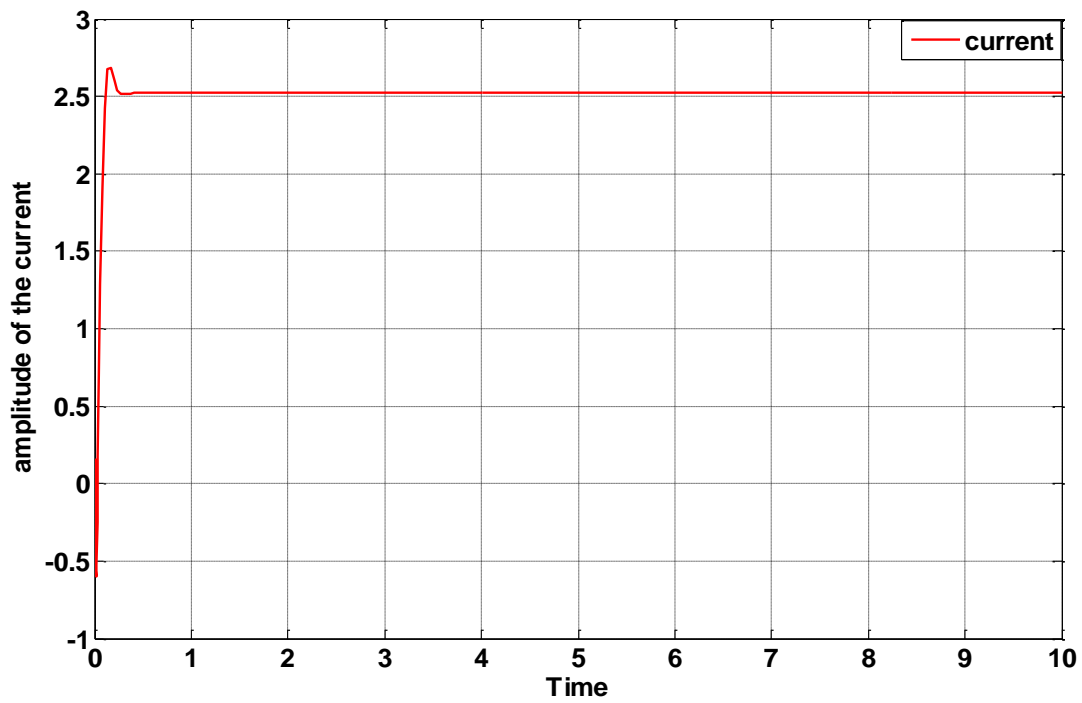


Figure 5.24: Current response with integral control with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

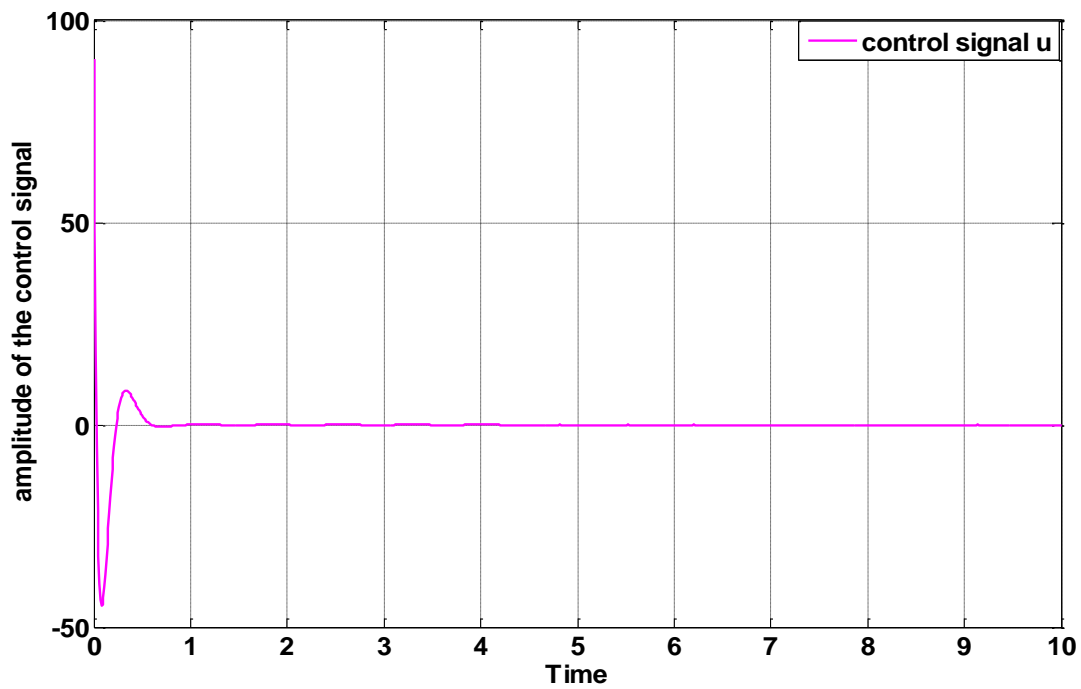


Figure 5.25: Control signal response with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

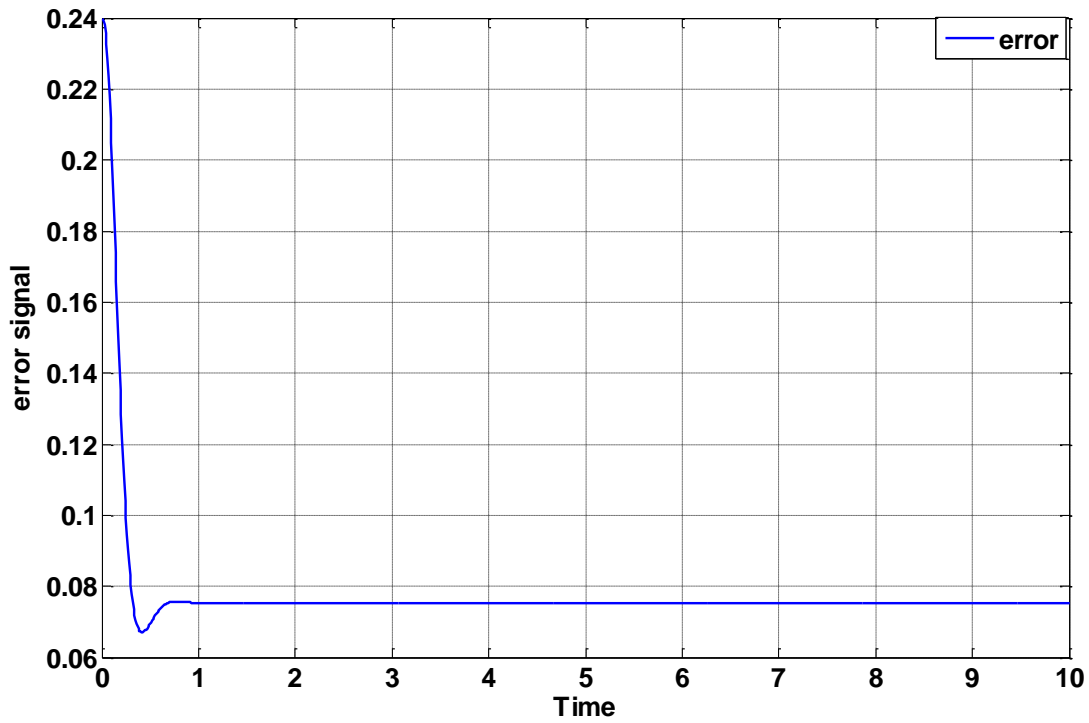


Figure 5.26: Error signal response with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

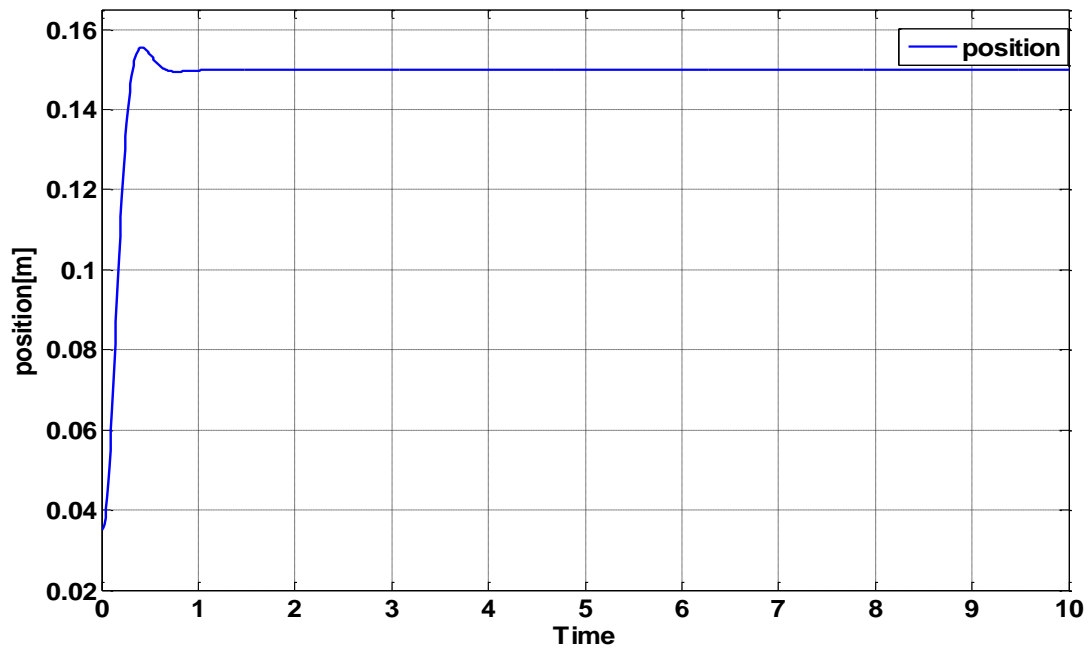


Figure 5.27: Position response with integral control with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

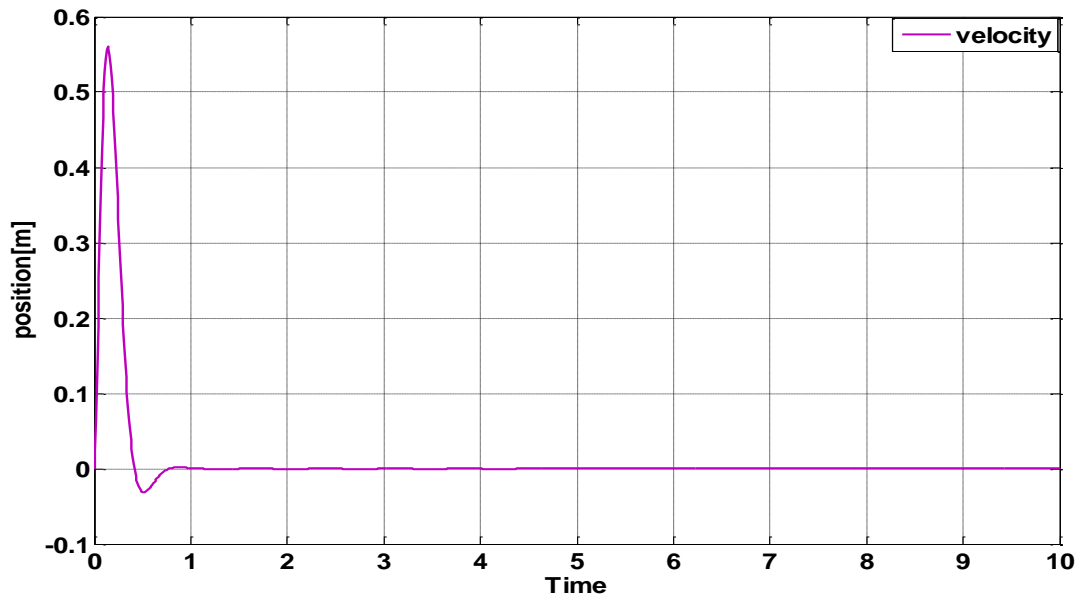


Figure 5.28: Velocity response with integral control with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

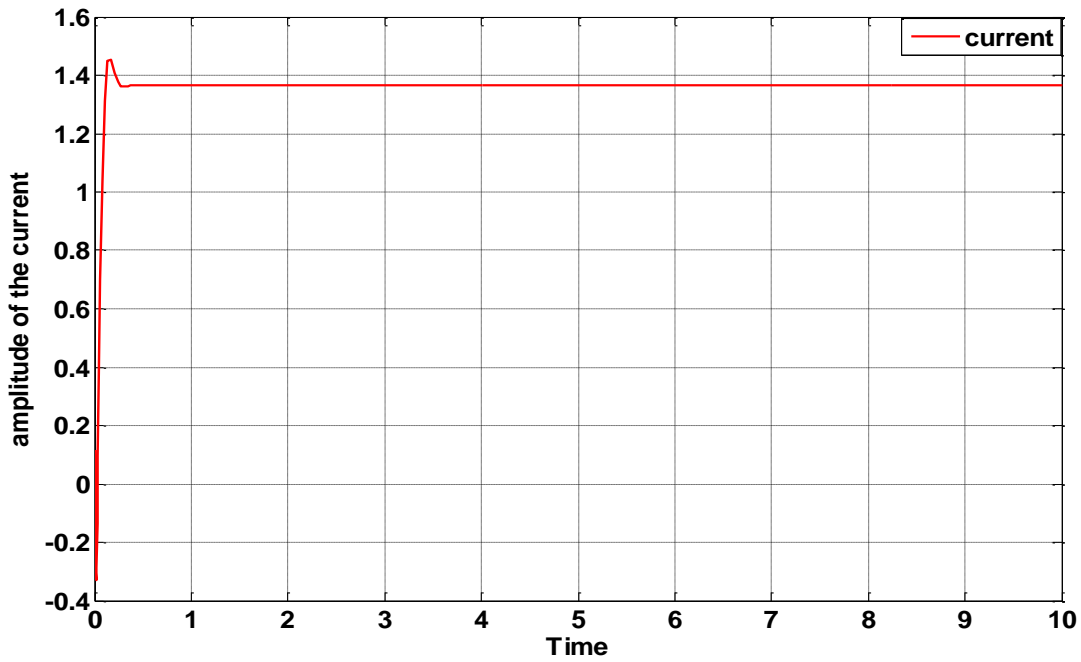


Figure 5.29: Current response with integral control with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$



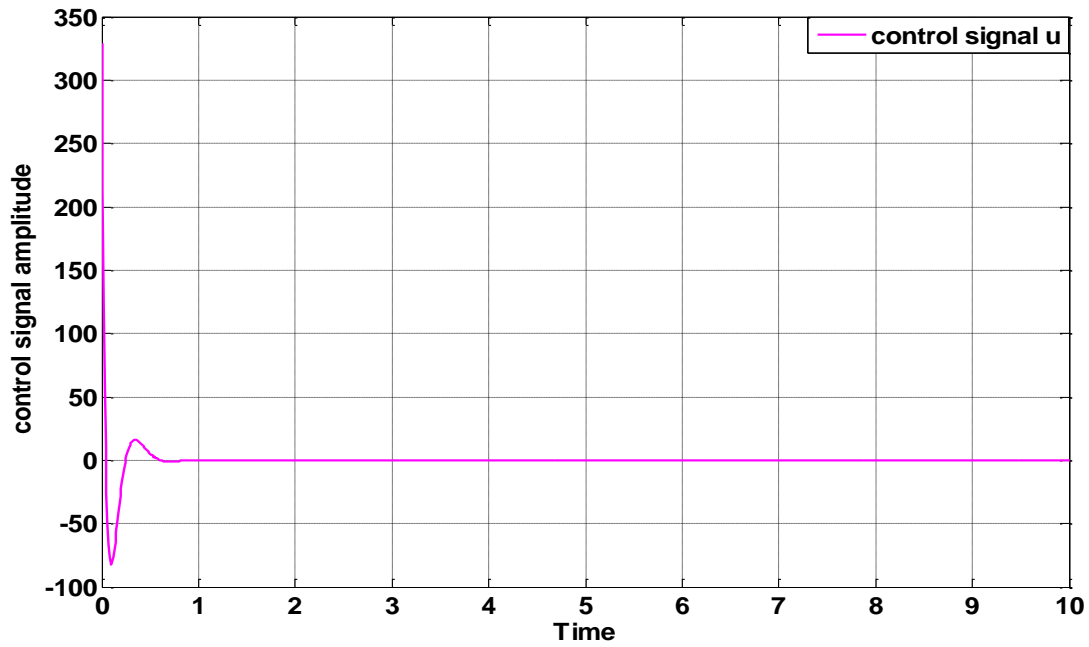


Figure 5.30: Control signal response with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

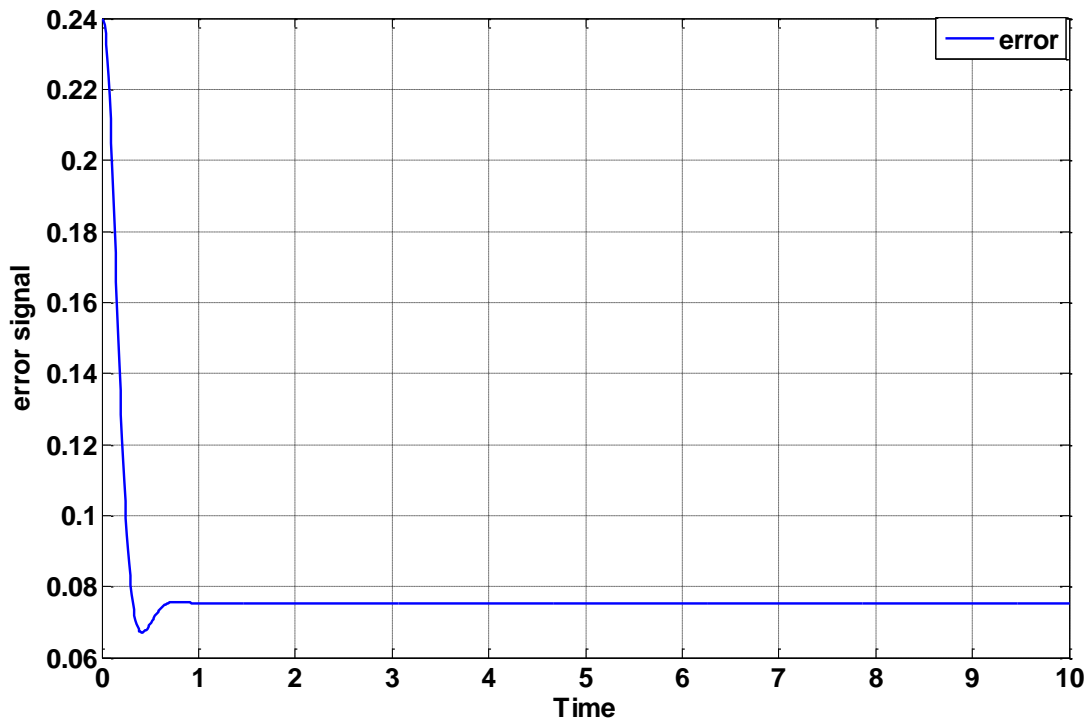


Figure 5.31: Error signal response with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

- Responses of the system when the set point is 0.2[m]:

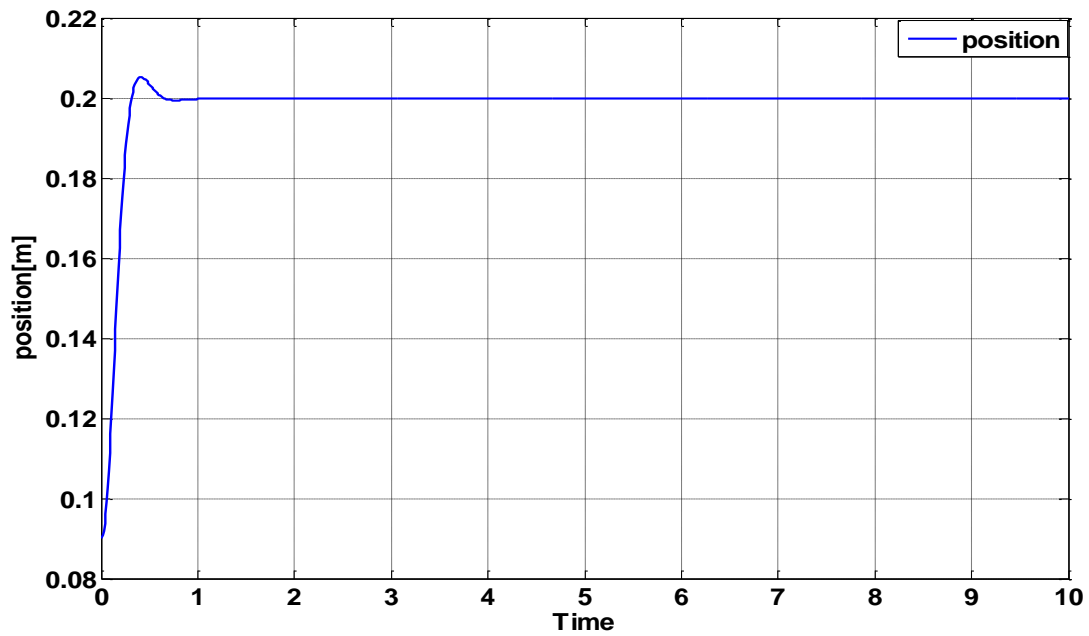


Figure 5.32: Position response with integral control with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

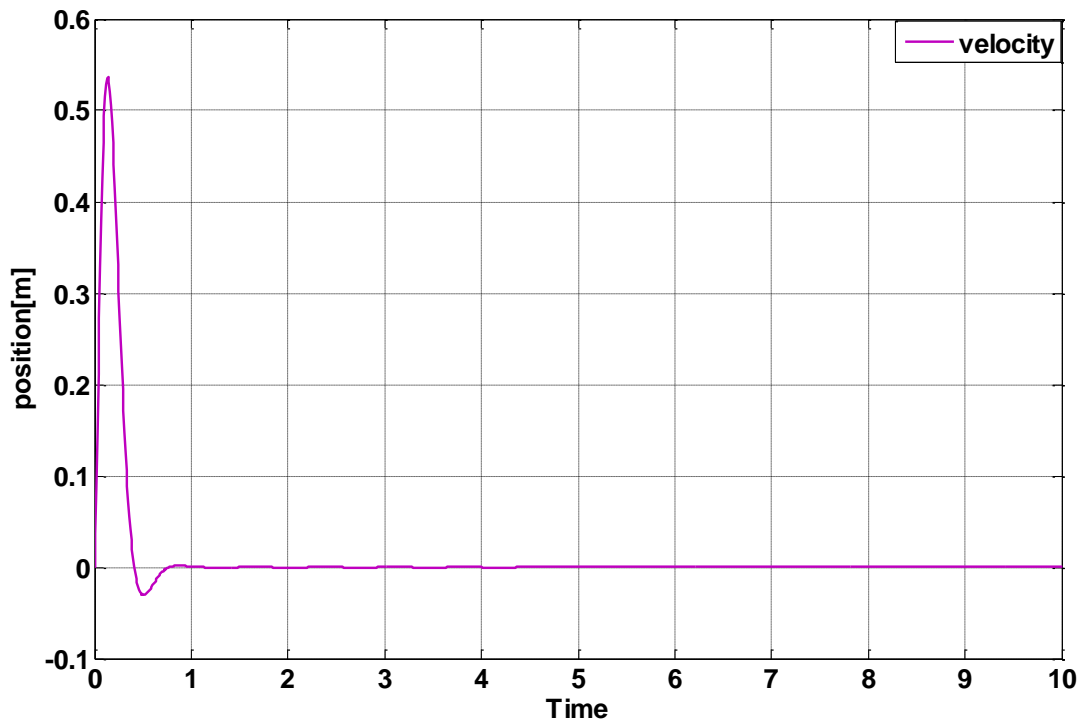


Figure 5.33: Velocity response with integral control with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

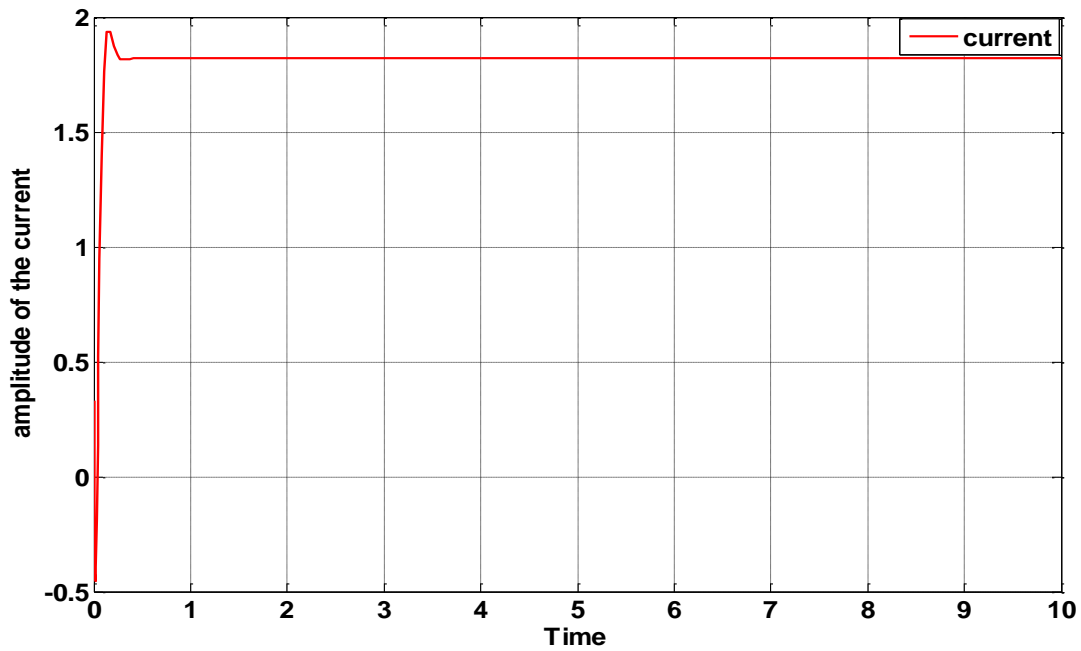


Figure 5.34: Current response with integral control with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

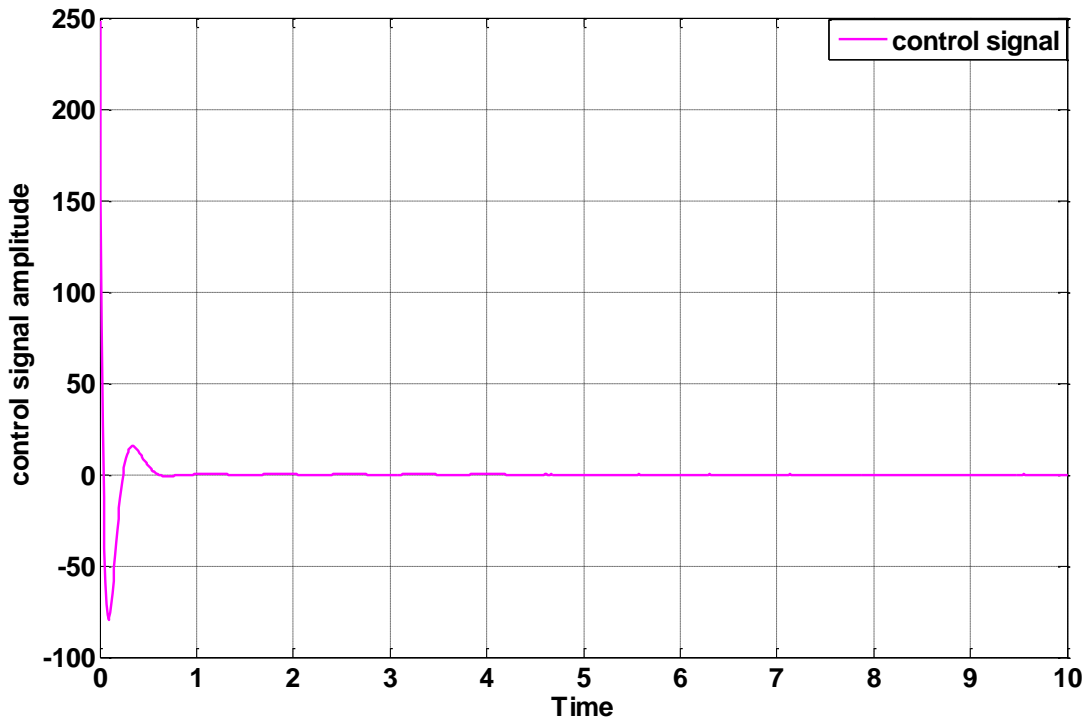


Figure 5.35: Control signal response with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

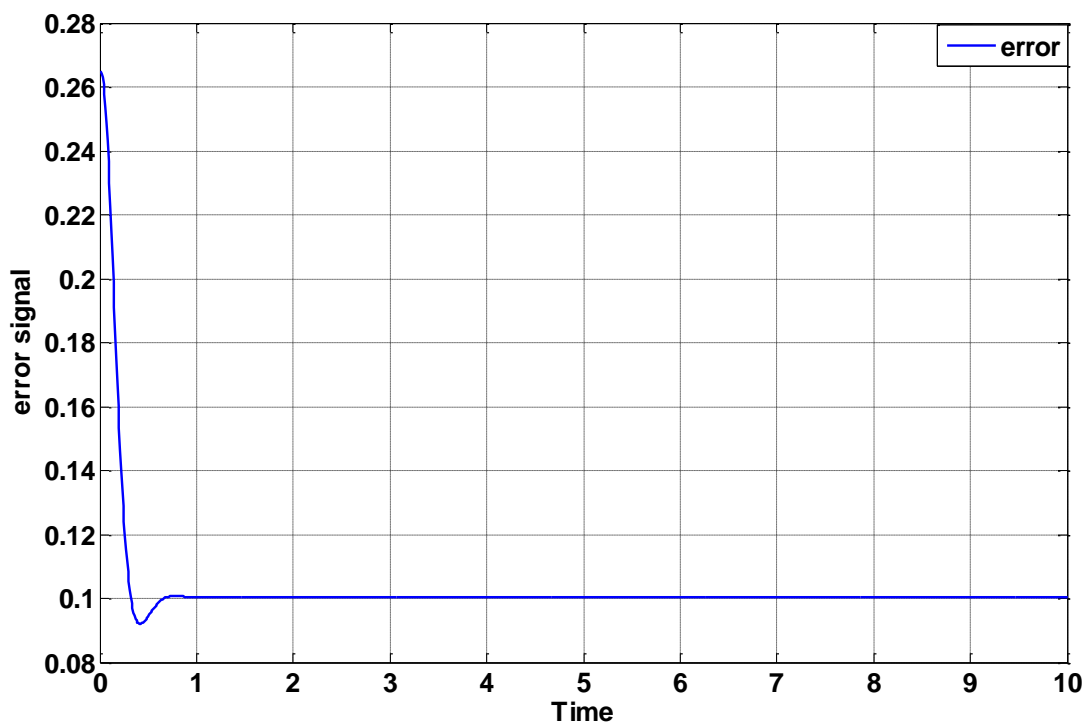


Figure 5.36: Error signal response with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

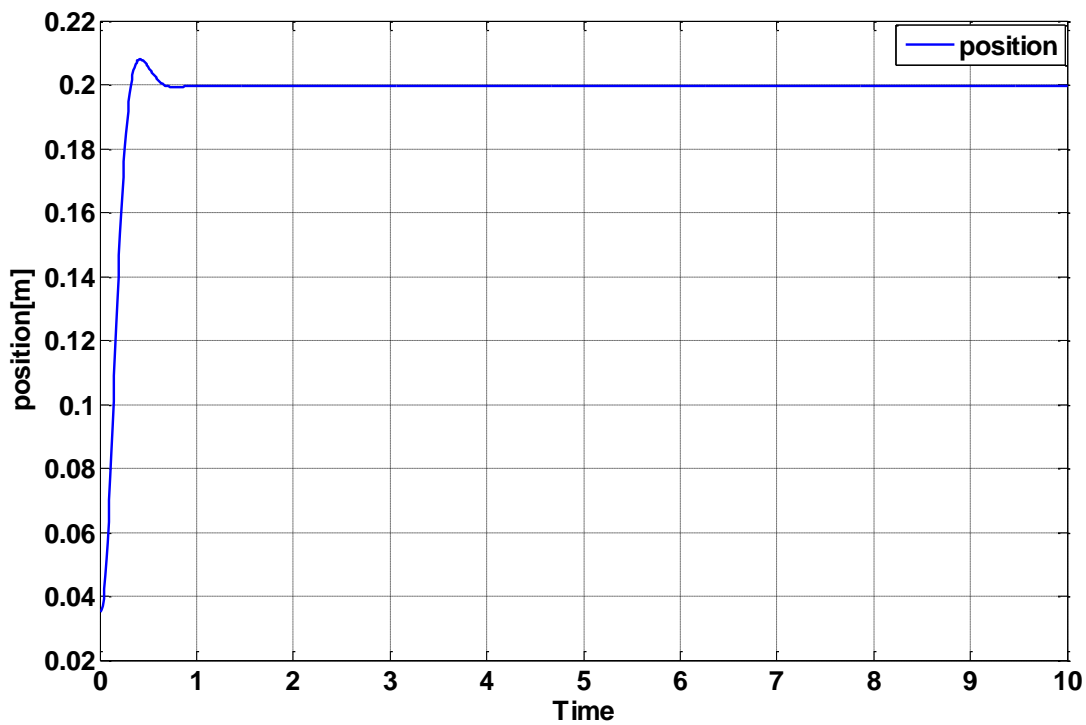


Figure 5.37: Position response with integral control with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

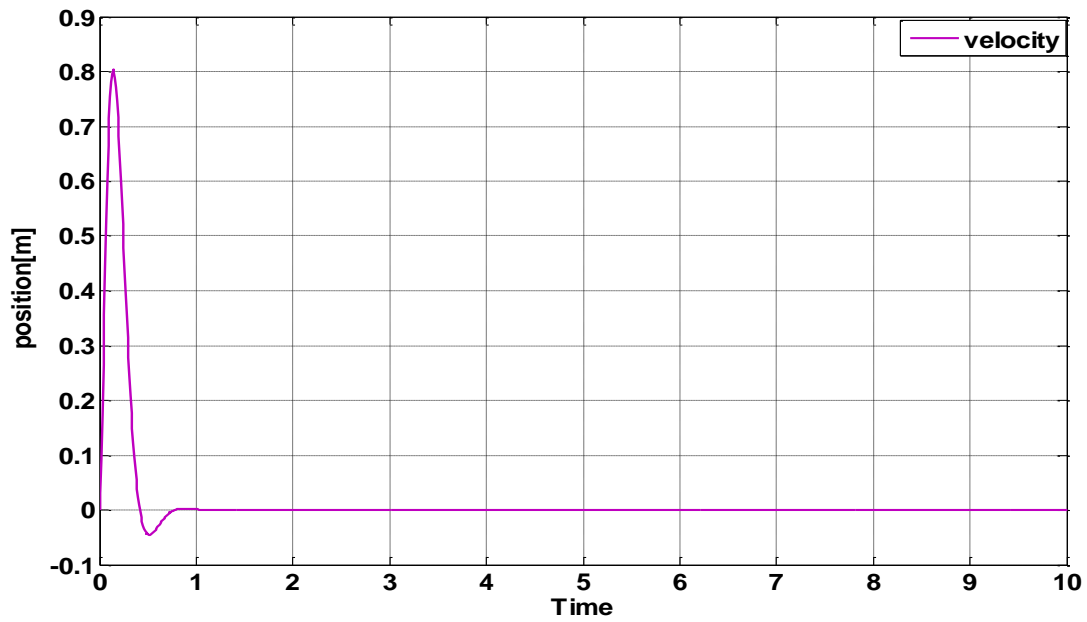


Figure 5.38: Velocity response with integral control with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

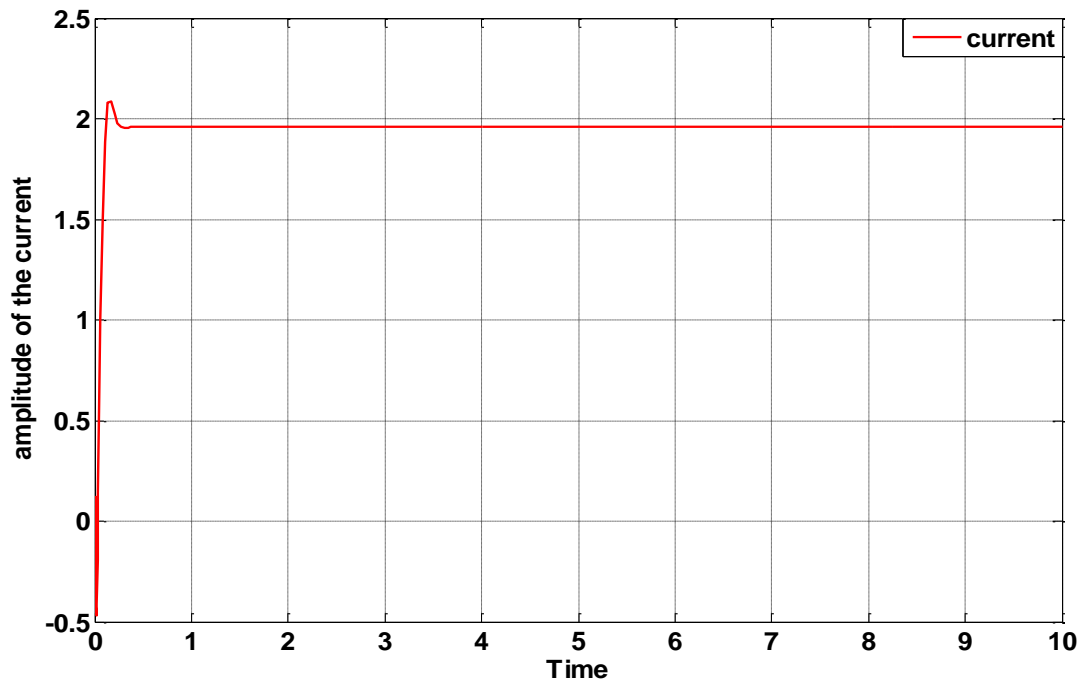


Figure 5.39: Current response with integral control with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

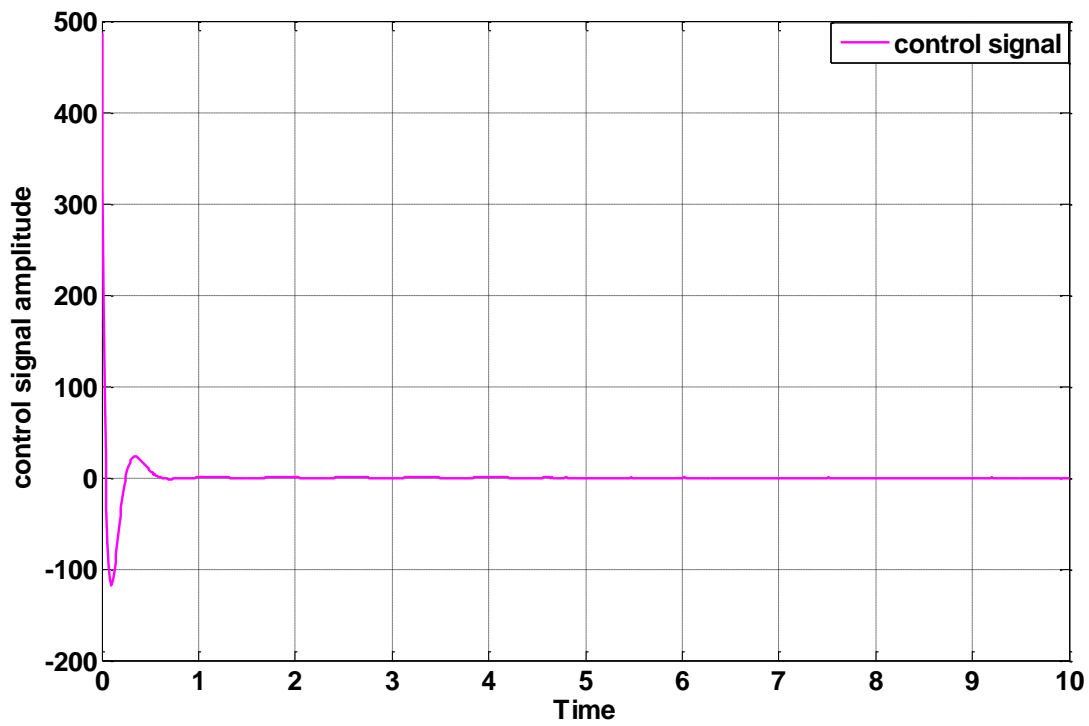


Figure 5.40: Control signal response with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

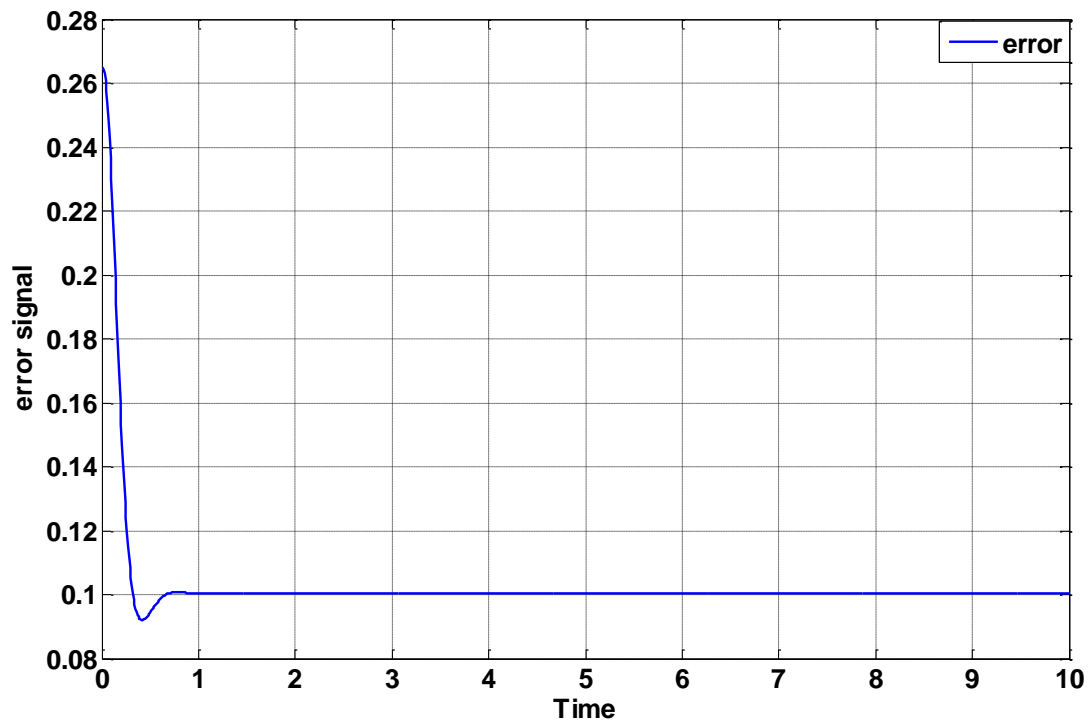


Figure 5.41: Error signal response with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

The obtained results are summarized in Table 5.2.

**Table 5.2: Characteristics of the position signal dynamic behaviour for the case of state feedback with integral action control**

Set point	Characteristics	Magnetic levitation initial conditions [0.09 0 2.56 0]'	Magnetic levitation initial conditions [0.035 0 0.89 0]'
0.15	Time Delay	0	0
	Overshoot	3%	5%
	Rising Time	0.4s	0.45s
	Steady State Error	0.075	0.075
	Settling Time	1.5s	1.5s
0.2	Time Delay	0	0
	Overshoot	3%	4%
	Rising Time	0.3s	0.3s
	Steady State Error	0.01	0.01
	Settling Time	1.7s	1.8s

The design of a linear integral controller based on pole-placement technique has been done in this section and different initials conditions have been used in the simulations. The addition of an integral action in the system has shown satisfactory results as the closed loop system was able to follow the reference input.

The next section describes the design of a linear quadratic optimal controller for the magnetic levitation system.

#### 5.4 Linear quadratic regulator LQR design

The linear quadratic control problem was discussed in the section 4.3.4 of the previous chapter. The design of the LQR is done according to the linearized model of the magnetic levitation system as it was done previously with the pole placement technique. The LQR problem consists of designing the gain matrix  $\bar{H}$  of the control input vector  $u(t) = -\bar{H}x(t)$  by minimizing the performance index J. The performance index is represented as:

$$J = \int_0^{\infty} \varepsilon(t)^T Q \varepsilon(t) + u(t)^T R u(t) dt$$

Where  $Q \in \mathbb{R}^{4 \times 4}$  is a positive definite (or semi-definite Hermitian) or real symmetric matrix.  $R \in \mathbb{R}^1$  is a positive definite Hermitian or real symmetric matrix.  $\bar{Q}$  is a square symmetric matrix called the state weighting matrix,  $\bar{R}$  is a square symmetric matrix called the control cost matrix. The solution of the optimal problem is found on Matlab using the function  $[\bar{H}, P, E] = lqr(\bar{A}, \bar{B}, \bar{Q}, \bar{R})$ . This function computes the feedback gain matrix  $\bar{H}$ . The closed loop representation of the magnetic levitation system with LQR control is the same as in the pole placement with the integral control.

#### 5.4.1 Determination and solution of the LQR problem parameters

According to theory, in the LQR design, the matrices  $\bar{Q}$  and  $\bar{R}$  determine the relative importance of the state and the expenditure of the energy. They are selected according to the best possible response of the system obtained during various simulations of the closed loop system. The following values were chosen to control the magnetic levitation system:

$$\bar{Q} = \begin{bmatrix} 10^4 & 0 & 0 & 0 \\ 0 & 250 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\bar{R} = 1$$

In these simulations, the initial conditions and the set points are the same as the one selected in the pole placement section. After the selection of the matrices  $\bar{Q}$  and  $\bar{R}$ , both are used in the Matlab *lqr* function along with the matrices  $\bar{A}$  and  $\bar{B}$  of the magnetic levitation system to find the feedback control gain. The feedback control gain found on Matlab for the closed loop system is:

$$\bar{H} = [-292.85 \quad -28.6 \quad 1.82 \quad 1]$$

#### 5.4.2 Closed-loop system behaviour simulation

The simulations of the closed-loop system with the obtained controller are done with the same Simulink diagram Figure 5.21 “*Linear\_MagLevPP.mdl*”, but it is associated with the m-file “*MagLev\_LQR.m*”. In these simulations the three states of the system and of the error are shown. The m-file of the parameters has been written and it is attached in the Appendix A.4.



The same set-points and initial conditions considered in the case of the pole placement controller with integral action are used in the simulations.

- Simulation results when the set point is 0.15 [m]:

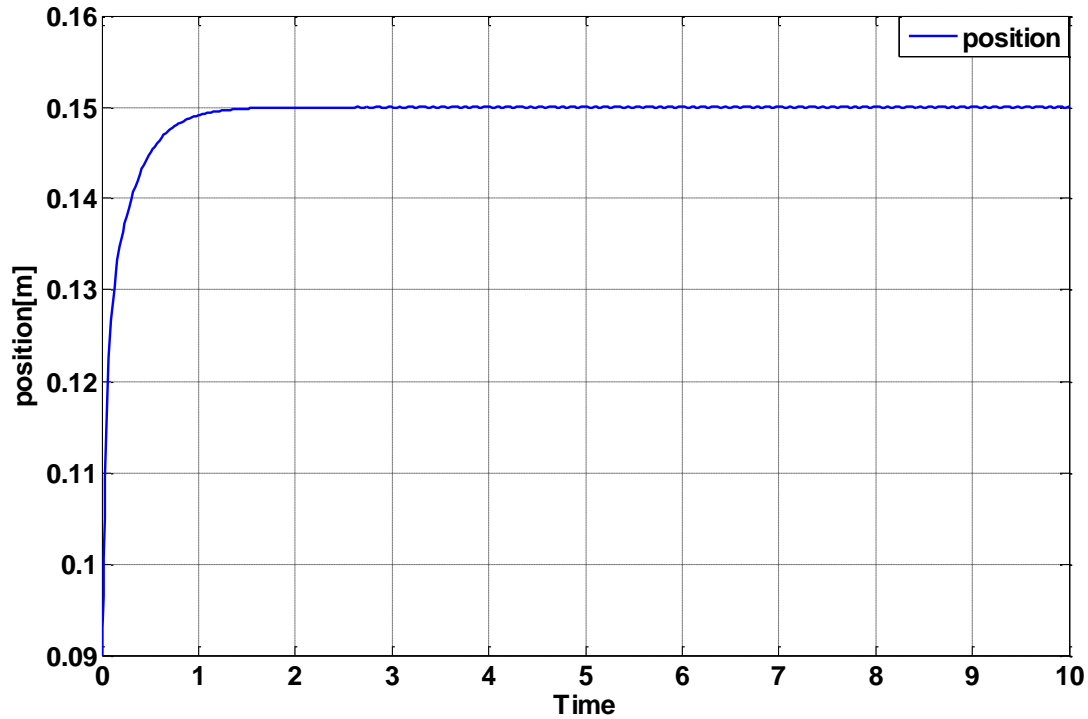


Figure 5.42: Position response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56]$

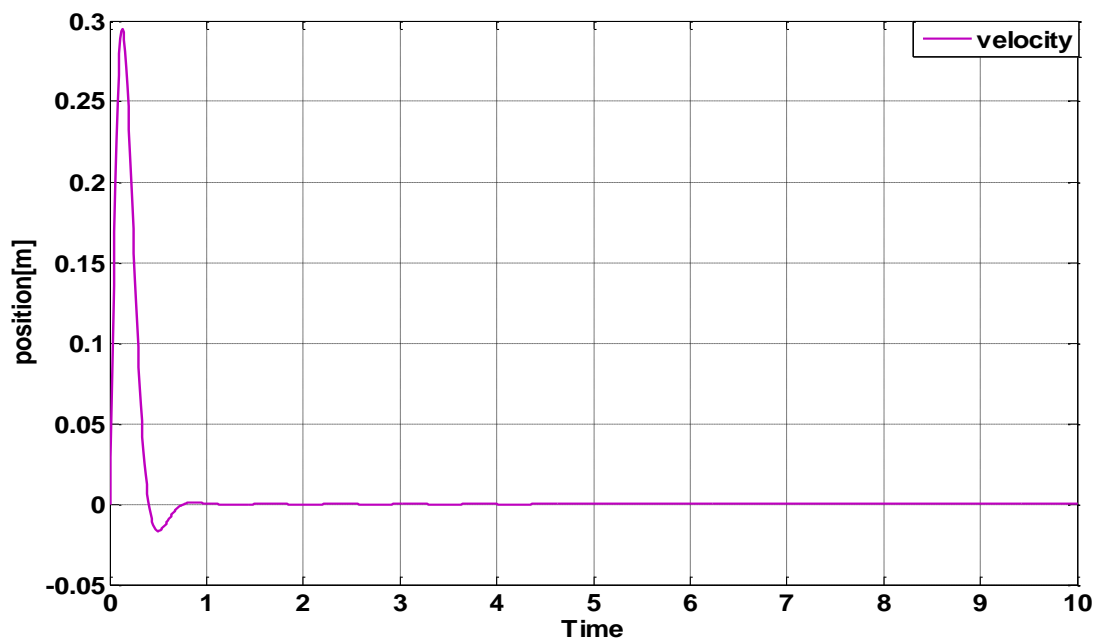


Figure 5.43: Velocity response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

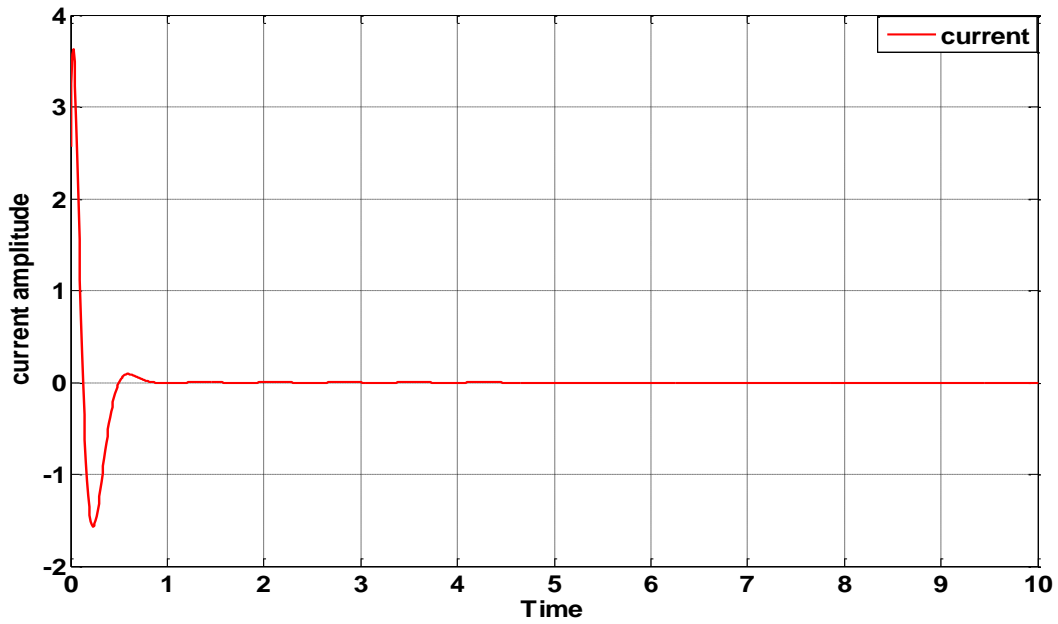


Figure 5.44: Current response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

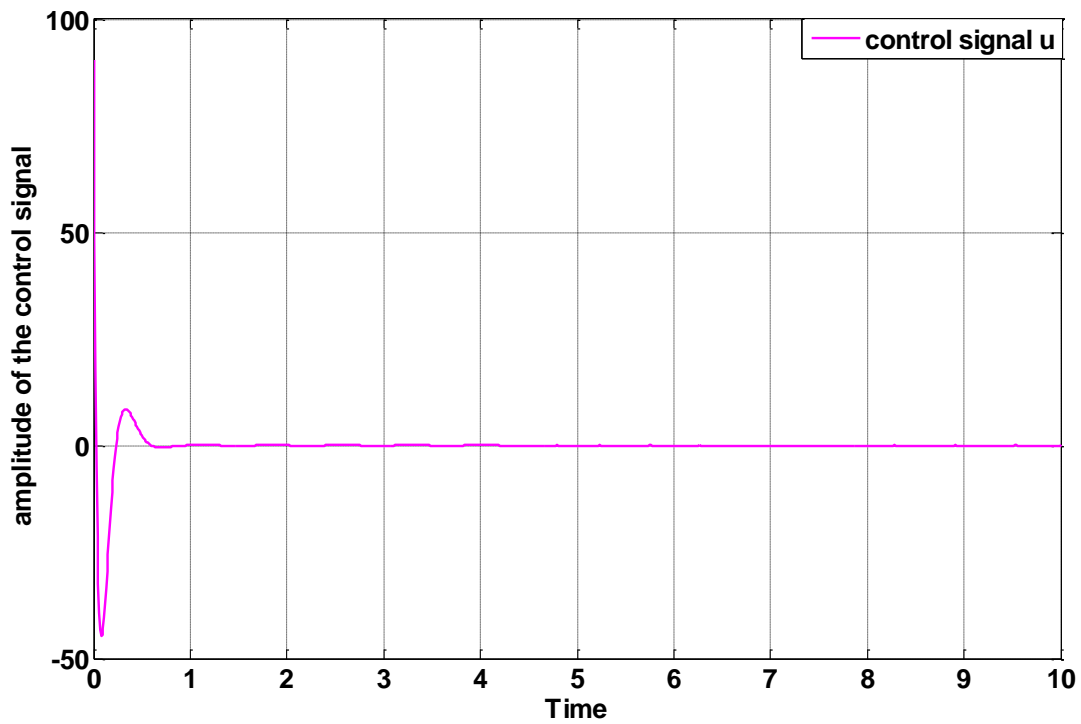


Figure 5.45: Control signal response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

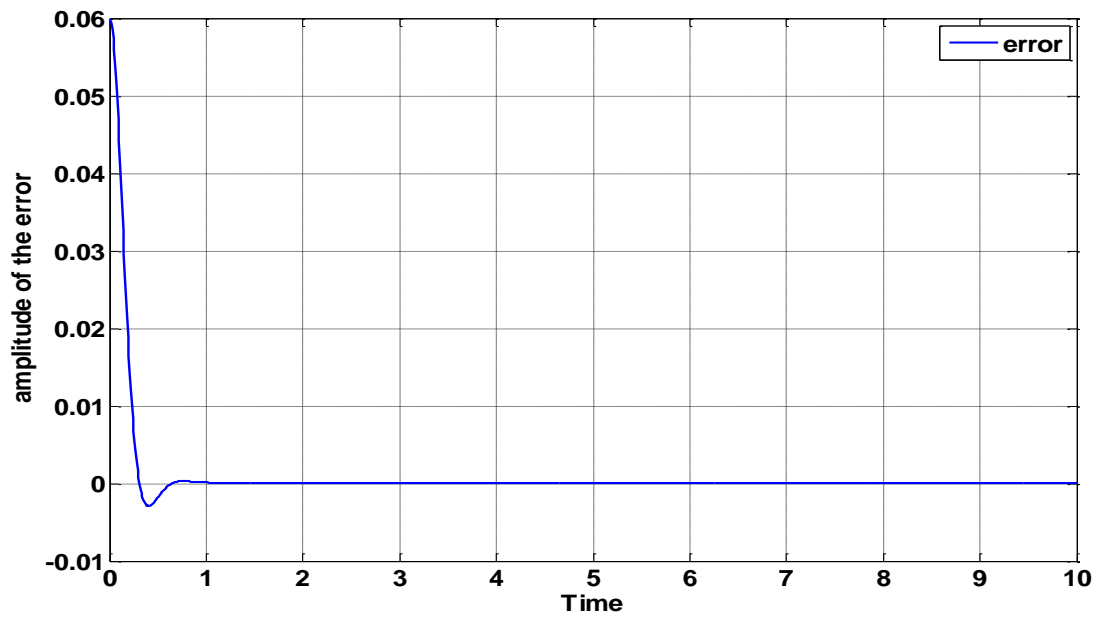


Figure 5.46: Error signal response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

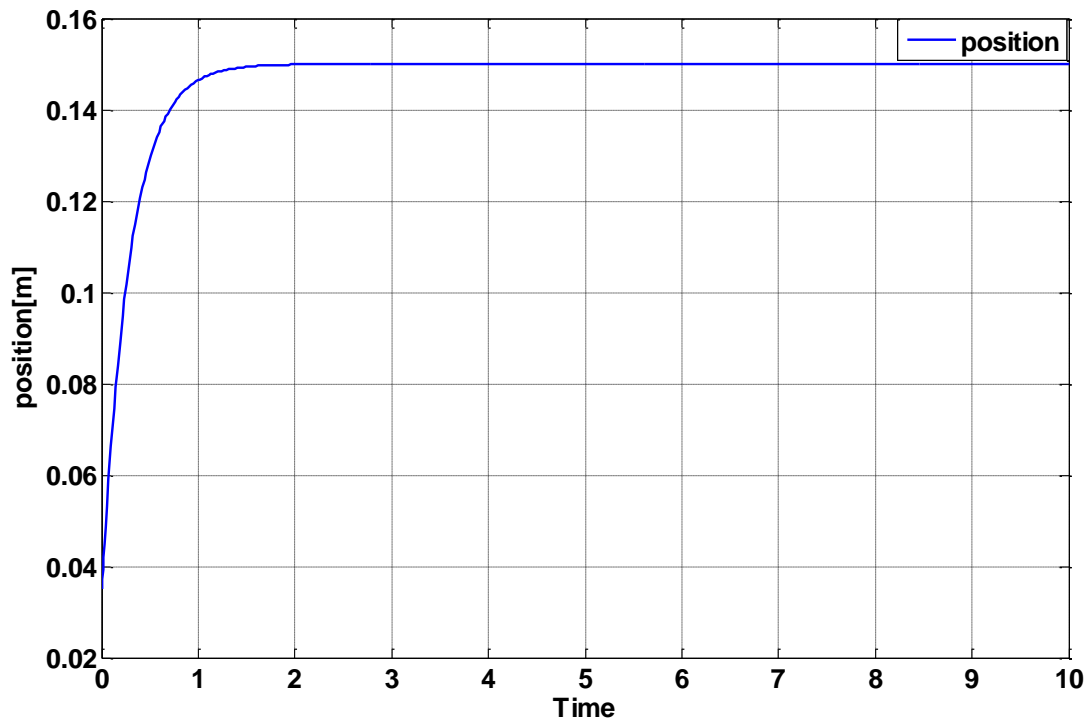


Figure 5.47: Position response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

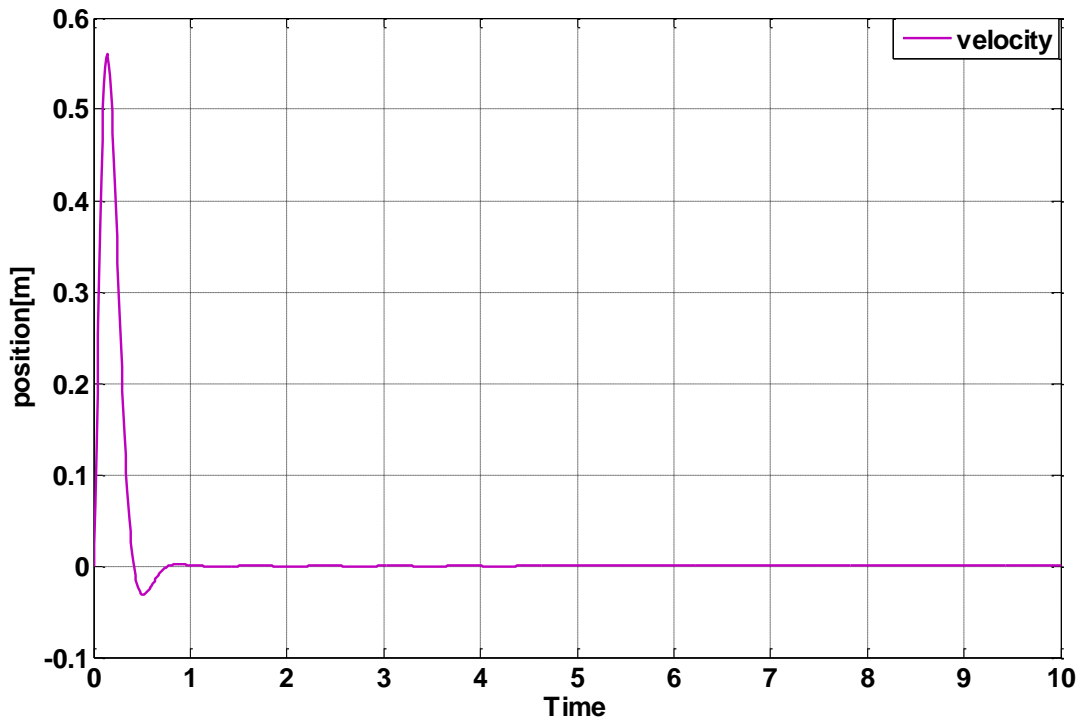


Figure 5.48: Velocity response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

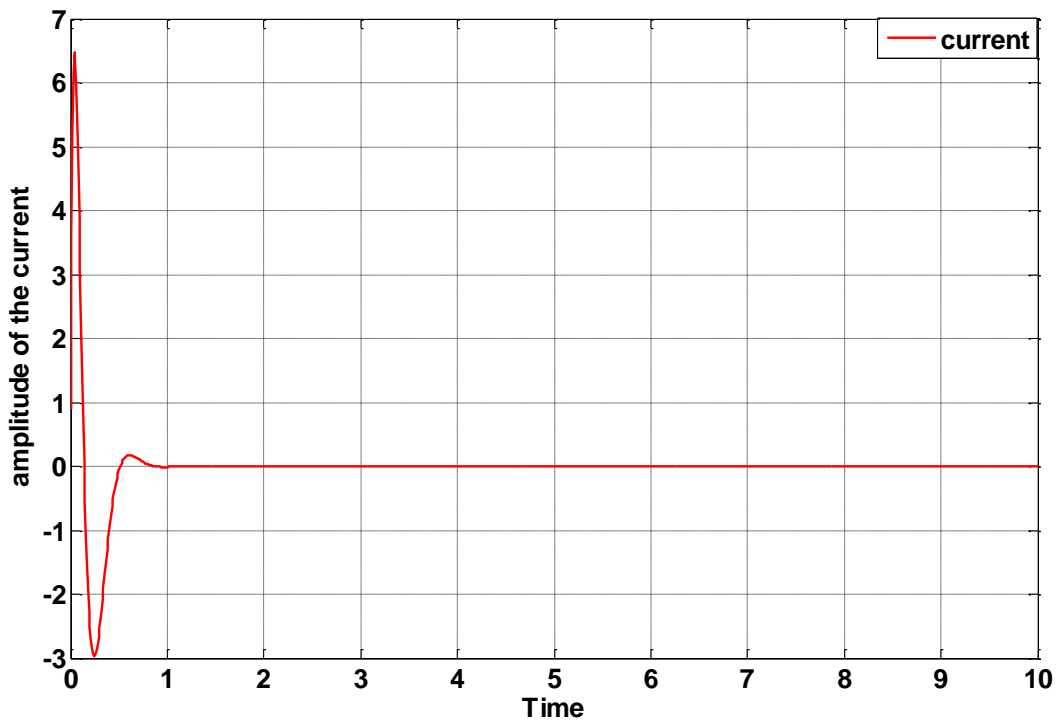


Figure 5.49: Current response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

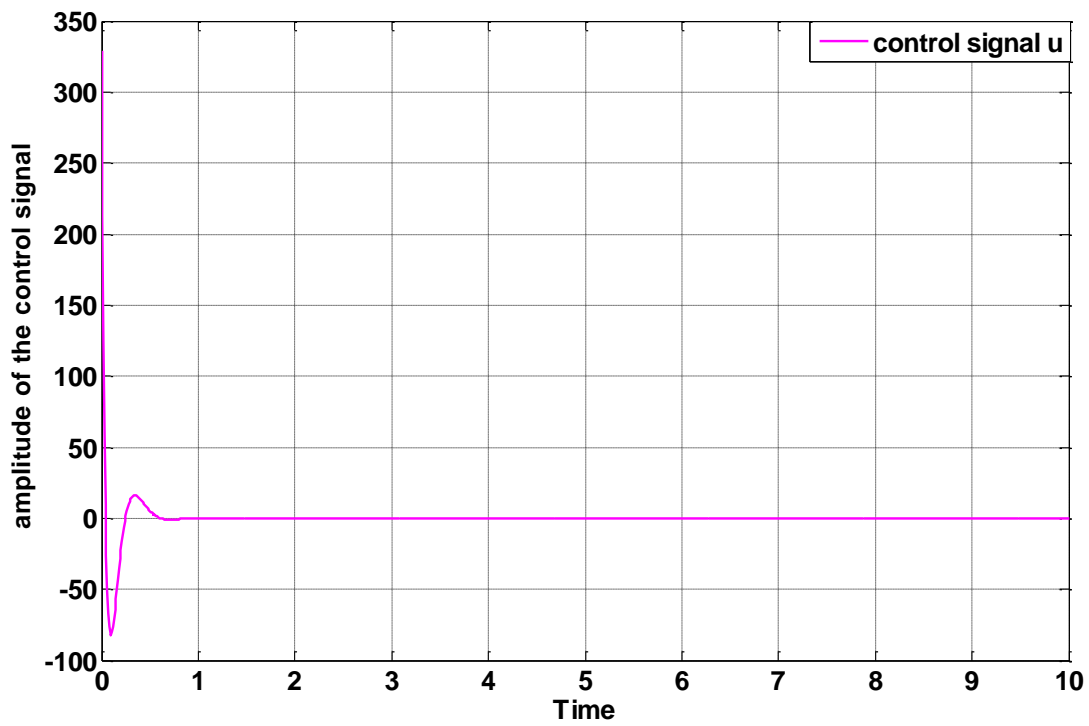


Figure 5.50: Control signal response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

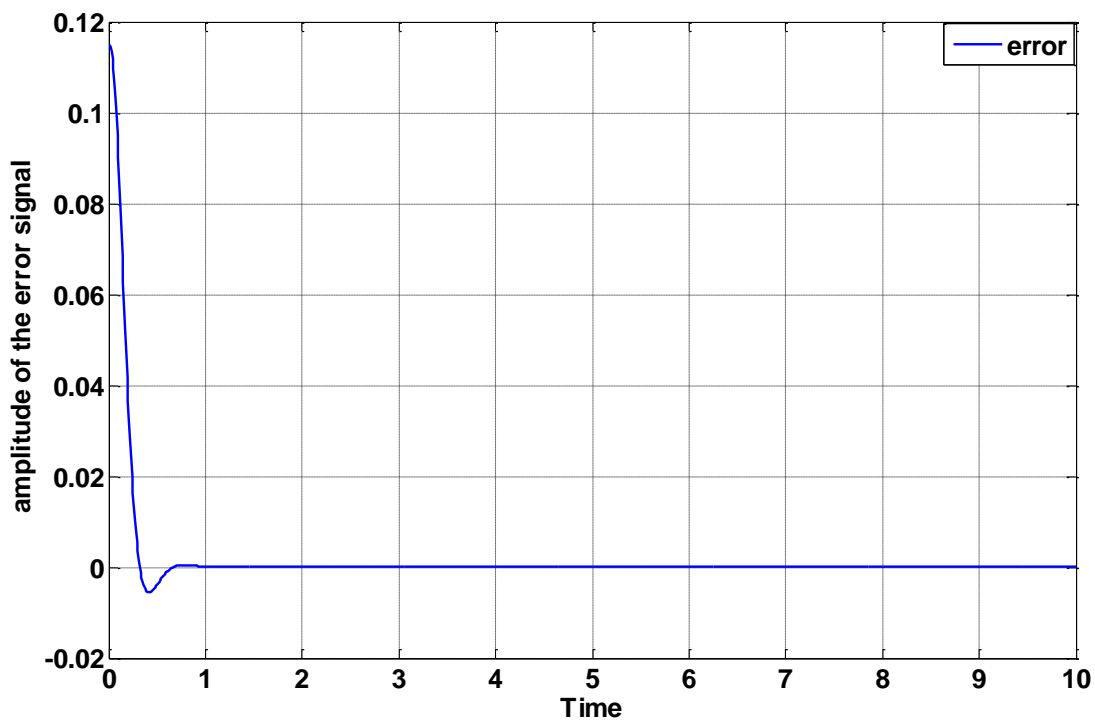


Figure 5.51: Error signal response with LQR control with  $r=0.15$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

- Simulation results when  $r=0.2$ [m]:

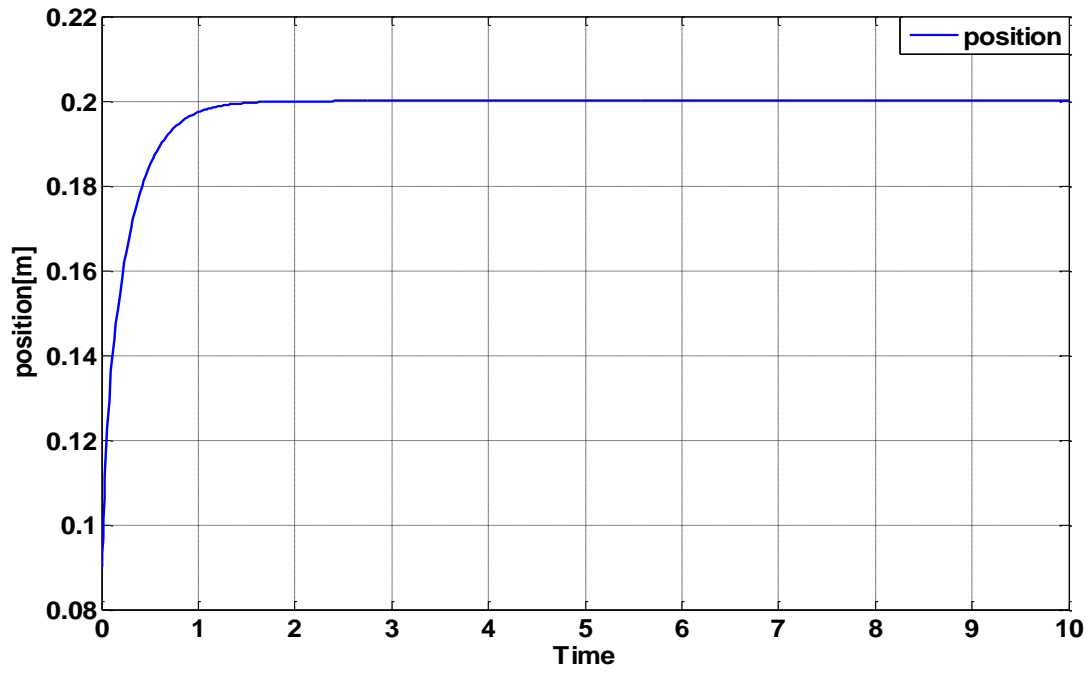


Figure 5.52: Position response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]^T$

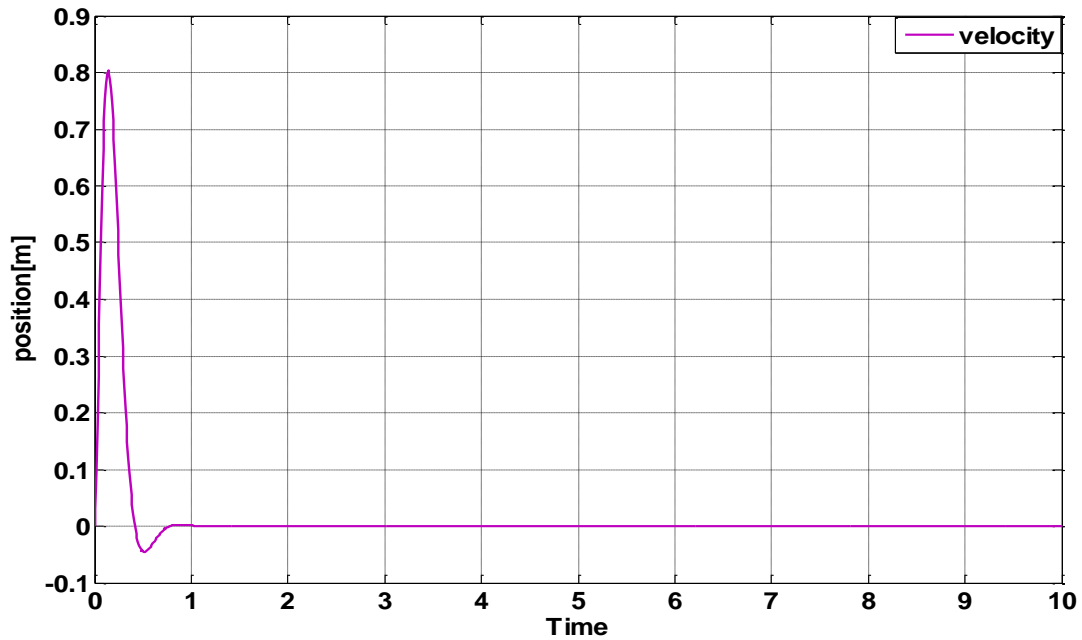


Figure 5.53: Velocity response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]^T$

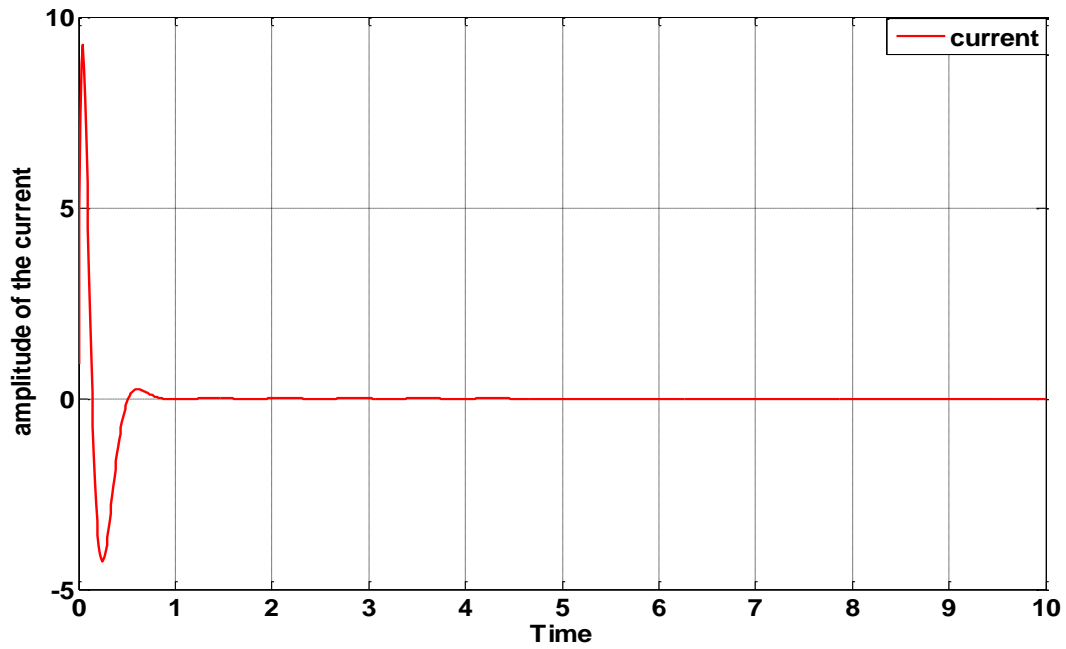


Figure 5.54: Current response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

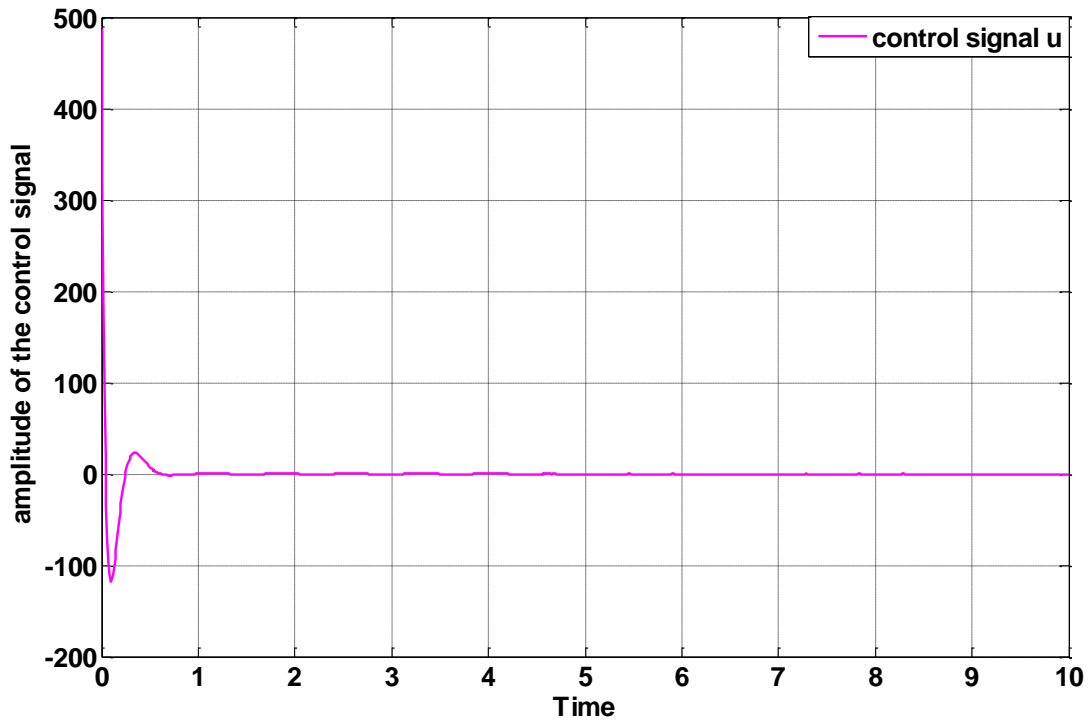


Figure 5.55: Control signal response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

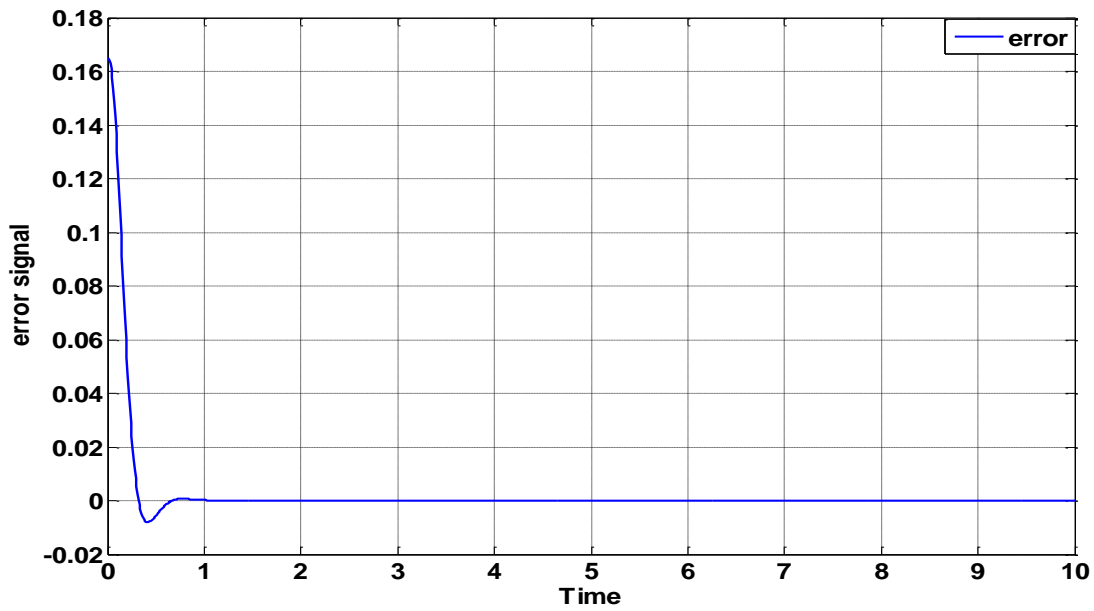


Figure 5.56: Error signal response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.09 \ 0 \ 2.56 \ 0]'$

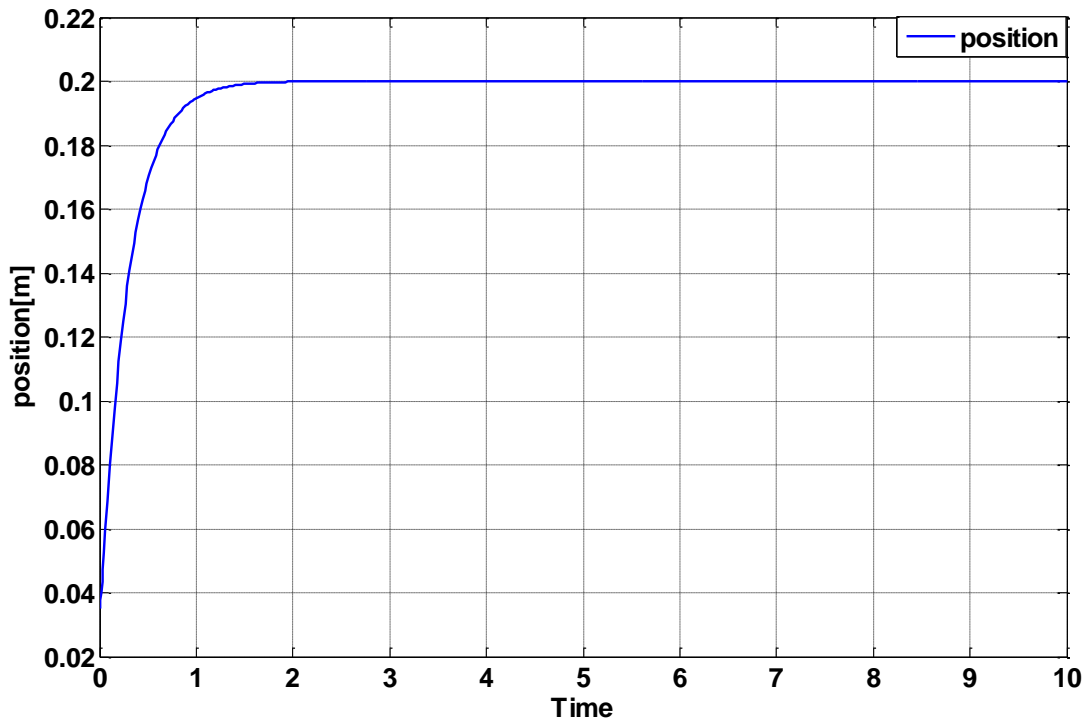


Figure 5.57: Position response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$



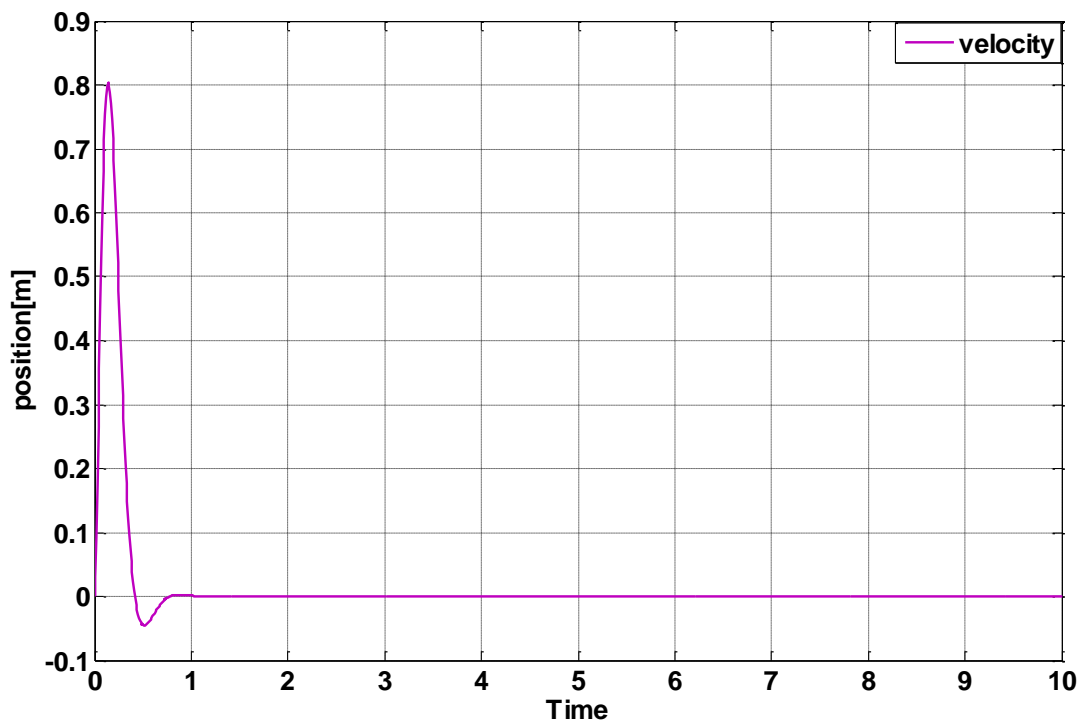


Figure 5.58: Velocity response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

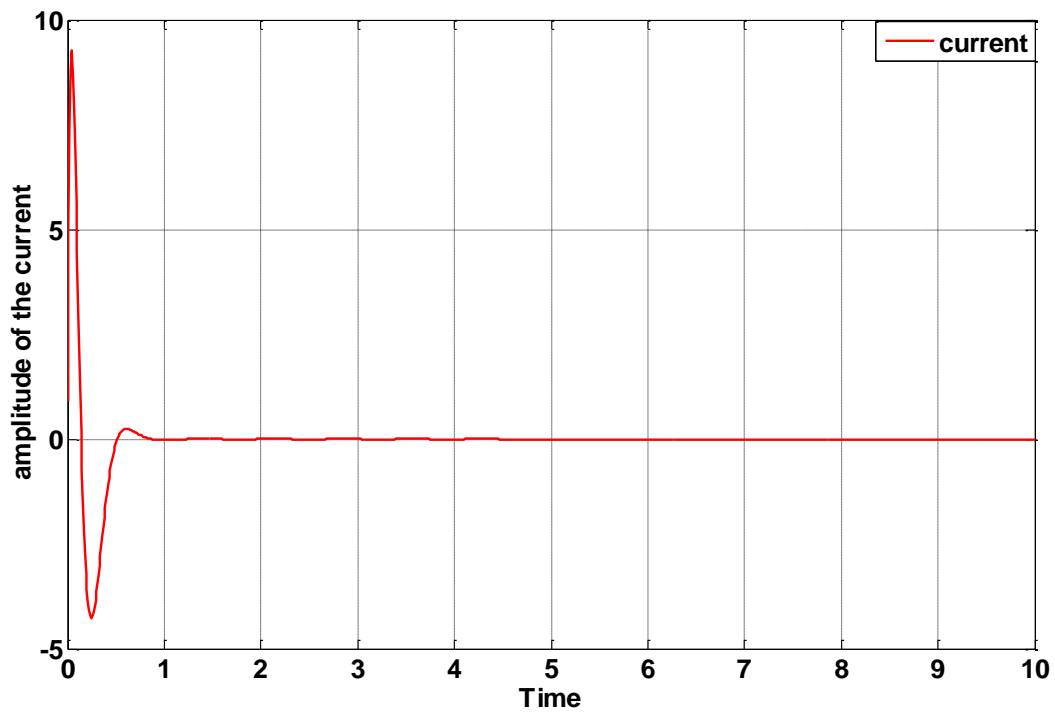


Figure 5.59: Current response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

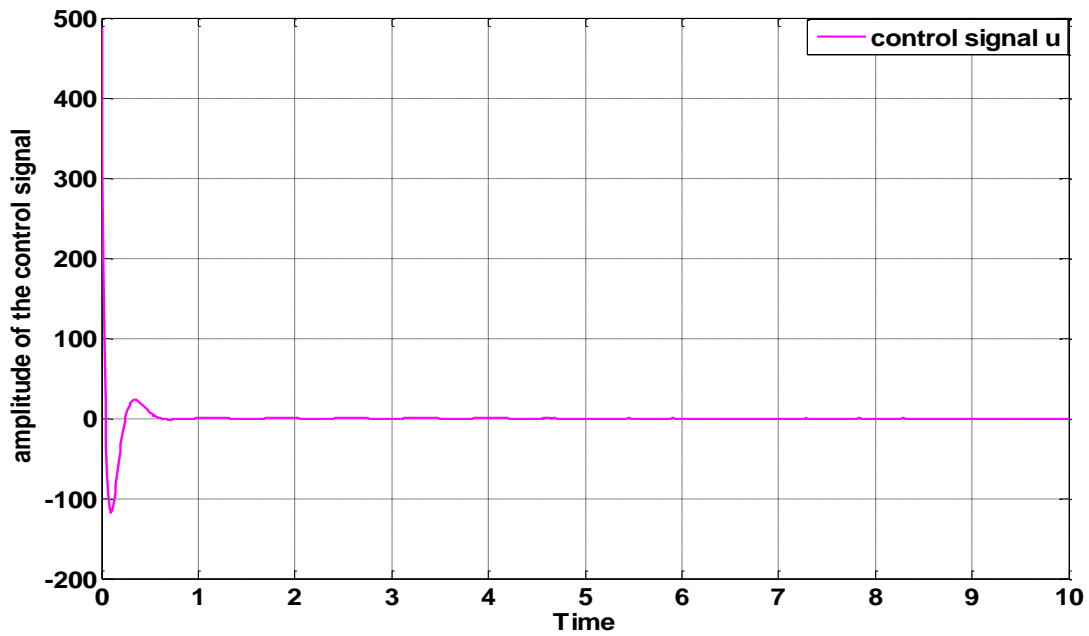


Figure 5.60: Control signal response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

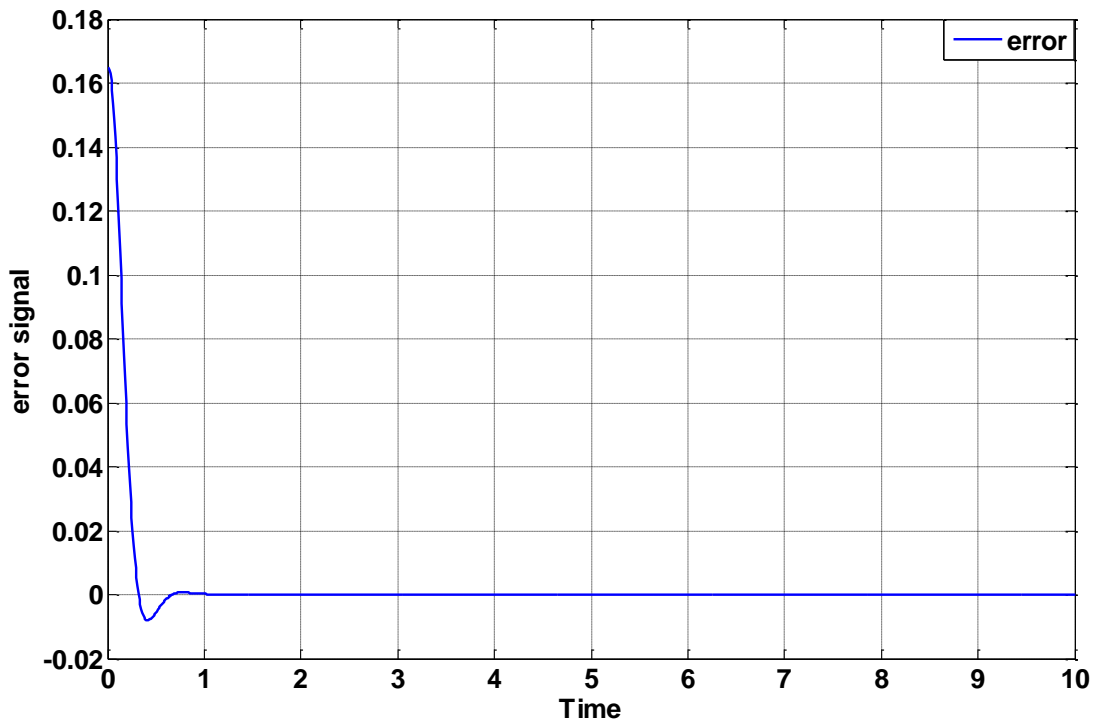


Figure 5.61: Error signal response with LQR control with  $r=0.2$  and  $\bar{x}_0 = [0.035 \ 0 \ 0.89 \ 0]'$

Table 5.3 shows the characteristics of the position transition behaviour.

**Table 5.3: Characteristics of the position transition behaviour for the case of LQR with integral action controller**

Set point	Characteristics	Magnetic levitation initial conditions [0.09 0 2.56 0]'	Magnetic levitation initial conditions [0.035 0 0.89 0]'
0.15	Time Delay	0	0
	Overshoot	0.9%	1%
	Rising Time	0.2s	0.4s
	Steady State Error	0.001	0.003
	Settling Time	1.5s	1.5s
0.2	Time Delay	0	0
	Overshoot	0.8%	1.2s
	Rising Time	0.5s	0.7s
	Steady State Error	0.01	0.0015
	Settling Time	1.7s	1.8s

In this section, the LQR controller with integral action was designed and simulated according to different initial conditions. The optimal values of the matrices  $\bar{Q}$  and  $\bar{R}$  to obtain a satisfactory response of the closed loop system were found after a certain number of trials. The next section compares the performances of the pole placement technique and LQR one to decide on the best control method to further design control of the linearized system by a nonlinear controller magnetic levitation system.

### 5.4.3 Results analysis

The obtained results described from Table 5.1 to Table 5.3 show that the LQR with integral action controller has better tracking performance than the pole placement one as it does not show any overshoot and steady state error which were the two main problems encountered in the pole placement technique.

### 5.5 Conclusion

In this chapter, two control techniques are discussed in details and applied to design controllers using the linearized model of the magnetic levitation system. The integral LQR controller showed better performances and is selected as control method to be used for the linearized

closed-loop system. Chapter 6 presents the feedback linearization of the magnetic levitation system.

## CHAPTER SIX: DESIGN OF A NONLINEAR LINEARIZING CONTROLLER ON THE BASIS OF INPUT/STATE INPUT/OUTPUT LINEARIZATION

### 6.1 Introduction

Feedback linearization is a method to design nonlinear controllers which has drawn massive interest in various research fields lately. The main idea of the concept is to mathematically transform the dynamics of a nonlinear system into a partly or fully linear one (Slotine and Li, 1991), (Isidori, 1994) and (Astom, 2000). In its global representation, linearization by feedback of a nonlinear system nullifies its contained nonlinearities such that the dynamics of a closed loop system are represented linearly. This is completely different from linearization performed conventionally, in which linearization by feedback is realized by accurate transformation of states and feedback, instead of linear approximation of the dynamics.

Feedback linearization techniques can be seen as methodologies of transformation of the system models derived originally into identical models of much simpler representation.

This concept has been applied with success to deal with very challenging control problems. These include the control of helicopters, high performance aircrafts, biomedical devices, and industrial robots. The advantages of feedback linearization are that the design of the control algorithm can be used generally, which means that the same concept can be used to all types of systems. However feedback linearization presents also a number of important shortcomings and limitations (Henson and Seborg, 1997) and (Guemghar, 2005).

The direct consequence of this approach is that feedback linearization is fundamentally subjected to the same restrictions as the direct inverse control and shares the same attributes such as:

The advantages:

- Simple implementation.
- Performs exact linearization not like the approximate linearization of nonlinear model equations.
- Only the model of the system to be controlled is needed.
- Adjustment of the closed loop behavior can be made without retraining of the model. An outer feedback can be included containing a linear controller,

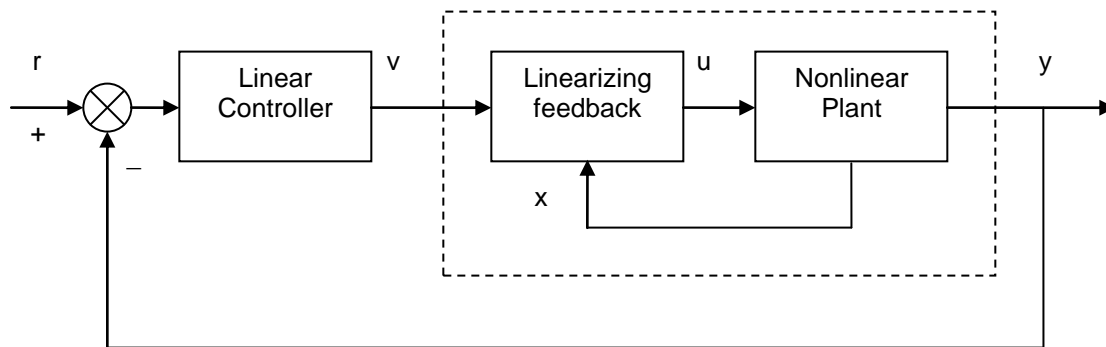
and the disadvantage:

- Lack of design parameters to tune the controller.

- Linearizing effect of the controller depends on the values of the model parameters, and they can vary.
- Very big control efforts sometimes have to be realised to linearize the closed loop system.
- Restricted to a specific category of systems. Hard to resolve whether an unknown system is part to this class.

The nonlinear linearizing controller linearizes a closed loop system consisting of the nonlinear plant and the nonlinear controller with output  $y$ .

This type of control system is very sensitive to model parameters variations. In order to overcome these problems, an additional linear controller  $v$  designed for the linearized system is needed. The fundamental idea of feedback linearization is to use a control law which is made of two components: one that nullifies the plant's nonlinearities and the other component controls the resulting linear system. The structure of feedback linearization control is shown in Figure 6.1.



**Figure 6.1: Feedback linearizing system with external linear controller**

This chapter is organised as follow: feedback linearization and the process model canonical form are presented in section 6.2. Mathematical tools for feedback linearization are described in section 6.3. Further the design of a feedback linearizing controller for magnetic levitation is discussed in section 6.4. The results of the simulation are presented in section 6.5, section 6.6 discusses its results. Finally section 6.7 gives the conclusion based on the results obtained.

## 6.2 Feedback linearization and the canonical form of the process model

In its basic aspect, feedback linearization amounts to cancelling the nonlinearities in a nonlinear system such that the dynamics of the closed loop system is represented linearly (Slotine and Li, 1991), (Chiasson, 1998) and (Ravi, 2013). Cancelling the nonlinearities and imposing a certain

linear dynamics, can be directly applied to a certain category of nonlinear systems, such systems are known as being in companion form, or controllability canonical form (Farzaneh, 2011).

The dynamics of a system represented in companion form are expressed by:

$$\dot{x}^n = f(x) + b(x)u \quad (6.1)$$

Where  $u$  is the scalar control input,  $x$  is the scalar output of interest,  $x^n = \left[ x, \dot{x}, \dots, x^{(n-1)} \right]^T$  is

the vector of states,  $f(x)$  and  $b(x)$  are nonlinear functions of states. This representation is unique because, although the derivatives of  $x$  is shown in the equation, there is no present derivative of the input  $u$ .

Equation (6.1) can be represented in the state space form:

$$\frac{d}{dt} \begin{bmatrix} x^1 \\ \vdots \\ x^{n-1} \\ x^n \end{bmatrix} = \begin{bmatrix} x^2 \\ \vdots \\ x^n \\ f(x) + b(x)u \end{bmatrix} \quad (6.2)$$

For a non-zero  $b$  the control input can be defined as:  $u = \frac{1}{b}(v - f)$ .

The nonlinearities can be cancelled to obtain the fundamental input-output relation (multiple integrator form):  $x^n = v$  (6.3)

Hence, the control law is:  $v = -h_0 x - h_1 \dot{x} - \dots - h_{n-1} x^{(n-1)}$  (6.4)

$h_i$  is selected so that the roots of the polynomial  $s^n + h_{n-1}s^{n-1} + \dots + h_0$  are strictly in the left side of the complex plane, that leads to exponentially stable dynamics:

$$x^n + h_{n-1}x^{(n-1)} + \dots + h_0x = 0 \quad (6.5)$$

For tracking performance of a specific desired output  $x_d$ , the control law is:

$$v = \ddot{x}_d - h_0 e - h_1 \dot{e} - \dots - h_{n-1} e^{(n-1)} \quad (6.6)$$

Where:  $e = x - x_d$

Similar results would be obtained if the scalar  $x$  was substituted by a vector and the scalar  $b$  by an invertible square matrix.

There are two approaches to convert a nonlinear system into a controllable canonical form: input state linearization and input output linearization. These approaches are discussed further in this chapter. The next section discusses on the different mathematical tools used to achieve feedback linearization.

### 6.3 Mathematical tools for feedback linearization

This section discusses the different mathematical theoretical concepts applied in feedback linearization. Lie derivatives and Lie brackets along with diffeomorphisms and state transformations are explained.

In describing mathematical theories, a vector function  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  is called a vector field in  $\mathfrak{R}^n$ , to be consistent with the terminology used in differential geometry. The intuitive reason for this term is that every vector function  $f$  corresponds to a field of vectors in a  $n$ -dimensional space.

Given a smooth vector function  $h(x)$  of the state  $x$ , the gradient of  $h$  is denoted by  $\bar{V}h$ :

$$\bar{V}h = \frac{\partial h}{\partial x} \quad (6.7)$$

The gradient is represented by a row-vector of elements  $(\bar{V}h)_i = \partial h / \partial x_j$ . Similarly, given a vector field  $f(x)$ , the Jacobian of  $f$  is denoted  $\bar{V}f$ :

$$\bar{V}f = \frac{\partial f}{\partial x} \quad (6.8)$$

It is represented by a  $n \times n$  matrix of elements  $(\bar{V}f)_{ij} = \partial f_i / \partial x_j$ ,  $i = \bar{1}, n, j = \bar{1}, n$ .

#### 6.3.1 Lie derivatives and Lie brackets

Given a scalar function  $h(x)$  and a vector field  $f(x)$ , the definition of a new scalar function  $L_f h$ , is called Lie derivatives (or simply, the derivatives) of  $h$  with respect to  $f$ .

**Definition 6.1:** Let  $h : \mathfrak{R} \rightarrow \mathfrak{R}$  be a smooth scalar function, and  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}^n$  be a smooth vector field on  $\mathfrak{R}^n$ , in addition the Lie derivative of  $h$  with respect to  $f$  is a scalar function defined by:

$$L_f h = \bar{V}hf \quad (6.9)$$



Therefore the Lie derivative  $L_f h$  is straightforwardly the directional derivative of  $h$  along the direction of the vector  $f$ . Reiterated the definition of Lie derivatives can be:

$$L_f^0 h = h \quad (6.10)$$

$$L_f^i h = L_f(L_f^{i-1} h) = \bar{V}(L_f^{i-1} h)f \quad \text{for: } i = 1, 2, \dots$$

In the same manner, if  $g$  is a new vector field, in addition the scalar function  $L_g L_f h(x)$  is:

$$L_g L_f h = \bar{V}(L_f h)g \quad (6.11)$$

The relevance of Lie derivatives can be appreciated in a dynamic system by considering a system of single output:

$$\dot{x} = f \quad (6.12)$$

$$y = h$$

The output's derivatives are:

$$\dot{y} = \frac{\partial h}{\partial x} \dot{x} = L_f h \quad (6.13)$$

$$\ddot{y} = \frac{\partial [L_f h]}{\partial x} \dot{x} = L_f^2 h \quad (6.14)$$

And so on, if  $V$  is a Lyapunov function candidate for the system, the derivative of  $\dot{V}$  can be expressed as  $L_f V$ .

Another important concept of this section is the Lie bracket, its definition is (Jara, 2003):

**Definition 6.2:** Let consider two vector fields  $f$  and  $g$  on  $\mathfrak{R}^n$ . The Lie bracket is considered as the third vector field of  $f$  and  $g$  described by:

$$[f, g] = \bar{V}gf - \bar{V}fg \quad (6.15)$$

The lie bracket  $[f, g]$  is generally written as  $ad_f g$  (where  $ad$  means adjoint). Reiterated Lie brackets can then be expressed as:

$$ad_f^0 g = g \quad (6.16)$$

$$ad_f^i g = [f, ad_f^{i-1} g], \text{ for } i = 1, 2, \dots \quad (6.17)$$

### 6.3.2 Diffeomorphisms and state transformation

The concept of diffeomorphism can be viewed as mean of transforming a nonlinear system into another nonlinear system by means of new set of states. It is defined as (Farrell and Polycarpou, 2006):

**Definition 6.3** (Fadali, 2011): A function  $\phi: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ , defined in a region  $\Omega$ , is called a diffeomorphism if it is smooth, and its inverse  $\phi^{-1}$  exists and is smooth.

If the region  $\Omega$  is the entire space  $\mathfrak{R}^n$ , then  $\phi(x)$  is called a global diffeomorphism. It is scarce to find global diffeomorphisms, and therefore one often looks for local diffeomorphism. Local diffeomorphism is for transformations defined only in a finite neighbourhood of a given point.

**Lemma 6.1** (Fadali, 2011): Let  $\phi(x)$  be a smooth function defined in the region  $\Omega$  in  $\mathfrak{R}^n$ . If the Jacobian matrix  $\bar{V}\phi$  is non-singular at a point  $x = x_0$  of  $\Omega$ , then  $\phi(x)$  defines a local diffeomorphism in a sub region of  $\Omega$ .

A diffeomorphism has the particularity to be utilized to convert a nonlinear system into another nonlinear system in the form of new set of states. The dynamic system expressed as follow is considered:

$$\dot{x} = f(x) + g(x)u, x(0) = x_0 \quad (6.18)$$

$$y = h(x)$$

And a new set of state defined by:

$$z = \phi(x) \quad (6.19)$$

Differentiation of  $z$  yields:

$$\dot{z} = \frac{\partial \phi}{\partial x} \dot{x} = \frac{\partial \phi}{\partial x} [f(x) + g(x)u] \quad (6.20)$$

The new representation of state space of the system can be written as:

$$\dot{z} = f^*(z) + g^*(z)u \quad (6.21)$$

$$y = h^*(z)$$

where:  $f^*(z)$  = state function,  $g^*(z)$  = control function,  $h^*(z)$  = output function.

### 6.3.3 Frobenius theorem

The Frobenius theorem is an important tool of feedback linearization for  $n^{\text{th}}$  order nonlinear system providing necessary and sufficient conditions for solvability of partial differential equations.

**Definition 6.4:** Let  $f_1, f_2, \dots, f_m$  be a set of linearly distinct vector fields. The set is fully integrable if, and only if, it is involutive.

### 6.3.4 Relative degree of a system

The relative degree of a system is the number of differentiation needed to be taken on the output  $y$  in order to create a clear relationship between the output  $y$  and the input  $u$ . For example, if the output  $y$  is necessary to be differentiated  $\rho$  times, in order a clear relationship between the output  $y$  and the input  $u$  to be obtained, the relative degree of the system is said to be  $\rho$ . This terminology is consistent with the notation of relative degree in linear system which is the excess of poles over zeros. It is possible to prove that for any system of order  $n$  that is controllable, it will require to differentiate any output at most  $n$  times for the control input to be found. 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), (Sun, 2002) and (Chopra, 2008).

### 6.3.5 Linearization techniques

#### 6.3.5.1 Input-state linearization

Slotine and Li, 1991 define the input-state linearization as follows:

**Definition 6.5:** A single-input nonlinear system expressed by (6.18), with

$f(x)$  and  $g(x)$  considered to be smooth vector fields on  $\mathfrak{R}^n$ , is considered to be linearizable in input-state if there is the existence of a region  $\Omega$  in  $\mathfrak{R}^n$  a diffeomorphism  $\phi: \Omega \rightarrow \mathfrak{R}^n$ , and a control law for nonlinear feedback:

$$u = \alpha(x) + \beta(x)v \tag{6.22}$$

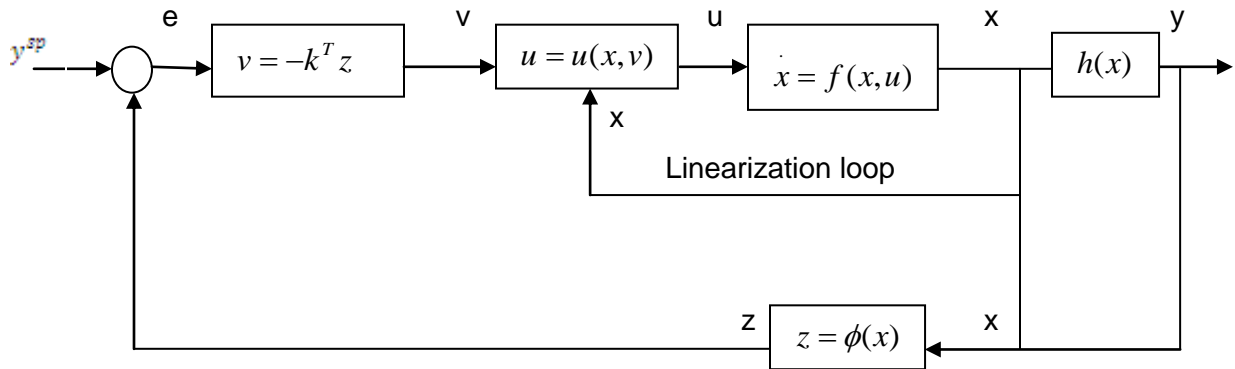
such that the newly derived state variables  $z = \phi(x)$  and the new input  $v$  fulfil a linear time

invariant relation represented as:  $\dot{z} = Az + bv$ .

$$\text{Where: } A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}; \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}; \quad z \text{ is the linearized state and } u \text{ is the linear control law.}$$

**Lemma 6.2** (Doyle, 1996): An  $n^{\text{th}}$  order nonlinear system is input state linearizable if, and only if, there is an existing scalar function  $z_1(x)$  such that the system's input-output linearization with  $z_1(x)$  as output function has relative degree  $n$ .

Figure 6.2 shows the closed loop diagram of an input-state linearization.



**Figure 6.2: Input-state linearization control system**

### 6.3.5.1.1 Input-state linearization procedure

To achieve input-state linearization, the following procedure is recommended (Doyle, 1997) and (Han, 2009):

- Build the vector fields  $g, ad_f g, \dots, ad_f^{n-1} g$  for the specific system
- Examine whether the controllability and involutivity requirements are fulfilled.
- If both are fulfilled, determine the first state  $z_1$  of relative degree  $n$  from the equation:

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

$$\bar{V}z_1 a d_f^i g = 0, \quad i = 0, \dots, n-2 \quad (6.23)$$

$$\bar{V}z_1 a d_f^{n-1} g \neq 0$$

- Calculate the state transformation  $z(x) = [z_1 \quad L_f z_1 \quad \dots \quad L_f^{n-1} z_1]^T$  and the input transformation  $u = \alpha(x) + \beta(x)v$ , with:

$$\alpha(x) = -\frac{L_f^n z_1}{L_g L_f^{n-1} z_1} \quad (6.24)$$

$$\beta(x) = \frac{1}{L_g L_f^{n-1} z_1}$$

### 6.3.5.2 Input output linearization

A single input nonlinear system expressed as:

$$\dot{x} = f(x) + g(x)u, \quad x(0) = x_0 \quad (6.25)$$

$$y = \bar{h}(x)$$

is considered.

It can be seen that the output  $y$  is not directly related to  $u$  through the nonlinear state equations and the state variable  $x$ . The relationship between the output  $y$  and the input  $u$  can be created by the continuous differentiation of output function  $y$  until the input  $u$  is found. After that, design  $u$  to make sure that the nonlinearities are nullified.

#### 6.3.5.2.1 Control design procedure for the input-output linearization

To achieve input-output linearization of a system, the following procedure has to be followed:

- Continuous differentiation of the output  $y$  until the input  $u$  is found.
- Select  $u$  to nullify nonlinearities and achieve convergence of tracking.
- Analyze the internal dynamics of the system to find out about its stability.

The internal dynamics is a portion of the system dynamics that has been made “unobservable” in the input-output linearization method (Isidori, 1984). It is called internal dynamics due to the fact that the external input-output relationship is not able to see it. The internal dynamics are stable, if the states remain bounded during tracking or stabilizing. If this condition is not met, 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 components.

## 6.4 Design of a feedback linearizing controller for magnetic levitation

### 6.4.1 Relative degree of the system

The nonlinear model of the magnetic levitation system was developed in Chapter 3 and was defined as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ g - \frac{k}{m} \frac{x_3^2}{x_1^2} \\ -\frac{R}{L} x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u, x(0) = x_0 \quad (6.26)$$

$$y = C[x_1 \quad x_2 \quad x_3]^T = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

In order to determine the system's relative degree, the derivatives of the output are taken until the input appears explicitly.

- First differentiation of the output:

$$y = x_1$$

$$\dot{y} = L_f h(x) + L_g h(x)u \quad (6.27)$$

$$L_f h(x) = x_2$$

$$L_g h(x)u(t) = 0.u$$

$$\dot{y} = x_2 + 0.u$$

The first differentiation shows that there is no straightforward relationship between the input and the output.

- Second differentiation of the output:

$$\dot{y} = x_2$$

$$\ddot{y} = L_f^2 h(x) + L_g L_f h(x)u \quad (6.28)$$

$$L_f^2 h(x) = g - \frac{k}{m} \left( \frac{x_3}{x_1} \right)^2$$

$$L_g L_f h(x) u = 0 \cdot u$$

$$\ddot{y} = g - \frac{k}{m} \left( \frac{x_3}{x_1} \right)^2 + 0 \cdot u$$

A third differentiation is required as the input and output are not directly related.

- Third differentiation of the output:

$$\ddot{y} = g - \frac{k}{m} \left( \frac{x_3}{x_1} \right)^2 + 0 \cdot u$$

$$\dddot{y} = L_f^3 h(x) + L_g L_f^2 h(x) u \quad (6.29)$$

$$L_f^3 h(x) = \frac{2k}{m} \left( \frac{x_2 x_3^2}{x_1^3} \right) + \frac{2kR}{mL} \left( \frac{x_3^2}{x_1^2} \right)$$

$$L_g L_f^2 h(x) u = -\frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right) u$$

$$\dddot{y} = \frac{2k}{m} \left( \frac{x_2 x_3^2}{x_1^3} \right) + \frac{2kR}{mL} \left( \frac{x_3^2}{x_1^2} \right) - \frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right) u$$

The third differentiation shows a direct relationship between the input and output, at the same time the system's relative degree is 3, because the output has to be differentiated three times to find its relationship with the input.

At the equilibrium  $x_{e1}$ , the term  $L_2 L_f^2 h(x) u \neq 0$  and for the relative degree to remain correctly defined, the system is allowed to operate in a region of state space bounded by:

$$x_{e1} > 0 \text{ and } x_{e3} > 0. \quad (6.30)$$

These are realistic restrictions because if  $x_{e1} \leq 0$  it means that the levitated steel ball touches the coil. And if  $x_{e3} < 0$ , the resulting current would be negative.

This case is of particular interest as the relative degree of the system is identical to the order of the system ( $r = n$ ), therefore the input-output linearization implementation actually yields input-state linearization.

The next phase of the nonlinear linearizing controller design is to check if the system is input-state linearizable by constructing the vector fields and checking of the controllability and involutivity of the system.

#### 6.4.2 Construction of the vector fields for input-state linearization

The construction of the vector fields is done as follows:

- Calculation of vector  $g$  at the equilibrium point  $x_e$ :

$$ad_f^0 g(x_e) = g(x_e) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \frac{1}{L} \end{bmatrix} \quad (6.31)$$

- Calculation of the vector  $ad_f g(x)$ :

$$ad_f g(x) = \frac{\partial g}{\partial x} f - \frac{\partial f}{\partial x} g \quad (6.32)$$

$$\frac{\partial g}{\partial x} f = \begin{bmatrix} 0 \\ g - \frac{k}{m} \left( \frac{x_3}{x_1} \right)^2 \\ -\frac{R}{L} x_3 \end{bmatrix}$$

$$-\frac{\partial f}{\partial x} g = - \begin{bmatrix} 0 & 1 & 0 \\ \frac{2k}{m} \left( \frac{x_3}{x_1^3} \right) & 0 & -\frac{2k}{m} \left( \frac{x_3}{x_1^2} \right) \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ \frac{1}{L} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right) \\ \frac{R}{L^2} \end{bmatrix} \quad (6.33)$$

$$ad_f g(x) = \begin{bmatrix} 0 \\ \frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right) \\ \frac{R}{L^2} \end{bmatrix}$$

- Calculation of vector  $ad_f^2 g(x)$ :



$$ad_f^2 g(x) = [f, ad_f g(x)] = \frac{\partial ad_f g}{\partial x} f - \frac{\partial f}{\partial x} ad_f g$$

$$\frac{\partial ad_f g}{\partial x} f = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{4k}{mL} \left( \frac{x_3}{x_1^3} \right) & 0 & \frac{2k}{mL} \left( \frac{1}{x_2} \right) \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ g - \frac{k}{m} \left( \frac{x_3}{x_1} \right)^2 \\ -\frac{R}{L} x_3 \end{bmatrix} \quad (6.34)$$

$$\frac{\partial f}{\partial x} ad_f g = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2k}{m} \left( \frac{x_3}{x_1^3} \right) & 0 & -\frac{2k}{m} \left( \frac{x_3}{x_1^2} \right) \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right) \\ \frac{R}{L^2} \end{bmatrix}$$

$$ad_f^2 g(x) = \begin{bmatrix} 0 & 0 & 0 \\ -\frac{4k}{mL} \left( \frac{x_3}{x_1^3} \right) & 0 & \frac{2k}{mL} \left( \frac{1}{x_2} \right) \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_2 \\ g - \frac{k}{m} \left( \frac{x_3}{x_1} \right)^2 \\ -\frac{R}{L} x_3 \end{bmatrix} - \begin{bmatrix} 0 & 1 & 0 \\ \frac{2k}{m} \left( \frac{x_3}{x_1^3} \right) & 0 & -\frac{2k}{m} \left( \frac{x_3}{x_1^2} \right) \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right) \\ \frac{R}{L^2} \end{bmatrix}$$

$$ad_f^2 g(x) = \begin{bmatrix} -\frac{2kx_3}{mLx_1^2} \\ -\frac{4kx_2x_3}{mLx_1^3} \\ \frac{R^2}{L^3} \end{bmatrix}$$

Once all the calculations are done, the formation of the vector field  $[g \quad ad_f g \quad ad_f^2 g]$  is done and the calculation of its rank is determined at the equilibrium point  $x_e$ .

$$\text{rank}[g \quad ad_f g \quad ad_f^2 g] = \text{rank} \begin{bmatrix} 0 & 0 & -\frac{2kx_3}{mLx_1^2} \\ 0 & \frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right) & -\frac{4kx_2x_3}{mLx_1^3} \\ \frac{1}{L} & \frac{R}{L^2} & \frac{R^2}{L^3} \end{bmatrix} = 0 \quad (6.35)$$

The rank of the vector field is full; therefore the system is controllable by both input-output and input-state linearizing controllers. On this basis, it could be concluded that all state variables are controllable.

### 6.4.3 Check for involutivity of the system

The involutivity of a system involves the Frobenius theorem which states that a span is considered to be involutive if it is completely integrable. This condition is checked by noting the Lie product of any two column vectors in the span. It is said that the span is involutive if that product can be constructed as a linear combination of those two vectors (Han, 2009).

$$[g(x), ad_f g(x)] = \delta g(x) + \gamma ad_f g(x) \quad (6.36)$$

Where:  $\delta$  and  $\gamma$  are constant.

- Calculation of  $[g(x), ad_f g(x)]$ :

All the Jacobians required for the formation of this vector were calculated in the previous section:

$$[g(x), ad_f g(x)] = \begin{bmatrix} 0 \\ \frac{2k}{mL^2} \left( \frac{1}{x_1^2} \right) \\ 0 \end{bmatrix} \quad (6.37)$$

This equation is the linear combination of  $g(x)$  and  $ad_f g(x)$  as illustrated:

$$[g(x), ad_f g(x)] = \begin{bmatrix} 0 \\ \frac{2k}{mL^2} \left( \frac{1}{x_1^2} \right) \\ 0 \end{bmatrix} = -\frac{R}{Lx_3} \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} + \frac{1}{x_3} \begin{bmatrix} 0 \\ \frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right) \\ \frac{R}{L} \end{bmatrix} \quad (6.38)$$

In this particular for  $\delta$  and  $\gamma$  can be identified as:

$$\delta = \frac{R}{Lx_3} \quad (6.39)$$

$$\gamma = \frac{1}{x_3}$$

The second condition is true which confirms the existence of a linearizing coordinate transformation and nonlinear feedback possible.

#### 6.4.4 Coordinate transformation

According to theory (Fadali, 2011) and (Hedrick and Gerard, 2005), the new coordinates of the system have the following representation:

$$z = \phi(x) = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \\ y \end{bmatrix} \quad (6.40)$$

The new system's states are: position, velocity and acceleration. The new states of the system expressed in terms of the dynamics of the system are:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \phi(x) = \begin{bmatrix} x_1 \\ x_2 \\ g - \frac{k}{m} \left( \frac{x_3}{x_1} \right)^2 \end{bmatrix} \quad (6.41)$$

The first derivative of the new system defines the input  $u$  (nonlinear linearizing controller) and allows the definition of the new input  $v$  (linear controller):

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} \dot{y} \\ \ddot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} x_2 \\ g - \frac{k}{m} \left( \frac{x_3}{x_1} \right)^2 \\ \frac{2k}{m} \left( \frac{x_2 x_3^2}{x_1^3} \right) + \frac{2kR}{mL} \left( \frac{x_3^2}{x_1^2} \right) - \frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right) u \end{bmatrix} \quad (6.42)$$

The last derivative  $\ddot{y}$  is the new input to the linear system and can be rewritten as:

$$\dot{z} = Az + bv \quad (6.43)$$

Where:  $A$  is a  $3 \times 3$  matrix and  $b$  is a  $3 \times 1$  vector.

$$\dot{z} = \begin{bmatrix} z_2 \\ z_3 \\ v \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \quad (6.44)$$

Equation (6.44) is in a special format called Bruvonsky form. The last row is of special interest because it is the nonlinear feedback controller. Based on that, it is possible to find a direct relationship between the input  $u$  and the linear system's input  $v$ :

$$v = L_f^3 h(x) + L_g L_f^2 h(x)u \quad (6.45)$$

Solving this equation for  $u$  would result to the determination of the nonlinear linearizing controller expression:

$$u = \frac{-L_f^3 h(x) + v}{L_g L_f^2 h(x)} = \frac{\frac{2k}{m} \left( \frac{x_2 x_3^2}{x_1^3} \right) + \frac{2kR}{mL} \left( \frac{x_3^2}{x_1^2} \right) - v}{\frac{2k}{mL} \left( \frac{x_3}{x_1^2} \right)} = \alpha(x) + \beta(x)v \quad (6.46)$$

Where:

$$\alpha(x) = \frac{-L_f^3 h(x)}{L_g L_f^2 h(x)} \quad (6.47)$$

$$\beta(x) = \frac{1}{L_g L_f^2 h(x)}$$

The linear control  $v$  is unknown in the Equation (6.46). It has to be designed using the obtained linearized system (6.44).

#### 6.4.5 Linear quadratic regulator design for the linearized model

The model of the linearized system by the input-state nonlinear controller is selected as the Bruvonsky model as it is directly related to the new states of the system. The task in hand is to design an Integral Linear Quadratic Regulator (ILQR) to stabilize the closed loop system.

The first step is to check the controllability and observability of the Bruvonsky model:

- Controllability:

$$\text{rank} \bar{C} = \text{rank} \begin{bmatrix} b & Ab & A^2 b \end{bmatrix} = 3, \text{ where } \bar{C} \text{ is the controllability matrix.}$$

- Observability:

$$\text{rank} \bar{O} = \text{rank} \begin{bmatrix} C^T & C^T A & C^T A^2 \end{bmatrix} = 3, \text{ where } \bar{O} \text{ is the observability matrix.}$$

The linear reference model is controllable and observable, the next step is to design the LQR controller to stabilize the system.

The ILQR design technique is discussed in Chapter 5. The main idea is to design the gain matrix  $\bar{H}$  for the linear state feedback control law  $u = -\bar{H}\bar{x}$  which is found by reducing the performance index in the form of:

$$J = \int_0^\infty \left( \bar{x}^T \bar{Q} \bar{x} + u^T \bar{R} u \right) dt \quad (6.48)$$

Where  $\bar{Q} \in \mathfrak{R}^{4 \times 4}$  is a positive definite (or positive semi-definite) Hermitian or real symmetric matrix.  $\bar{R} \in \mathfrak{R}^1$  is a positive definite Hermitian or real symmetric matrix.  $\bar{Q}$  and  $\bar{R}$  are matrices made of weighting parameters used to penalize certain states or control inputs and  $y^{sp}$  is given different heights values to check the ability of the closed loop system to follow a particular input. The model of the extended system needed for design of the integral optimal control is:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \\ \dot{\varepsilon} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \varepsilon \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} v = \bar{A}z + \bar{b}v, \quad \bar{A} \in \mathfrak{R}^{4 \times 4}, \bar{b} \in \mathfrak{R}^4 \quad (6.49)$$

$$\varepsilon = y^{sp} - y$$

$$\dot{\varepsilon} = -\dot{y} = -\dot{z}_1 = -z_2$$

The check of controllability and observability of the extended Brunovsky model is:

- Controllability:

$$\text{rank} \bar{C} = \text{rank} \begin{bmatrix} b & Ab & A^2b & A^3b \end{bmatrix} = 4, \text{ where } \bar{C} \text{ is the controllability matrix.}$$

- Observability:

$$\text{rank} \bar{O} = \text{rank} \begin{bmatrix} C & CA & CA^2 & CA^3 \end{bmatrix}^T = 4, \text{ where } \bar{O} \text{ is the observable matrix.}$$

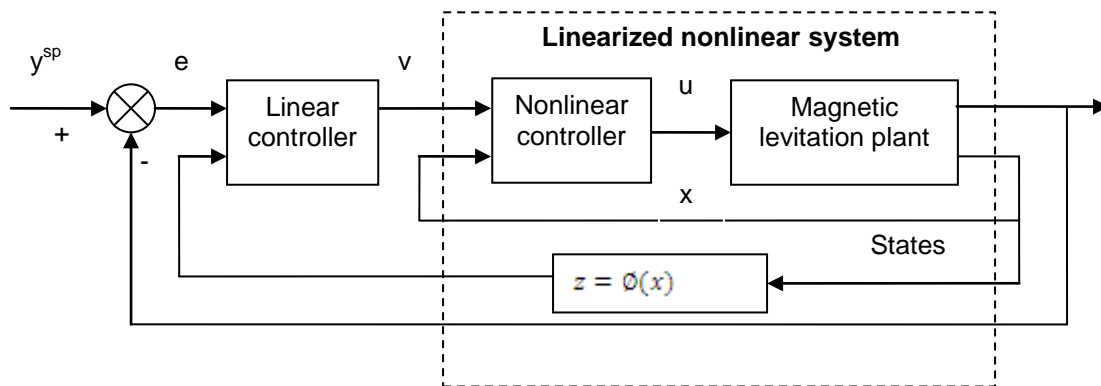
The extended Brunovsky model is controllable and observable therefore an Integral Linear Quadratic Regulator can be designed to make the magnetic levitation system stable.

In MATLAB, the command “lqr” is used to solve the continuous-time, linear, quadratic regulator problem. This function computes the feedback gain matrix  $\bar{H}$ .  $\bar{Q}$  and  $\bar{R}$  are weighting matrices selected to control the magnetic levitation system. Table 6.1 shows the parameters chosen to achieve integral linear control of the linearized nonlinear closed loop system. The simulations are performed for the same initial conditions of the ball for various set-points of the ball position. The matrix  $\bar{R}$  of the criterion (6.48) is selected to be equal to 1 and the matrix  $\bar{Q}$  is changed for every set-point in order to receive the best behaviour of the ball position. The parameters of the calculated linear integral controller are given in Table 6.1.

Figure 6.3 represents the structure of the block diagram of the input-state linearization control of the magnetic levitation system.

**Table 6.1: Parameters of simulation**

Set points	Initial conditions	Matrix $\bar{Q}$	Matrix $\bar{R}$	Feedback controller gain $\bar{H}$
0.55m	[0.2 0 0.894 0]'	$\begin{bmatrix} 10^6 & 0 & 0 & 0 \\ 0 & 2500 & 0 & 0 \\ 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$	1	[3163.9 523.3 39.3 -10]
0.6m	[0.2 0 0.894 0]'	$\begin{bmatrix} 80 \times 10^6 & 0 & 0 & 0 \\ 0 & 9300 & 0 & 0 \\ 0 & 0 & 750 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$	1	[8945.4 967.7 51.8 -10]
0.75m	[0.2 0 0.894 0]'	$\begin{bmatrix} 10^6 & 0 & 0 & 0 \\ 0 & 3500 & 0 & 0 \\ 0 & 0 & 350 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$	1	[3163.8 484.9 36.3 -10]
0.85	[0.2 0 0.894 0]'	$\begin{bmatrix} 10^6 & 0 & 0 & 0 \\ 0 & 8300 & 0 & 0 \\ 0 & 0 & 350 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}$	1	[3163.9 488.7 36.4 -10]



**Figure 6.3: Structure of the input-state/input-output linearization of the magnetic levitation system**

## 6.5 Simulation

The simulation is done in Matlab/Simulink environment. The closed-loop diagram based on input-state linearization method is shown in Figure 6.4. This closed loop diagram is made of four important subsystems:

- Linear controller (Figure 6.5)
- Nonlinear linearizing controller (Figure 6.6)
- Magnetic levitation model (Figure 3.4 of Chapter 3)
- State transformation model (Figure 6.7)

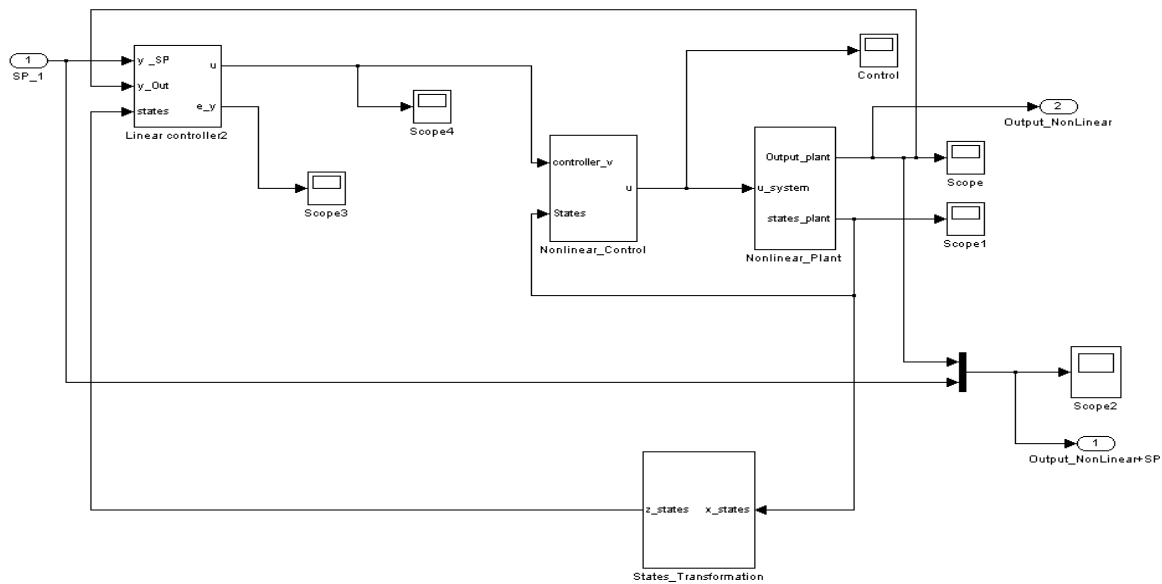


Figure 6.4: Simulink diagram of the input-state linearization method

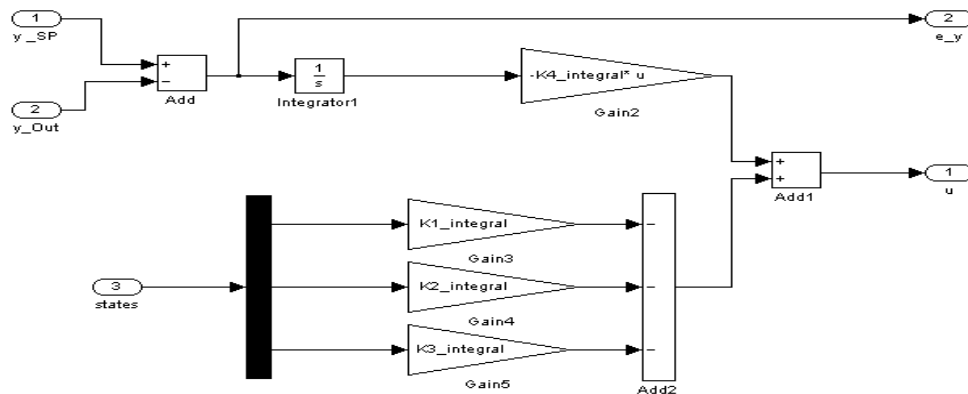
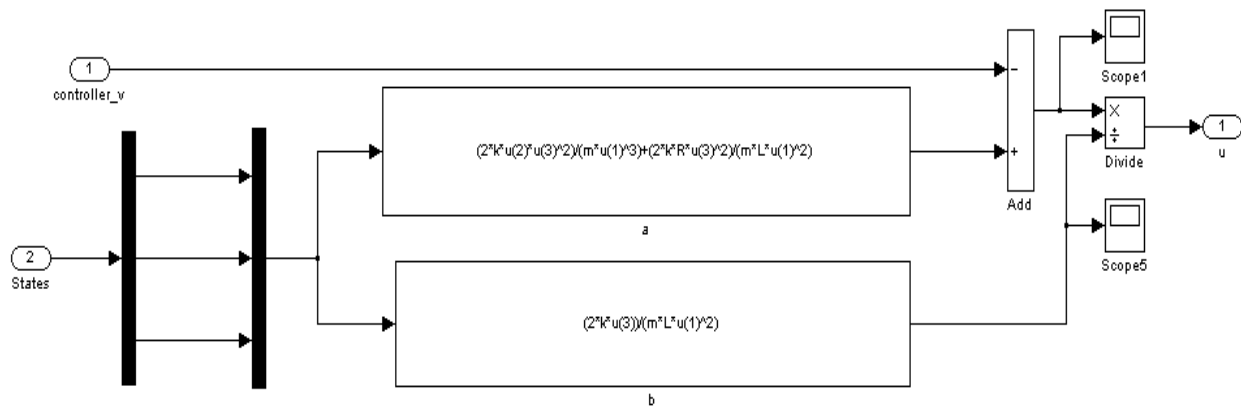
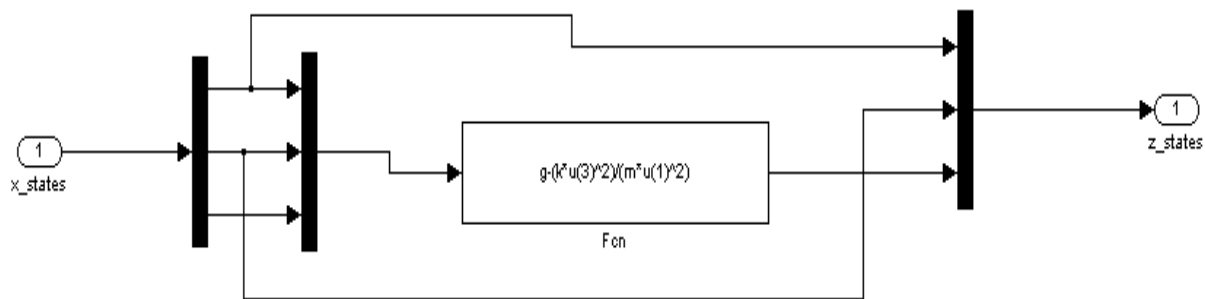


Figure 6.5: Simulink block diagram of the linear state feedback integral controller



**Figure 6.6: Nonlinear linearizing controller structure**



**Figure 6.7: Input-state transformation function**

The parameters of the simulation are found in Matlab file named “*yohan\_linearization.m*”, and the program is given in the Appendix A.5. The corresponding Simulink model “*Linearization\_maglev.mdl*” is shown in Figure 6.4.

### 6.5.1 Simulation results

The simulation results are presented in this section. The results are grouped according to Table 6.2 and show the following trajectories:

- Linear controller signal
- Error signal between the set point and the nonlinear plant output
- State of the velocity of the ball
- Nonlinear linearizing controller signal



- Position of the ball under linear and linearizing controls of the nonlinear plant model.
- Set-point  $y^{SP}=0.55[m]$

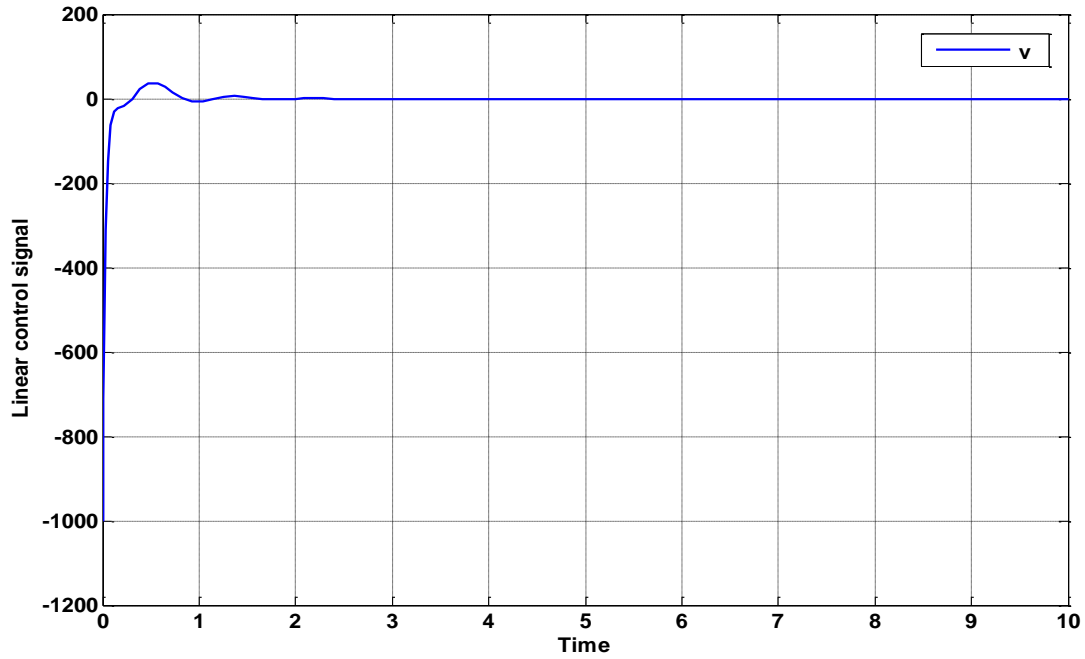


Figure 6.8: Linear control signal when the set point is at 0.55[m]

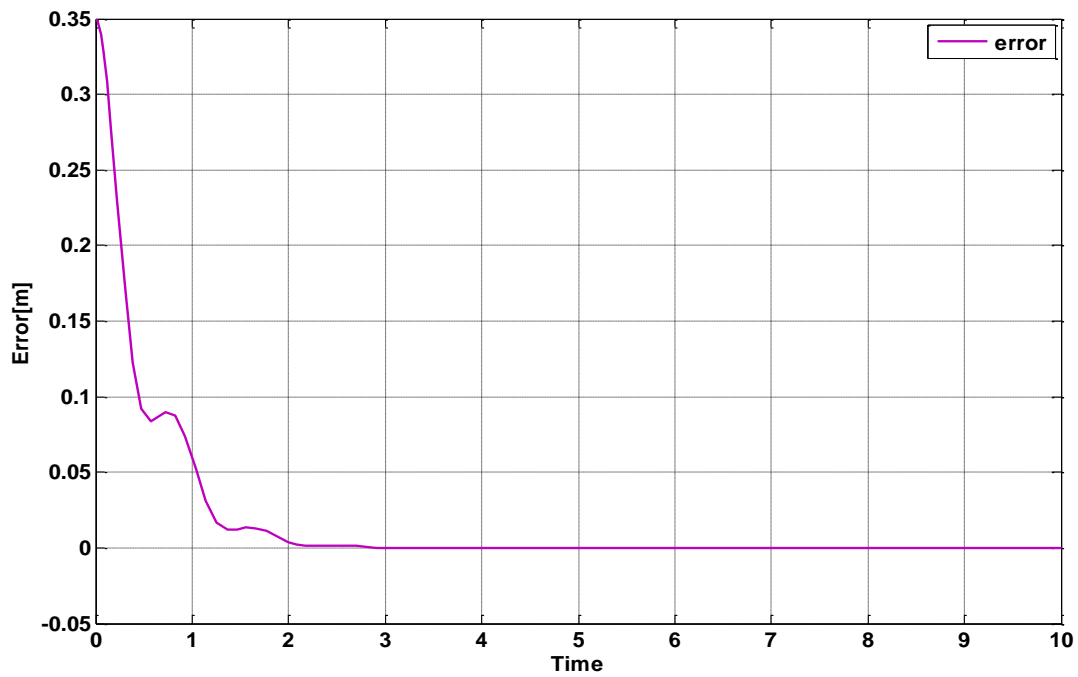


Figure 6.9: Error signal between the set point and the nonlinear plant output when the set point is 0.55[m]

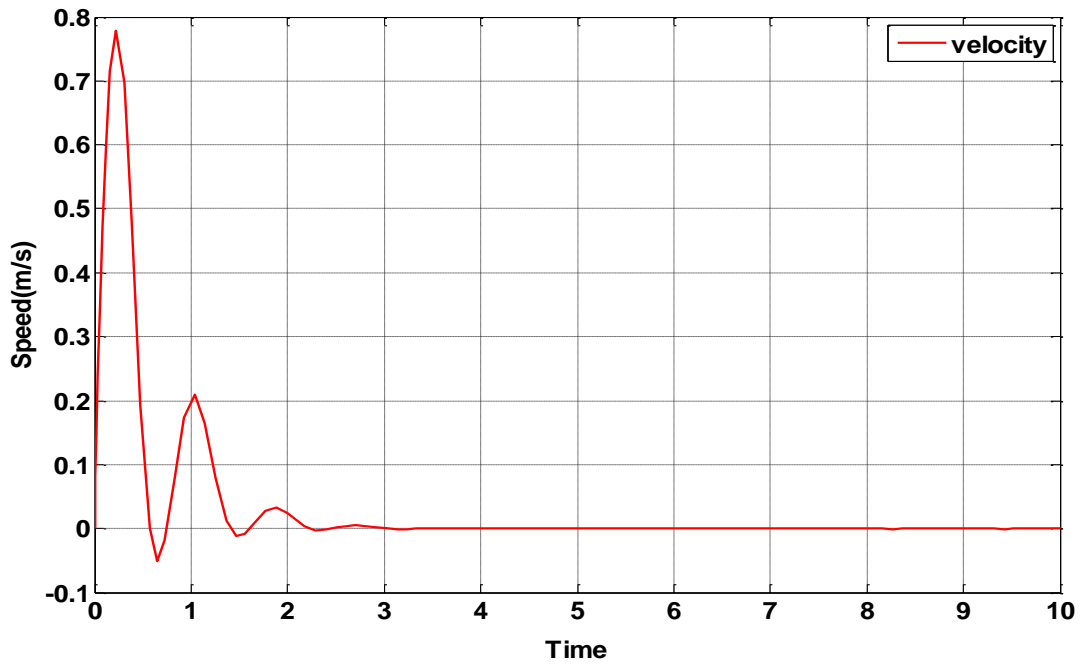


Figure 6.10: Velocity of the ball when the set point is 0.55[m]

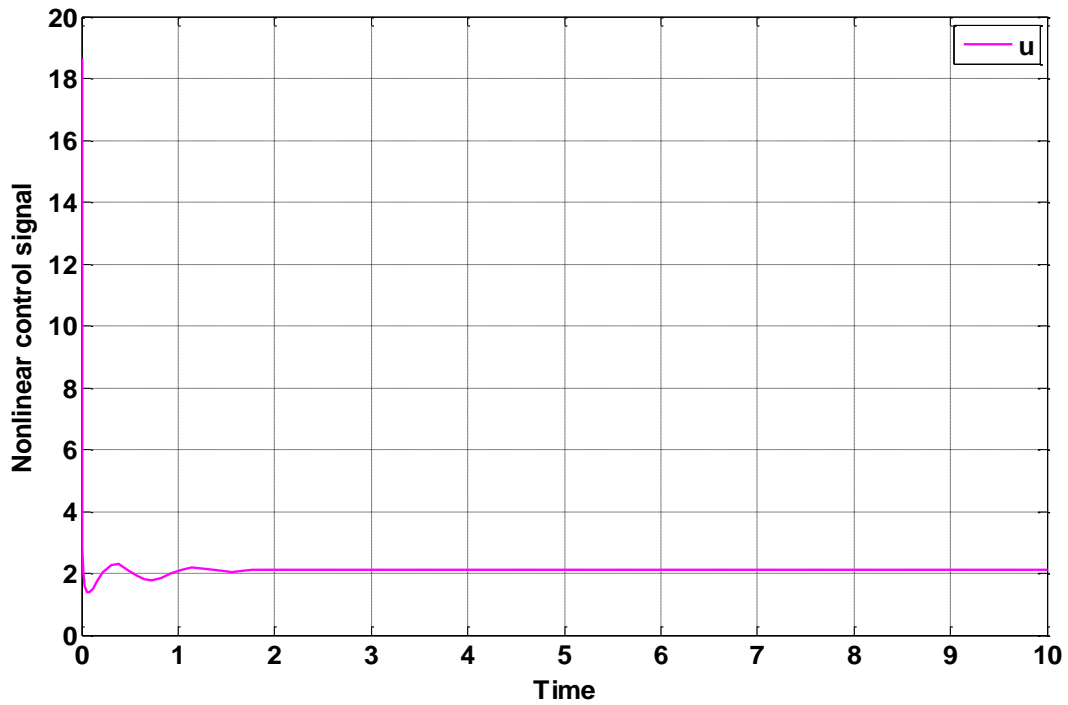


Figure 6.11: Nonlinear linearizing control signal when the set point is 0.55[m]

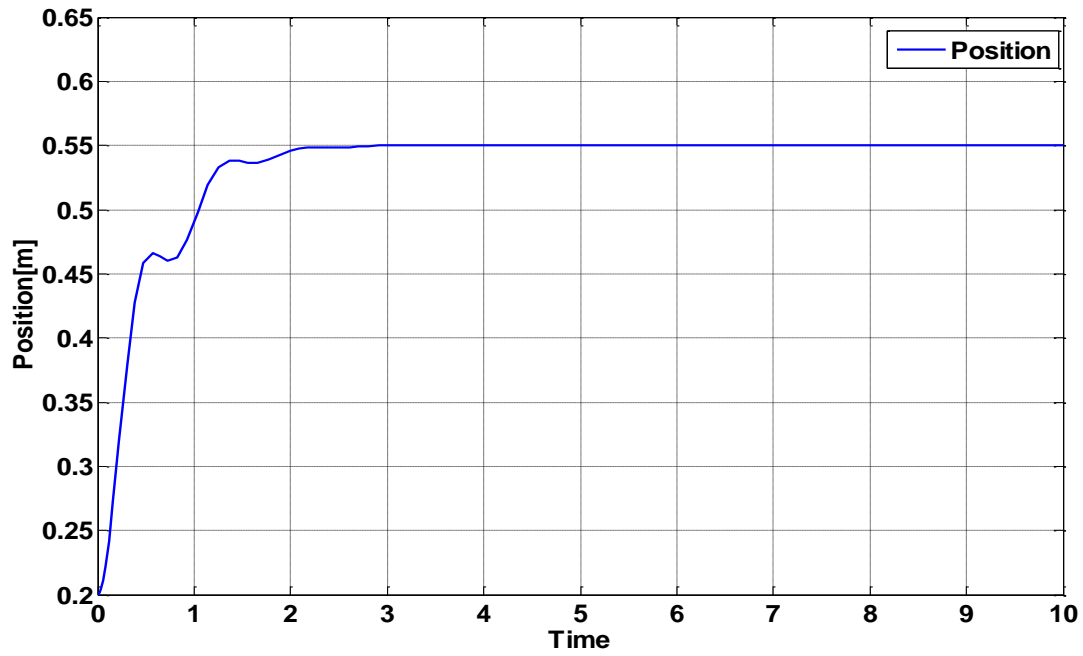


Figure 6.12: Nonlinear system response under linear and linearizing controls when the set-point is 0.55[m]

- Set-point  $y^{sp}=0.6[m]$ :

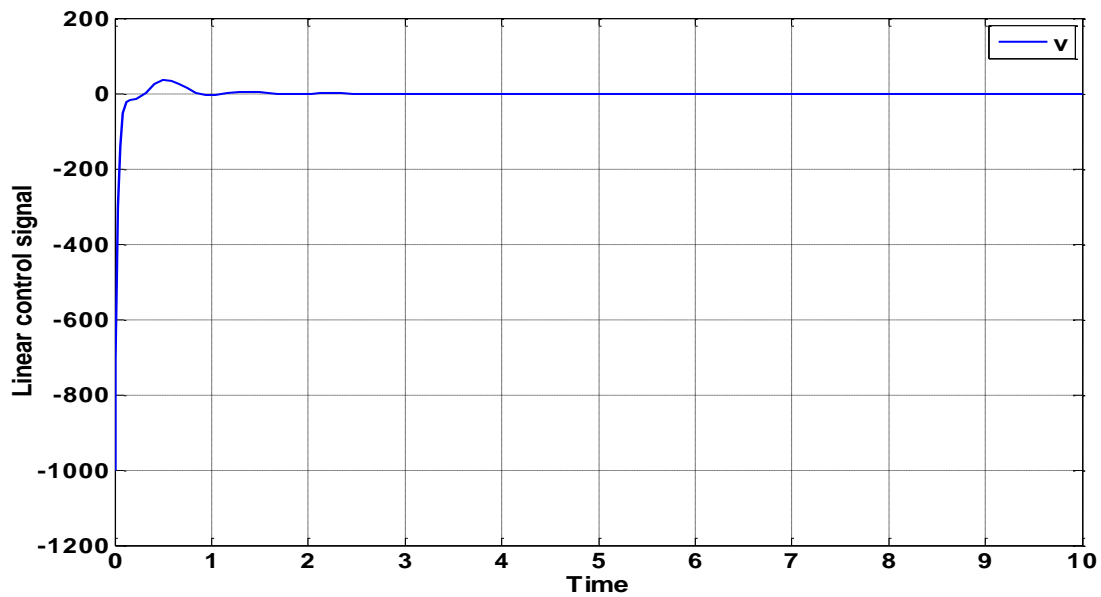


Figure 6.13: Linear control signal when the set point is 0.6[m]

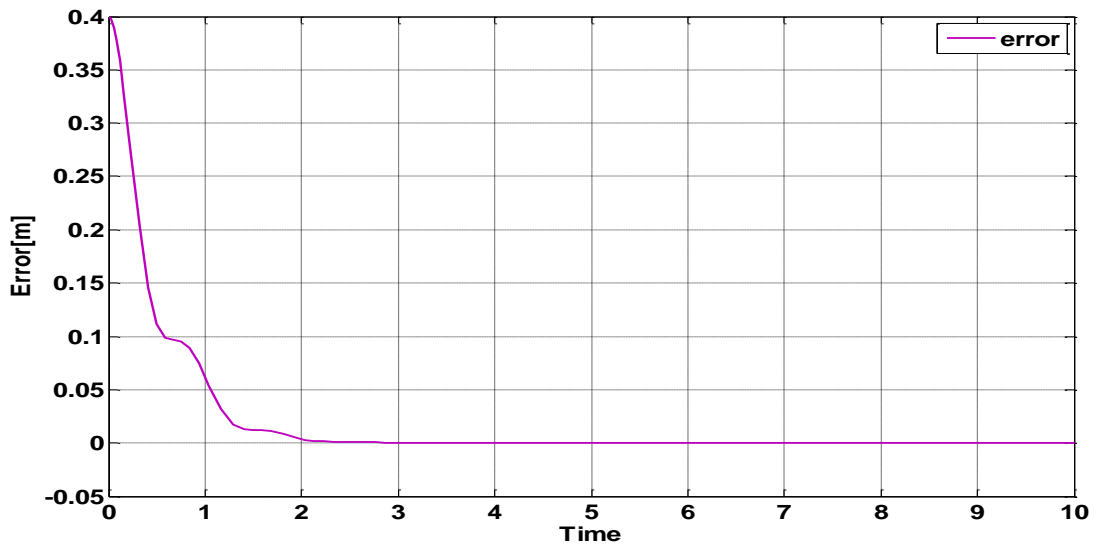


Figure 6.14: Error signal between the set point and the nonlinear plant output when the set point is 0.6[m]

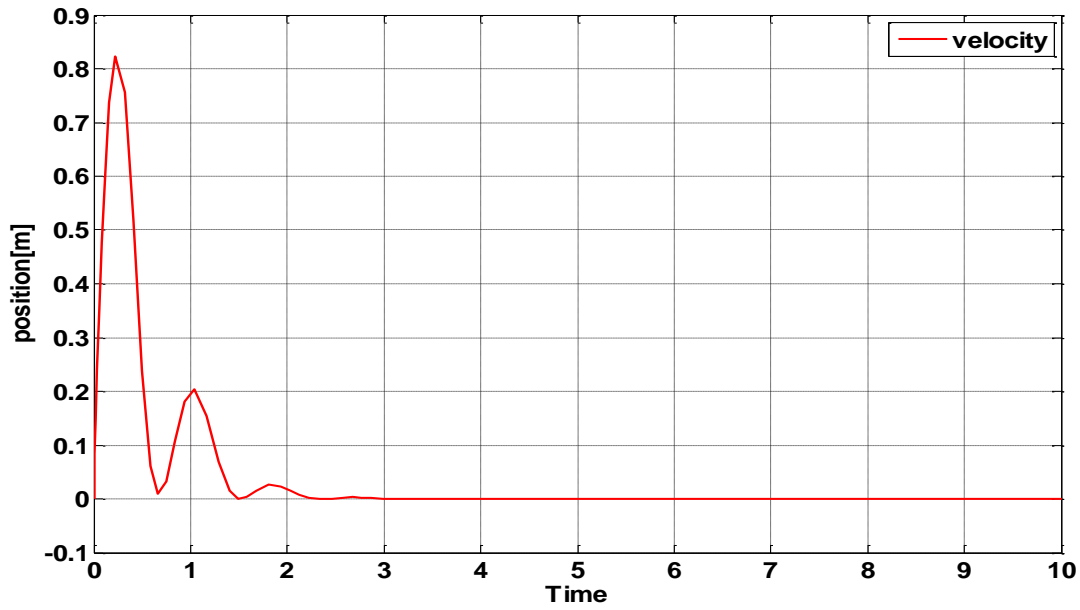


Figure 6.15: Velocity of the ball when the set point is 0.6[m]

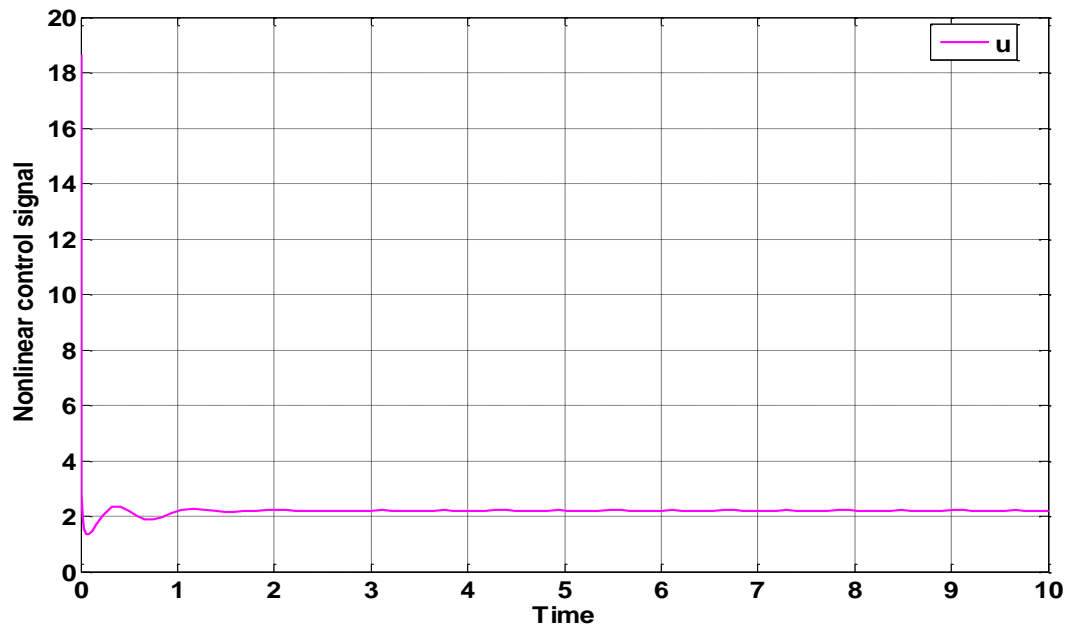


Figure 6.16: Nonlinear linearizing control signal when the set point is 0.6[m]

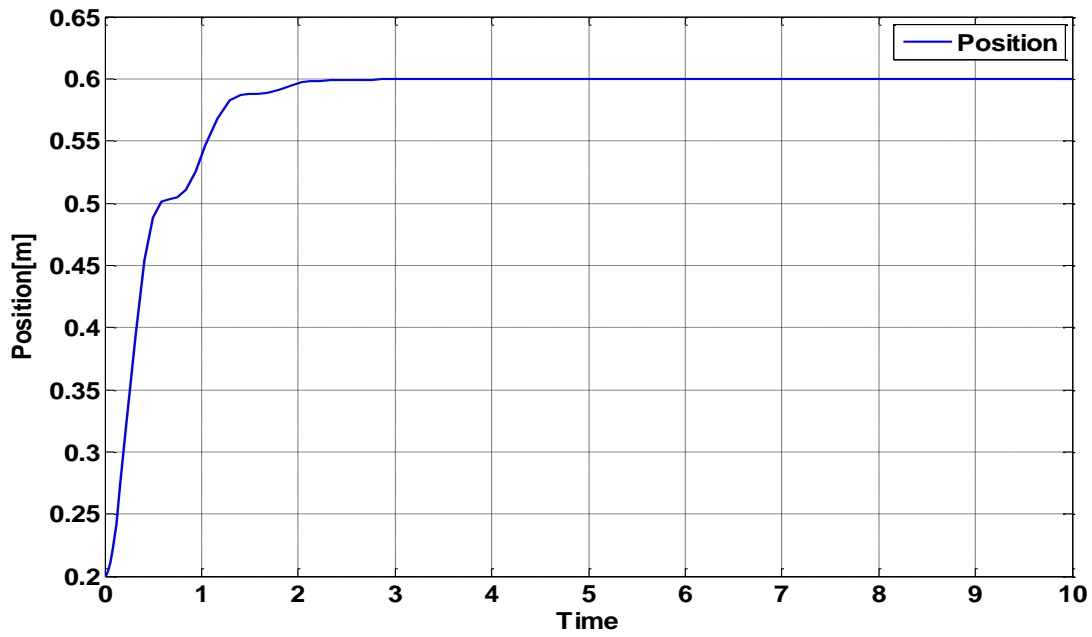


Figure 6.17: Nonlinear system response under linear and linearizing controls when the set-point is 0.6[m]

- Set-point  $y^{sp}=0.75[m]$ :

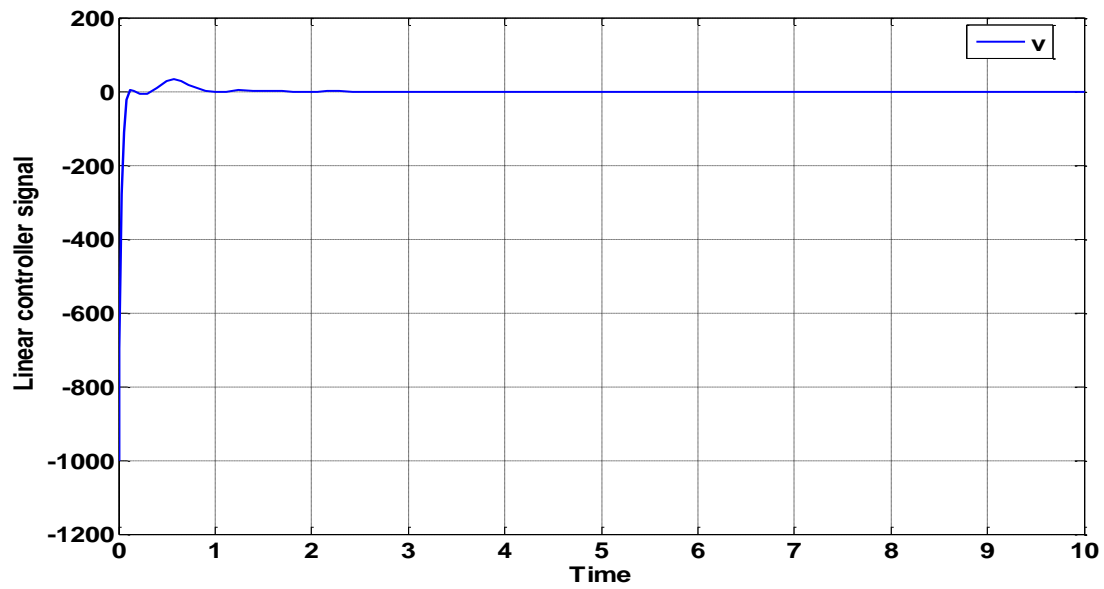


Figure 6.18: Linear control signal when the set point is 0.75[m]

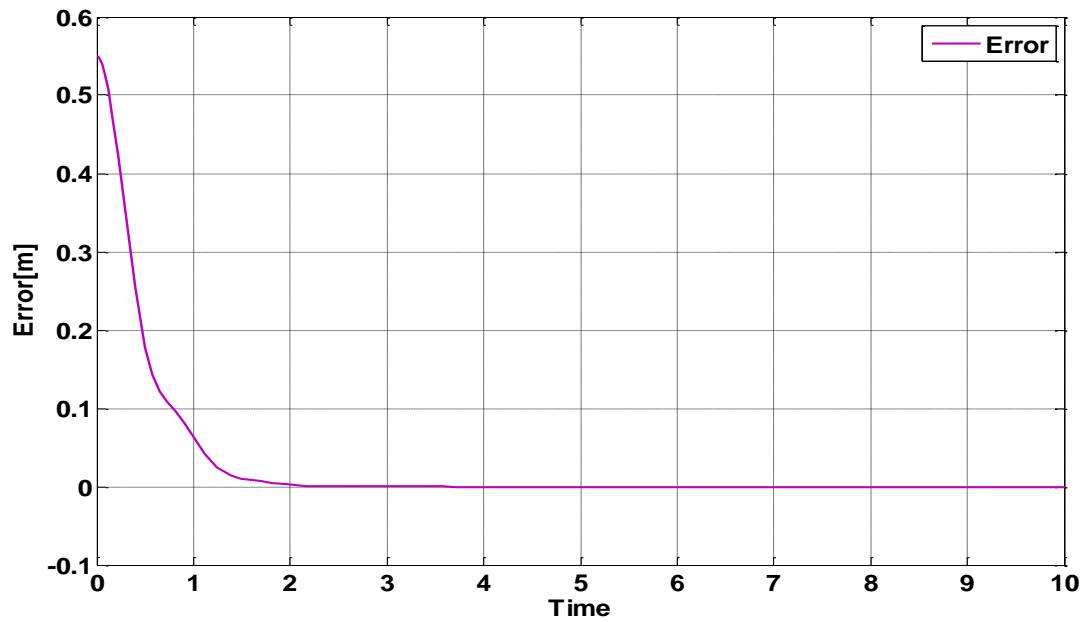


Figure 6.19: Error signal between the set point and the nonlinear plant output when the set point is 0.75[m]

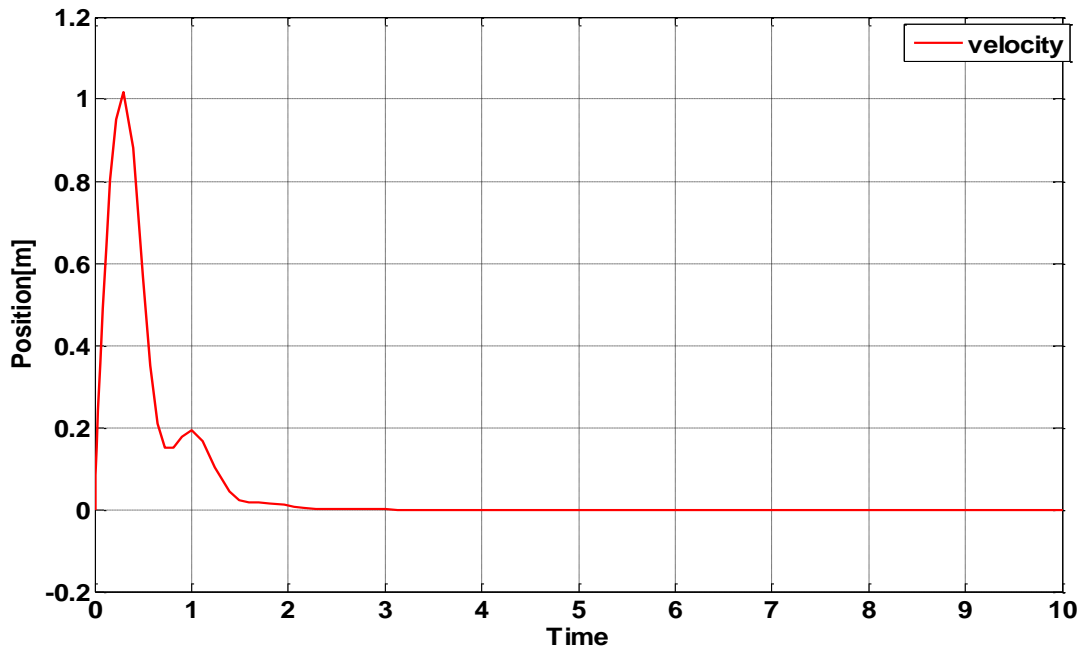


Figure 6.20: Velocity of the ball when the set point is 0.75[m]

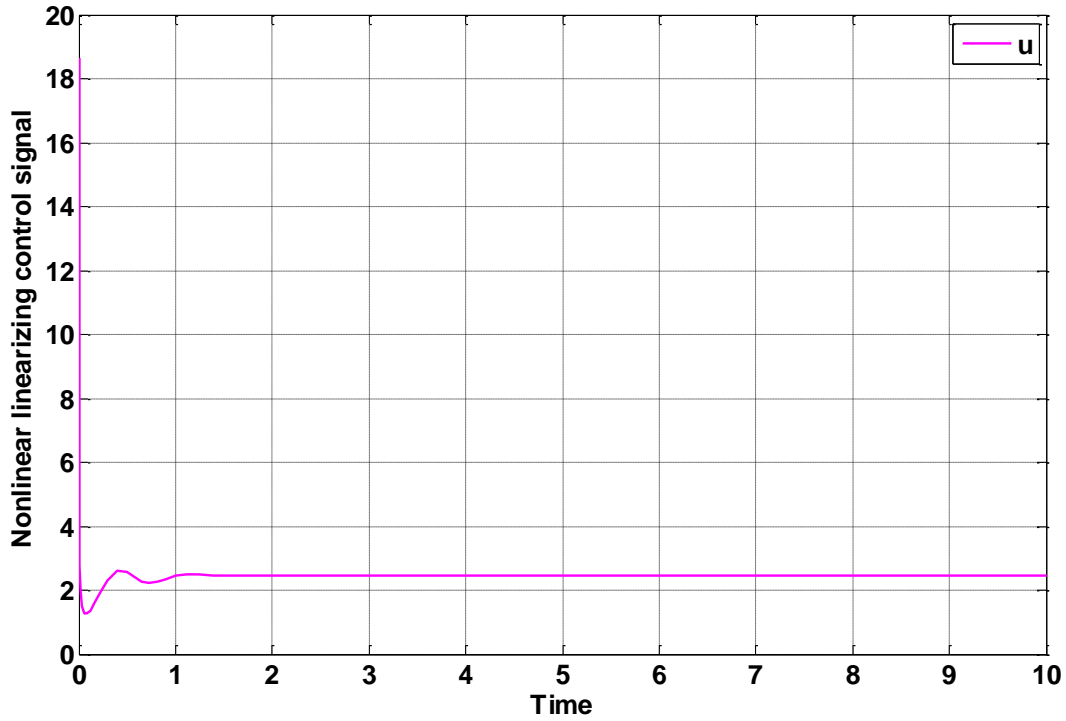


Figure 6.21: Nonlinear linearizing control signal when the set point is 0.75[m]

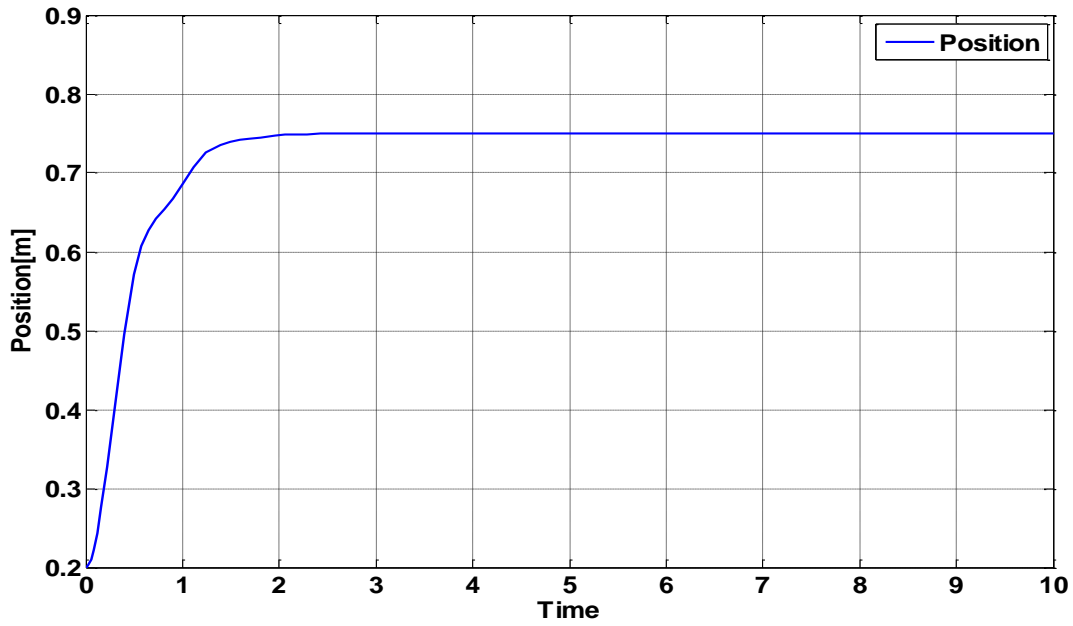


Figure 6.22: Nonlinear system response under linear and linearizing controls when the set-point is 0.75[m]

- Set-point  $y^{sp}=0.85[m]$ :

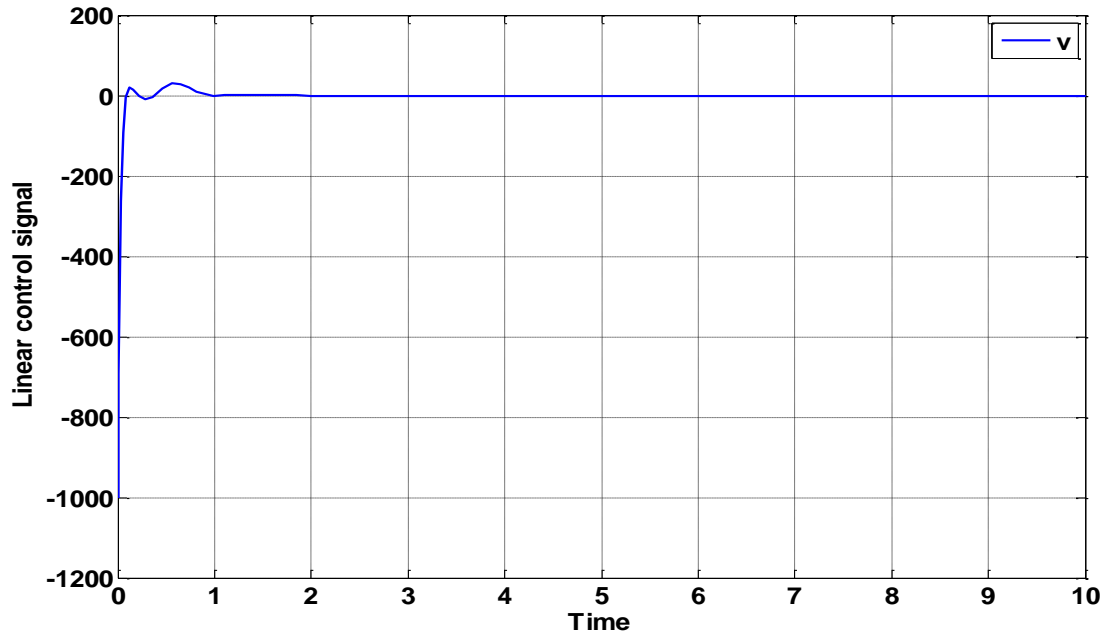


Figure 6.23: Linear control signal when the set point is at 0.85[m]



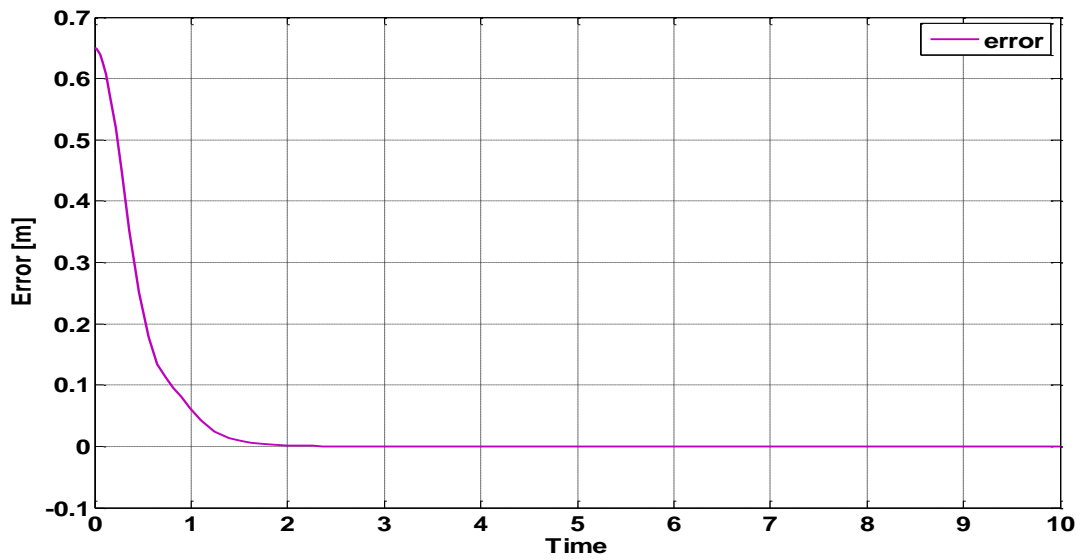


Figure 6.24: Error signal between the set point and the nonlinear plant output when the set point is 0.85[m]

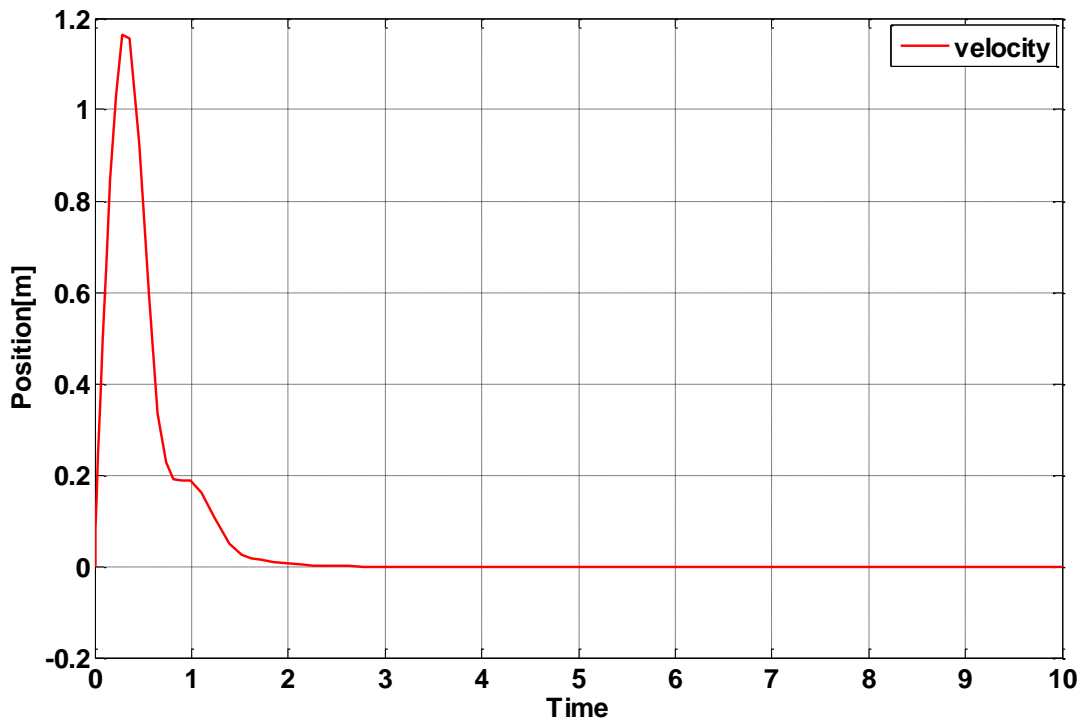


Figure 6.25: Velocity of the ball when the set point is 0.85[m]

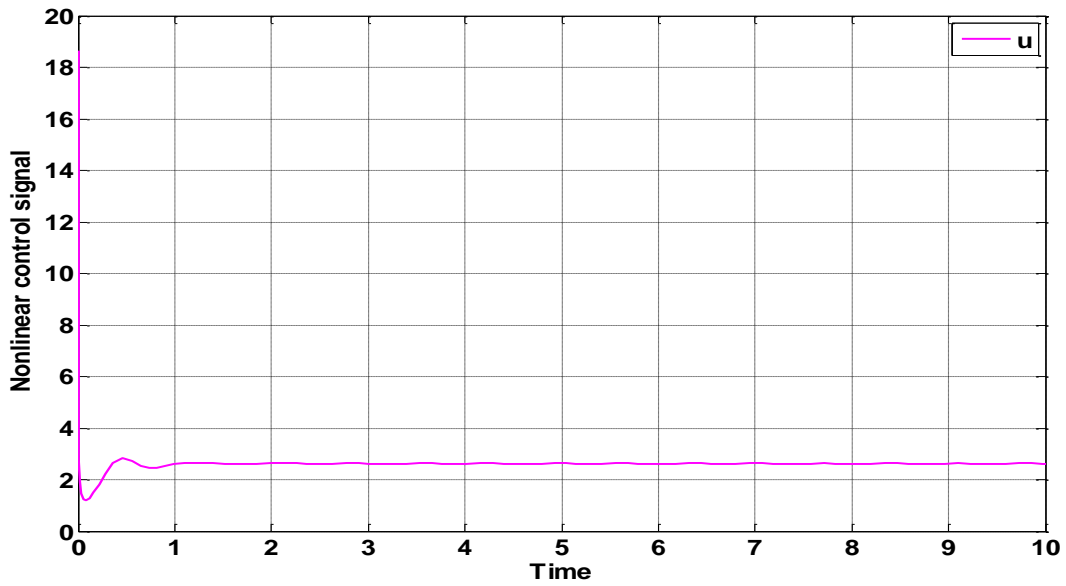


Figure 6.26: Nonlinear linearizing control signal when the set point is at 0.85[m]

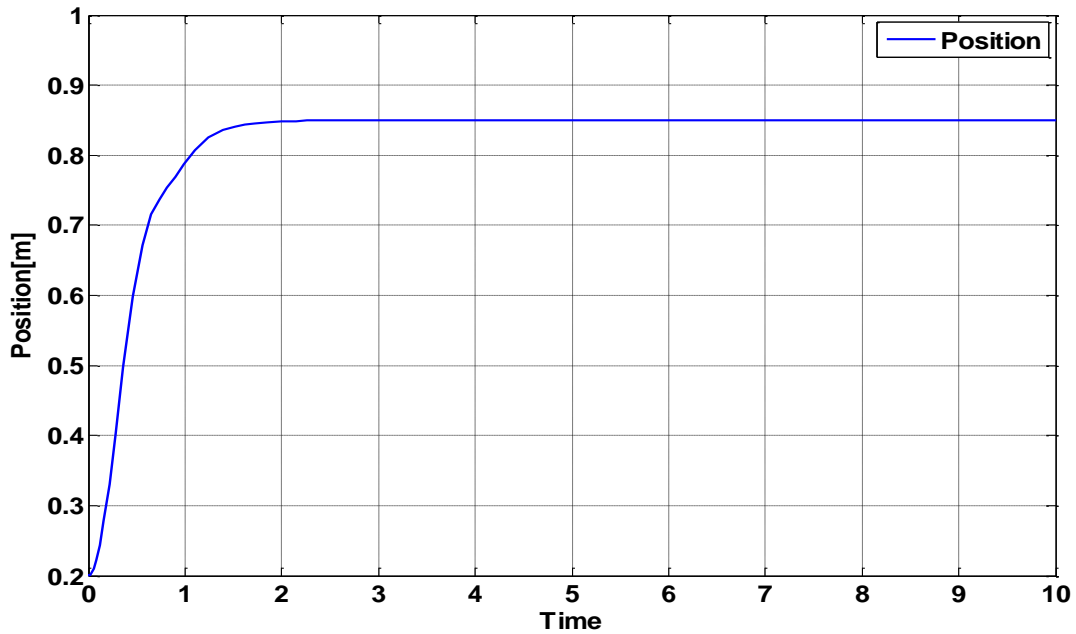


Figure 6.27: Nonlinear system response under linear and linearizing controls when the set-point is 0.85[m]

## 6.6 Discussion of the results

The different simulation results for the magnetic levitation system behaviour under nonlinear linearizing control show the following:

- The system is stable.
- There is no time delay.
- The errors signals go to zero.
- The plant output always follows the reference model and the set points trajectories.
- All the states of the system are stabilized.
- The steady state error varies for different values of the set-points.
- The rising time also varies with the changes in the values of the set-points. For bigger set-points values the rising time is smaller.

The characteristics of the dynamic output behaviour of the closed-loop system are given below in Table 6.2.

**Table 6.2: Characteristics of the dynamic behaviour of the closed loop system output for the case of input-state/input-output and linear controls**

Set point	Characteristics	Magnetic levitation results
0.55	Time Delay	0
	Overshoot	0
	Rising Time	2s
	Steady State Error	0.015m
	Settling Time	4s
0.6	Time Delay	0
	Overshoot	0
	Rising Time	2.1s
	Steady State Error	0.008m
	Settling Time	3s
	Time Delay	0

0.75	Overshoot	0
	Rising Time	2s
	Steady State Error	0.002m
	Settling Time	2.2s
0.85	Time Delay	0
	Overshoot	0
	Rising Time	1.8s
	Steady State Error	0.001m
	Settling Time	2s

## 6.7 Conclusion

The developed method for design of linear integral and nonlinear linearizing controllers based on the linear quadratic regulation (LQR) and the input-state closed loop linearization techniques is described in this chapter. The simulation results show that the designed linear integral and nonlinear linearizing controllers can control the magnetic levitation process according to the proposed requirements. Given by the values of the matrices  $\bar{Q}$  and  $\bar{R}$  in the criterion for optimality. The method allows:

- The linearized by the nonlinear controller closed-loop system to be equivalent to a linear stable desired system.
- To control the process at a certain range of ball positions. In the considered case, the system is stable on relatively high values of ball position.
- The behaviour of the system is optimal according to the selected values of matrices  $\bar{Q}$  and  $\bar{R}$ .

Nonlinear controller based on Lyapunov direct method is designed in Chapter 7 for a benchmark reduced order levitation system to control the system at relatively low values of ball position. Then the implementation of the nonlinear linearizing control in real-time is done in Chapter 8.

## **CHAPTER SEVEN: DESIGN OF A NONLINEAR LINEARIZING CONTROLLER ON THE BASIS OF MODEL REFERENCE CONTROL AND LYAPUNOV THEORY**

### **7.1 Introduction**

The linearization method draws deductions about the local stability of a nonlinear system around an operating point from the stability characteristics of the system linear estimation. The stability of dynamic systems can be analyzed in a very precise way with Lyapunov methods if the equivalent mathematical models are expressed as systems of normal differential equations (Hachicho, 2006).

The model reference control (MRC) approached is selected as a technique to achieve the stability of the nonlinear system based on Lyapunov theory approach. In control theory, the MRC is a proposed method to solve problems in which the specifications of the performance are given in terms of the reference model behaviour. This reference model gives the ideal response of the process output to the command signal. In this thesis, one reference model is needed thus the first step toward the completion of this work is to find the desired model that will generate the ideal trajectory that the actual model will follow under nonlinear control signal. Based on MRC, the first target is to find a reference model (Ge, 2004).

For the purpose of MRC, linear models are selected as reference models. In order to achieve model reference control, an ideal second order system is considered and controlled by a linear controller designed using the quadratic optimal control method. The idea is to study the behaviour of the linear system then the reference model is derived which allows a better understanding of the plant behaviour.

In this chapter, the Lyapunov stability theory based on the model reference control technique is applied to the magnetic ball levitation system. In section 7.2, the design of nonlinear controllers based on Lyapunov direct method and model reference control is presented. The design of a Lyapunov controller for the magnetic levitation process is described in section 7.3. Further the simulation results are presented in section 7.4. Finally the discussion of results and the conclusion are given respectively in section 7.5 and section 7.6.

## 7.2 Design of nonlinear control based on Lyapunov direct method

One convenient approach for defining system performance is by means of a model that will generate the desired output for a given input (Han, 2009). The model needs not be actual hardware, it can be exclusively a mathematical model simulated on a personal computer. In theory, the output of the model and that of the plant are compared and the difference is used to produce the control signals for the plant. Model reference control (MRC) has been used to get acceptable performance in some very harsh control situations involving nonlinearities and/or time varying parameters (Landau, 1979). MRC and Lyapunov second method for stability are used in this thesis to design a linearizing controller for the magnetic levitation system.

The method for the design of the nonlinear controller follows the procedure described in (Slotine, 1991):

### 7.2.1 Plant model

The plant is characterized by the nonlinear state equation:

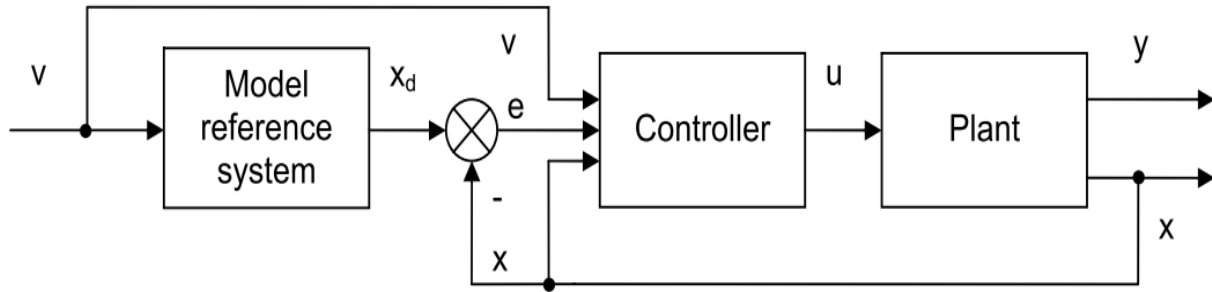
$$\dot{x} = f(x) + g(x)u \quad (7.1)$$

$$y = Cx \quad (7.2)$$

Where  $x \in \mathfrak{R}^n$  is the state vector (n-vector);  $u \in \mathfrak{R}^m$  is the control vector;  $f \in \mathfrak{R}^n$  is the vector valued function;  $y \in \mathfrak{R}^1$  is the plant output;  $C \in \mathfrak{R}^{1 \times n}$  is the output matrix.

### 7.2.2 Design of the reference model. Error between states of the reference and plant models

It is required that the control system tracks closely some model system. The design problem is to develop a controller that always produces a signal that forces the plant state toward the model state (Hans, 2009) and (Nketoane, 2009). Figure 7.1 shows the block diagram of the closed-loop MRC system configuration, where  $v$  is the control input of the reference model.



**Figure 7.1: Model-reference control system (Slotine, 1991)**

The reference model can be different, linear or nonlinear, time invariant or time variant, and so on. In this thesis, it is assumed that the reference model is linear and described by:

$$\dot{x}_d = Ax_d + Bv \quad (7.3)$$

$$y = Cx_d$$

Where  $x_d \in \mathfrak{R}^n$  is the state vector of the model;  $v \in \mathfrak{R}^m$  is the control vector for the reference model;  $A \in \mathfrak{R}^{n \times n}$  is the constant state matrix;  $B \in \mathfrak{R}^{n \times m}$  is the constant control matrix and  $C \in \mathfrak{R}^{1 \times n}$  is the constant output matrix.

It is assumed that the eigenvalues of  $A$  have negative real parts so that the model-reference system has an asymptotically stable state of equilibrium. The control input  $v$  can be selected in such a way that  $x_d$  follows some desired trajectory, which then will be followed by the plant.

The error vector  $\varepsilon$  is defined by:

$$\varepsilon = x_d - x \quad (7.4)$$

Where  $\varepsilon \in \mathfrak{R}^n$ ,  $x$  is the actual state of the plant.

The requirements towards the closed loop systems are that the error  $\varepsilon$  has to be reduced to zero by a suitable control vector  $u$ . In order to include the model equation and the plant equation in the error Equation (7.4) it is necessary to differentiate the error Equation (7.4) according to the time:

$$\begin{aligned} \dot{\varepsilon} &= \dot{x}_d - \dot{x} = Ax_d + Bv - f(x) - g(x)u \\ &= Ax_d - Ax + Ax + Bv - f(x) - g(x)u \end{aligned}$$

$$= A(x_d - x) + Ax + Bv - f(x) - g(x)u$$

$\therefore \varepsilon = x_d - x$ , then the above equation can be simplified as:

$$\dot{\varepsilon} = A\varepsilon + Ax + Bv - f(x) - g(x)u \quad (7.5)$$

Equation (7.5) is a differential equation for the error vector.

Then a nonlinear controller can be designed such that at steady state  $x = x_d$  and  $\dot{x} = \dot{x}_d$ , or  $\varepsilon = \dot{\varepsilon} = 0$ . Thus the equilibrium  $\varepsilon = 0$  will be the origin of the coordinate system.

### 7.2.3 Design of the nonlinear linearizing 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.

#### 7.2.3.1 Construction of a Lyapunov function for the system and determination of its first derivative

An ideal point to start the design of the control vector  $u$  is to construct a Lyapunov function system. In this thesis, the Lyapunov function is assumed to be done by a quadratic form:

$$V(\varepsilon) = \varepsilon^T P \varepsilon \quad (7.6)$$

Where  $P \in \mathfrak{R}^{n \times n}$  is a positive-definite Hermitian or real symmetric matrix. Because the function  $V(\varepsilon)$  is in quadratic form and the matrix  $P$  is positive definite, it is true that  $V(\varepsilon)$  is positive definite.

Differentiating the positive definite function  $V(\varepsilon)$  along the system trajectory, its time derivative is obtained as follow:

$$\begin{aligned} \dot{V}(\varepsilon) &= \dot{\varepsilon}^T P \varepsilon + \varepsilon^T P \dot{\varepsilon} \\ &= [A\varepsilon + Ax + Bv - f(x) - g(x)u]^T P \varepsilon + \varepsilon^T P [A\varepsilon + Ax + Bv - f(x) - g(x)u] \\ &= [A^T \varepsilon^T + A^T x^T + B^T v^T - f^T(x) - g^T(x)u^T] P \varepsilon + \varepsilon^T P [A\varepsilon + Ax + Bv - f(x) - g(x)u] \end{aligned}$$



$$\begin{aligned}
&= A^T \varepsilon^T P \varepsilon + A^T x^T P \varepsilon + B^T v^T P \varepsilon - f^T(x) P \varepsilon - g^T(x) u^T P \varepsilon + \varepsilon^T P A \varepsilon + \varepsilon^T P A x + \varepsilon^T P B v - \\
&\quad - \varepsilon^T P f(x) - \varepsilon^T P g(x) u \\
&= \varepsilon^T [A^T P + P A] \varepsilon + 2N
\end{aligned} \tag{7.7}$$

Where:

$$\begin{aligned}
2N &= A^T x^T P \varepsilon + \varepsilon^T P A x - f^T(x) P \varepsilon - g^T(x) u^T P \varepsilon - \varepsilon^T P f(x) - \varepsilon^T P g(x) u + B^T v^T P \varepsilon + \varepsilon^T P B v \\
&= \varepsilon^T P A x + \varepsilon^T P A x - \varepsilon^T P f(x) - \varepsilon^T P g(x) u - \varepsilon^T P f(x) - \varepsilon^T P g(x) u + \varepsilon^T P B v + \varepsilon^T P B v \\
&= 2\varepsilon^T P [A x - f(x) - g(x) u + B v]
\end{aligned} \tag{7.8}$$

This derivation is based on the fact that  $P$  is a symmetrical matrix and  $P^T = P$  or

$$N = \varepsilon^T P [A x - f(x) - g(x) u + B v] \tag{7.9}$$

$N$  is a scalar quantity.

### 7.2.3.2 Calculation of the nonlinear linearizing control

$V(\varepsilon)$  is assumed to be a Lyapunov function, if its first derivative is negative definite then the system (7.7) is stable. The first derivative of  $V(\varepsilon)$  is the sum of two expressions:

$$\dot{V}(\varepsilon) = \varepsilon^T [A^T P + P A] \varepsilon + 2N \tag{7.10}$$

In order for  $\dot{V}(\varepsilon)$  to be negative definite, the two terms of equation (7.10) have to be negative definite.

$$1. \varepsilon^T [A^T P + P A] \varepsilon < 0 \text{ or } A^T P + P A = -Q$$

Where  $Q$  is a positive definite matrix  $\dot{V}(\varepsilon) = \varepsilon^T [A^T P + P A] \varepsilon + 2N$  arix.

$$2. N \leq 0.$$

Based on Equations (7.9) and (7.10), it can be concluded that  $N$  can be made negative or equal to zero through suitable selection of the plant control vector  $u$  which is part of the first derivative of the Lyapunov function  $\dot{V}(\varepsilon)$ . Then from noting that  $V(\varepsilon) \rightarrow \infty$  as  $\|\varepsilon\| \rightarrow \infty$ , it can be seen that the equilibrium state  $\varepsilon = 0$  is asymptotically stable in the larger range. The fulfilment of condition (1) can be achieved by an ideal choice of the matrix  $P$  since the eigenvalues of the state matrix  $A$  are selected to be with negative real parts. The problem to solve now is to select an appropriate vector  $u$  so that  $N$  is either zero, or negative scalar quantity. The determination of the nonlinear linearizing controller  $u$  can be done with proper

selected values of the matrix  $P$  or the matrix  $Q$ . The obtained nonlinear linearizing controller  $u$  makes the system stable and follows the desired trajectory determined by the reference model. In this section, the general procedure for designing a nonlinear linearizing controller on the basis of Lyapunov stability theory, a linear reference model and MRC method are described. In the next section, the method is applied to a reduced model of the magnetic levitation system developed by Education Control Products (ECP) (ECP, 1999) and (Silva, 2009). The adequate control developed should always keep the ball levitated at a desired position.

### 7.3 Design of a Lyapunov-based and MRC based nonlinear linearizing controller for the magnetic levitation system

On the basis of the study and understanding of the MRC theory, the Lyapunov stability theory, the Lyapunov direct method and the LQR control method, the following sub-sections cover the general procedures as described previously, and applied to the magnetic levitation system. The nonlinear linearizing controller for controlling the magnetic levitation system is developed step by step.

#### 7.3.1 The nonlinear model of the magnetic levitation system

The nonlinear reduced model of the magnetic levitation system according to ECP is expressed as follow (Torres et al., 2010):

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -g - \frac{c}{m}x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ \frac{1}{ma(x_1 + b)^4} \end{bmatrix} u, x(0) = x_0 \quad (7.11)$$

$$y = Cx \quad (7.12)$$

$x_1$  = ball position

$x_2$  = velocity of the ball

Where:  $C = [1 \ 0]$ ;  $a$ ,  $b$  and  $c$  are constants related with the magnetic coil properties. The values of the parameters of the process are:

$$\begin{aligned} g &= 9.81N / Kg \\ m &= 0.12Kg \end{aligned} \quad (7.13)$$

$$\begin{aligned} a &= 0.95 \\ b &= 6.28 \end{aligned} \quad (7.14)$$

$$c = 0.15N / Kg$$

The nonlinear model can be rewritten in the common form as:

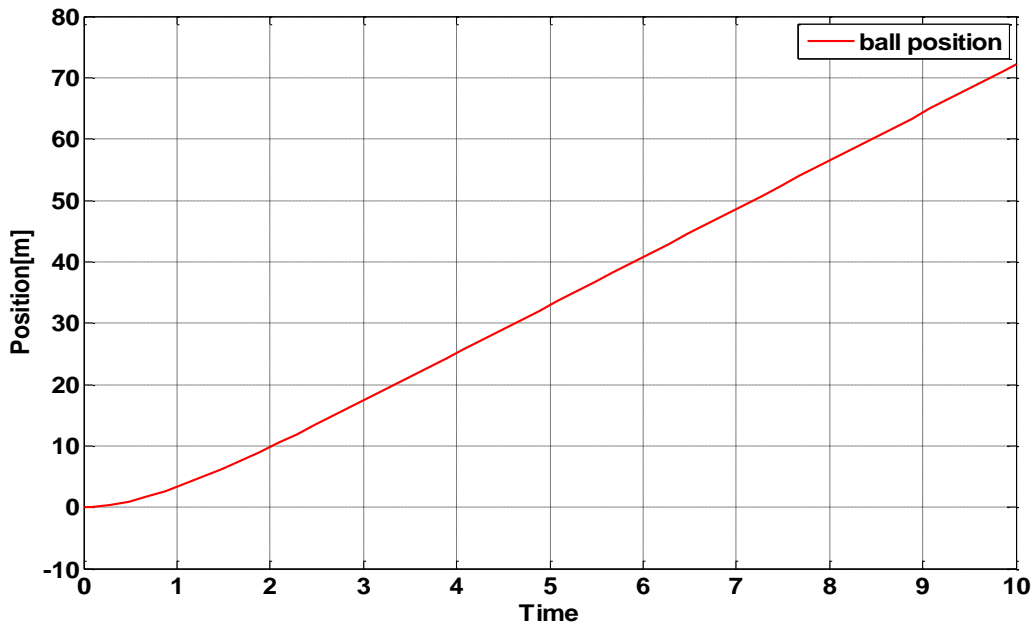
$$\dot{x} = f(x) + g(x)u \quad (7.15)$$

$$y = Cx \quad (7.16)$$

Figure 7.2 shows the behaviour of the nonlinear reduced order model of the levitation system.

The simulation is done with the following parameters:

- Initial conditions: [0.05 0]'
- Step-input: 0.3[volts]



**Figure 7.2: Open loop response of the nonlinear reduced-order system when the step input is at 0.3[volts]**

The open loop response of the reduced order system is approximately the same as the one of the full system shown in Figure 3.6 of Chapter 3. Therefore the reduced-order system is also open loop unstable.

### 7.3.2 Model of the desired linear system (reference model)

The linear reference model for this project can be written in the following form:

$$\dot{x}_d = Ax_d + Bv, x_d(0) = x_{d0} \quad (7.16)$$

$$y_d = Cx \quad (7.17)$$

Where  $x_d \in \mathfrak{R}^2$  is the desired state space vector,  $v \in \mathfrak{R}^1$  is the control vector for the reference model,  $A \in \mathfrak{R}^{2 \times 2}$  and  $B \in \mathfrak{R}^{2 \times 1}$  are the state space and control matrices of the reference model,  $x_{d0}$  is the initial state.

The model of the magnetic levitation is of second order. Therefore the desired model is selected to be of second order too. The eigenvalues of the state matrix  $A_d$  are selected to be with negative real parts to ensure stability of the reference model.

### 7.3.2.1 Determination of the error between the reference model and the plant states

The error between the desired and the current model of the magnetic levitation is:

$$\varepsilon = x_d - x, \varepsilon \in \mathfrak{R}^{2 \times 2} \quad (7.18)$$

The error signal  $\varepsilon$  has to be reduced to zero by a suitable control vector  $u$ . The differential equation of the error is:

$$\begin{aligned} \dot{\varepsilon} &= \dot{x}_d - \dot{x} = Ax_d + Bv - f(x) - g(x)u \\ &= Ax_d + Ax - Ax + Bv - f(x) - g(x)u \\ &= A[x_d - x] + Ax + Bv - f(x) - g(x)u \\ &= A\varepsilon + Ax + Bv - f(x) - g(x)u \end{aligned} \quad (7.19)$$

The problem is to design a control vector  $u$ , such that at the equilibrium state  $x = x_d$ ,  $\dot{x} = \dot{x}_d$ ,  $\varepsilon = \dot{\varepsilon} = 0$  is achieved.

### 7.3.2.2 Design of the nonlinear controller

#### 7.3.2.2.1 Construction of the Lyapunov function

The construction of the Lyapunov function for the error differential equation shown in equation (7.18) is:

$$V(\varepsilon) = \varepsilon^T P \varepsilon \quad (7.20)$$

Where  $P$  is a symmetrical positive definite matrix,  $P \in \mathfrak{R}^{2 \times 2}$ .

### 7.3.2.2 Calculation of the first derivative of the Lyapunov function

$$\begin{aligned}
 \dot{V}(\varepsilon) &= \dot{\varepsilon}^T P \varepsilon + \varepsilon^T P \dot{\varepsilon} \\
 &= [A\varepsilon + Ax + Bv - f(x) - g(x)u]^T P \varepsilon + \varepsilon^T P [A\varepsilon + Ax + Bv - f(x) - g(x)u] \\
 &= [A^T \varepsilon^T + A^T x^T + B^T v^T - f^T(x) - g^T(x)u^T] P \varepsilon + \varepsilon^T P [A\varepsilon + Ax + Bv - f(x) - g(x)u] \\
 &= A^T \varepsilon^T P \varepsilon + A^T x^T P \varepsilon + B^T v^T P \varepsilon - f^T(x) P \varepsilon - g^T(x)u^T P \varepsilon + \varepsilon^T P A \varepsilon + \varepsilon^T P A x + \varepsilon^T P B v - \\
 &\quad - \varepsilon^T P f(x) - \varepsilon^T P g(x)u \\
 &= \varepsilon^T [A^T P + PA] \varepsilon + 2\varepsilon^T P [Ax + Bv - f(x) - g(x)u] \\
 &= -\varepsilon^T Q \varepsilon + 2N
 \end{aligned}$$

$Q = A^T P + PA$  is a positive definite matrix,  $Q$  is symmetrical.

$$N = \varepsilon^T P [Ax + Bv - f(x) - g(x)u] \quad (7.21)$$

The derived Equation (7.21) is the expression of the first derivative of the Lyapunov function. In order to make the error in the closed loop system to go to zero as time goes to infinity ( $t \rightarrow \infty$ ), it is fundamental for this equation to be negative definite. The first expression of this equation is negative definite as  $Q$  is selected to be positive definite. Then the second expression  $N$  can be made zero or negative  $N \leq 0$  by a convenient selection of the control  $u$ .

### 7.3.2.2.3 Calculation of the nonlinear linearizing controller $u$

The calculation of the nonlinear linearization controller  $u$  is done by some transformations of the expression for  $N$ :

$$\begin{aligned}
 N &= \varepsilon^T P [Ax + Bv - f(x) - g(x)u] \leq 0 \\
 N &= \varepsilon^T P [Ax + Bv - f(x)] - \varepsilon^T P [g(x)u] \leq 0 \\
 \varepsilon^T P [Ax + Bv - f(x)] &\leq \varepsilon^T P [g(x)u] \quad (7.22)
 \end{aligned}$$

The expressions from both sides of the equation are scalars, which depend on time. That is the reason why it is possible to divide both sides by  $\varepsilon^T P [g(x)]$  and obtain:

$$u \geq \frac{\varepsilon^T P [Ax + Bv - f(x)]}{\varepsilon^T P [g(x)]} \quad (7.23)$$

### 7.3.2.3 Representation of the diagram of the closed loop system

On the basis of Equation (7.23), a diagram of the closed loop system can be drawn. The expression of the nonlinear linearizing controller is multiplied by a number  $I > 0$  in order to make the realization in the Equation (7.23) stronger. The nonlinear controller developed makes the first derivative of the Lyapunov function negative, linearizes the closed loop system consisting of the nonlinear controller and the plant, and makes the behaviour of the closed-loop system to follow the behaviour of the reference model. The block diagram of the closed loop system is shown in Figure 7.3.

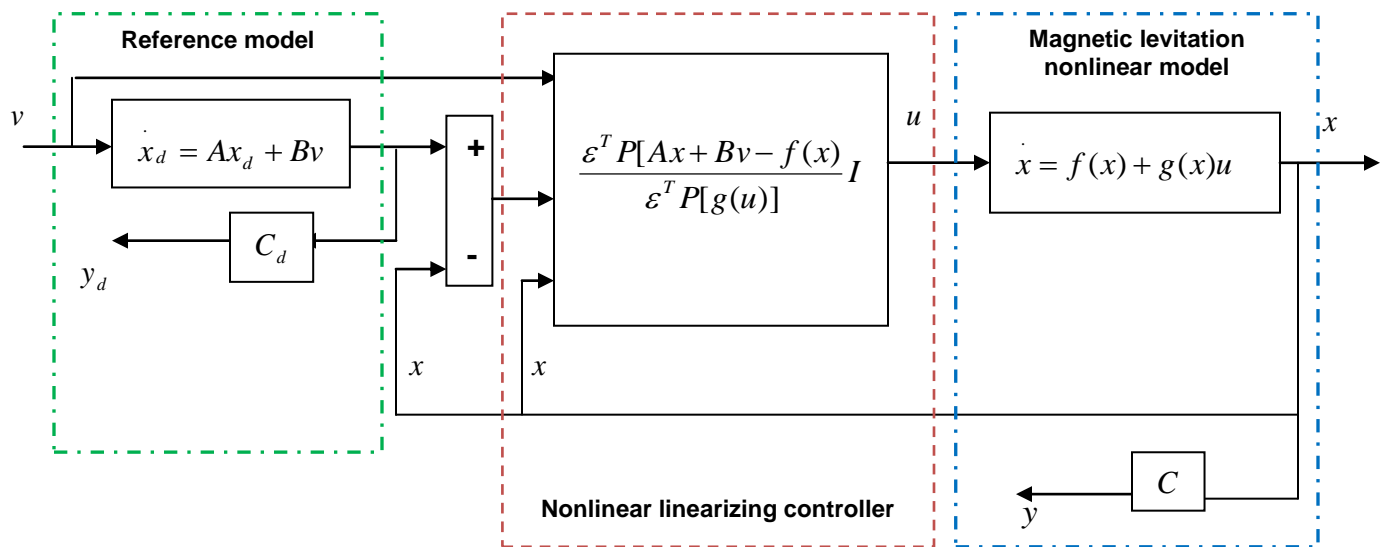


Figure 7.3: Block diagram of the closed loop system

### 7.3.3 Design of a linear control v for the linearized closed loop system

Figure 7.3 shows that the desired vector  $x_d$  depends on the input control vector  $v$  for the reference model. Different values of  $v$  will give different values of  $x_d$ . From the expression of the nonlinear control given by Equation (7.23), it can be seen that the values of  $u$  depend on the parameters of the nonlinear plant model. The implementation of the nonlinear linearizing controller then cannot be very successful because of the influence of the disturbances, and the changes of the plant parameters. The linearizing and stabilizing effects could be lost, and could make the system unstable. This means that an additional linear control has to be designed to

make the closed loop system more robust and its output exactly to follow the desired behaviour of the reference model. It is possible to apply the theory of optimal linear quadratic control, in order to design an optimal controller for the system of the reference model.

### 7.3.3.1 Specification of the closed loop system with the reference model and for the linearized closed loop system

It is assumed that the desired output of the entire closed loop system is a set point value  $y^{sp}$ . On the basis of this assumption, it is crucial to determine the optimal controller such that:  $y = y^{sp}$  or  $\varepsilon = y^{sp} - y$  when  $t \rightarrow \infty$ .

### 7.3.3.2 Design of the linear quadratic controller

The design of the linear integral controller follows the steps described in Chapter 5, point 5.3.3.1:

$$\dot{\varepsilon}_d = A\varepsilon_d + Bv, \quad \varepsilon_{d0} = y^{sp} - Cx_{d0}$$

The aim is to design linear integral quadratic control in order to make the error between the set-point and the current value of the system output to go to zero. Then the extended version of the model is built as follows:

$$\begin{aligned} \dot{x}_d &= A_d x_d + Bv \\ \dot{x}_{n+1} &= y^{sp} - y_d, \quad x_d(0) = x_{d0} \\ y_d &= Cx_d \end{aligned} \quad (7.24)$$

State space type of integral quadratic controller is necessary to be designed of the form:

$$v = \bar{H}_\phi \bar{x}_d = Hx_d + H_{n+1}x_{n+1}, \quad \bar{H}_\phi \in \mathfrak{R}^{m \times (n+1)} \quad (7.25)$$

Where:  $\dot{\bar{x}}_d = \bar{A}\bar{x}_d + \bar{B}v + \begin{bmatrix} 0_{2 \times 1} \\ 1 \end{bmatrix} y^{sp}$

$$\bar{A} = \begin{bmatrix} A_d & 0 \\ -C_d & 0 \end{bmatrix}, \quad \bar{B} = \begin{bmatrix} B_d \\ 0 \end{bmatrix}, \quad \bar{H} = [H \quad H_1] \quad (7.26)$$

The fundamental idea of Equations (7.24), (7.25) is that the servo problem is converted to a problem for design of a linear quadratic regulator in which the set point is zero. The problem to find the matrix controller  $\bar{H}$  can be formulated as follows:

Find a feedback state space controller (7.25) such that:

$$J_p = \int_0^{\infty} \left[ \|\bar{x}_d\|^2 \bar{Q} + \|v\|^2 \bar{R} \right] dt, \bar{Q} \in \mathfrak{R}^{(n+1) \times (n+1)}, \bar{R} \in \mathfrak{R}^{m \times m} \quad (7.27)$$

is minimized under the model Equation (7.24).

### 7.3.3.3 Solution of the linear quadratic regulator problem

The resolution of linear quadratic regulator problem is given by the following equation:

$$v = -\bar{H}_{\phi} \bar{x}_d = -\bar{H} \bar{x}_d + H_1 \varepsilon = -\bar{R}^{-1} \bar{B}^T \bar{P} \bar{x}_d \quad (7.28)$$

In equation (7.28),  $\bar{P}$  is the solution of the Riccati equation;  $\bar{H}_{\phi} = [H \ H_1] \in \mathfrak{R}^{1 \times 2}$

The solution of the problem can be found in Matlab using the 'lqr' function, its structure is as follow:

$$[\bar{H}_{\phi}, \bar{P}, \bar{E}] = lqr(\bar{A}, \bar{B}, \bar{Q}, \bar{R})$$

$\bar{H}_{\phi}$  is the matrix of the regulator;  $\bar{P}$  is the matrix of the Riccati equation and  $\bar{E}$  is the vector of the poles of the closed-loop matrix  $[\bar{A} - \bar{B}\bar{H}_{\phi}]$ . To make the system stable, all the poles have to be with real negative parts.

The control  $v$  is obtained as follow:

$$v = -Hx_d + H_1 \varepsilon \quad (7.29)$$

The augmented matrices with the additional integrator states can be expressed as follow:

$$\bar{A} = \begin{bmatrix} A_d & 0 \\ -C_d & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -2 & -3 & 0 \\ -1 & 0 & 0 \end{bmatrix};$$

$$\bar{B} = \begin{bmatrix} B_d \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Where:

$$\bar{A} = \begin{bmatrix} A_d \in \mathfrak{R}^{2 \times 2} & 0 \in \mathfrak{R}^{2 \times 1} \\ C_d \in \mathfrak{R}^{1 \times 2} & 0 \in \mathfrak{R}^{1 \times 1} \end{bmatrix}; \text{ and } \bar{B} = \begin{bmatrix} B \in \mathfrak{R}^{2 \times 1} \\ 0 \notin \mathfrak{R}^{1 \times 1} \end{bmatrix}$$

The values of the weighting matrices  $\bar{Q}$  and  $\bar{R}$  are summarized in Table 7.1.

Table 7.1 shows the different values of  $\bar{H}$  with regard to different set points, the values of the weighting matrices  $\bar{Q}$  and  $\bar{R}$ .

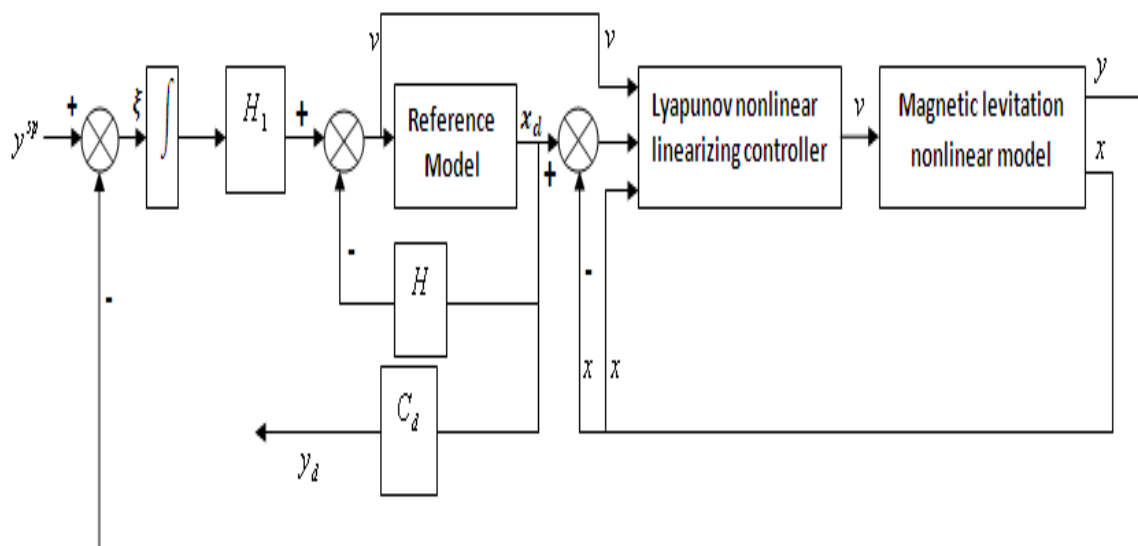


**Table 7.1: Parameters obtained for the LQR**

Set points	Initial conditions	Matrix $\bar{Q}$	Matrix $\bar{R}$	Feedback controller gain $\bar{H}$
0m	[0.05 0 0]'	$\begin{bmatrix} 91000 & 0 & 0 \\ 0 & 110 & 0 \\ 0 & 0 & 4 \end{bmatrix}$	0.1	[299.85 23.81 -2]
0.01m	[0.05 0 0]'	$\begin{bmatrix} 91000 & 0 & 0 \\ 0 & 300 & 0 \\ 0 & 0 & 10 \end{bmatrix}$	0.1	[952.67 67.1 -10]
0.09m	[0.05 0 0]'	$\begin{bmatrix} 69000 & 0 & 0 \\ 0 & 750 & 0 \\ 0 & 0 & 25 \end{bmatrix}$	1	[261.37 31.8 -5]
0.15m	[0.05 0 0]'	$\begin{bmatrix} 100000 & 0 & 0 \\ 0 & 650 & 0 \\ 0 & 0 & 90 \end{bmatrix}$	1	[315.31 32.91 -9.79]

**7.3.3.4 Application of the linear integral controller to the closed-loop system with the nonlinear controller system and the reference model.**

The structure of the block diagram with the nonlinear linearizing MRC based on Lyapunov second method is shown in Figure 7.4.



**Figure 7.4: Block diagram of the Lyapunov stability based on model reference control system**

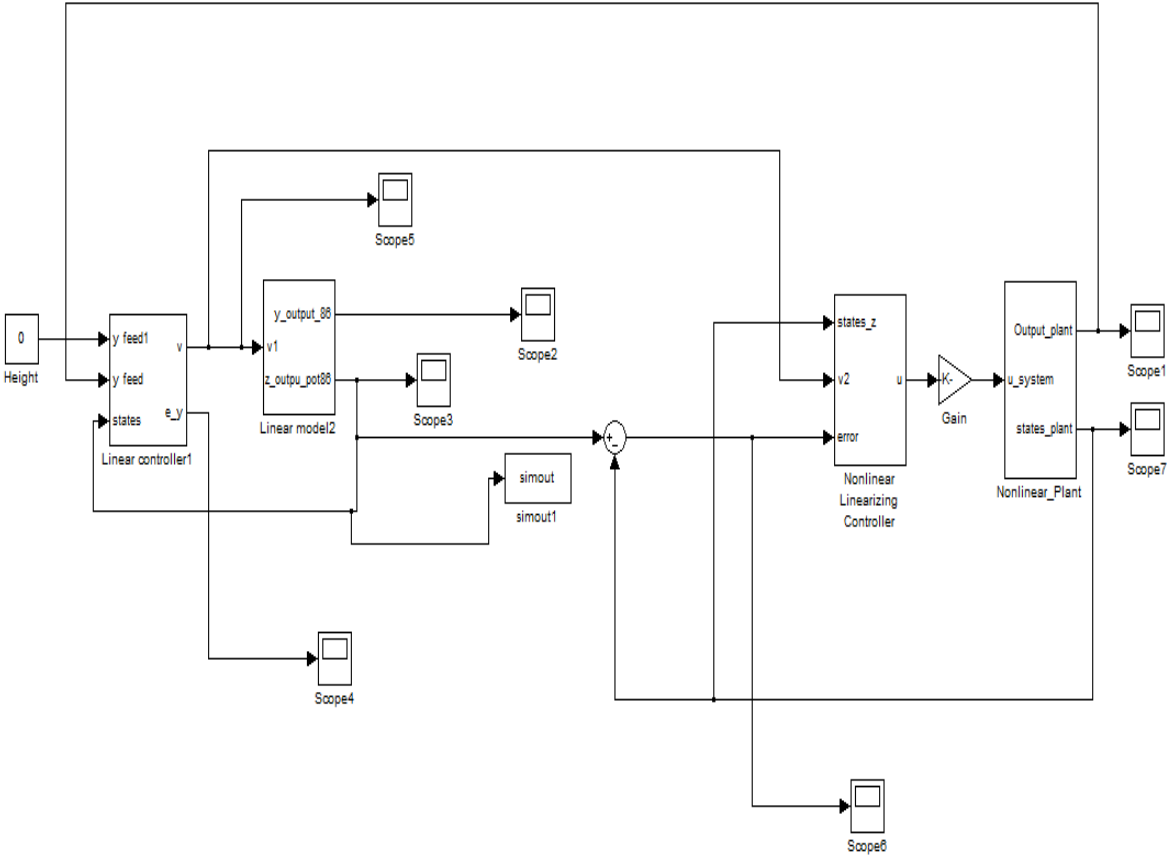
It is important that the feedback for the linear controller implementation is not taken from the output of the reference model but from the output of the magnetic levitation nonlinear model.

Using the process real output will lead to better results as the integral controller compensates for disturbances over the real process.

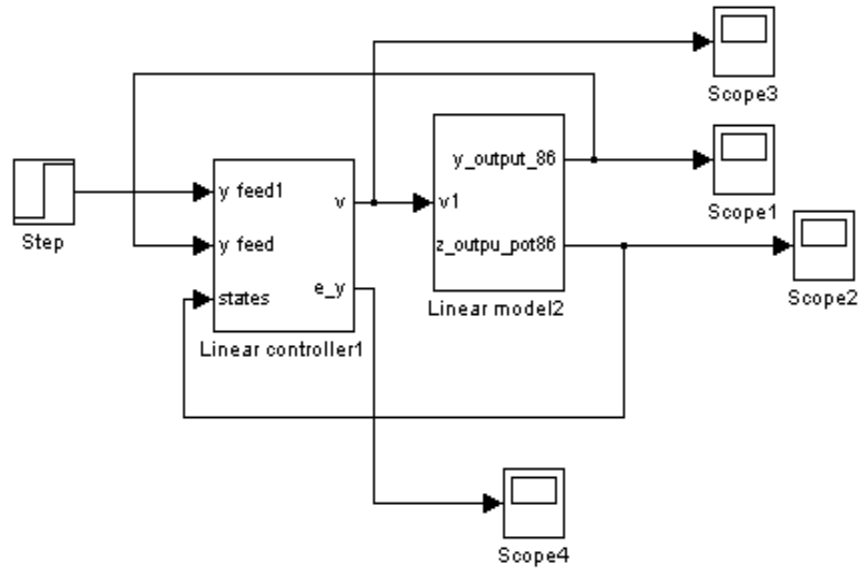
### 7.4 Simulation

The simulation is done in Matlab/Simulink environment. The closed-loop diagram based on Lyapunov direct method is shown in figure 7.4. This closed loop diagram is made of four important subsystems:

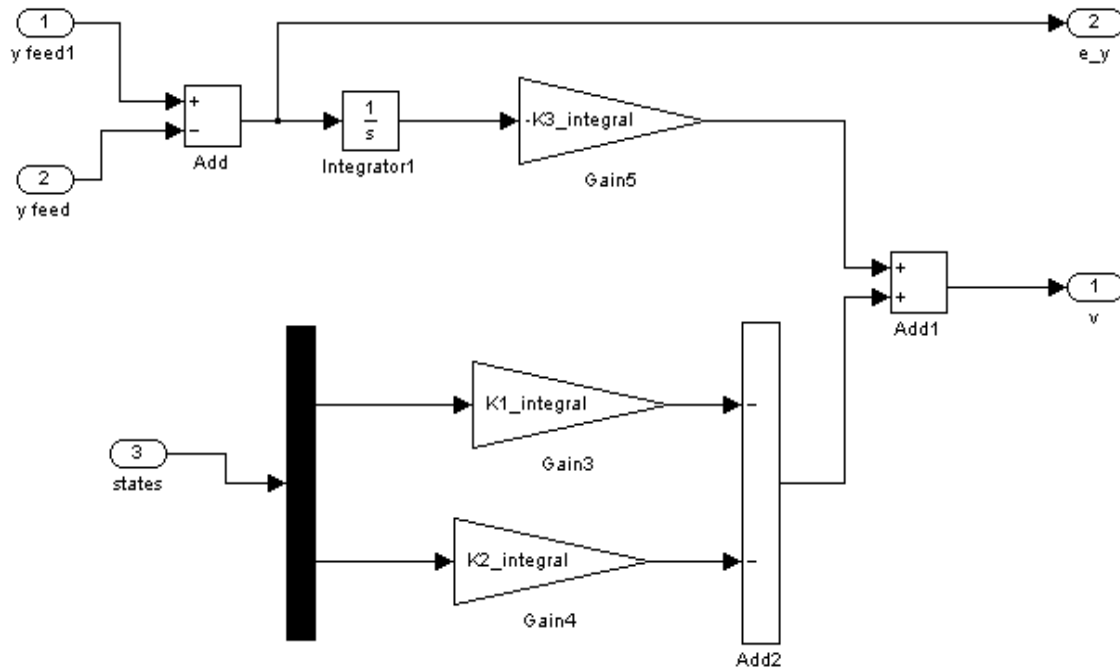
- Reference model (Figure 7.6)
- Linear controller (Figure 7.7)
- Nonlinear linearizing controller (Figure 7.8)
- Magnetic levitation model (Figure 3.4 of Chapter 3)



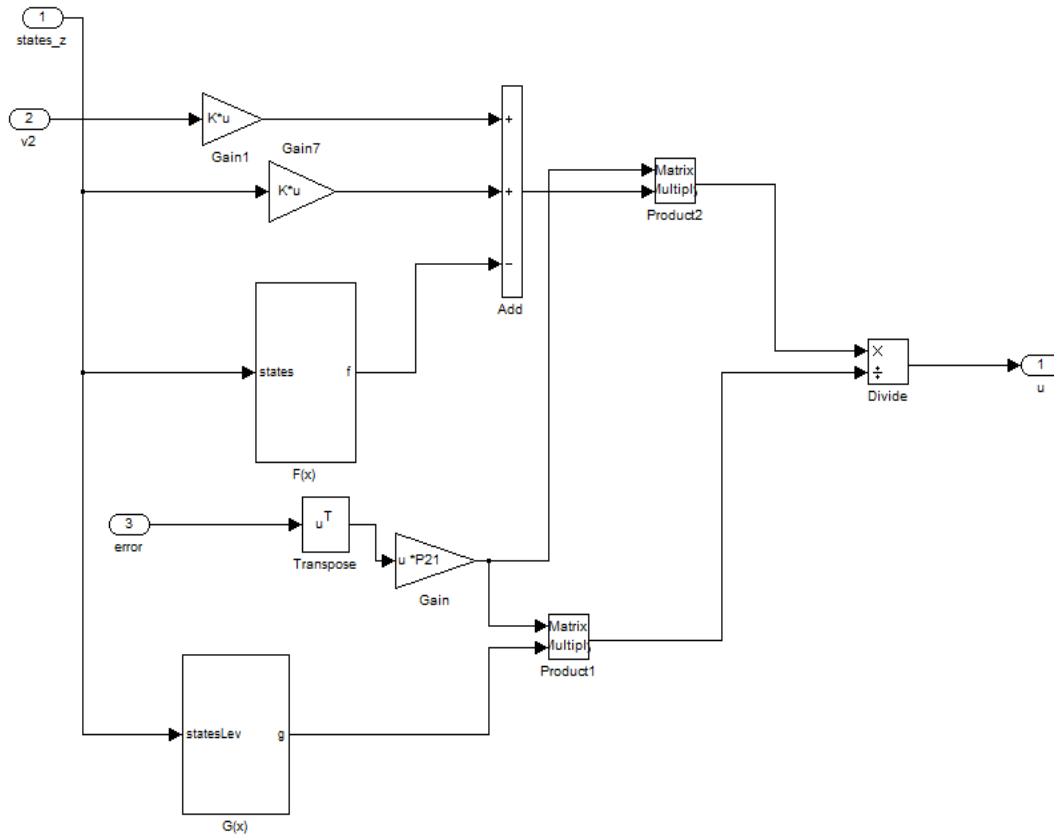
**Figure 7.5: Simulink diagram of the Lyapunov direct method based on MRC**



**Figure 7.6:** Simulink block diagram of the linear reference model and its controller



**Figure 7.7:** Simulink block diagram of the linear integral controller



**Figure 7.8: Simulink block diagram of the nonlinear linearizing controller on the basis of Lyapunov second method**

The reference model is formed by the combination of the linear controller and the linear plant. The nonlinear controller produces the nonlinear control signal determined in Equation (7.23). The parameters of the simulation are found in Matlab file named “*yohan\_Lyapunov.m*”, and the program is given in the Appendix A.6. The corresponding Simulink model “*Lyapunov\_maglev.mdl*” is shown in Figure 7.5.

### 7.4.1 Simulation results

The simulation results are presented in this section. The results show the following results:

- Linear integral controller signal
- Error signal between the set point and the nonlinear plant output
- Error signal between the reference model states and the nonlinear system states
- Nonlinear linearizing control signal
- Position of the ball
  
- Initial conditions  $[0.05 \ 0 \ 0]'$  and  $y^{sp} = 0[m]$ .

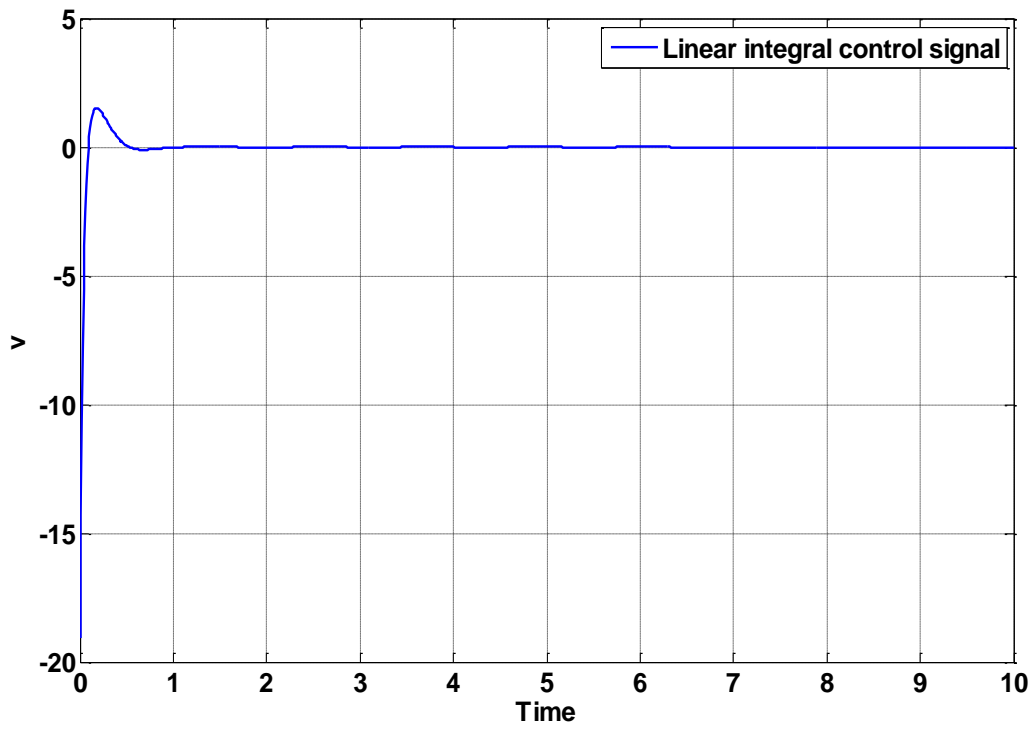


Figure 7.9: Linear control signal when the set point is  $y^{sp} = 0[m]$

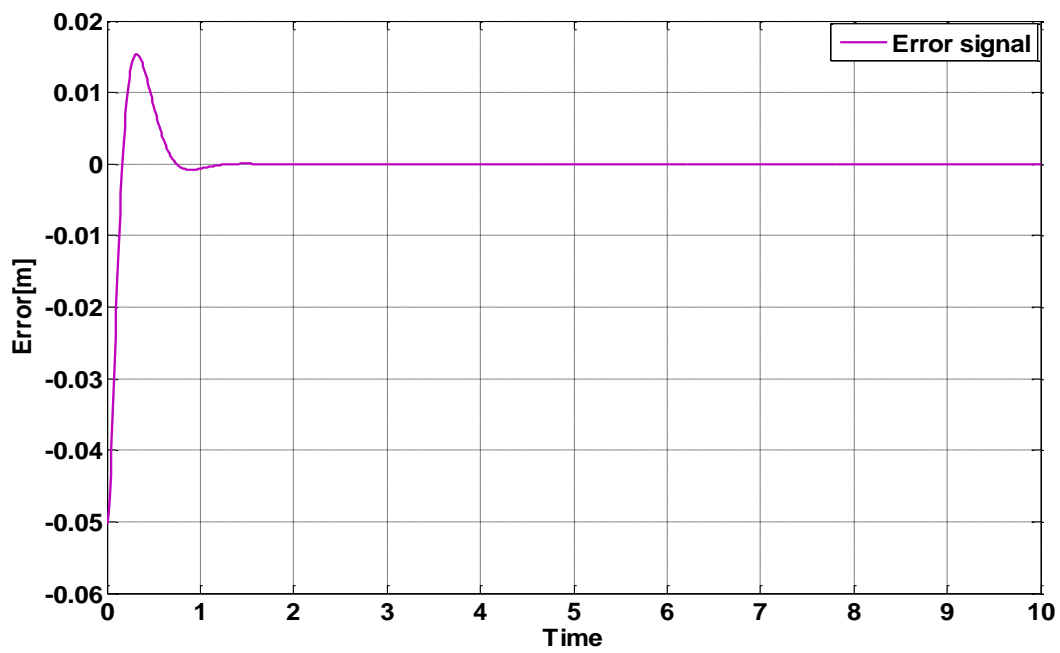


Figure 7.10: Errors between the reference model and the plant states when the set point is  $y^{sp} = 0[m]$

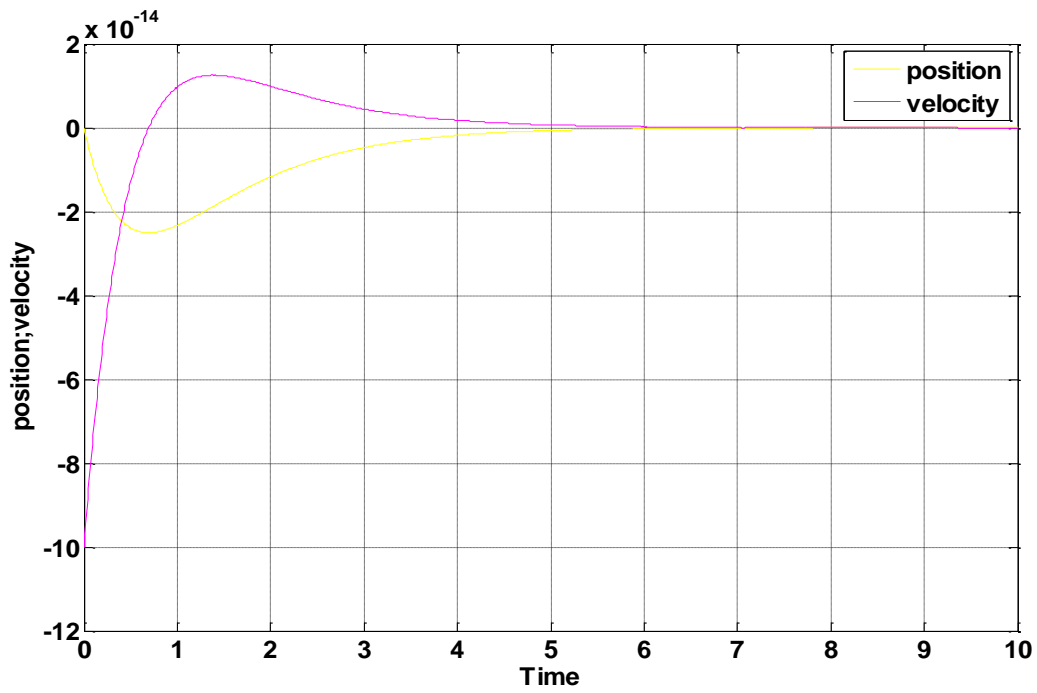


Figure 7.11: Error for the position and velocity of the magnetic levitation system when the set point is  $y^{sp} = 0[m]$

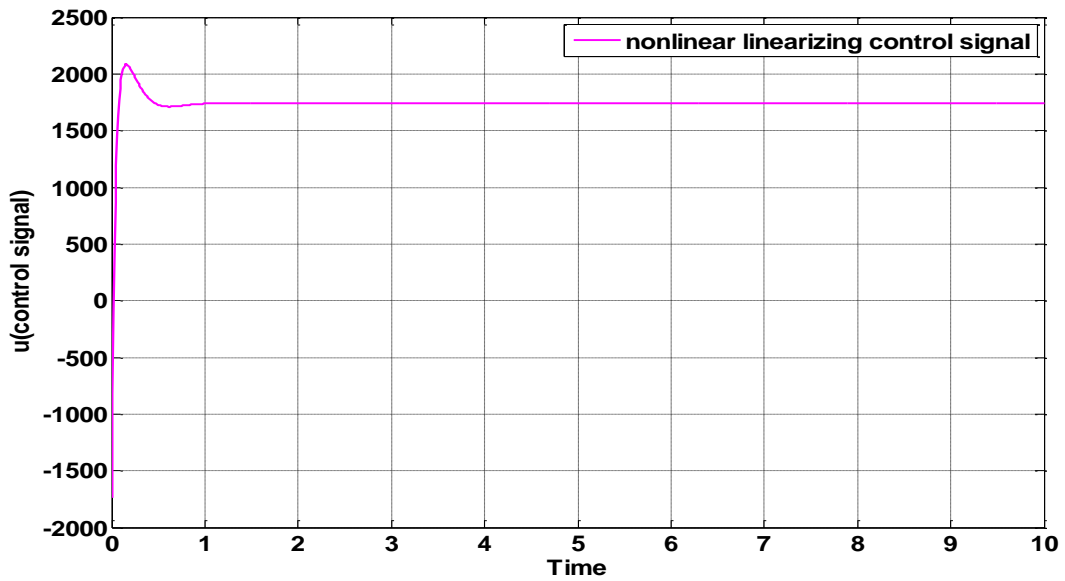


Figure 7.12: Nonlinear linearizing controller signal when the set point is  $y^{sp} = 0[m]$

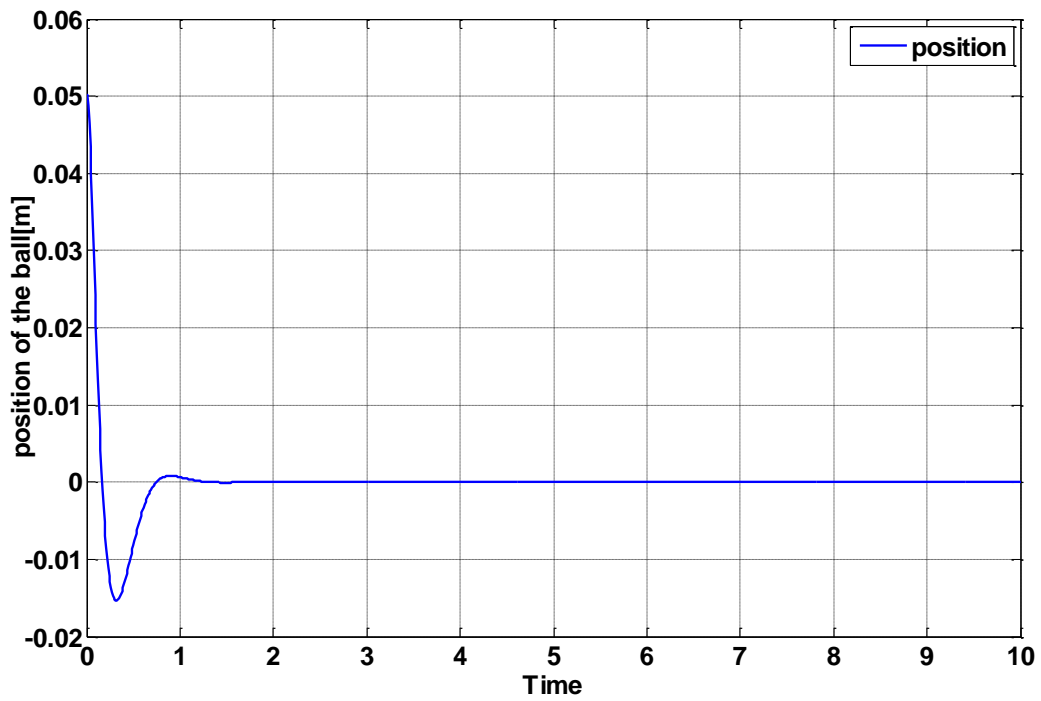


Figure 7.13: Position of the ball when the set point is  $y^{sp} = 0[m]$

- Initial conditions  $[0.05 \ 0 \ 0]'$  and  $y^{sp} = 0.01[m]$

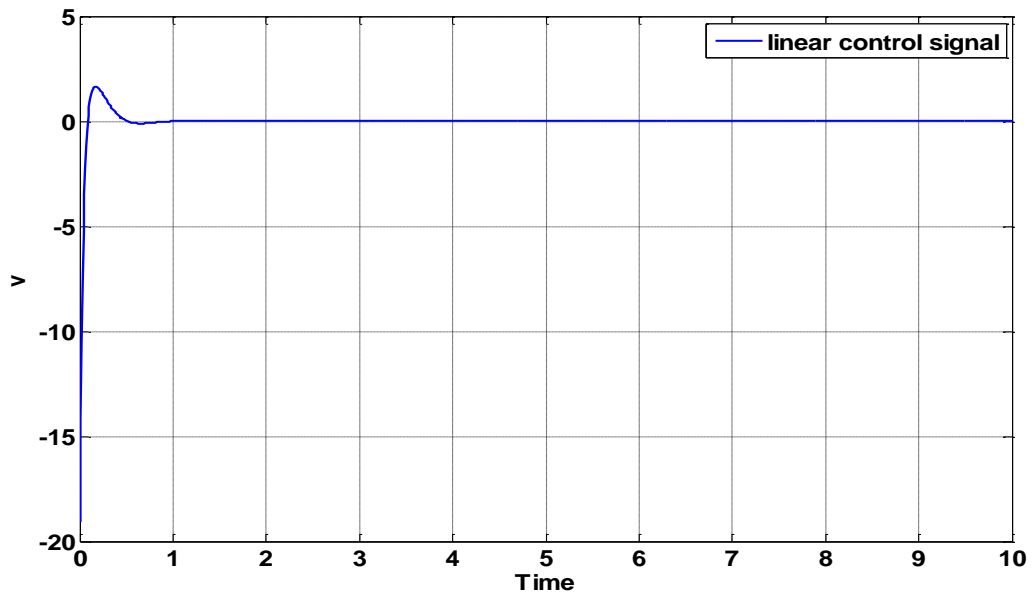


Figure 7.14: Linear control signal when the set point is  $y^{sp} = 0.01[m]$

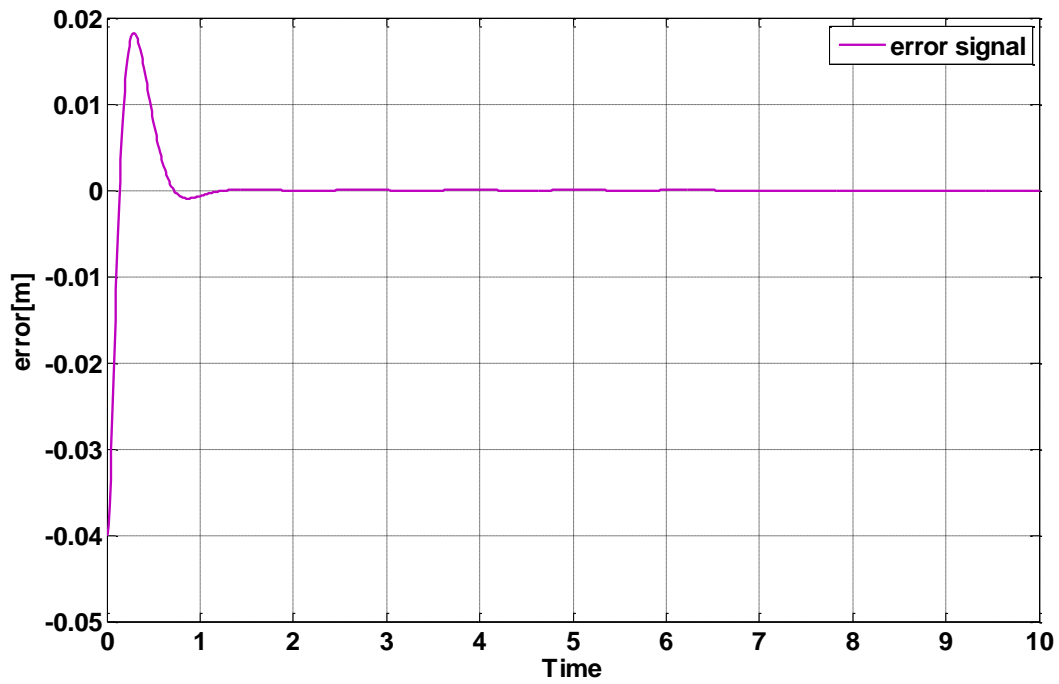


Figure 7.15: Error signal between the set point and the nonlinear plant output when the set point is  $y^{SP} = 0.01[m]$

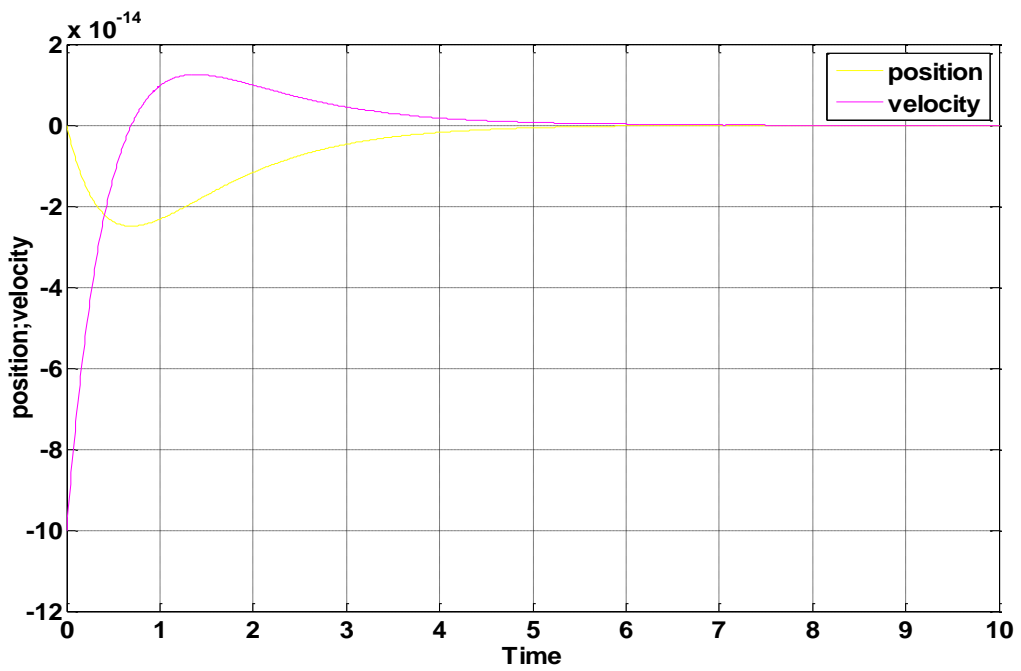


Figure 7.16: Errors between the reference model and plant states when the set point is  $y^{SP} = 0.01[m]$



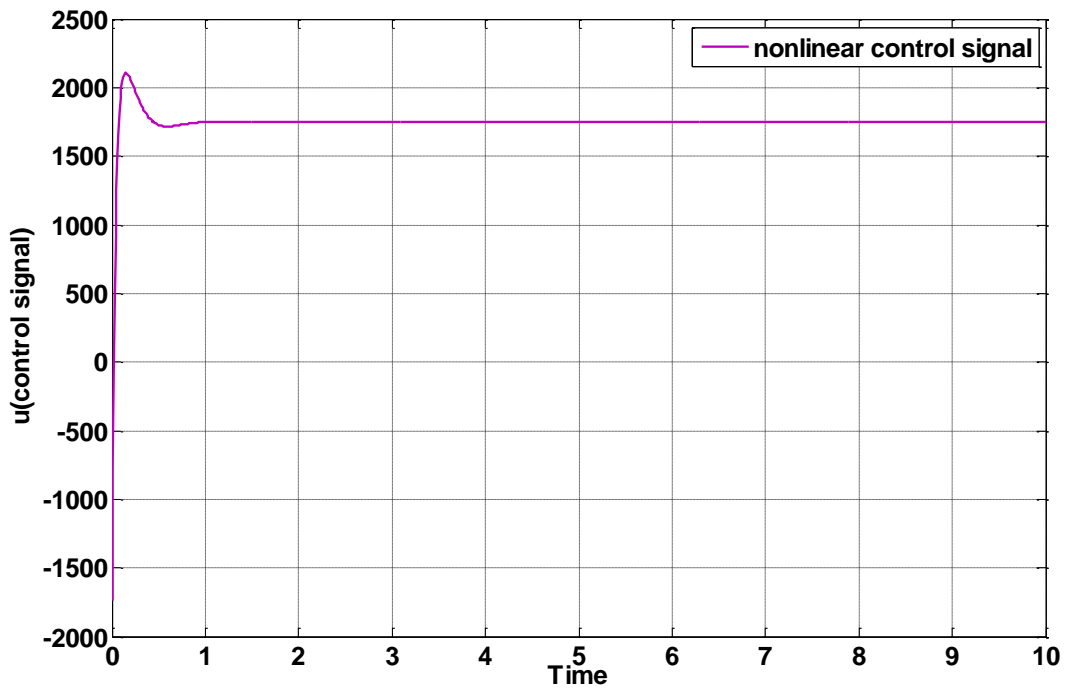


Figure 7.17: Nonlinear linearizing controller signal when the set point is  $y^{sp} = 0.01[m]$

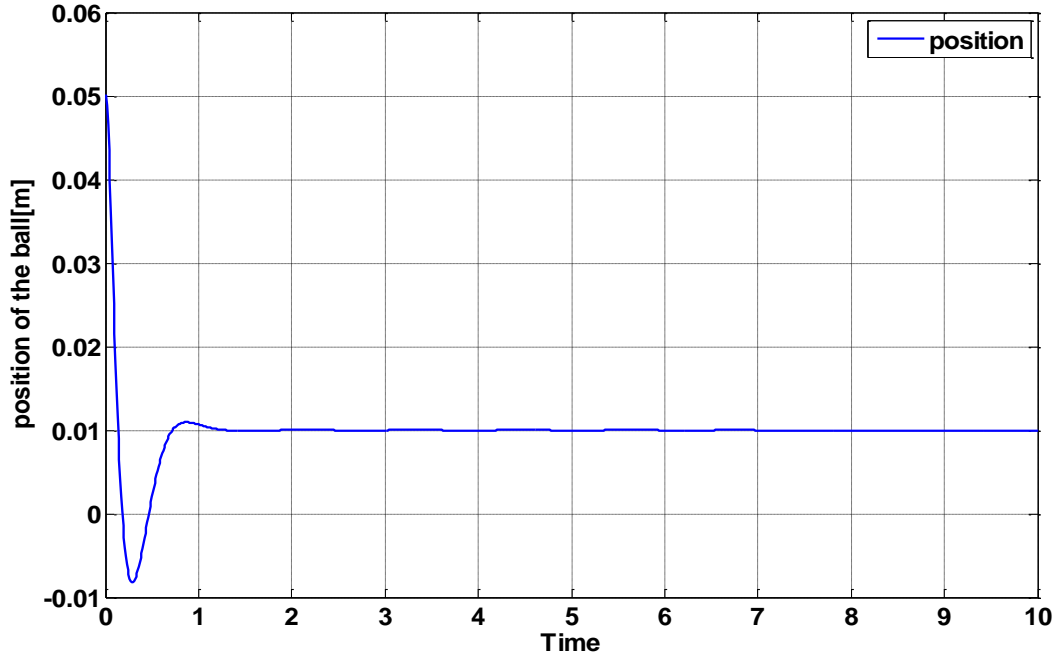


Figure 7.18: Position of the ball when the set point is  $y^{sp} = 0.01[m]$

- Initial conditions  $[0.05 \ 0 \ 0]'$  and  $y^{sp} = 0.09[m]$ :

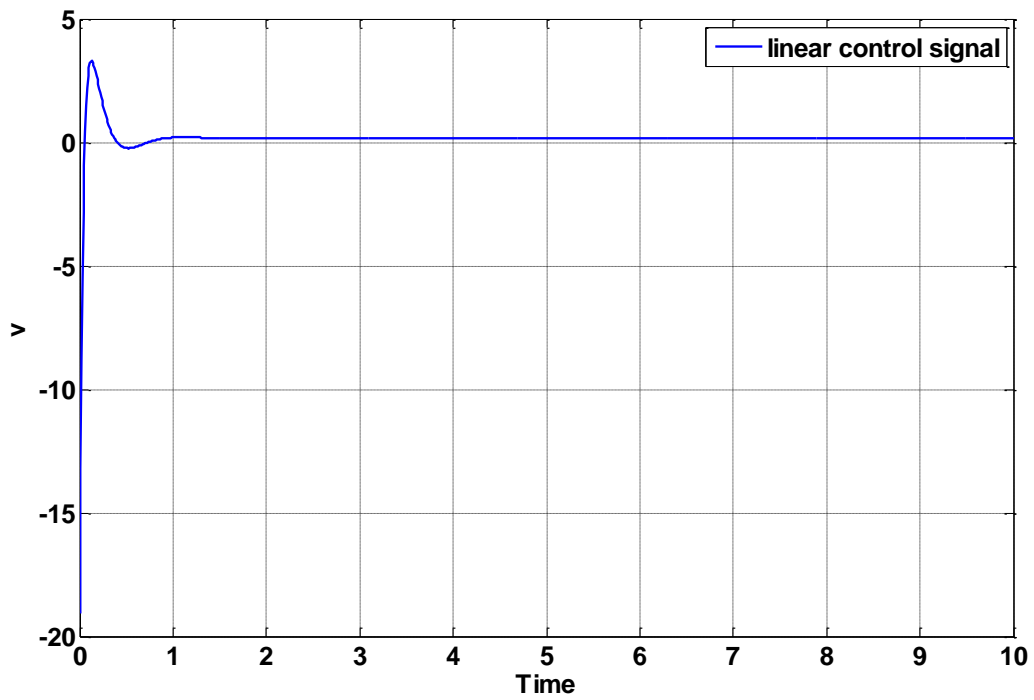


Figure 7.19: Linear control signal when the set point is  $y^{sp} = 0.09[m]$

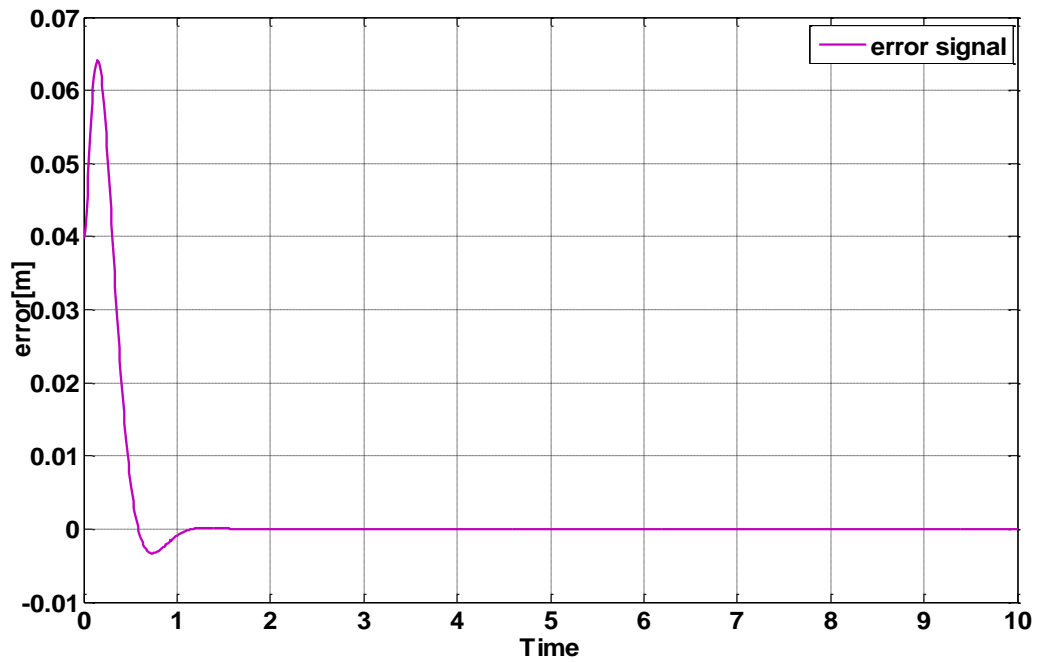


Figure 7.20: Error signal between the set point and the nonlinear plant output when the set point is  $y^{sp} = 0.09[m]$

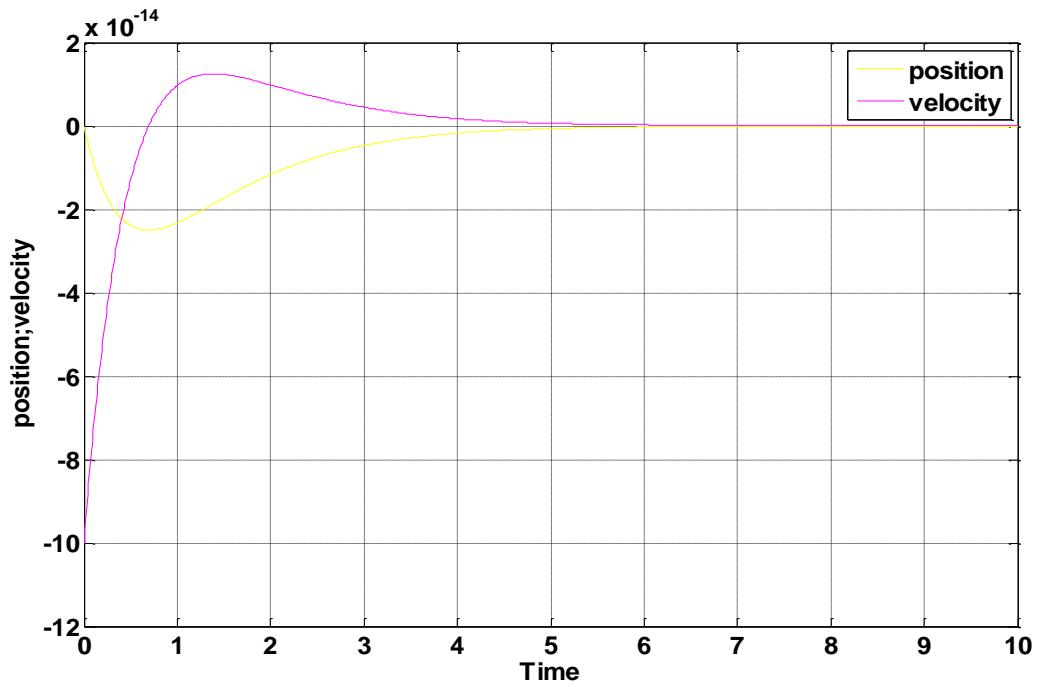


Figure 7.21: Errors between the reference model and the plant states when the set point is  $y^{sp} = 0.09[m]$

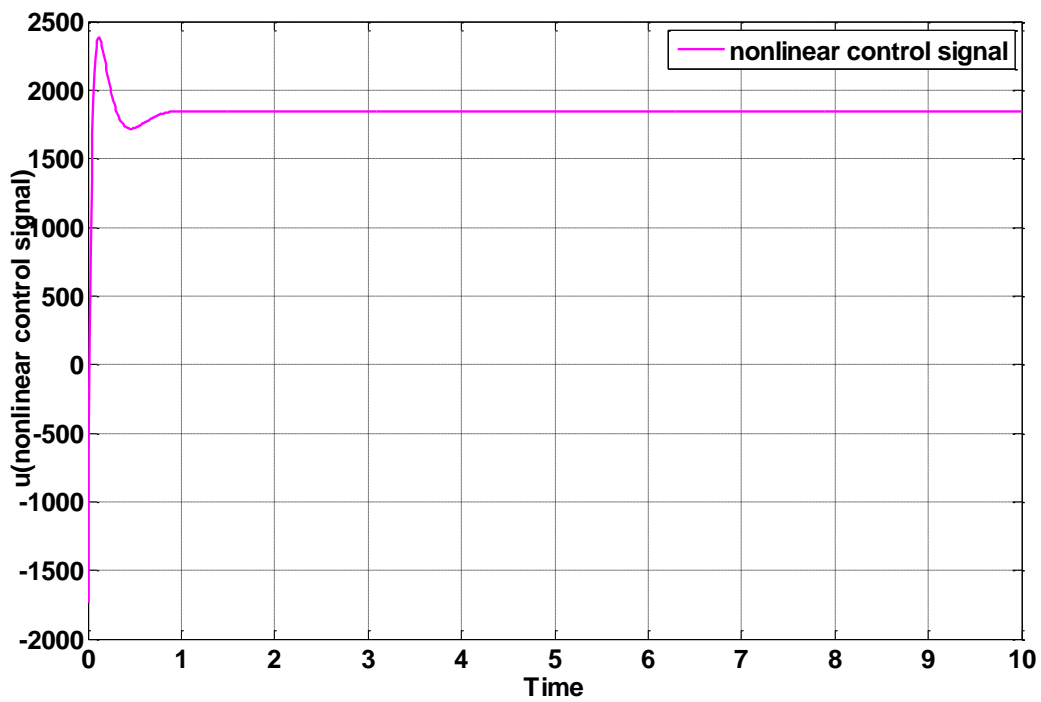


Figure 7.22: Nonlinear linearizing controller signal when the set point is  $y^{sp} = 0.09[m]$

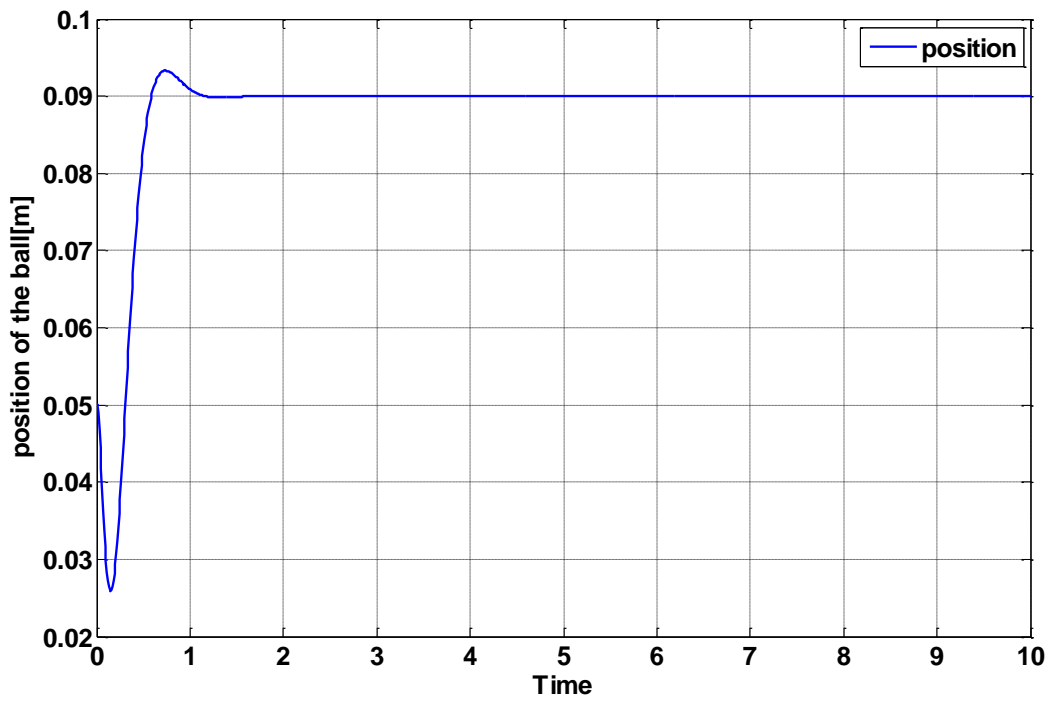


Figure 7.23: Position of the ball when the set point is  $y^{sp} = 0.09[m]$

- Initial conditions  $[0.05 \ 0 \ 0]'$  and  $y^{sp} = 0.15[m]$ :

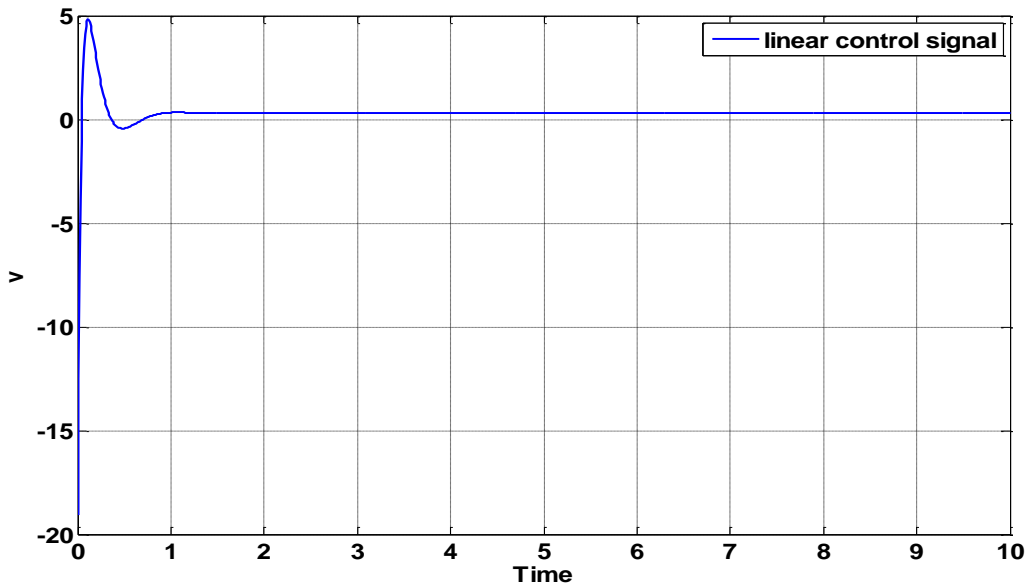


Figure 7.24: Linear control signal when the set point is  $y^{sp} = 0.15[m]$

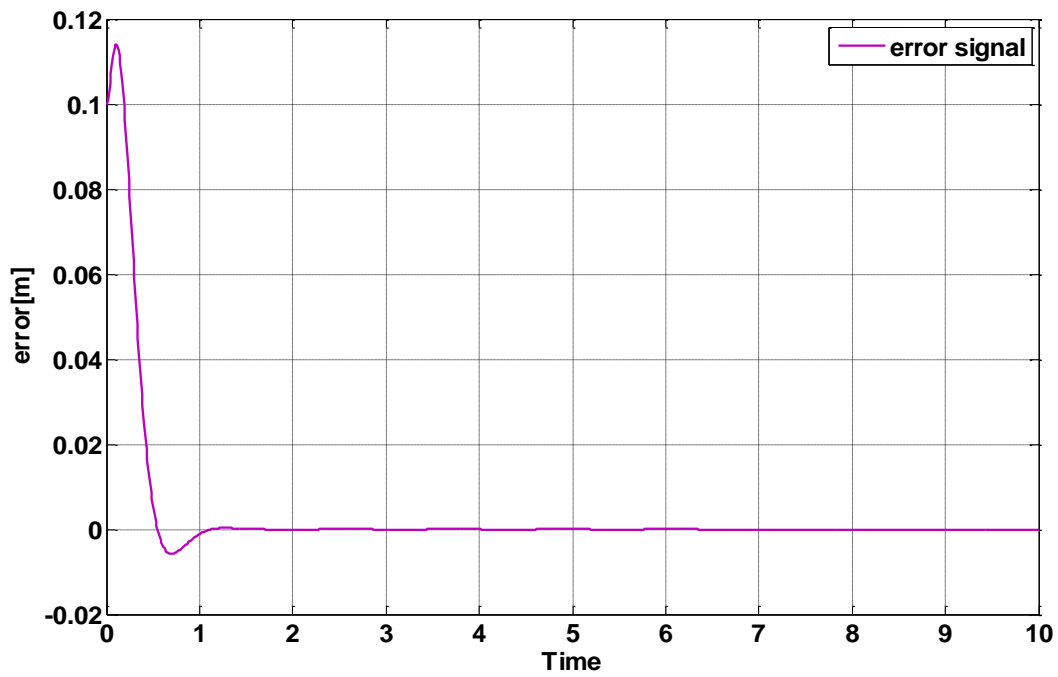


Figure 7.25: Error signal between the set point and the nonlinear plant output when the set point is  $y^{SP} = 0.15[m]$

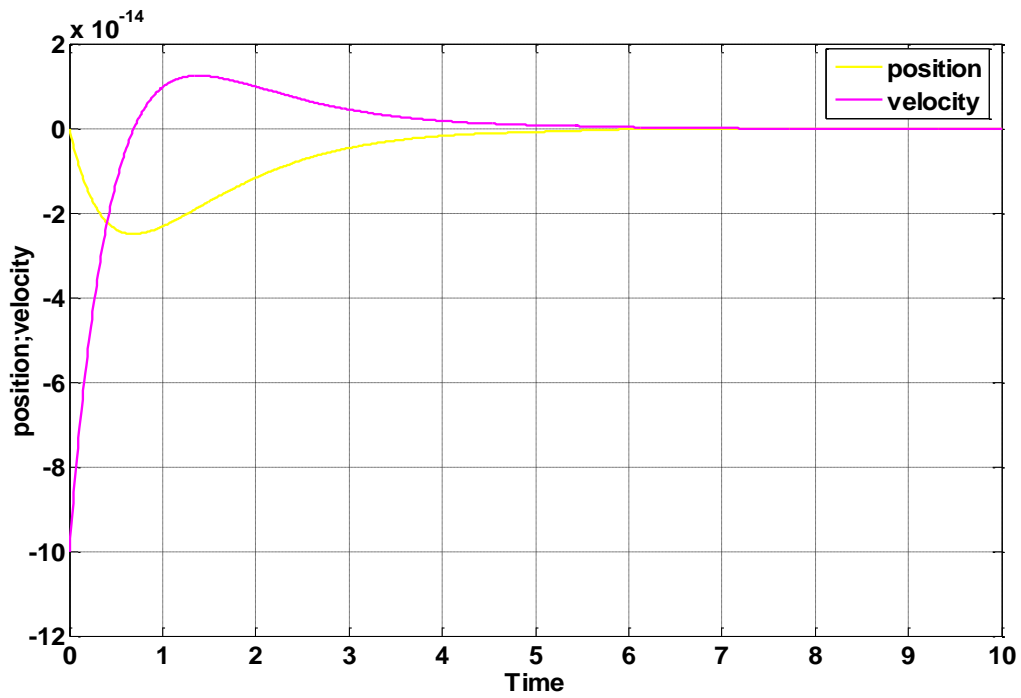


Figure 7.26: Error between the reference model and plant states when the set point is  $y^{SP} = 0.15[m]$

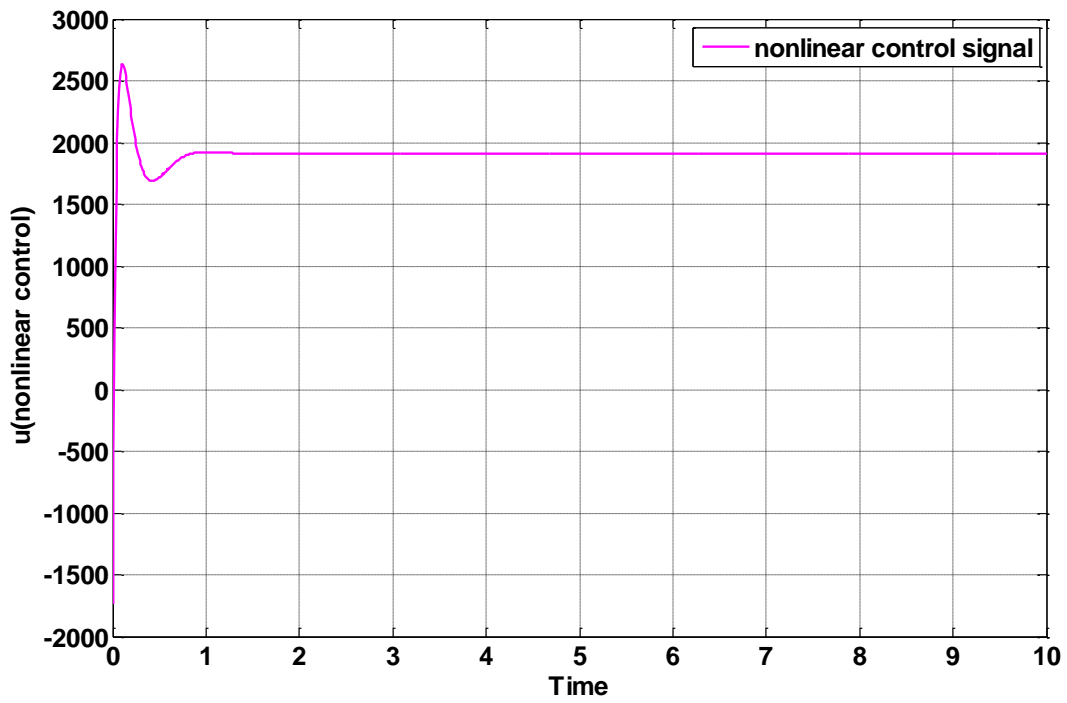


Figure 7.27: Nonlinear linearizing controller signal when the set point is  $y^{sp} = 0.15[m]$

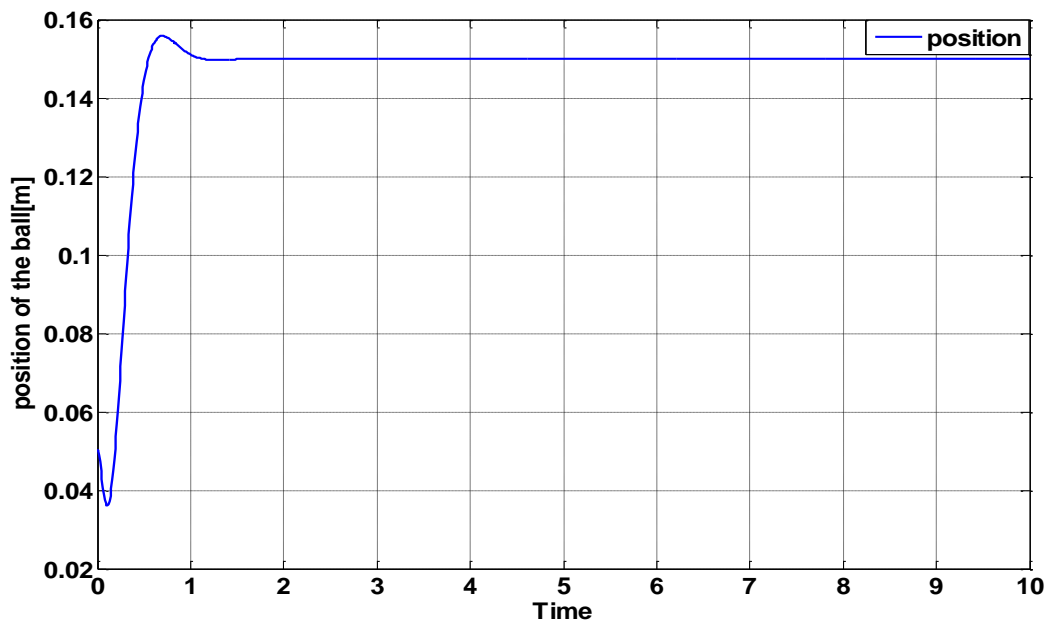


Figure 7.28: Position of the ball when the set point is  $y^{sp} = 0.15[m]$

## 7.5 Discussion of results

The different simulation results of the magnetic levitation system show the following observations:

- The system is stable.
- The errors signals go to zero.
- The plant output always follows the reference model and the set points trajectories.
- All the states of the system are stabilized.

The specifications of the dynamic output behaviour of the closed-loop system are given below in Table 7.2.

**Table 7.2: Simulation results comparison**

Set point	Characteristics	Magnetic levitation results
0	Time Delay	0
	Overshoot	0
	Rising Time	0.2s
	Steady State Error	0.001
	Settling Time	2s
0.01	Time Delay	0
	Overshoot	0
	Rising Time	0.4s
	Steady State Error	0.001
	Settling Time	2.3s
0.09	Time Delay	0
	Overshoot	1%
	Rising Time	1.3s
	Steady State Error	0.001
	Settling Time	1.7s
0.15	Time Delay	0
	Overshoot	3%
	Rising Time	0.8s
	Steady State Error	0.001
	Settling Time	1.1s

## **7.6 Conclusion**

In this chapter, the developed nonlinear linearizing controller for a reduced order of a magnetic levitation system based on Lyapunov direct method and linear reference model is described. The proper selection of the linear reference model plays a vital part in the system design as the plant output follows a desired model output. The simulation results demonstrate that the nonlinear linearizing controller based on a Lyapunov second method has been designed successfully.

The closed loop system is implemented on TwinCAT 3 engineering development platform in Chapter 8 to achieve real-time simulation using a programmable logic controller.



## **CHAPTER EIGHT: IMPLEMENTATION OF THE DEVELOPED CLOSED-LOOP SYSTEM MODELS IN A PROGRAMMABLE LOGIC CONTROLLER (PLC) BASED ON MODEL TRANSFORMATION APPROACH**

### **8.1 Introduction**

The increasing demand of low cost computers in industry has brought the development of Programmable Logical Controller (PLC). They are designed as computers for the purpose of industrial use. The first Programmable Logic Controller was developed in 1968 by General Motors as a project to substitute hard wired relay systems with much more flexible and sophisticated electronic controller (Schneider, 2010). Their utilization in the industrial sector started a year later in 1969 and since that time they have become a key component of controlling the operation of plants and machinery. From that time till today, PLCs have greatly evolved both in hardware and software (Clements and Jeffcoat, 1996). In 1974 microprocessors became the brain of the PLC and this, along with advanced electronic circuits and components helped to develop, cheaper, smaller and more reliable and powerful units. Nowadays, PLCs are vastly superior to those developed in the 1970s. This evolution is due to different factors such as:

- Hardware advances.
- New means of programming.
- Functionality.
- Communication features.
- Documenting programs.
- Fault finding.

This chapter introduces programmable logic controllers (PLCs) in section 8.2. In section 8.3 discusses PLCs hardware. Further the programming languages are discussed in section 8.4. Total Windows Control and Automation Technology (TwinCAT) development software is presented in section 8.5. Beckhoff CX5020 PLC used in conjunction with TwinCAT 3 is presented in section 8.6. Section 8.7 discusses the communication method to link the development PC and the PLC. Model transformation of the closed loop systems developed in Chapters 6 and 7 are implemented and simulated in real-time in section 8.8. Section 8.9 presents a distributed implementation of two closed-loop models of the magnetic levitation systems operating with different parameters and finally section 8.9 is the conclusion.

## 8.2 Programmable Logic Controller

PLC is a device consisting of a programmable microprocessor able to store commands and perform the implementation of functions such as logic, arithmetic, sequencing, timing and counting in order to control processes or machineries, and is programmed using a specialized computer language. PLCs are used principally to replace electromechanical relay systems which are not programmable. In the early years of PLCs, they were programmed in ladder logic, which is similar to schematic of relay logic. More advanced PLCs are programmed most of the time in languages, starting from ladder to C or Visual Basic. Generally, writing the program is done in a special development environment installed on a personal computer (PC), and later downloaded onto a PLC platform straight through a cable connection. The written program is then stored in the programmable logic controller through a non-volatile memory (Jack, 2008).

PLCs have the following advantages (Bolton, 1996):

- Cost effective for controlling complex systems.
- Easier to troubleshoot.
- Answer more rapidly than computers.
- Remote control capability.
- Enhance reliability.
- Communication capability.
- More flexibility.

PLCs disadvantages are:

- Most PLCs manufacturers offer only closed architecture for their products.
- Portability which consists of making software tools able to accept and correctly interpret vendors' libraries elements from different manufacturers is still an issue.

PLCs have the following characteristics:

- Interfaces between inputs and outputs are already into the controller.
- Easily programmed and possess comprehensive programming language which fundamentally deals with operations that involve switching and logic.
- They are robust and designed to resist very hostile environments.

Programmable Logic Controllers have inputs and outputs ports that are typically based on Reduced Instruction Set Computer (RISC). They are built for real-time applications, and must

always resist hostile environment on the floor. PLC circuitry oversees the positions of several sensor inputs, and controls output actuators, like lights, valves, solenoids and motors starters. In the factory automation, programmable logic controller has made a very big contribution. In the past, automation systems were using thousands of individual relays and timers. Nowadays, timers and relays within a factory are replaceable by one PLC. In modern industry, programmable logic controllers provide a vast range of functionality, in the industry of basic relay, motion control, process control and much more difficult networking, as well as being in distributed systems as a measure of control. Programmable logic controllers see digital signals that give an on and off signal as Boolean values. Analog signal can also be employed, from devices like volume controls, and these signals are identified by programmable logic controllers as floating point values.

Operators have a wide range of interfaces to choose from and interact with programmable logic controllers. These interfaces may take the forms of simple light bulbs, text displays, switches, or more challenging systems, a computer Web interface and a Supervisory Control and Data Acquisition (SCADA) system. Short description of the PLC hardware and software programming languages are given in appendices A.7, A.8 and A.9.

### **8.3 Programmable Logic Controller Hardware**

A programmable logic controller is made of the following basic components:

- Programming devices (Personal computer).
- Central Processing Unit (CPU).
- Power Supply (PS).
- Input and Output modules (both Discrete and Analog).

#### **8.3.1 Programming Devices**

The programming devices are used to enter the necessary programs into the memory of the processor.

#### **8.3.2 Central Processing Unit**

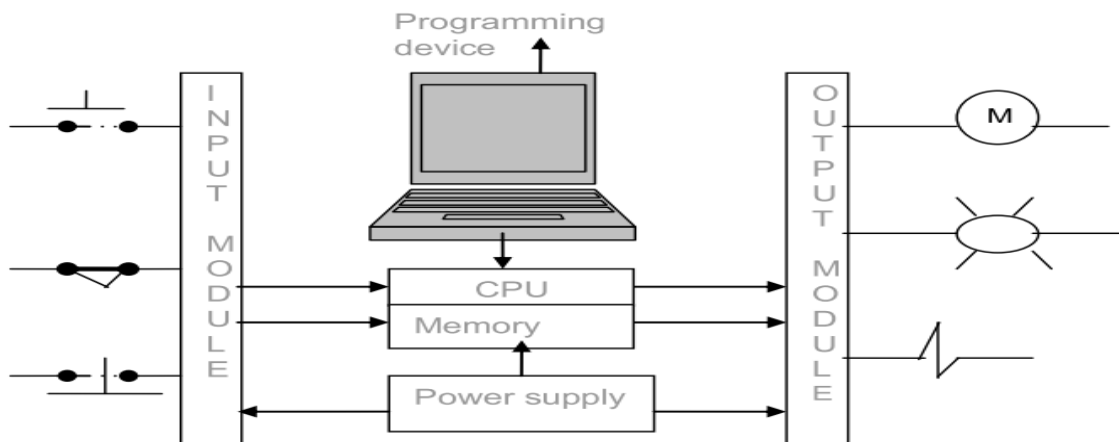
The CPU is the brain of the PLC. It consists of one or more microprocessor chips with all the required support to allow communication with the programmer terminal, the inputs, the outputs, and the memory.

### 8.3.3 Power Supply

The power supply is a unit that is required to transform the main A.C voltage to low D.C voltage of about 5V needed for the processor and the circuits in the interface modules of the input and output.

### 8.3.4 Input and Output Modules

In a programmable logic controller, the inputs and outputs modules are the connections to the real world. Input devices such as push buttons, switches and sensors accept signals from the machine and convert them into signal that can be used by the controller, which means that the input interface transmits status information regarding the processes to be sent to the CPU. Outputs devices such as relay contacts, solenoid valves, signal devices and motors convert controller signals that can be used to control the machine or process, which allows the CPU to communicate operating signals to the process devices under its control. The block diagram of the standard components of a PLC is presented in Figure 8.1 (NIIT, 2005).



**Figure 8.1: Block diagram of the Standard components of a PLC (NIIT, 2005)**

### 8.4 Programming languages

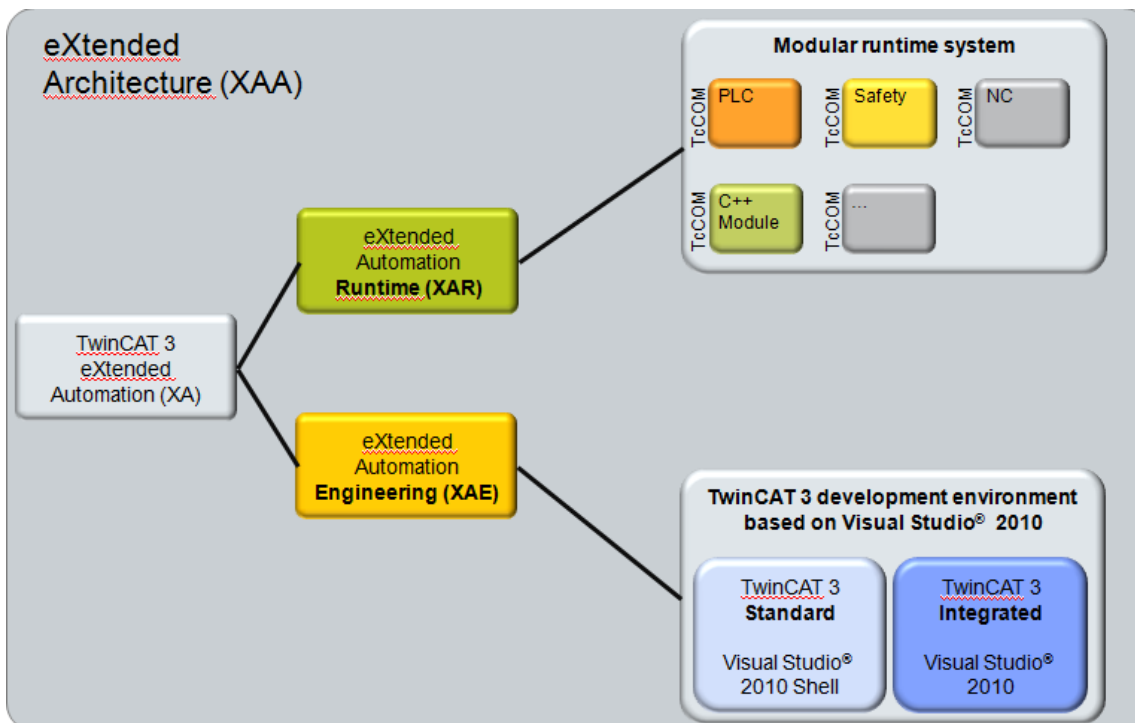
PLC programs can be developed in one or more of the languages recognized in the IEC 61131-3 standard. The languages used to program PLCs are the following (Jack, 2008):

- Structured Text (ST)
- Ladder Diagram (LD)
- Function Block Diagram (FBD)
- Instruction List (IL)

- Sequential Function Chart (SFC)

### 8.5 Total Windows Control and Automation Technology (TwinCAT) development software

The Total Windows Control and Automation Technology (TwinCAT) is a software platform developed by Beckhoff which enables the user to deal with complex control system applications. It turns almost any compatible personal computer (PC) into a real-time controller with a multi PLC system. TwinCAT PLC offers all types of programming languages of IEC 61131-3 standard and possesses an extremely advanced development environment for programs whose code size and data regions go beyond the memory sizes of conventional PLC systems. Figure 8.2 shows the architecture of the version 3 of TwinCAT software (Beckhoff, 2013) which is the development environment used in this thesis.



**Figure 8.2:** TwinCAT 3 extended architecture (TwinCAT, 2013)

TwinCAT3 is composed of two platforms:

- eXtended Automation Engineering (XAE)
- eXtended Automation Runtime (XAR)

### **8.5.1 eXtended Automation Engineering (XAE)**

The main approach of TwinCAT 3 is to simplify the software engineering environment. It is a development environment done on Microsoft Visual studio and has the following characteristics:

- Visual Studio 2010 for real time programming implementation in IEC 61131-3 and C/C++.
- Visual Studio 2010 for the configuration of the complete system.

TwinCAT 3 eXtended Automation Engineering has the following advantages:

- One programming environment
- One project folder
- One debugging environment for IEC and C/C++ code
- Real-Time Programming in C++
- Link to Matlab/Simulink
- TwinCAT 2 PLC projects can be converted into a TwinCAT 3 PLC project
- The 'old' TwinCAT system Manager file (\*.tsm file) can be converted
- Embedded in Microsoft Visual Studio 2010.

Depending on the system level TwinCAT 3 XAE complies with the following system requirements:

- Windows XP with SP3 (x86) or Windows 7 (x86 or x64)
- 2 GB RAM
- Processor running at 1,6 GHz or higher
- 3 GB free hard disk space if Visual Studio 2010 shell not already installed
- Graphics adapter supporting DirectX9, running at a minimum resolution of 1024x768
- 500 MB free hard disk space if Visual Studio 2010 shell already installed

TwinCAT 3 eXtended Automation Engineering software is made of two variants (TwinCAT, 2012):

- TwinCAT 3 standard
- TwinCAT 3 Integrated

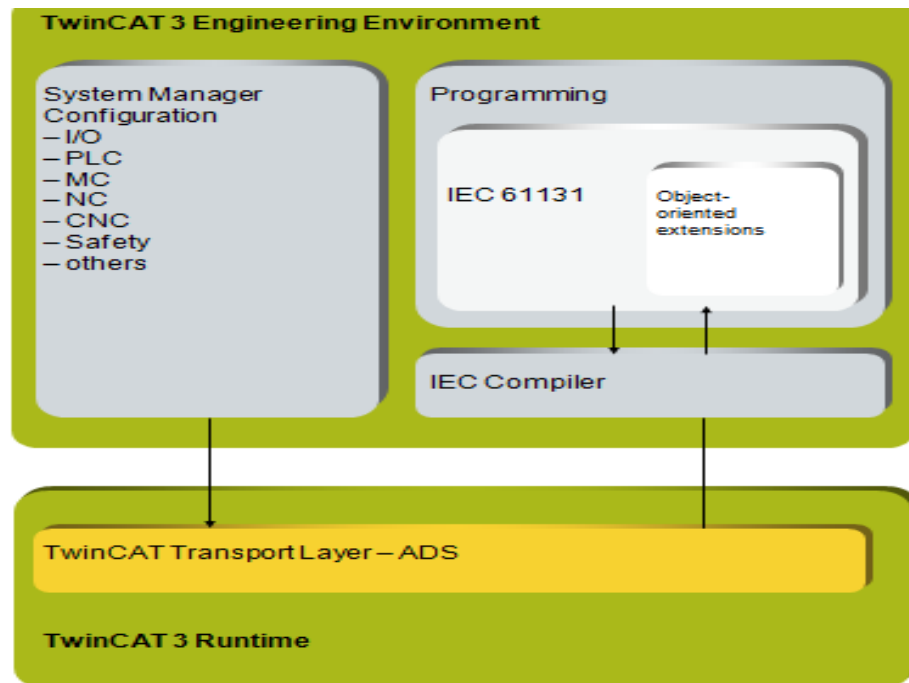
#### **8.5.1.1 TwinCAT 3 standard**

The TwinCAT 3 standard is for PLC programmers and users of existing, compiled modules such as Matlab/Simulink and C/C++. It allows the configuration, parameterisation and diagnosis, as well as debugging of the PLC code.

The TwinCAT 3 standard is based on four parts:

- Microsoft Visual Studio Shell
- Integrated System Manager
- Integrated IEC 61131-3<sup>rd</sup> edition programming
- Integrated Safety PLC

Figure 8.3 shows the architecture of the TwinCAT 3 standard (TwinCAT, 2012).



**Figure 8.3:** TwinCAT 3 standard (TwinCAT, 2012)

### 8.5.1.2 TwinCAT 3 integrated

The characteristics of TwinCAT 3 integrated are as follow:

- Microsoft Visual Studio 2010 existing integration
- Integrated system manager (Figure 8.4)
- Integrated IEC 61131-3<sup>rd</sup> edition programming
- Integrated Safety PLC
- C/C++ programming
- Matlab/Simulink
- C# and .NET programming for non real time applications
- Optional: integration of third party software tool

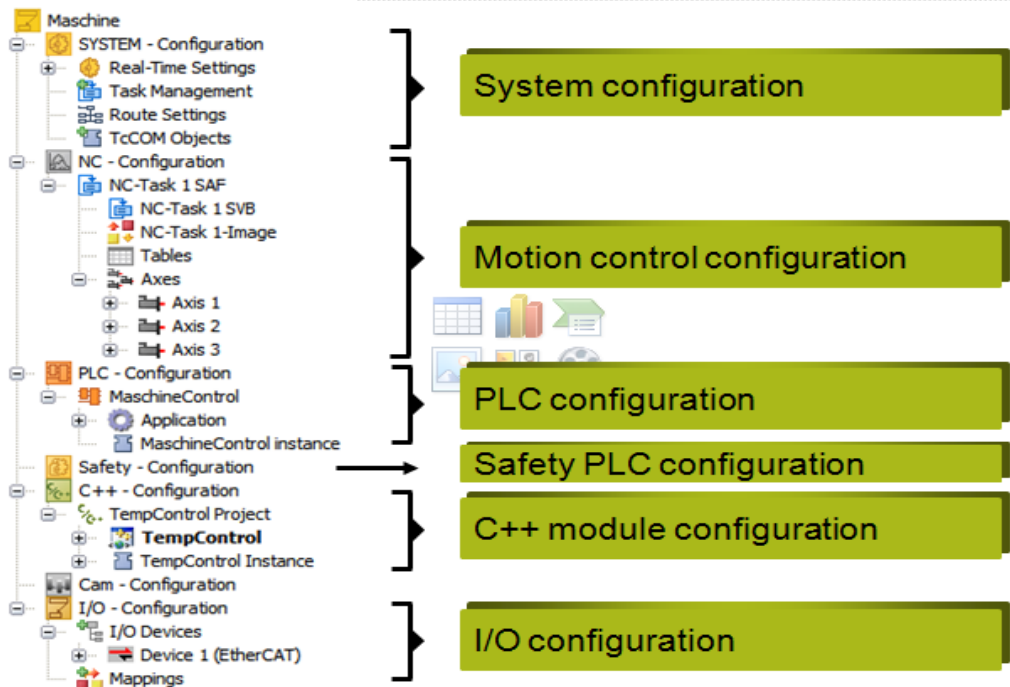


Figure 8.4: Integrated system manager directory tree (TwinCAT, 2012)

The architecture of TwinCAT 3 integrated is presented in Figure 8.13.

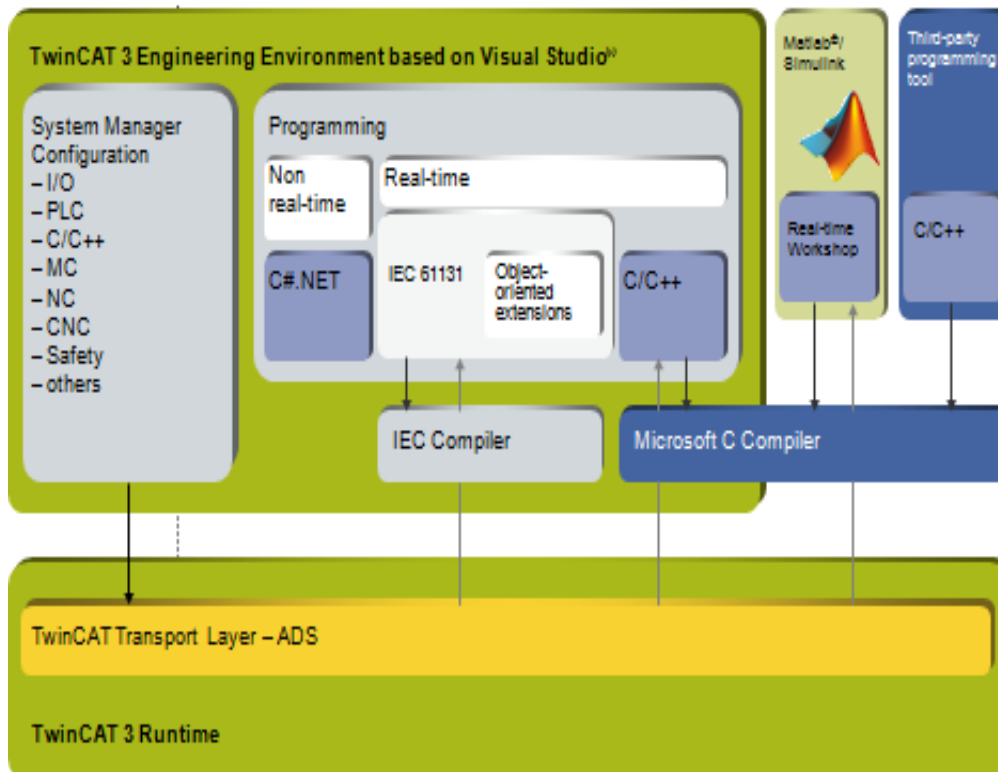
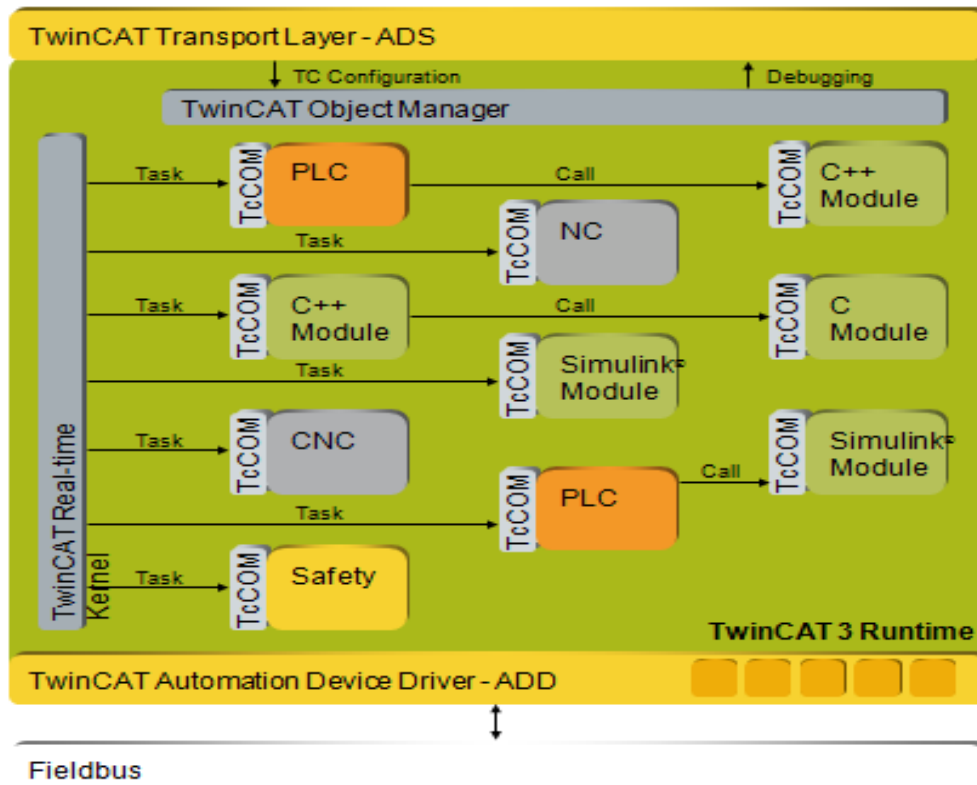


Figure 8.5: TwinCAT 3 integrated (TwinCAT, 2012)



### 8.5.2 eXtended Automation Runtime (XAR)

TwinCAT 3 eXtended Automation Runtime is a real-time environment in which modules developed on TwinCAT can be loaded, executed or administrated. One of the advantages of the XAR is that the individual modules can be developed with different compilers and therewith can be programmed independently and by different manufacturers or developers. Figure 8.6 presents TwinCAT XAR modular runtime system. The XAR generates cyclic modules from the task and a wide range of tasks can run on one control PC.



**Figure 8.6:** TwinCAT 3 runtime modular system (TwinCAT, 2014)

TwinCAT 3 XAR supports a multicore central processing unit (CPU) which allows individual tasks to be allocated on different core of the CPU. This feature makes the newest multicore industrial and embedded personal computer (PC) achieves higher performance. Figures 8.7 and 8.8 show the architecture of TwinCAT 3 multi-core support and the multicore assignment (TwinCAT 3, 2012).

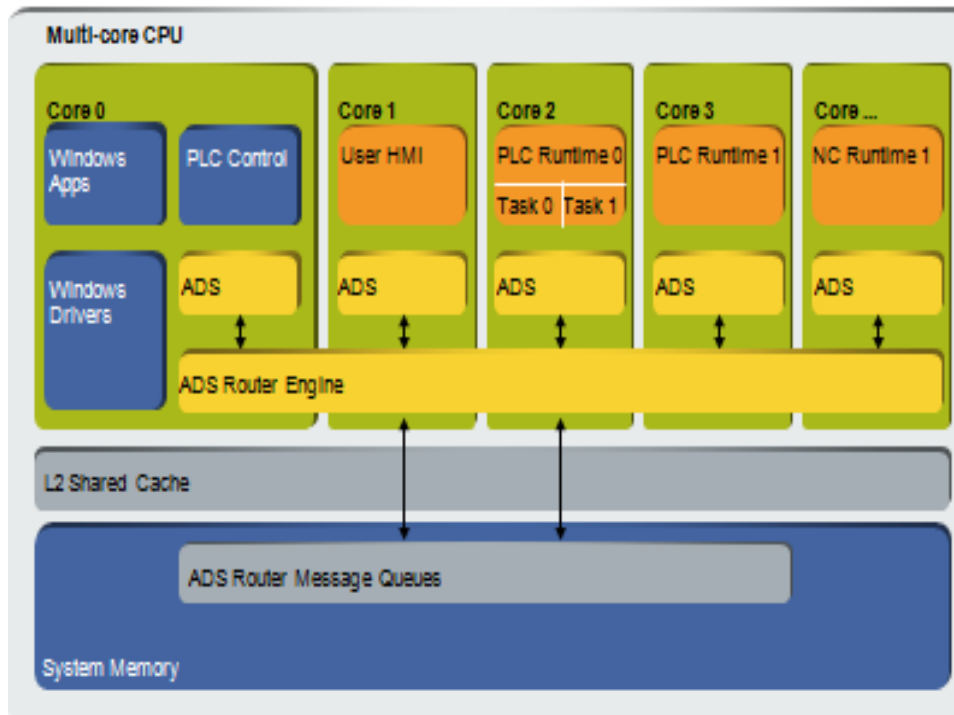


Figure 8.7: TwinCAT 3 multicore support (TwinCAT 3, 2014)

Activation of the cores

Definition of the CPU limits

Definition of the basic time

Assignment of the task to the core/CPU

CPU	RT-CPU	Base Time	Fast Clock	CPU Limit	Latency Warning
0	<input checked="" type="checkbox"/>	1 ms	(none)	80 %	(none)
1	<input type="checkbox"/>				
2	<input type="checkbox"/>				
3	<input type="checkbox"/>				
4	<input type="checkbox"/>				
5	<input type="checkbox"/>				
6	<input type="checkbox"/>				
7	<input type="checkbox"/>				

Type	Object	RT-CPU	Base Time	Cycle Time	Cycle Ticks	Priority
TASK	PlcTask	Default (0)	1 ms	10 ms	10	20

Figure 8.8: TwinCAT 3 multicore assignment (TwinCAT 3, 2013)

## 8.6 Beckhoff CX5020 PLC

The Beckhoff CX5020 PLC is an embedded personal computer (PC) (CX5020, 2013); it can be used for the implementation of PLC or PLC/Motion control project. The CX5020 is used in synchronicity with TwinCAT 3 software from Beckhoff and offers the same functionalities as large industrial personal computers (PCs). In terms of PLC, it possesses up to four virtual IEC 61131 CPUs that can be programmed to up to four tasks each, with a minimum cycle time of 12.5 $\mu$ s. It has a wide range of operating temperatures that varies between -25 and 60°C to enable applications in climatically challenging conditions.

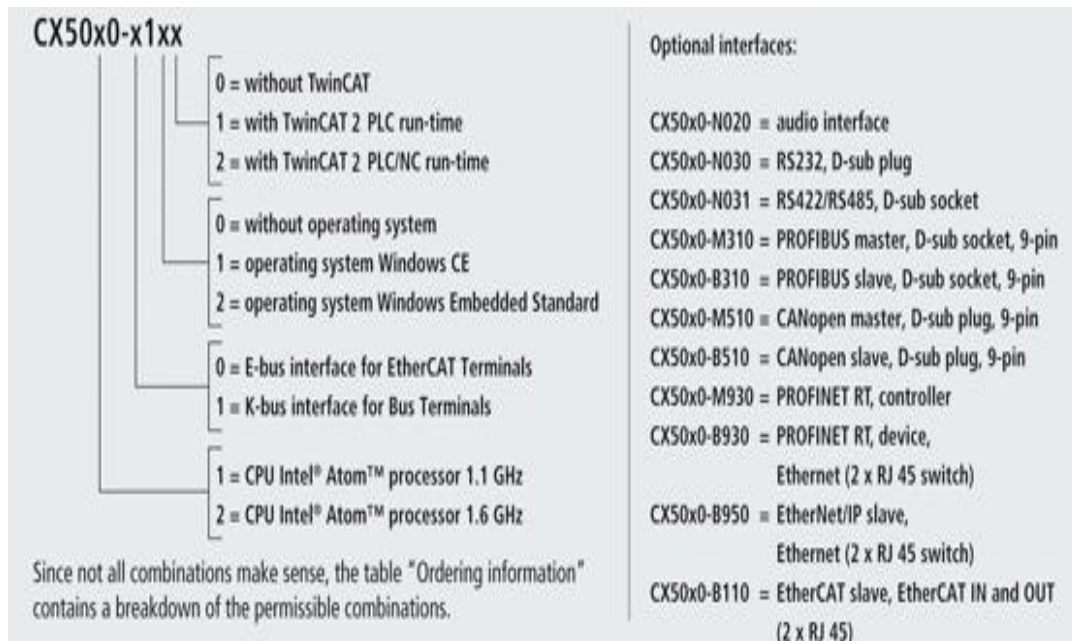
The CX5020 is DIN rail mountable, fanless embedded PC with direct connection for Beckhoff Bus Terminals or EtherCAT terminals. Table 8.1 shows a technical datasheet of the CX5020.

**Table 8.1: CX5020 technical data (CX5020, 2013)**

Technical data	CX5020-x1xx
Processor	processor Intel® Atom™ Z530, 1.6 GHz clock frequency
Flash memory	64 MB Compact Flash card (optionally extendable)
Internal main memory	512 MB RAM (optionally 1 GB installed ex factory)
Persistent memory	Integrated 1sec UPS (1 MB on Compact Flash card)
Interfaces	2 x RJ 45, 10/100/1000 Mbit/s, DVI-D, 4 x USB 2.0, 1 x optional interface
Diagnostics LED	1 x power, 1 x TC status, 1 x flash access, 2 x bus status
Clock	internal battery-backed clock for time and date (battery exchangeable)
Operating system	Microsoft Windows CE 6 or Microsoft Windows Embedded Standard 2009
Control software	TwinCAT 2 PLC runtime, TwinCAT 2 NC PTP runtime or TwinCAT 3
Power supply	24 V DC (-15 %/+20 %)
Dielectric strength	500 V (supply/internal electronics)
Current supply I/O terminals	2A
Max. power loss	12.5 W (including the system interfaces)

Dimensions (W x H x D)	100 mm x 100 mm x 91 mm
Operating/storage temperature	-25...+60 °C/-25...+85 °C
Relative humidity	95 %, no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2/EN 61000-6-4
Protection class	IP 20
TC3 performance class	Performance (40)

CX5020 offers a wide range of options depending on the process to be controlled and communication protocol to be used to link the PLC to the real plant. Figure 8.9 presents the different types of CX5020.



**Figure 8.9:** Order of different CX5000 devices (Beckhoff, 2013)

Beckhoff PLC CX5020 has the advantage of having almost unlimited number of inputs and outputs cards, the user makes the decision to select the number of inputs and outputs required in his applications. A clear diagram of the CX5020 is presented in Figure 8.10.

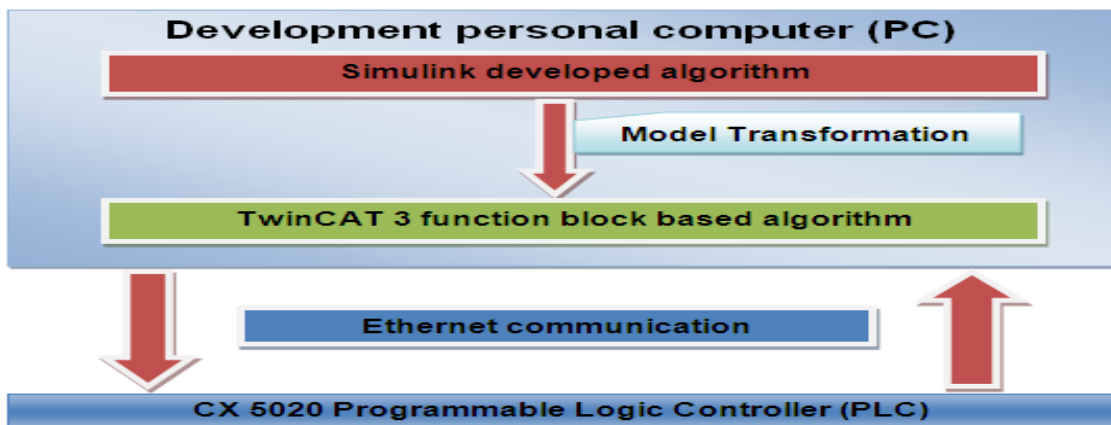


**Figure 8.10: Diagram of the CX5020**

### 8.7 Algorithm for real-time communication

In this thesis, the closed-loop simulation model is transformed from Matlab/Simulink environment to TwinCAT 3 via full model transformation. To achieve real-time simulation of the closed-loop system obtained in TwinCAT 3, the closed-loop algorithm is downloaded into the PLC CX5020.

The CX5020 serves as a real-time platform to execute the application developed in TwinCAT3. The most important factor to link the development Persona Computer (PC) and the PLC is the communication driver module. Ethernet is used as a mean of communication to link the personal computer used as development platform and the PLC via fast duplex communication. Ethernet module scans the PLC continuously for values of the input variables and writes them in an appropriate database in TwinCAT 3. Figure 8.11 shows the closed-loop algorithm for real-time communication.



**Figure 8.11: Closed-loop algorithm for real-time communication**

## 8.8 Real-Time Implementation

Based on the study of model transformation with TwinCAT 3 software, Beckhoff PLC CX5020 hardware and configurations, the real-time implementations are executed on these two platforms by using the developed control algorithms in the previous chapters, including: the input-state/input-output feedback linearization and the Lyapunov stability-based nonlinear MRC. The TE1400 module of TwinCAT 3 software transforms Simulink blocks to PLC function blocks. The transformation software automatically translates each state flow block into customized basic IEC 61499 function block with exactly, the same inputs, outputs and parameters as their Simulink counter parts. The advantages of this solution in TwinCAT 3 are:

- No Beckhoff specific blocks are required in Simulink for the module generation.
- Graphical representation used in the TwinCAT 3 engineering environment can be used.
- Call of the TwinCAT 3 module out of a task or other modules is possible.
- Support of all toolboxes which are supported by the Simulink Coder.
- Access to the process image of the TwinCAT 3 module by mappings of the transformed model.
- Changing of the developed module cycle time in TwinCAT 3 without a new compiler is needed.

The amount of time for performing model transformation depends on the complexity of the model in Simulink (Vyatkin and Yang, 2010). The transformation technique has some limitations such as:

- Function Blocks are not event driven.
- The Simulink solver type is set at a fixed step instead of a variable step.
- The fundamental sample time is at 10ms.

### 8.8.1 Real-Time Implementation on the CX5020 and PC real time platform

The models of the closed-loop systems developed in Matlab/Simulink software environment based on the control algorithm and Simulation block diagram developed in Chapter 6 and Chapter 7 are transformed and implemented in the PLC CX5020 and PC real-time implementation platform. Further, they are simulated in real-time. The steps of the procedure are discussed below.

### 8.8.1.1 Real-Time Implementation of the closed-loop system model based on feedback linearization system

Real-time implementation of the closed-loop system based on feedback linearization method is done according to the full model transformation approach. A code generation is done to transform the Simulink program developed in Figure 6.4 of Chapter 6 to a real-time algorithm deployed on the PLC.

#### 8.8.1.1.1 Code generation process for the closed-loop system based on feedback linearization control

The code generation to move the closed-loop system model from a Simulink program to a TwinCAT 3 one is as follow (TwinCAT, 2013):

- Design and build the system model on Simulink
- Automatic generate of C/C++ code by the Simulink Coder of the developed model
- Compilation with Visual Studio C Compiler
- Parameterize the generated code in the TwinCAT System Manager
- Download and extend the developed model in the TwinCAT 3 runtime.

Figure 8.12 shows the code generation process.

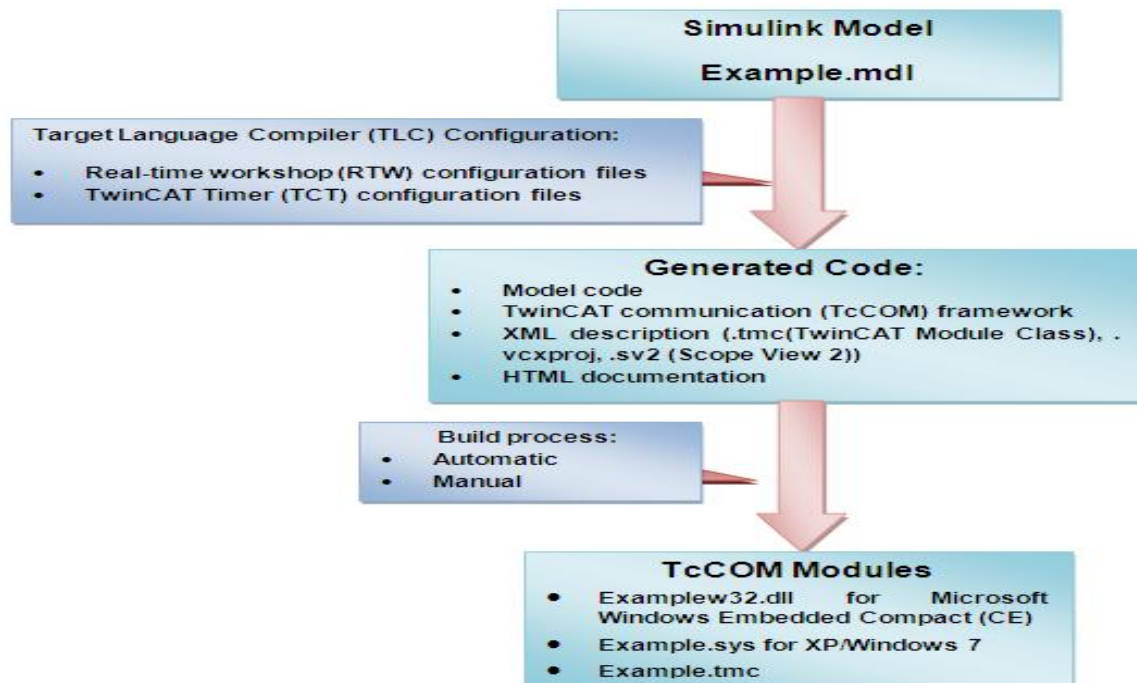


Figure 8.12: Code Generation process

### 8.8.1.1.2 Requirements for proper model transformation and simulation

To achieve model transformation of the Lyapunov model, it is very important to ensure the correctness of the model developed in Simulink. The wrong mapping of the data type will make the algorithm for computation to behave incorrectly. Simulink and TwinCAT 3 have their own primitive data type defined. Table 8.2 gives an overview of the data type mapping between both environments.

**Table 8.2: Primitive data mapping between Simulink and TwinCAT3**

Data Type	Comment	Memory Space	TwinCAT 3 Support	Simulink Support
byte	short integer	8 bits	SINT	int8
Int	integer	16 bits	INT	int16
int	double integer	32 bits	DINT	int32
short	Unsigned short integer	8 bits	USINT	uint8
int	unsigned integer	16 bits	YUBT	uint16
long	Unsigned double integer	32 bits	UDINT	uint32
single		32 bits	REAL	single
boolean		8 bits	BOOL	boolean
double		64 bits	LREAL	double
string		Variable size	WSTRING	-
byte	unsigned	8 bits	BYTE	-

- Discretization and sampling time

Any model running on a computer always involves some discretization process in order to represent the flow of data in a physical system. The sampling time is the rate at which a physical device or development software checks their inputs and output.



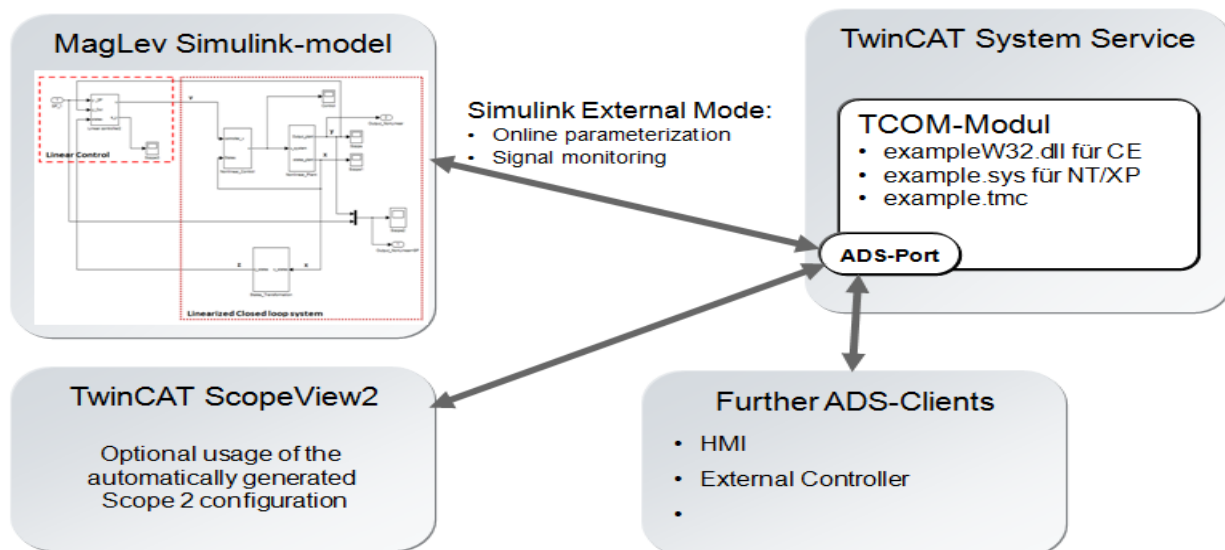
This is exactly the case in both TwinCAT 3 and Simulink. In both environments the discretization and sampling time are the same to ensure exact behaviour or model correctness. Table 8.3 shows the discretization and sampling time selected to achieve model transformation.

**Table 8.3: Discretization and Sampling Times**

	Discretization Time	Sampling Time
<b>Simulink</b>	$10^{-2}$ ms	10ms
<b>TwinCAT 3</b>	$10^{-2}$ ms	10ms

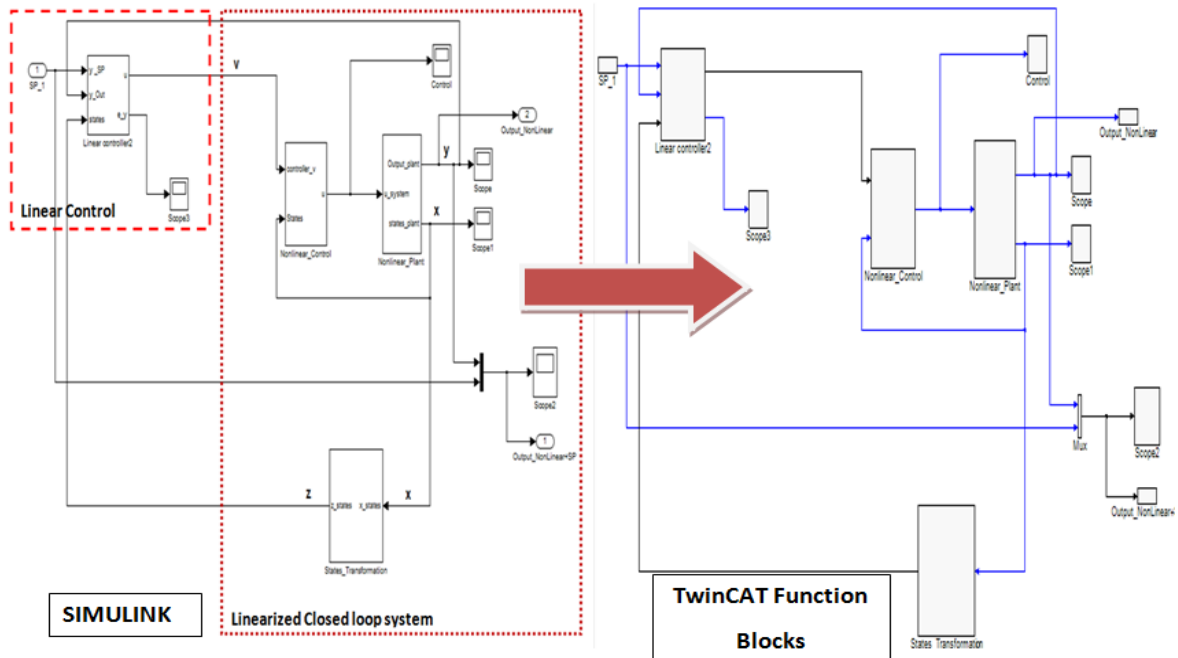
- Simulink connectivity to the TwinCAT 3 software for model transformation

The connectivity process of the Simulink to TwinCAT 3 developed module for the magnetic levitation system is shown in Figure 8.13. The closed-loop model of the magnetic levitation system developed in Simulink exchanges data with TwinCAT system service through an Automated Device Specification (ADS) external port. The TwinCAT System Service is the platform in which the transformed modules from Simulink to TwinCAT 3 are saved. Further, the developed TwinCAT3 modules are visualized in TwinCAT 3 with an embedded Scope to analyse the behaviour of the closed-loop system. The TwinCAT system service gives the possibility to other ADS users to communicate with the developed modules.



**Figure 8.13: Connectivity of a Simulink TwinCAT 3 module**

The model transformation technique keeps the same data and parameter connections in order for Simulink and TwinCAT to produce the same output data. Figure 8.14 shows the transformation from Simulink to TwinCAT PLC function blocks of the feedback linearization closed-loop system.



**Figure 8.14: Closed-loop feedback linearization-based system Model Transformation from Simulink to TwinCAT 3 PLC Function Blocks**

### 8.8.1.1.3 Real-time simulation of the closed-loop system

The real-time simulation has a set of priority of 5 and a 10ms cycle time. The real-time simulations are done with the same parameters selected in Table 6.1 of Chapter 6:

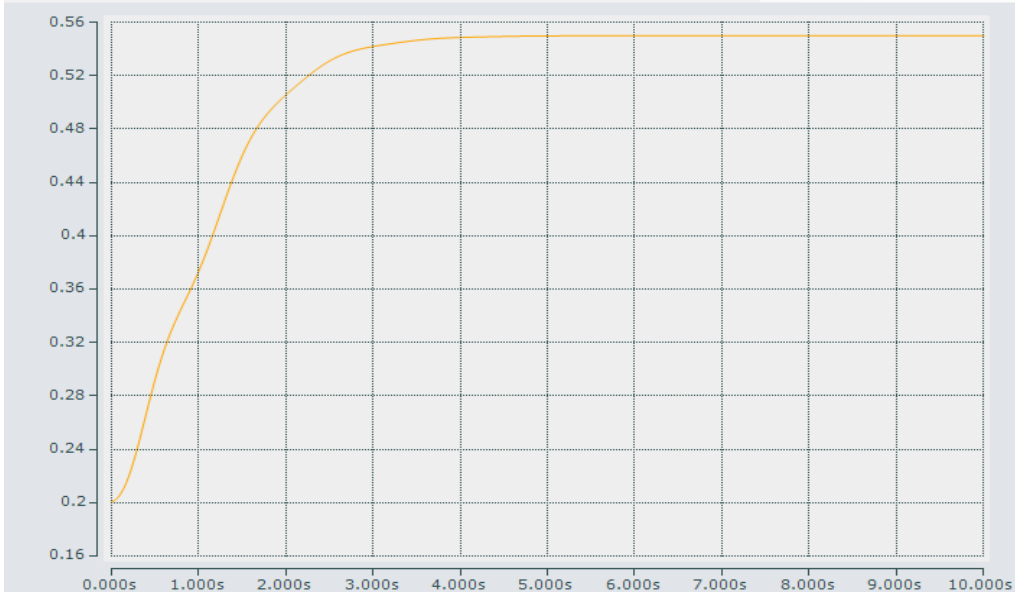
- The same initial conditions: [0.2 0 0.894 0]'
- The same set points:  $y^{sp1} = 0.55[m]$ ,  $y^{sp2} = 0.6[m]$ ,  $y^{sp3} = 0.75[m]$  and  $y^{sp4} = 0.85[m]$ .

For the feedback linearization method based on real-time implementation on the PLC, one signal is taken on the graph:

- Position of the ball under the feedback linearization control

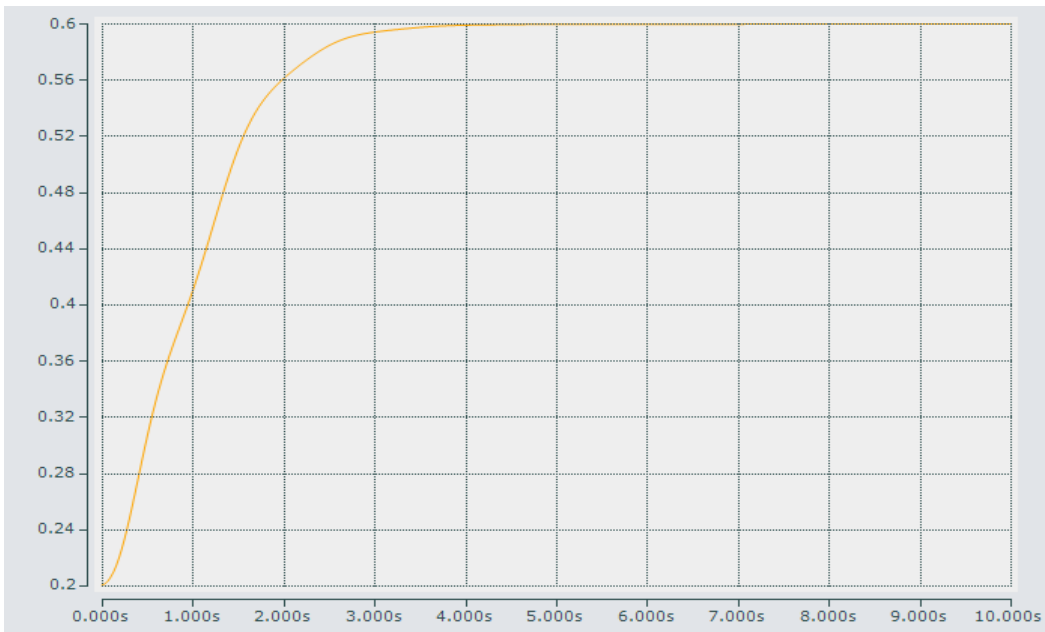
From Figure 8.15 to Figure 8.18, the following results are presented:

- Real-time simulation when the set-point is  $y^{sp1} = 0.55[m]$ :



**Figure 8.15:** Magnetic levitation real-time position when the set point is  $y^{sp1} = 0.55[m]$

- Real-time simulation when the set-point is  $y^{sp2} = 0.6[m]$ :



**Figure 8.16:** Magnetic levitation real-time position when the set point is  $y^{sp2} = 0.6[m]$

- Real-time simulation when the set-point  $y^{sp3} = 0.75[m]$ :

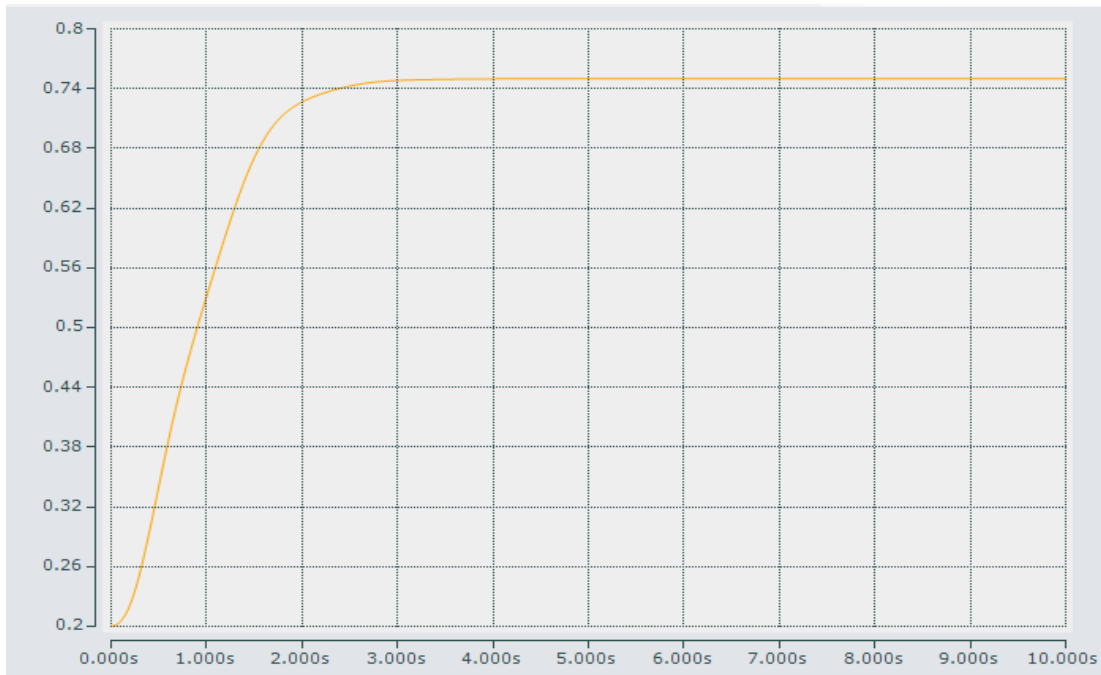


Figure 8.17: Magnetic levitation real-time position when the set point is  $y^{sp3} = 0.75[m]$

- Real-time simulation when the set-point  $y^{sp4} = 0.85[m]$ :

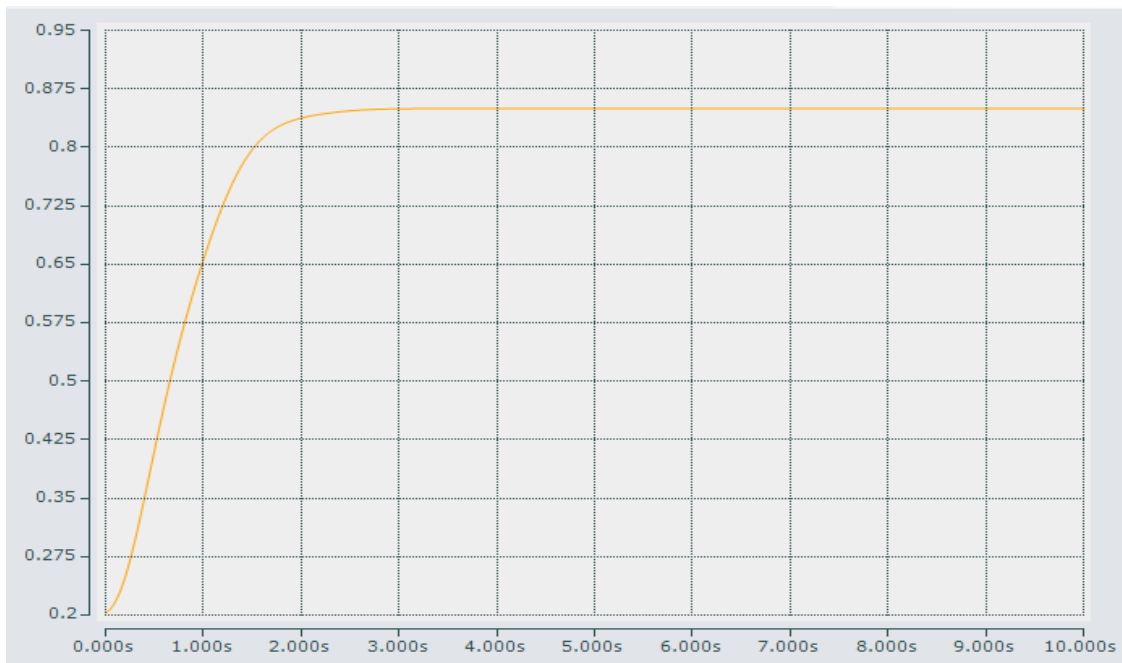


Figure 8.18: Magnetic levitation real-time position when the set point is  $y^{sp4} = 0.85[m]$

The comparison of the dynamic output behaviours of the closed-loop systems in Simulink and the PLC real-time results are given below in Table 8.4.

**Table 8.4: Simulink versus real-time results comparison**

Set point	Characteristics	Magnetic levitation Simulink results	Magnetic levitation Real-time results
0.55	Time Delay	0	0
	Overshoot	0	0
	Rising Time	2s	4s
	Steady State Error	0.015	0
	Settling Time	4s	6s
0.6	Time Delay	0	0
	Overshoot	0	0
	Rising Time	2.1s	3.8s
	Steady State Error	0.008	0
	Settling Time	6s	4.2s
0.75	Time Delay	0	0
	Overshoot	0	0
	Rising Time	2.3s	3s
	Steady State Error	0.002	0
	Settling Time	2.2s	3.6s
0.85	Time Delay	0	0
	Overshoot	0	0
	Rising Time	1.8s	2.8s
	Steady State Error	0.001	0
	Settling Time	2s	3s

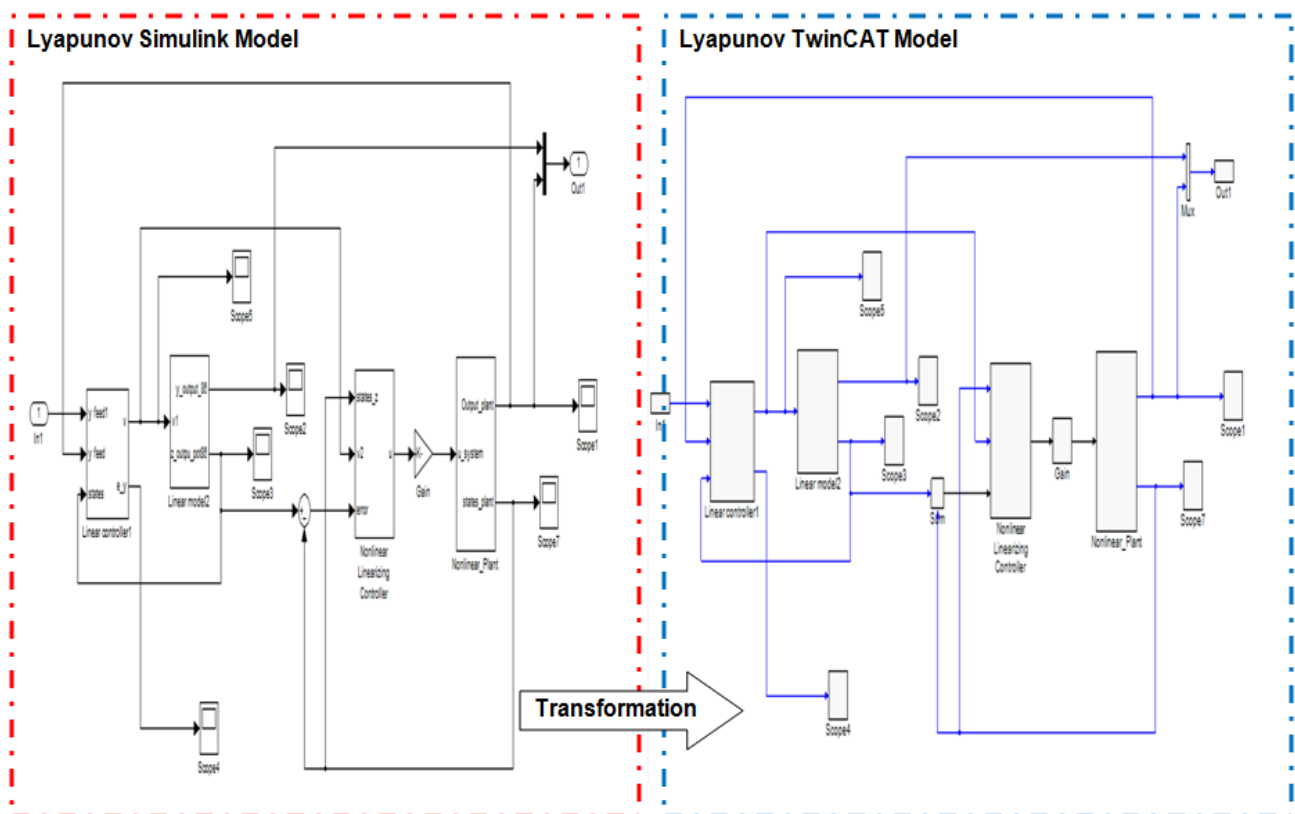
The real-time implementation results are shown from Figure 8.15 to Figure 8.18. The results confirm that the magnetic levitation system is well under control as it was the case in the Simulink simulations in Chapter 6.

Table 8.4 shows that the magnetic levitation system behaves within more or less the same time delay, overshoot, rising time, steady state error, and settling time in both real-time and Simulink environments. These results prove that the model transformation technique was successful to

transform a closed-loop magnetic levitation system based on input-state feedback linearization from Simulink to function block based on IEC 61499 standard real-time environment.

### 8.8.1.2 Real-time implementation of the closed-loop system based on Lyapunov stability and MRC

Figure 8.19 present the transformation of the closed-loop system model from Simulink to TwinCAT 3 to achieve real-time implementation of the Lyapunov stability based MRC closed-loop system.



**Figure 8.19: Transformation of the model of the closed-loop system based on the Lyapunov and MRC from Simulink to TwinCAT**

### 8.8.1.3 Real Time simulation of the closed-loop system

The real-time simulation of the Lyapunov stability based MRC closed-loop system model is done with respects to the same initial conditions and set points as done in Table 7.1 of Chapter 7:

- Initial condition:  $[0.05 \ 0 \ 0]'$
- Same set-points:  $y^{sp1}=0[m]$ ,  $y^{sp2}=0.01[m]$ ,  $y^{sp3}=0.09[m]$ , and  $y^{sp4}=0.15[m]$ .

The following signal is taken in consideration in the real-time simulations:

- Position of the ball.

From figure 8.20 to figure 8.23, the simulation results of the real-time control simulation of the magnetic levitation system based on Lyapunov stability are presented. The system presents good behaviour as the ball follows the reference trajectory to achieve stability of the closed loop system.

- Real-time simulation when the set-point is  $y^{sp1}=0[m]$ :



**Figure 8.20:** Position of the magnetic levitation system when the set-point is  $y^{sp1}=0[m]$

- Real-time simulation when the set-point is  $y^{sp2}=0.01[m]$ :

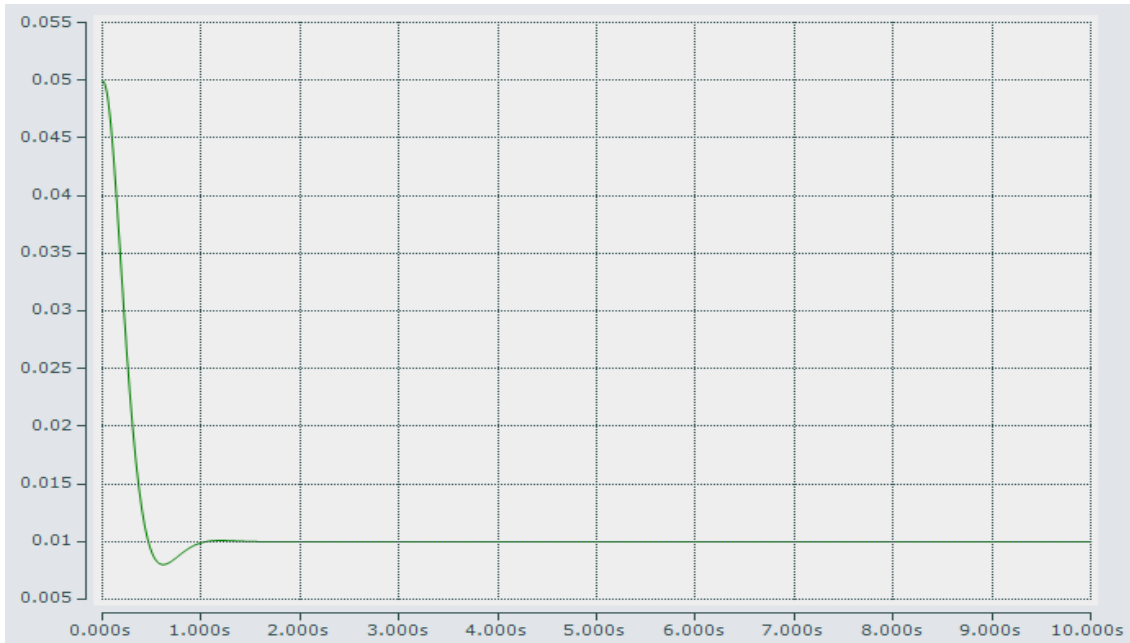


Figure 8.21: Position of the magnetic levitation system when the set-point is  $y^{sp2}=0.01$ [m]

- Real-time simulation when the set-point is  $y^{sp3}=0.09$ [m]:

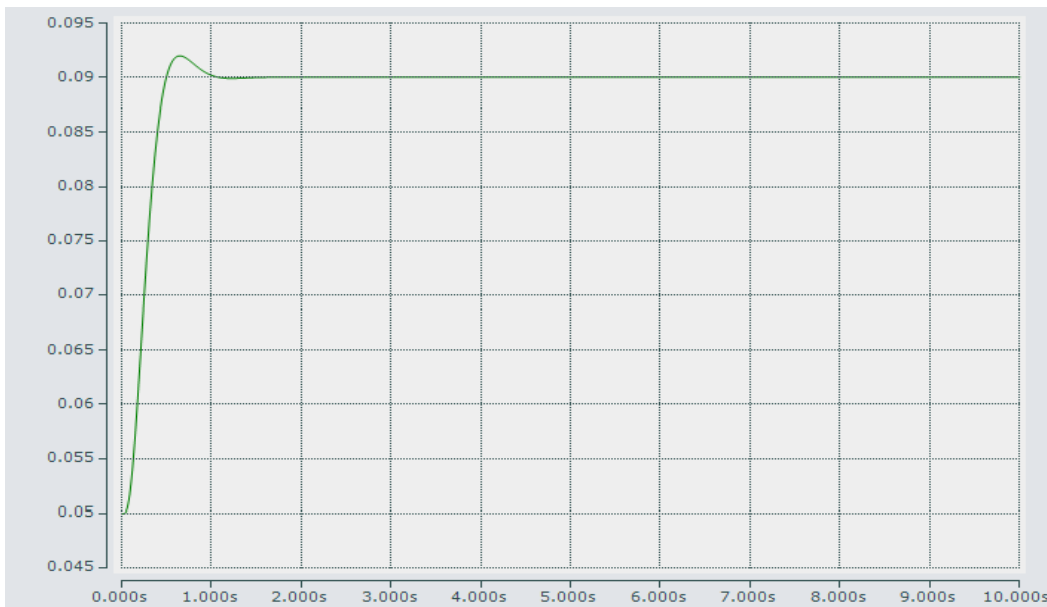
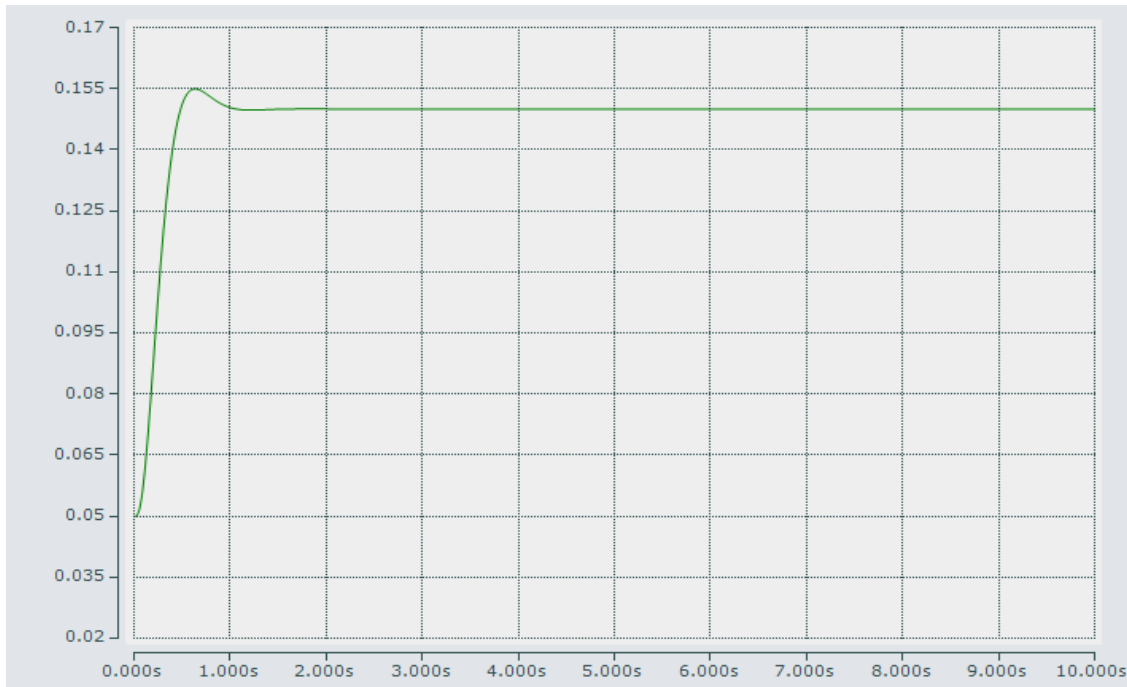


Figure 8.22: Position of the magnetic levitation system when the set-point is  $y^{sp3}=0.09$ [m]

- Real-time simulation when the set-point is  $y^{sp4}=0.15$ [m]:





**Figure 8.23:** Position of the magnetic levitation system when the set-point is  $y^{sp4}=0.15[m]$

The comparison of the dynamic output behaviour of the closed-loop systems in the Simulink and the real-time PLC environment are given below in Table 8.5.

**Table 8.5:** Simulink versus TwinCAT 3 real-time simulation results

Set point	Characteristics	Magnetic levitation Simulink results	Magnetic levitation Real-time results
0	Time Delay	0	0
	Overshoot	0	0
	Rising Time	2s	4s
	Steady State Error	0.015	0
	Settling Time	4s	6s
0.01	Time Delay	0	0
	Overshoot	0	0
	Rising Time	2.1s	3.8s
	Steady State Error	0.008	0
	Settling Time	6s	4.2s

0.09	Time Delay	0	0
	Overshoot	0	0
	Rising Time	2.3s	3s
	Steady State Error	0.002	0
	Settling Time	2.2s	3.6s
0.15	Time Delay	0	0
	Overshoot	0	0
	Rising Time	1.8s	2.8s
	Steady State Error	0.001	0
	Settling Time	2s	3s

The real-time implementation results are shown from Figure 8.20 to Figure 8.23. The results confirm that the stability of the magnetic levitation system is achieved as it was the case in the Simulink simulations in Chapter 7.

Table 8.5 shows that the magnetic levitation system behaves within approximately the same characteristics in both real-time and Simulink environments. These results prove that the model transformation technique was successful to transform a closed-loop magnetic levitation system based on Lyapunov stability and MRC from Simulink to the function block based on IEC 61499 standard real-time environments.

### 8.8.2 Results analysis

On the Beckhoff PLC CX5020 physical platform and TwinCAT 3 real-time implementation software, feedback linearization and Lyapunov stability MRC-based closed-loop system are simulated. It can be seen that when both methods are applied to control the magnetic levitation system, the feedback linearization method shows good steady state behaviour, but a larger rising time. The Lyapunov stability based MRC method has small overshoot, and a faster rising time.

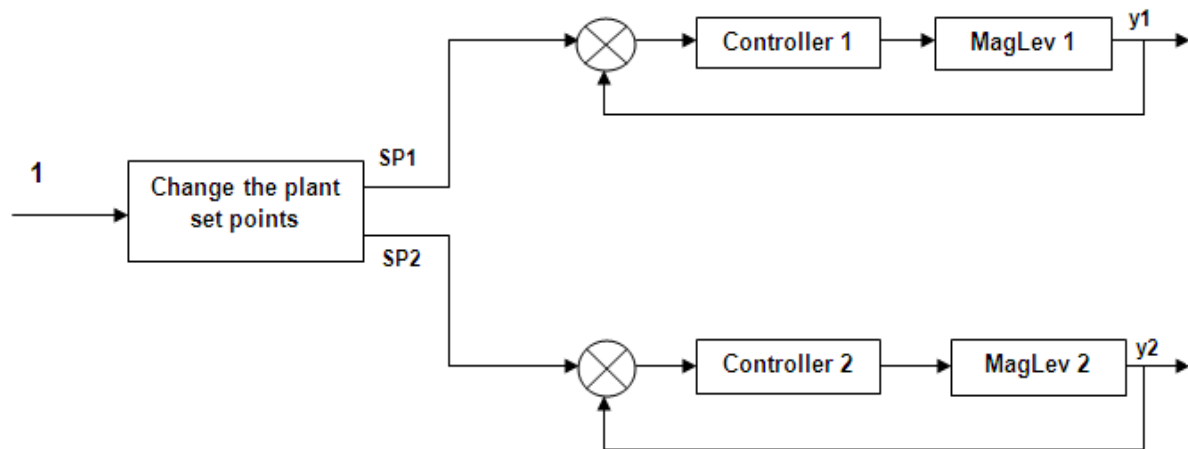
### 8.9 Real-time implementation and simulation of a distributed system

A distributed system of parallel distributed control of two levitation systems based on the same controllers and plants parameters but with different initial conditions and set-points is developed.

- MagLev 1 and MagLev 2 have the same parameters.

- Controller 1 and controller 2 are the combination of Integral Linear Quadratic Regulator (ILQR) and input-state/input-output feedback controllers. Both have the same parameters.
- MagLev 1 and MagLev 2 have different initial conditions.
- The set-point 1 and the set-point 2 are different.

The two closed-loop models of the magnetic levitation system controlled by feedback input-state linearization technique are implemented in the CX5020 PLC to achieve real-time distributed control. The set-points and initial conditions are different for both systems, and an algorithm to select the appropriate set-point for each system is developed. Figure 8.24 shows the block diagram of the distributed control methodology applied in this thesis.

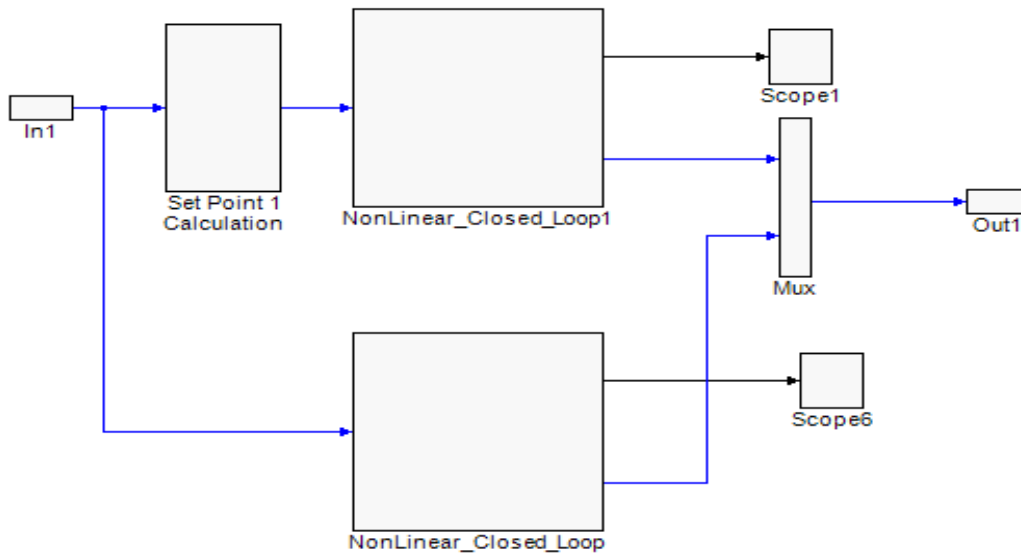


**Figure 8.24: Block diagram of the distributed solution**

Table 8.6 gives the parameters of the simulation of both systems, and Figure 8.25 shows the function block algorithm developed for the distributed implementation.

**Table 8.6: Parameters of simulation for the distributed system**

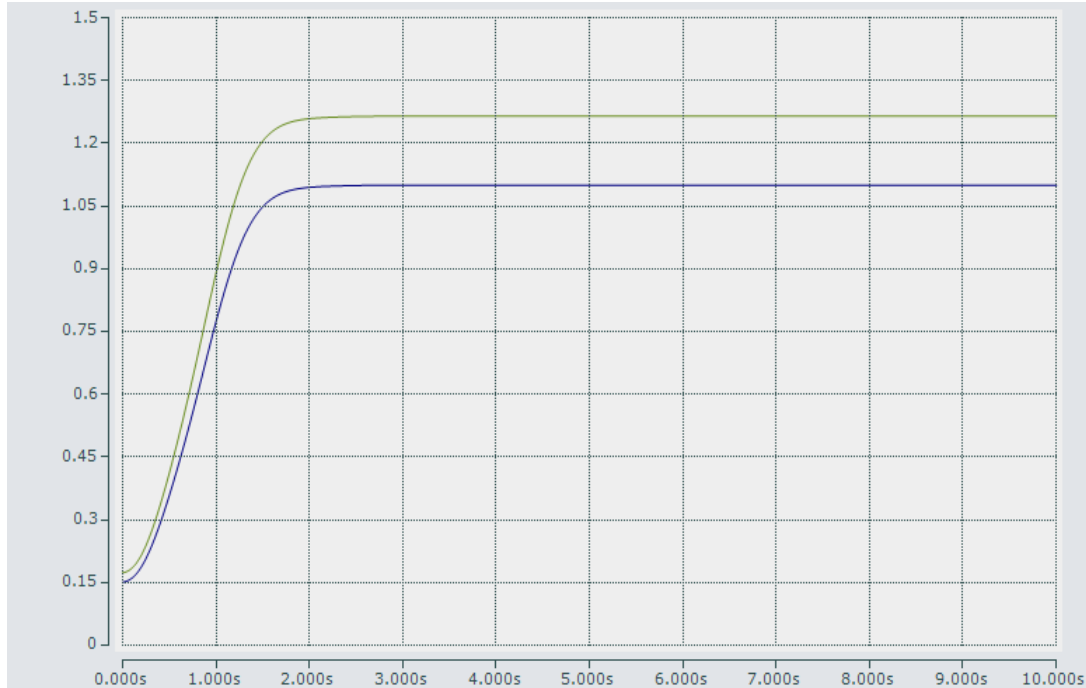
Magnetic levitation 1		Magnetic levitation 2	
Set points	Initial conditions	Set points	Initial conditions
1.1[m]	[0.15 0 0.8 0]'	1.24 [m]	[0.2 0 0.86 0]'
1.5[m]	[0.2 0 0.86 0]'	1.45[m]	[0.25 0 0.9 0]'
1.2[m]	[0.4 0 0.96 0]'	1.4[m]	[0.45 0 0.97 0]'



**Figure 8.25:** Block diagram algorithm developed for the distributed implementation

The results of the real-time distributed system are shown from Figure 8.26 to Figure 8.28.

- Simulation when the set-points are:  $y_{Maglev1}^{sp1} = 1.1[m]$  and  $y_{Maglev2}^{sp1} = 1.24[m]$ :



**Figure 8.26:** Simulation of the distributed system when the set-points are:

$$y_{Maglev1}^{sp1} = 1.1[m] \text{ and } y_{Maglev2}^{sp1} = 1.24[m]$$

- Simulation when the set-points are:  $y_{Maglev1}^{sp2} = 1.15[m]$  and  $y_{Maglev2}^{sp2} = 1.45[m]$ :

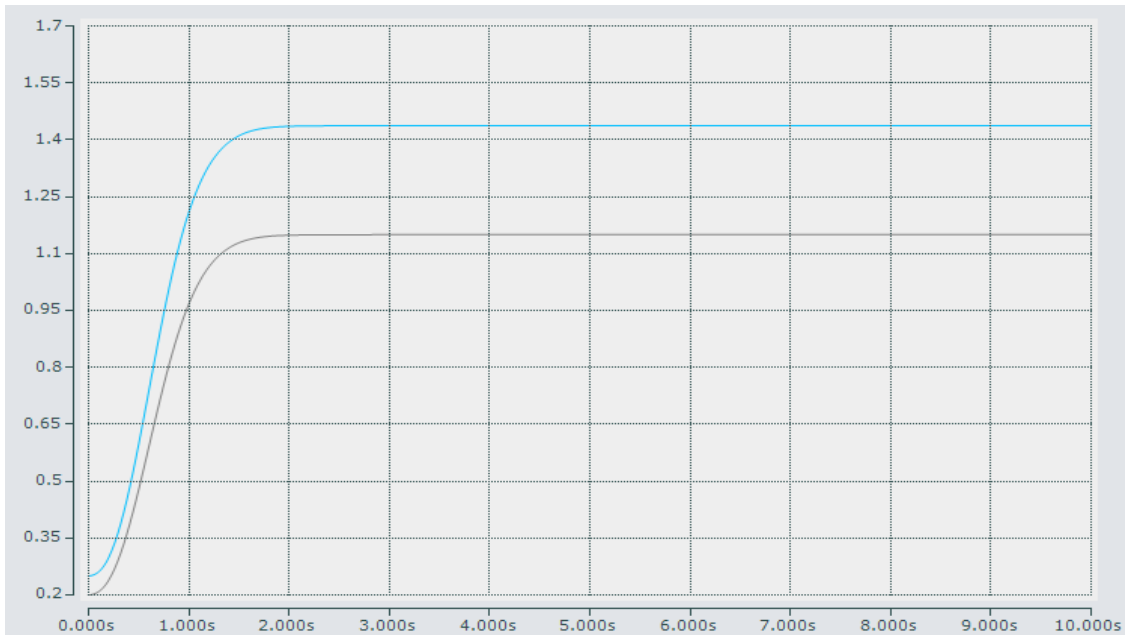


Figure 8.27: Simulation of the distributed system when the set-points are:

$$y_{Maglev1}^{sp2} = 1.15[m] \text{ and } y_{Maglev2}^{sp2} = 1.45[m]$$

- Simulation when the set-points are:  $y_{Maglev1}^{sp3} = 1.2[m]$  and  $y_{Maglev2}^{sp3} = 1.4[m]$ :

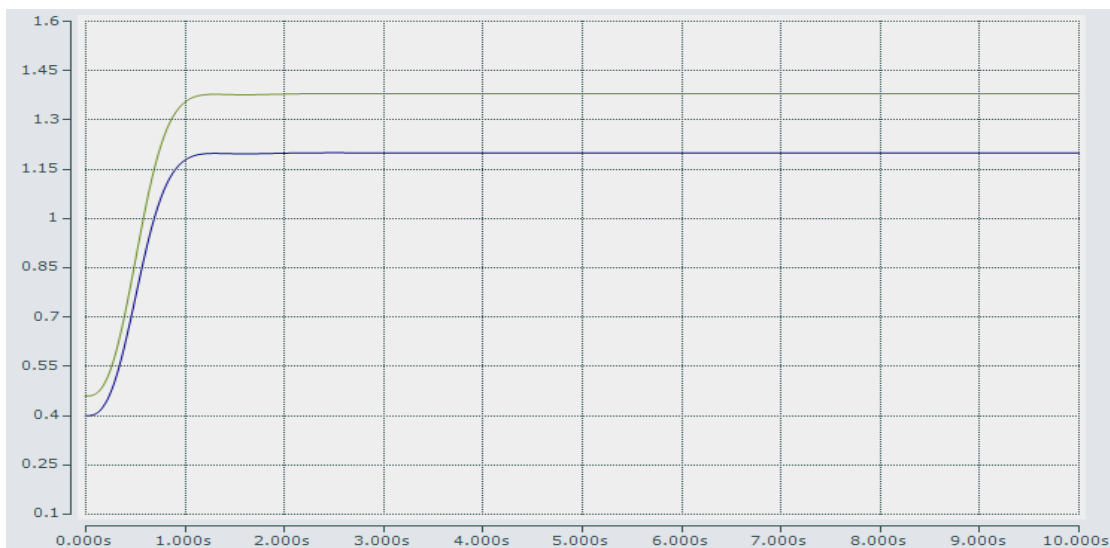


Figure 8.28: Simulation of the distributed system when the set-points are:  $y_{Maglev1}^{sp3} = 1.2[m]$

$$\text{and } y_{Maglev2}^{sp3} = 1.4[m]$$

The simulation of the distributed implementation of the magnetic levitation systems shows satisfactory results. These results prove that it is possible to implement distributed control algorithms in PLC.

### **8.10 Conclusion**

In this chapter, the general introduction of PLCs, and the model transformation technique based on IEC 61499 standard to achieve real-time implementation of the magnetic levitation system are discussed. Beckhoff CX 5020 PLC together with TwinCAT 3 software were used for the development of linearizing and nonlinear Lyapunov reference based model controllers. The Real-Time implementation results are acquired, and plotted in TwinCAT 3. The results of real time implementation in both control techniques show that the output signal is able to follow the input signal, and achieve stability. The Beckhoff PLC CX 5020 is a very reliable platform and it is built according to new industrial standard. In next chapter, the deliverables of thesis are reviewed and the future directions of research are outlined.

## **CHAPTER NINE: CONCLUSION: THESIS DELIVERABLES, APPLICATIONS AND FUTURE RESEARCH**

### **9.1 Introduction**

Nonlinear control of nonlinear plants has become more and more important in industry. For this thesis, exact input-state linearization and Lyapunov stability based nonlinear model reference control methods to stabilize and optimize a nonlinear magnetic levitation system are proposed. The control algorithms and the models of the closed-loop systems are both implemented in Matlab/Simulink. The models of the closed-loop systems are then transformed into TwinCAT function blocks language based on IEC 61499 standard concept to perform real-time simulation with a Beckhoff CX5020 PLC. The output signals track the set-points, therefore the developed methodology can be considered as adequate for the control of the magnetic levitation system. The real-time simulation results show that the developed complex nonlinear control algorithms can be implemented into a modern PLC. According to the aims and objectives of the thesis, the obtained deliverables are:

### **9.2 Thesis deliverables**

#### **9.2.1 Mathematical model development and simulation**

- A nonlinear mathematical model of the magnetic levitation plant is developed based on physical laws.
- The nonlinear mathematical model of the magnetic levitation is linearized.
- The nonlinear mathematical model is simulated in Matlab/Simulink software environment.
- The obtained linear mathematical model of the magnetic levitation is simulated in Matlab/Simulink software environment.
- A nonlinear reduced order model of the magnetic levitation developed by ECP industry is derived.
- The nonlinear reduced order model of the magnetic levitation system is simulated in Matlab/Simulink environment.
- According to (Wong, 1986), (Torres et al., 2010) and (Matlab, 2010), the nonlinear and linearized models of the magnetic levitation system are accurate.

## **9.2.2 Linear controller design and closed-loop simulation**

- A linear integral controller for the magnetic levitation is designed based on the pole-placement method.
- The closed-loop system which consists of the magnetic levitation linearized model and the pole-placement controller is simulated in Matlab/Simulink software environment for the selected output set-points and initial conditions
- A linear controller for the magnetic levitation is designed based on the linear quadratic optimal control method.
- The closed-loop system which consists of the magnetic levitation linearized model and the linear quadratic controller is simulated in Matlab/Simulink software environment for the selected output set-points and initial conditions.
- The simulation results are analyzed and compared to find the better linear controller.

## **9.2.3 Nonlinear controllers design and closed-loop system simulation**

### **9.2.3.1 Feedback linearization controller design and closed-loop system simulation**

- The input-state feedback linearization method based nonlinear controller is designed for nonlinear magnetic levitation model.
- The linear integral optimal controller is designed for the input-state linearized closed loop system.
- The closed-loop system which consists of the feedback linearization method based nonlinear controller, the linear optimal controller, and the nonlinear model of the magnetic levitation plant is simulated in Matlab/Simulink for different set-points and initial conditions.
- The simulation results are discussed.

### **9.2.3.2 Lyapunov stability based and Model Reference (MR) based linearizing controller design and closed-loop system simulation**

- A Lyapunov stability based MR linearizing controller is designed for the nonlinear reduced order model of the magnetic levitation plant.
- The linear integral optimal controller is designed for the reference model based linearized closed-loop system.



- The closed-loop system which consists of the Lyapunov stability and MR linearizing controller, the linear optimal controller, and the reduced order of the magnetic levitation plant is simulated in Matlab/Simulink for different output set-points and same initial conditions.
- The simulation results are discussed.

## **9.2.4 Real-time implementation on TwinCAT 3 software and deployment on Beckhoff CX5020 PLC**

### **9.2.4.1 Real-time implementation and simulation of the closed-loop system based on the nonlinear input-state feedback linearizing controller**

- Model transformation from Simulink to TwinCAT 3 function blocks of the closed-loop system based on nonlinear input-state feedback linearizing controller is developed and implemented.
- The Real-time deployment of the transformed model of the closed-loop system in Beckhoff CX5020 PLC is done.
- Real-time simulation results are acquired and plotted on TwinCAT 3 measurement scope project for the magnetic levitation closed-loop behavior.
- The real-time implementation results are discussed.

### **9.2.4.2 Real-time implementation and simulation of the closed-loop system based on the Lyapunov stability MR nonlinear linearizing controller**

- Model transformation from Simulink to TwinCAT 3 function blocks of the closed-loop system based on the Lyapunov stability and MR nonlinear linearizing controller is developed and implemented.
- The real-time deployment in Beckhoff CX5020 PLC is done.
- Real-time simulation results are acquired and plotted on TwinCAT 3 measurement scope project for the magnetic levitation closed-loop behavior.
- Real-time implementation results are discussed.

### 9.2.4.3 Real-time implementation and simulation of a distributed nonlinear system

- The magnetic levitation systems have the same parameters.
- The controllers are the combination of Integral Linear Quadratic Regulator (ILQR) and input-state/input-output feedback controllers. Both have the same parameters.
- The magnetic levitations systems have different initial conditions.
- The set-points of both magnetic levitation systems are different.

### 9.2.5 Software deployment

The methods for design described above are examined through simulation and PLC real-time implementation using two software environments Matlab/Simulink and Beckhoff TwinCAT 3. Separate software programs are developed for every of the deliverables from 9.1.1 till 9.1.4. The developed software for Matlab simulation and real-time simulation in PLC are grouped in Table 9.1 and Table 9.2 respectively.

**Table 9.1: Programs used for pole-placement, LQR, feedback linearization and Lyapunov controller designs for Matlab simulation of the closed-loop systems behaviours**

Methods	Design	Simulation	Appendices
Pole-placement	MagLevPPI.m	Pplevitation.mdl	A.3
LQR	MagLev_LQR.m	Lqrlevitation.mdl	A.4
Feedback linearization	yohan_linearization.m	Linearization_maglev.mdl	A.5
Lyapunov	yohan_Lyapunov.m	Lyapunov_maglev.mdl	A.6

**Table 9.2: Programs for linear and nonlinear controllers and plant implementation in the TwinCAT 3 development of Beckhoff CX5020 PLC**

Method	Design	Subsystems	Appendices
Linearization	InputOutputLinearization	Linear_controller.tmc	A.7
		NonLinear_Controller.tmc	A.8
		NonLinear_Plant.tmc	A.9

		State_Transformation.tmc	A.10
Lyapunov based linearization	LyapunovModelReference	Linear_ControllerLyap.tmc	A.11
		LinearSystem_model.tmc	A.12
		Linearizing_controller.tmc	A.13
		Nonlinear_plant.tmc	A.14

In this thesis, based on the above deliverables a methodology for design and implementation of the magnetic levitation nonlinear control systems in real-time environment based on model transformation and IEC 61499 standard-based function block programming is developed. The developed methodology can be used to guide the researchers and students into the field of distributed control real-time implementation of nonlinear control algorithms in programmable logic controllers.

For the detailed research outcomes, the complete linear and nonlinear mathematical models of the magnetic levitation 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, the feedback linearization method and Lyapunov stability based MRC method. The simulation results show that the above mentioned controllers are designed successfully.

The real-time implementation results on TwinCAT 3 and their deployment on Beckhoff CX5020 PLC show that the project aim and objectives are achieved.

### 9.3 Applications

The developments in the thesis can be applied in research laboratories, for education and research at universities, and in industry.

The proper control of the magnetic levitation plants has the potential to improve the plant behaviour in numerous sector of industry such as aerodynamics, levitation vehicles, magnetic bearings, noise mitigation and fiber reinforced plastics for vehicles and structural concretes.

### 9.4 Future research

The future research is seen to be useful in the following directions:

- The function blocks in TwinCAT 3 can be improved to be event driven.
- A better algorithm translation tool can be developed to transform Matlab/Simulink code into language supported by the TwinCAT 3 function block development software.

- Model transformation can be tested with more industrial examples to boost the acceptance of IEC 61499 in industry.
- Development of a hybrid system for distributed control and data exchange based on joint implementation of IEC 61499 and IEC 61850 standards respectively.

## **9.5 Publications**

1. Y.D Mfoumboulou, M. Adonis (2011). Design of a PID-PWM controller for a single-input multiple-output DC-DC buck converter using LabVIEW, presented at the 15<sup>th</sup> annual BTech conference.
2. Y.D Mfoumboulou, R. Tzoneva (2013). Development of a nonlinear control algorithm for implementation in Programmable Logic Controller (PLC), SACAC control system open day conference, University of Cape Town, Cape Town.
3. Y.D Mfoumboulou, R. Tzoneva (2014). Development and implementation of a distributed system based on IEC 61499 for the control of two magnetic levitation systems in real-time, submitted to the South African Journal of Industrial Engineering.

## APPENDIX A:

### THE DEVELOPED MATLAB M-FILES

The m-files used in this 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 parameters and coefficients for Simulink files to run. The Simulink files include the magnetic levitation mathematical models, the closed-loop system and the real-time implementation system.

#### A.1: MATLAB script-file for levitation parameters-levitationparameters.m

```
% Magnetic levitation system parameters for nonlinear simulation
% Names: Yohan Darcy
% Surname: Mfoumboulou
% Student No: 207214107

clear
clc
g=9.81 % Gravitational force
k=0.0001
m=8.27*10^-3 % Mass of the ball
R=1 % Internal resistance of the inductor
L=0.01 % Inductance
i=0.84, % input current
x=0.012, % position of the ball at equilibrium point
yo=[0.001 0 0]

i0=x*(m*g/k)^0.5, % Current at equilibrium point

% State space representation of the magnetic levitation system

A=[0 1 0;(2*k/m)*(i^2/x^3) 0 -(2*k/m)*(i/x^2);0 0 -R/L], % State Matrix A
B=[0;0;1/L], % Control Matrix B
C=[1 0 0], % Output Matrix C
D= 0, % Matrix D
```

## A.2: MATLAB Script file-MagLevPF.m

```
% Magnetic levitation system parameters for Pole Placement simulation
% Names: Yohan Darcy
% Surname: Mfoumboulou
% Student No: 207214107

clear
clc
g=9.81 % Gravitational force
k=0.0001
m=8.27*10^-3 % Mass of the ball
R=1 % Internal resistance of the inductor
L=0.01 % Inductor Value
i=0.84, % input current
x=0.012, % position of the ball at linearization point
i0=x*(m*g/k)^0.5
% =====State space representation of the magnetic levitation system

A=[0 1 0;(2*k/m)*(i^2/x^3) 0 -(2*k/m)*(i/x^2);0 0 -R/L], % State Matrix A
B=[0;0;1/L], % Control Matrix B
C=[1 0 0], % Output Matrix C
D= 0, % Matrix D

% =====Controllability of the system
Cl=ctrb(A,B)
Rank_Cl=rank(Cl)

p1=-5.97+4.47i;
p2=-5.97-4.47i;
p3=-50;

K= place (A,B,[p1 p2 p3]);
K1=K(1)
K2=K(2)
K3=K(3)

sys_cl=ss(A-B*K,B,C,0);

%=====
```

### A.3: MATLAB script file-MagLevPPI.m

% Magnetic levitation system parameters for simulation of the linear closed loop system based on linear place placement controller

% Names: Yohan Darcy

% Surname: Mfoumboulou

% Student No: 207214107

clear

clc

g=9.81 % Gravitational force

k=0.0001

m=8.27\*10<sup>-3</sup> % Masse of the ball

R=1 % Internal resistance of the inductor

L=0.01 % Inductor Value

i=0.84, % input current

x=0.012, % position of the ball at linearization point

i0=x\*(m\*g/k)<sup>0.5</sup>

% State space representation of the magnetic levitation system

A=[0 1 0;(2\*k/m)\*(i<sup>2</sup>/x<sup>3</sup>) 0 -(2\*k/m)\*(i/x<sup>2</sup>);0 0 -R/L], % State Matrix A

B=[0;0;1/L], % Control Matrix B

C=[1 0 0], % Output Matrix C

D= 0, % Matrix D

=====

% Controllability of the system

Cl=ctrb(A,B)

Rank\_Cl=rank(Cl)

p1=-5.97+4.47i;

p2=-5.97-4.47i;

p3=-50;

K= place (A,B,[p1 p2 p3]);

K1=K(1)

K2=K(2)

K3=K(3)

sys\_cl=ss(A-B\*K,B,C,0);

=====

Ai=[0 1 0 0;(2\*k/m)\*(i<sup>2</sup>/x<sup>3</sup>) 0 -(2\*k/m)\*(i/x<sup>2</sup>) 0;0 0 -R/L 0;-1 0 0 0], % State Matrix A

Bi=[0;0;1/L;0], % Control Matrix B

Ci=[1 0 0 0], % Output Matrix C

Di= 0, % Matrix D

=====

% Controllability of the system

Cil=ctrb(Ai,Bi)

Rank\_Cil=rank(Cil)

p4=-10

Ki= place (Ai,Bi,[p1 p2 p3 p4]);

Ki1=Ki(1)

Ki2=Ki(2)

Ki3=Ki(3)

Ki4=Ki(4)



#### A.4: MATLAB Script file-MagLev\_LQR.m

```
% Magnetic levitation system parameters for simulation of the linear closed loop system based  
on linear integral optimal controller
```

```
% Names: Yohan Darcy  
% Surname: Mfoumboulou  
% Student No: 207214107
```

```
clear
```

```
clc
```

```
g=9.81 % Gravitational force
```

```
k=0.0001
```

```
m=8.27*10^-3 % Masse of the ball
```

```
R=1 % Internal resistance of the inductor
```

```
L=0.01 % Inductor Value was 0.01
```

```
i=0.84, % input current
```

```
x=0.09, % position of the ball at linearization point was 0.012
```

```
% State space representation of the magnetic levitation system
```

```
A=[0 1 0;(2*k/m)*(i^2/x^3) 0 -(2*k/m)*(i/x^2);0 0 -R/L], % State Matrix A
```

```
B=[0;0;1/L], % Control Matrix B
```

```
C=[1 0 0], % Output Matrix C
```

```
D= 0, % Matrix D
```

```
poles=eig(A)
```

```
%=====
```

```
% Controllability of the system
```

```
Cl=ctrb(A,B)
```

```
Rank_Cl=rank(Cl)
```

```
% Design of the linear-quadratic(LQ) state-feedback regulator by using the
```

```
% function lqr in matlab
```

```
% The Q matrix is selected as positive definite and R is positive definite
```

```
% Real symetric matrix
```

```
Q=[100 0 0;0 1 0;0 0 1]
```

```
R=0.1
```

```
[K, P, E]=lqr(A,B,Q,R), % feedback control gain
```

```
K1=K(1)
```

```
K2=K(2)
```

```
K3=K(3)
```

```
A_opt=A-B*K
```

```
%=====
```

```
%Augmented Matrices
```

```
%Design of an Integral controller for the linearized model using LQR method
```

```
A_integral=[0 1 0 0;(2*k/m)*(i^2/x^3) 0 -(2*k/m)*(i/x^2) 0;0 0 -R/L 0;-1 0 0 0], % State Matrix A
```

```
B_integral=[0;0;1/L;0], % Control Matrix B
```

```
C_integral=[1 0 0 0], % Output Matrix C
```

```
D_integral= 0, % Matrix D
```

```
Q_integral=[10^4 0 0 0;0 250 0 0;0 0 5 0;0 0 0 1]
R_integral=1
[K_integral, P_integral, E_integral]=lqr(A_integral,B_integral,Q_integral,R_integral), % feedback
control gain

K1_integral=K_integral(1)
K2_integral=K_integral(2)
K3_integral=K_integral(3)
K4_integral=K_integral(4)

A_opt_integral= A_integral-B_integral*K_integral
```

## A.5: MATLAB Script file-yohan\_linearization.m

```
% Magnetic levitation system parameters for simulation of the linearized by input-state nonlinear
controller closed loop system
% Names: Yohan Darcy
% Surname: Mfoumboulou
% Student No: 207214107

clear
clc
g=9.81 % Gravitational force
k=0.0001
m=8.27*10^-3 % Masse of the ball
R=1 % Internal resistance of the inductor
L=0.01 % Inductor Value was 0.01
i=0.84, % input current
x=0.09, % position of the ball at linearization point

% State space representation of the Brunovsky reference system

A=[0 1 0;0 0 1;0 0 0], % State Matrix A
B=[0;0;1], % Control Matrix B
C=[1 0 0], % Output Matrix C
D= 0, % Matrix D
poles=eig(A)

%=====
% Controllability of the system
Cl=ctrb(A,B)
Rank_Cl=rank(Cl)

% Design of the linear-quadratic(LQ) state-feedback regulator by using the
% function lqr in matlab
% The Q matrix is selected as positive definite and R is positive definite
% Real symetric matrix

Q=[100 0 0;0 1 0;0 0 1]
R=0.1
[K, P, E]=lqr(A,B,Q,R), % feedback control gain
K1=K(1)
K2=K(2)
K3=K(3)

A_opt= A-B*K

%=====
% Augmented matrices
% Design of an integral controller for the reference model using lqr method
```

```
A_integral=[0 1 0 0;0 0 1 0;0 0 0 0;-1 0 0 0]
B_integral=[0;0;1;0]
C_integral=[1 0 0 0]
D_integral=0
```

```
%=====
% Controllability of the augmented system
Cil=ctrb(A_integral,B_integral)
Rank_Cil=rank(Cil)
```

```
Q_integral=[10000000 0 0 0;0 2500 0 0;0 0 500 0;0 0 0 10]
R_integral=1
[K_integral, P_integral, E_integral]=lqr(A_integral,B_integral,Q_integral,R_integral), % feedback control gain
```

```
K1_integral=K_integral(1)
K2_integral=K_integral(2)
K3_integral=K_integral(3)
K4_integral=K_integral(4)
```

```
A_opt_integral= A_integral-B_integral*K_integral
```

## A.6: MATLAB Script file-yohan\_Lyapunov.m

```
% Magnetic levitation system parameters for simulation of the linearized by Lyapunov and MRC
based nonlinear controller
% Names: Yohan Darcy
% Surname: Mfoumboulou
% Student No: 207214107

clear
clc
g=9.81 % Gravitational force
m=0.12% Masse of the ball
a=0.95 % Internal resistance of the inductor
b=6.28 % Inductor Value was 0.01
c=0.15, % input current
x=0.05, % position of the ball at linearization point was 0.012
Gain=1

% State space representation of the magnetic levitation system
A_I=[0 1;-2 -3], % State Matrix A
B_I=[0;1], % Control Matrix B
C_I=[1 0], % Output Matrix C
D_I= 0, % Matrix D
A_Eigen=eig(A_I)

% Form the linear model
sys_I = ss(A_I,B_I,C_I,D_I)
CI=ctrb(A_I,B_I)
R_I = rank(CI)

%=====
===
% 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   8]
R=1;

[K_I_hat, S_I_hat,E_I_hat]= lqr (A_I, B_I, Q, R)

%=====
===
%=====
===
% Form the desired model matrix A
% 4 by 4 matrix
A_d_n = A_I
EigenVal=eig(A_d_n)
```

```

%=====
===

% Formation of a matrix A for closed-loop system
A_d_hat = [A_d_n zeros(2,1); -C_l 0]

% Calculation and formation of matrix B
B_d_hat = [B_l ; 0]

% Formation of a matrix C D
C_d = [1 0 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 = 4 proves the system is controllable

% Form the desired system
sys_d = ss (A_d_n,B_l,C_l,D_l)

% 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 = [ 100000  0  0;
      0  650  0;
      0  0  90]
R=1;

[Khat, Shat,Ehat]= lqr (A_d_hat, B_d_hat, Q, R)

K1_integral=Khat(1)
K2_integral=Khat(2)
K3_integral=Khat(3)

%=====
===
% Lypunov matrix

```

```

% set the positive definite matrix Q1

Q1 = [1 0; 0 1]

% By using Q to calculate P, where  $-Q1 = A'P+PA$ 
%syms P2

A_transp=A_1';
P2 = -Q1 / (A_transp + A_1)
P21=lyap(A_1',Q1)

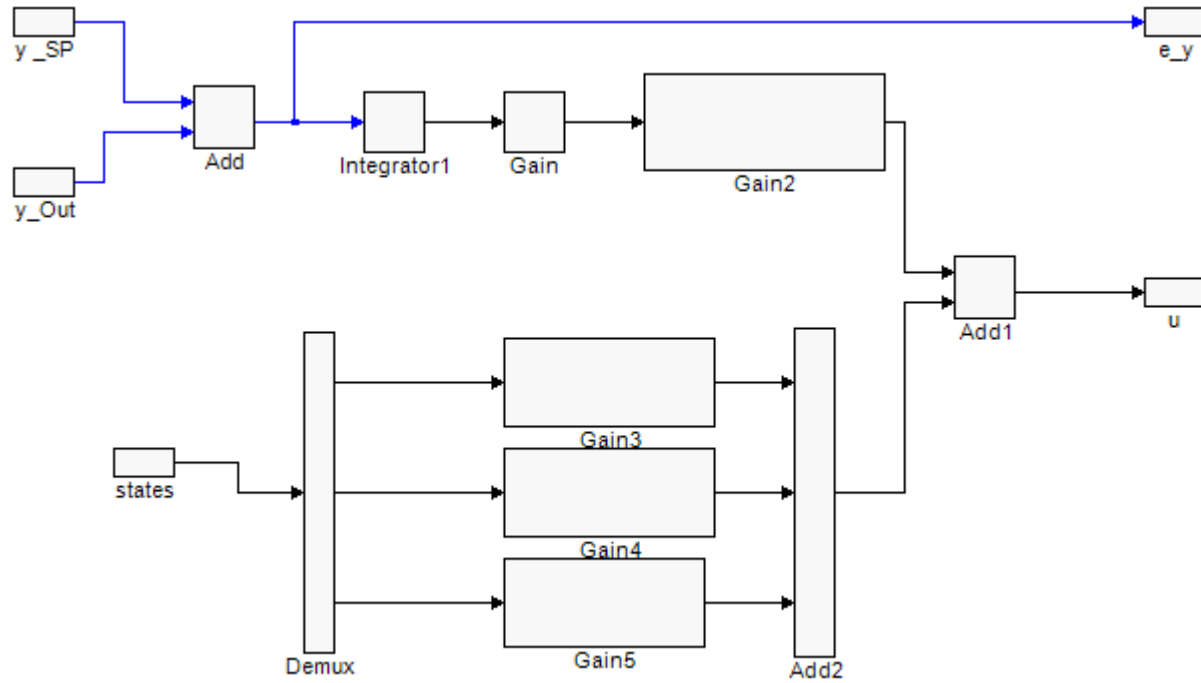
EigenValues_P21=eig(P21)
EigenValues_P2=eig(P2)
% Check P2 whether positive definite

[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

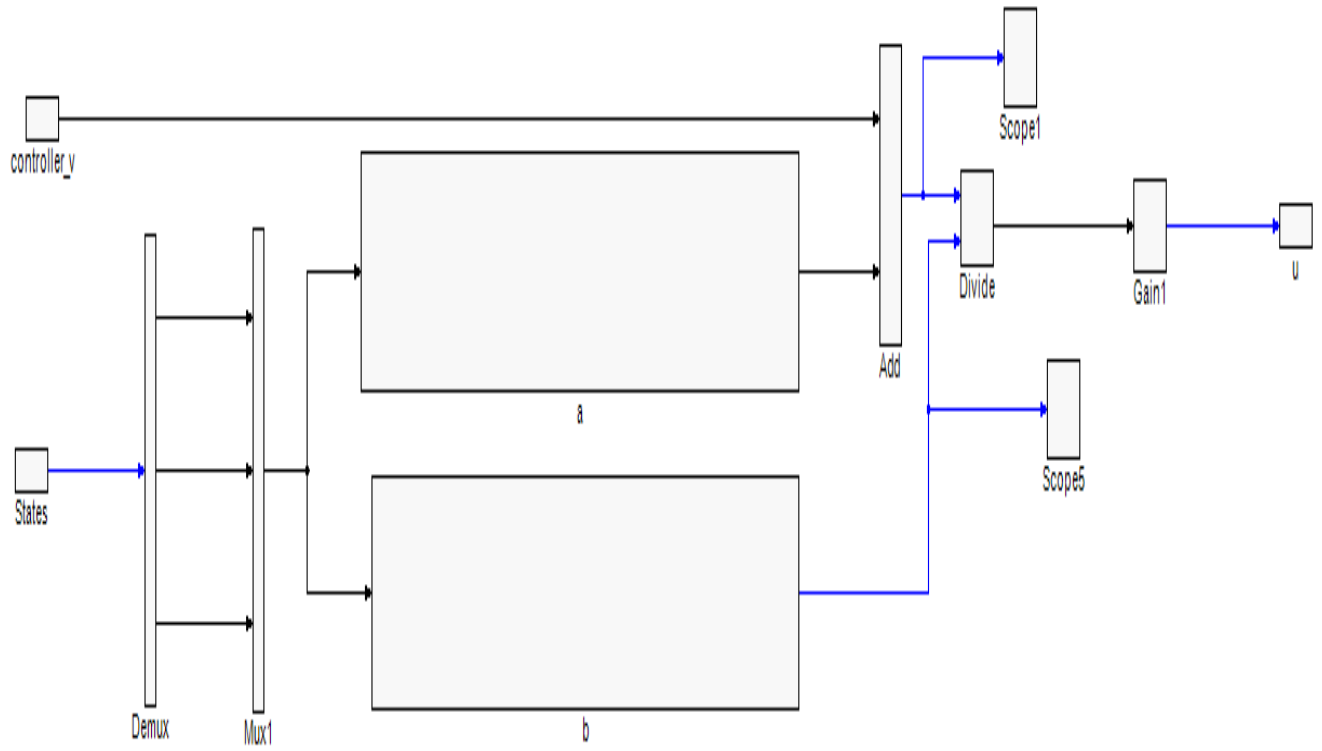
```

**A.7: TwinCAT Model of the linear integral optimal controller used in the input-state linearized closed loop system-Linear\_controller.tmc**

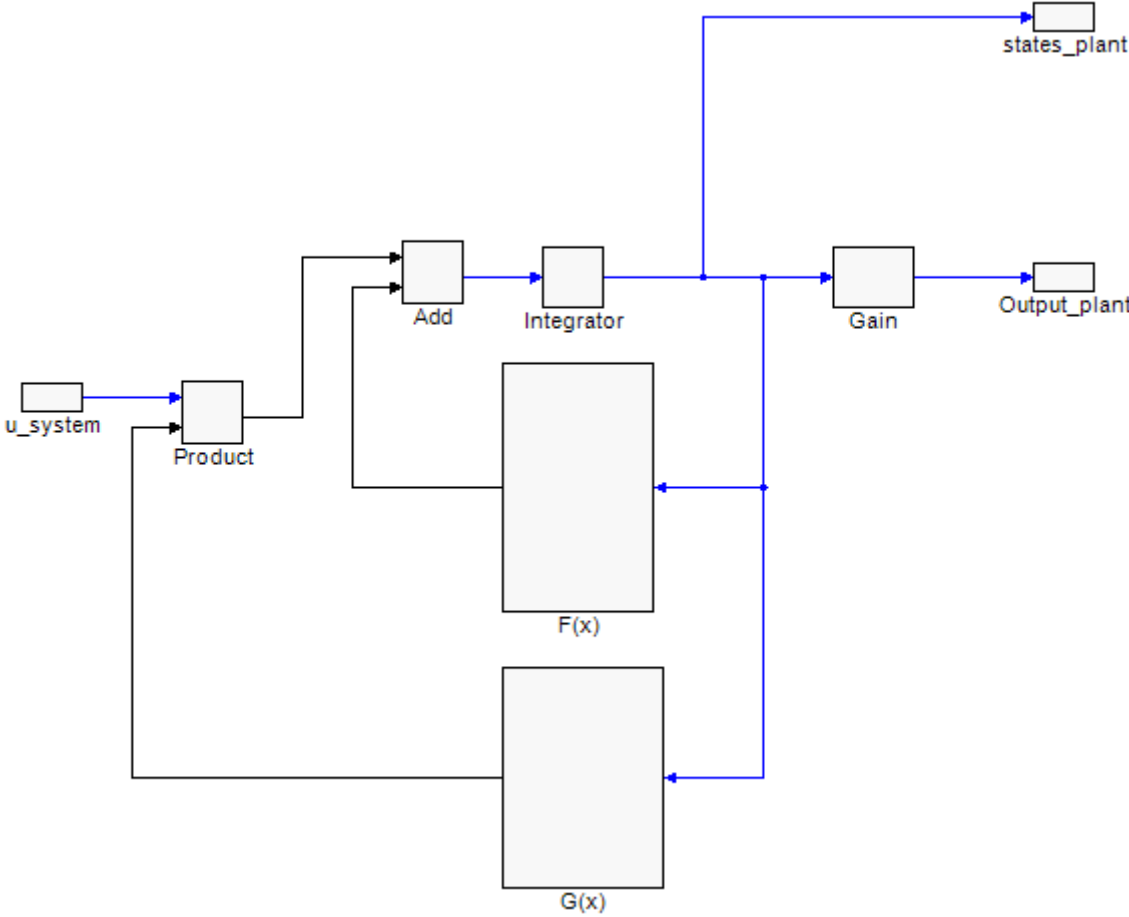




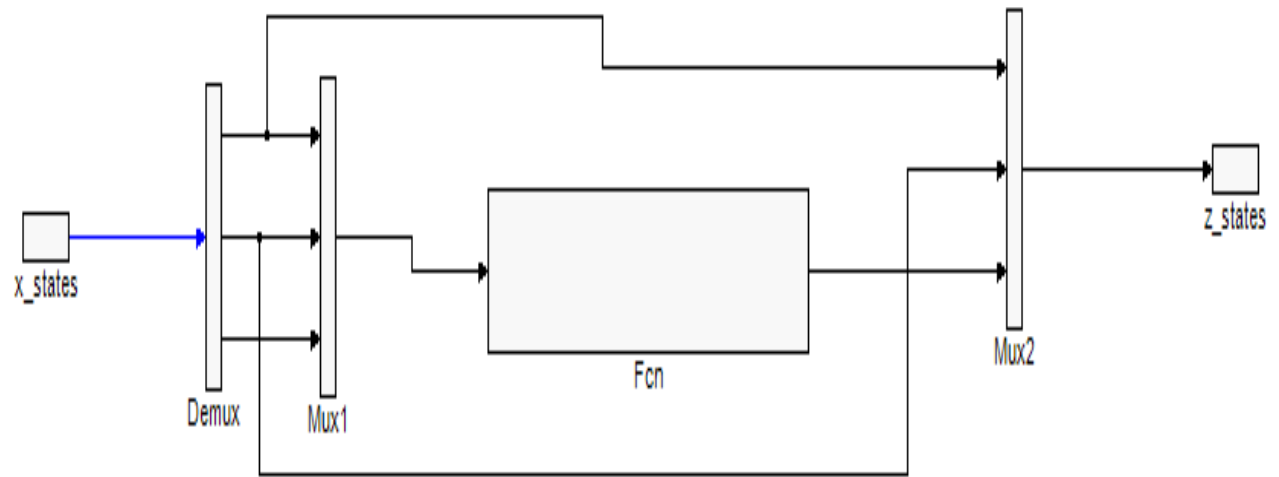
**A.8: TwinCAT Model of the nonlinear input-state linearizing controller-  
NonLinear\_Controller.tmc**



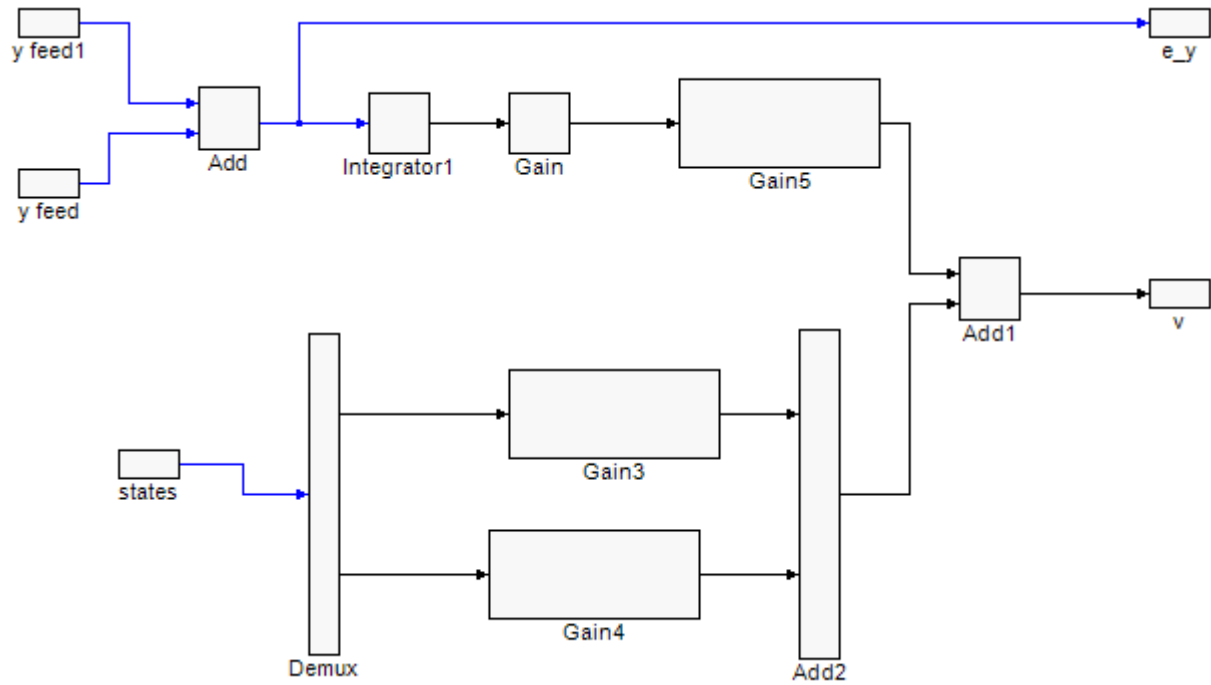
**A.9: TwinCAT Model of the nonlinear model of the magnetic levitation system-  
NonLinear\_Plant.tmc**



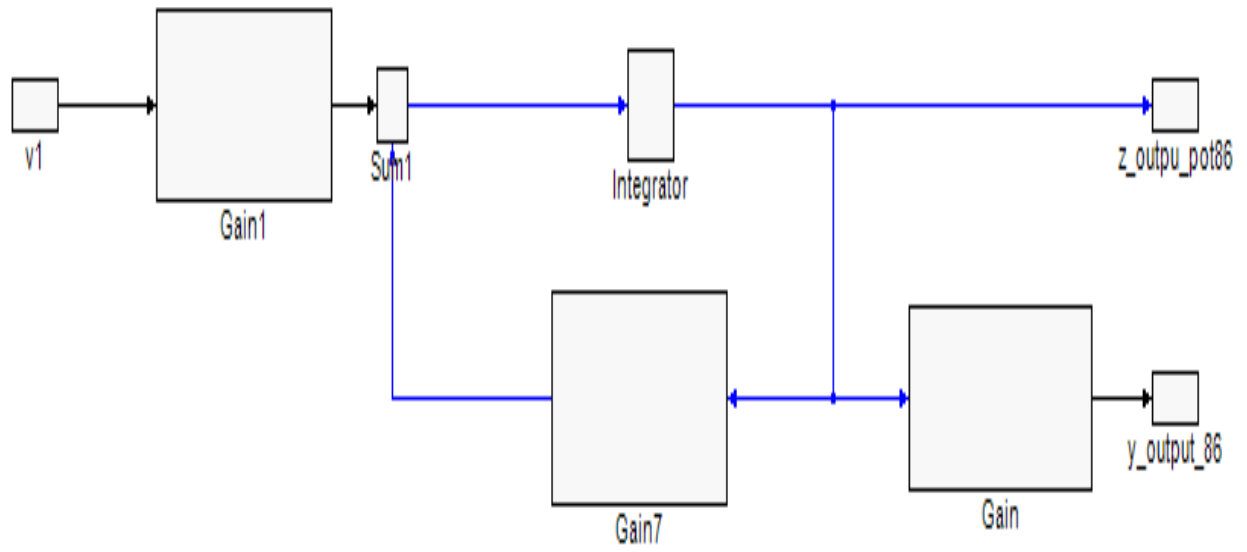
**A.10: TwinCAT Model of the state transformation unit in the input-state linearized closed loop system-State\_Transformation.tmc**



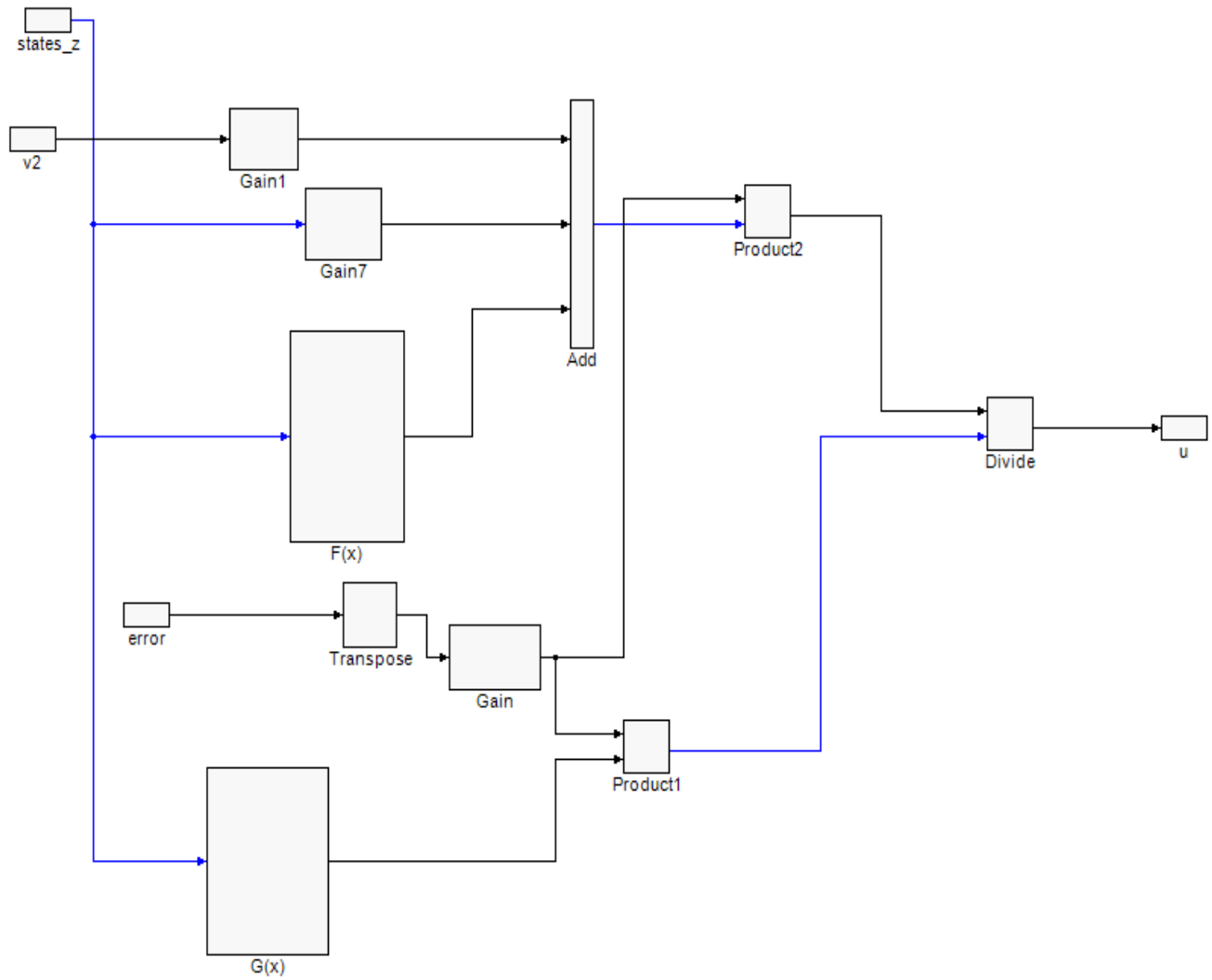
**A.11: TwinCAT Model of the integral optimal controller used in the Lyapunov and MRC based linearized closed loop system-Linear\_ControllerLyap.tmc**



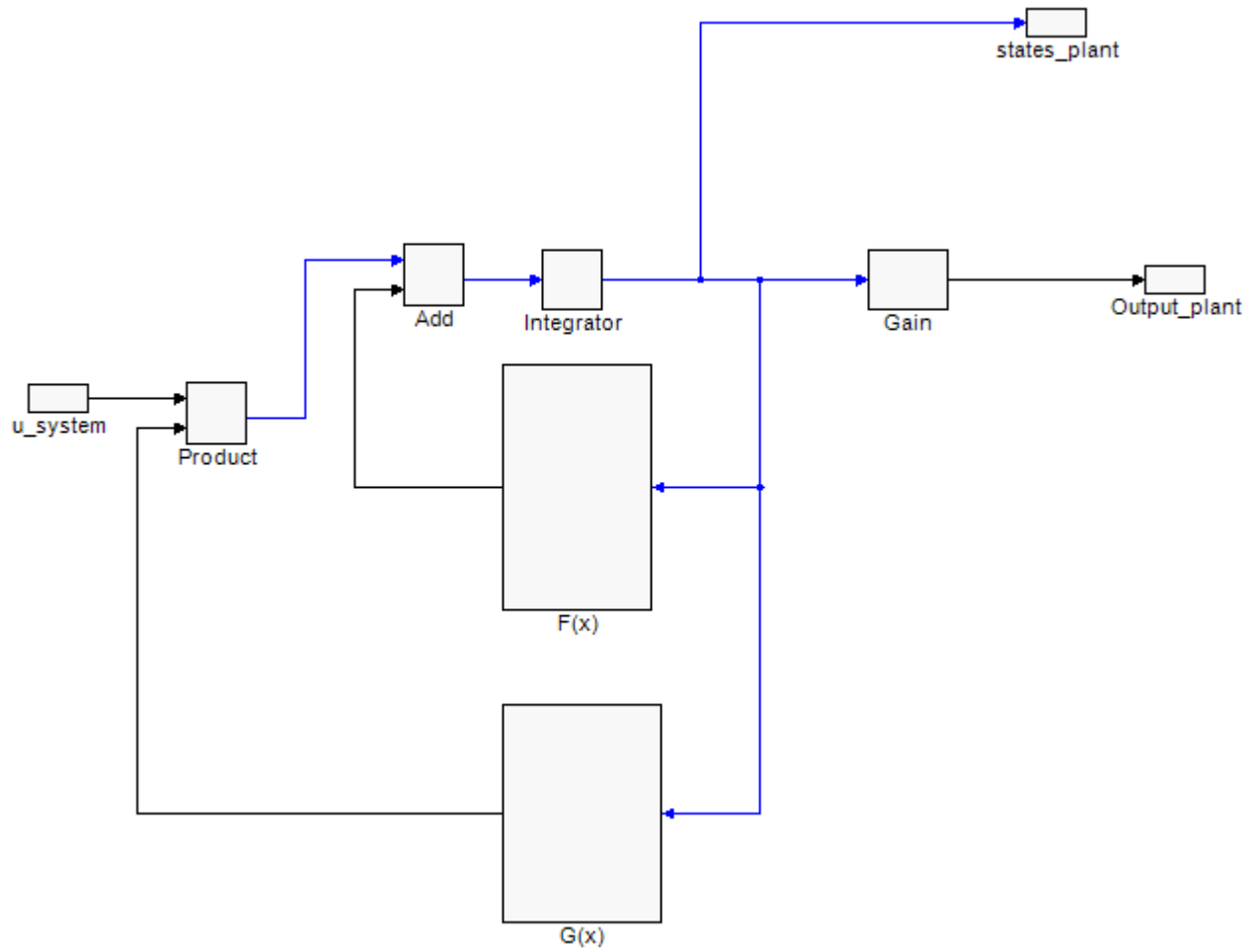
**A.12: TwinCAT Model of the linear reference system-LinearSystem\_model.tmc**



**A.13: TwinCAT Model of the Lyapunov and MRC based nonlinear linearized controller-  
Linearizing\_controller.tmc**

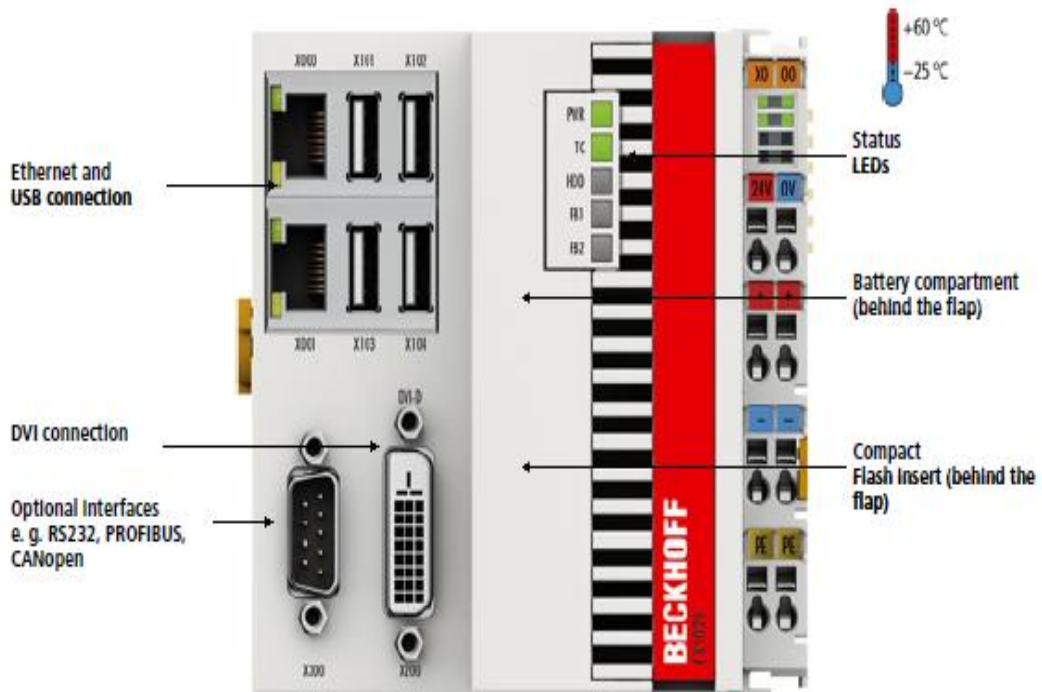


**A.14: TwinCAT Model of the nonlinear plant as part of the Lyapunov and MRC based linearized closed loop system-Nonlinear\_plant.tmc**



# APPENDIX B

## BECKHOFF CX5020





## APPENDIX C

### BECKHOFF CX5020: TECHNICAL DATA

Technical data	CX5010-x1xx	CX5020-x1xx
Processor	processor Intel® Atom™ Z510, 1.1 GHz clock frequency	processor Intel® Atom™ Z530, 1.6 GHz clock frequency
Flash memory	64 MB Compact Flash card (optionally extendable)	
Internal main memory	512 MB RAM (internal, not expandable)	512 MB RAM (optionally 1 GB installed ex factory)
Persistent memory	integrated 1-second UPS (1 MB on Compact Flash card)	
Interfaces	2 x RJ 45, 10/100/1000 Mbit/s, DVI-D, 4 x USB 2.0, optional 1 x RS232/RS422/RS485	
Diagnostics LED	1 x power, 1 x TC status, 1 x flash access, 2 x bus status	
Clock	internal battery-backed clock for time and date (battery exchangeable)	
Operating system	Microsoft Windows CE or Microsoft Windows Embedded Standard	
Control software	TwinCAT 2 PLC runtime or TwinCAT 2 NC PTP runtime	
Power supply	24 V DC (-15 %/+20 %)	
Dielectric strength	500 V (supply/internal electronics)	
Current supply I/O terminals	2 A	
Max. power loss	12 W (including the system interfaces)	12.5 W (including the system interfaces)
Dimensions (W x H x D)	100 mm x 100 mm x 91 mm	
Operating/storage temperature	-25...+60 °C/-40...+85 °C	
Relative humidity	95 %, no condensation	
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27/29	

## APPENDIX D

### D.1 Data type overview, instruction list operators, modifiers and relevant meanings

Data type overview (Beckhoff, 2013)

Data Type	Description
BOOLEAN	8 Bit Boolean value. 1=true, 0= false
SIGNED8	8 Bit signed Integer
SIGNED16	16 Bit signed Integer
SIGNED32	32 Bit signed Integer
UNSIGNED8	8 Bit unsigned Integer
UNSIGNED16	16 Bit unsigned Integer
UNSIGNED32	32 Bit unsigned Integer
UNSIGNED64	64 Bit unsigned Integer
REAL32	32 Bit floating point unit
VISIBLE STRING	ASCII string, variable length, not null terminated

Instruction List operators, modifiers and relevant meaning (TwinCAT, 2013)

Operators	Modifiers	Meaning
LD	N	Make current result equal to the operand
ST	N	Save current result at the position of the Operand
S		Put the Boolean operand exactly at TRUE if the current result is TRUE
R		Put the Boolean operand exactly at FALSE if the current result is TRUE
AND	N, (	Bitwise AND
OR	N, (	Bitwise OR
XOR	(	Bitwise exclusive OR
ADD	(	Addition
SUB	(	Subtraction
MUL	(	Multiplication

DIV	(	Division
GT	(	>
GE	(	>=
EQ	(	=
NE	(	<>
LT	(	<=
JMP	CN	Jump to label
CAL	CN	Call function block
RET	CN	Return from call a function block
)		Evaluate deferred operation

## D.2: Qualifiers of the actions with IEC steps

<b>N</b>	Non-stored	The action is active as long as the step.
<b>R</b>	overriding Reset	The action is deactivated.
<b>S</b>	Set (Stored)	The action is activated and remains active until a Reset.
<b>L</b>	time Limited	The action is activated for a certain time.
<b>D</b>	time Delayed	The action becomes active after a certain time as long as the step is still active.
<b>P</b>	Pulse	The action is executed just one time if the step is active.
<b>SD</b>	Stored and time Delayed	The action is activated after a certain time and remains active until a Reset.
<b>DS</b>	Delayed and Stored	The action is activated after a certain time as long as the step is still active and remains active up to a Reset.
<b>SL</b>	Stored and time Limited	The action is activated for a certain time.

**D.3: Structure Text Operators in order of their binding strength (Bolton, 1996)**

<b>Operation</b>	<b>Symbol</b>	<b>Binding Strength</b>
Put in parentheses	(expression)	Strongest binding
Function call	Function name (parameter list)	
Exponentiation	EXPT	
Negate	-	
Building of complements	NOT	
Multiply	*	
Divide	/	
Modulo	MOD	
Add	+	
Subtract	-	
Compare	<, >, <=, >=	
Equal	=	
Not Equal to	<>	
Boolean AND	AND	
Boolean XOR	XOR	
Boolean OR	OR	Weakest binding

#### D.4: Instruction in ST with examples (TwinCAT, 2013)

Instructions	Examples
Assignment	A:=B; CV := CV + 1; C:=SIN(X);
Calling a function block and use of the FB version	CMD_TMR(IN:=%IX5,PT:=300);A:=CMD_TMR.Q;
RETURN	RETURN;
IF	IF D<0.0 THEN C:=A; ELSIF D=0.0 THEN C:=B; ELSE C:=D; END_IF;
CASE	CASE INT1 OF 1: BOOL1 := TRUE; 2: BOOL2 := TRUE; ELSE BOOL1 := FALSE; BOOL2 := FALSE; END_CASE;
FOR	FOR I:=1 TO 100 BY 2 DO IF ARR[I] = 70 THEN J:=I; EXIT; END_IF; END_FOR;
WHILE	WHILE J<= 100 AND ARR[J] <> 70 DO J:=J+2; END_WHILE;
REPEAT	REPEAT J:=J+2; UNTIL J= 101 OR ARR[J] = 70 END_REPEAT
EXIT	EXIT;
Empty instruction	;

## REFERENCES

Abdelmalek, I., Golea, N. and Hadjili, M. 2007. A New Fuzzy Lyapunov Approach to NonQuadratic Stabilization of Takagi-Sugeno Fuzzy Models. *Journal of Systems Engineering and Electronics*. Volume 18, Issue 3, 2007, pp.598-602.

Ahmad, I. & Javaid, M. 2010 "Nonlinear Model & Controller Design for Magnetic Levitation System," in *Recent Advances in Signal Processing, Robotics and Automation- 9<sup>th</sup> WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA '10)*, Cambridge, pp. 324-328.

Aldea, *et al.*, 2004. "The scope of application of multi-agent systems in the process industry: three cases studies," *Expert Systems with Applications*, vol. 26, pp 39-47.

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.

Astom, K. & McAvoy, T. 1992 "Intelligent control," *Journal of Process Control*, vol.2, pp. 115-127, 1992.

Balasubramanian, S., Brennan, R.W. & Norrie, D.H. 2001. An architecture for metamorphic control of holonic manufacturing systems. *Computers in Industry*, 46(1):13-31.

Baquero, R., Rodriguez, J., Mendoza, S., Decouchant, D. & Papis, A.P. 2012. FunBlocks. A modular framework for Aml system development. *Sensors (Basel, Switzerland)*, 12(8):10259-10291.

Baran, W. 1971 "Optimizing a permanent magnet suspension system for high speed ground transport", *Zeitschrift, fur angewandt physic*, 32;3, pp. 216-18.

Barie, W. & Chiasson, J. 1996 "Linear and Nonlinear state-space controllers of magnetic levitation," *International Journal of Systems Science*, vol. 27, number 11, pp. 1153-1163.

Batzel, J. 2001. Introduction to Linear Control Theory. Lecture 3 Maths notes. University of Graz, Graz, Austria.

Berkelman, P.J & HOLLIS, R.L. 2000: Lorentz magnetic levitation of haptic interaction: Device design, performance, and integration with physical simulations, *International Journal of Robotics Research*, Vol. 19, No 7, pp. 644-667.

Beyhan, S. 2013. Runge-Kutta model-based nonlinear observer for synchronization and control of chaotic systems. *ISA transactions*, 52(4):501-509.

Black, G. 2007. *Multi-agent control of airport baggage handling using IEC 61499 function block*, Computer Systems Engineering, University of Auckland, New Zealand.

Brennan, R.W., Fletcher, M. & Norrie, D.H. 2002. An agent-based approach to reconfiguration of real-time distributed control systems. *IEEE Transactions on Robotics and Automation*, 18(4):444-451.

Camahan, D. & delaHostria, E. 2000. Application integration framework for machinery/process diagnostics. *ISA TECH/EXPO Technology Update Conference Proceedings*, 400:99-107.

CE 152 Magnetic levitation model - educational manual, Humusoft s.r.o., Prague, Czech Republic, 2002.

CE 152 Magnetic levitation model - user's manual, Humusoft s.r.o., Prague, Czech Republic, 2002.

CE 152 Magnetic levitation model [Online]. Humusoft product data sheet. Available: <http://www.humusoft.cz/produkty/models/ce152>.

Chang, Y.H. & Chan, W.S. 2013. Adaptive Dynamic Surface Control for Uncertain Nonlinear Systems With Interval Type-2 Fuzzy Neural Networks. *IEEE transactions on cybernetics*, 44: 293-304.

Chemachema, M. 2012. Output feedback direct adaptive neural network control for uncertain SISO nonlinear systems using a fuzzy estimator of the control error. *Neural networks : the official journal of the International Neural Network Society*, 36: 25-34.

Chen, B., Liu, K., Liu, X., Shi, P., Lin, C. & Zhang, H. 2013. Approximation-Based Adaptive Neural Control Design for a Class of Nonlinear Systems. *IEEE transactions on cybernetics*, Vol.44, No 5.



Chen, H.Y., Liang, J.W. & Wu, J.W. 2013. Active pneumatic vibration control by using pressure and velocity measurements and adaptive fuzzy sliding-mode controller. *Sensors (Basel, Switzerland)*, 13(7):8431-8444.

Christensen, J. 2000. "Design patterns for systems engineering with IEC 61499," in *Verteilte Automatisierung - Modelle und Methoden für Entwurf, Verifikation, Engineering und Instrumentierung*, Magdeburg, Germany.

Citi, L., Valenza, G. & Barbieri, R. 2012. Instantaneous estimation of high-order nonlinear heartbeat dynamics by Lyapunov exponents. *Conference proceedings : ...Annual International Conference of the IEEE Engineering in Medicine and Biology Society.IEEE Engineering in Medicine and Biology Society.Conference*, 2012:13-16.

CLEMENTS-JEWERY, K., & JEFFCOAT, W. 1996. The PLC workbook. Programmable Logic Controllers made easy. *Prentice Hall international UK) limited Campus 400, Maylands Avenue. ISBN-0-13-489840-0.*

COSTANZO, M.P.B. 1997. Programmable Logic Controllers. *Halesowen College, UK. ISBN 0 340 69258 8.*

Cornelius, S.P., Kath, W.L. & Motter, A.E. 2013. Realistic control of network dynamics. *Nature communications*, 4:1942.

Dai, W. 2012. *On migration of scan cycle based PLC programs to distributed component-based event driven software architecture with semantic correctness assurance*, PhD Thesis, University of Auckland, New Zealand.

Dey, C., Mudi, R.K. & Simhachalam, D. 2014. A simple nonlinear PD controller for integrating processes. *ISA transactions*, 53(1):162-172.

Diab, M.G.H.S. 2009. *Automated services restaurant utilizing robotics, automation, and embedded technologies*. Software Engineering, University of Auckland, New Zealand.

Ferrarini, L. & Veber, C. 2004, "Implementation approaches for the execution model of IEC 61499 applications", *2nd IEEE International Conference on Industrial Informatics: Collaborative Automation - One Key for Intelligent Industrial Environments, INDIN'04, June 24, 2004 - June 26* Institute of Electrical and Electronics Engineers Computer Society, Berlin, Germany, pp. 612.

Filho, M and Munaro, E. 2001 "A Design Methodology of Tracking Controllers for Magnetic Levitation Systems," Proceeding of the 2001 IEEE International Conference and Systems, E84-D(10):1398-1401.

Fletcher, M. & Brennan, R.W. 2001. Designing holonic manufacturing systems using the IEC 61499 (function block) architecture. *IEICE Transactions on Information and Systems*, E84-D(10):1398-1401.

Fletcher, M., Brennan, R.W. & Norrie, D.H. 2003, "Modeling and reconfiguring intelligent holonic manufacturing systems with Internet-based mobile agents", Kluwer Academic Publishers, pp. 7.

Fuente, L.A., Lones, M.A., Turner, A.P., Stepney, S., Caves, L.S. & Tyrrell, A.M. 2013. Computational models of signalling networks for non-linear control. *Bio Systems*, 112(2):122-130.

Function blocks: *International Standard IEC61499*, International Electrotechnical Commission, 2005.

Gachter, S. 2000 "Optimization of the magnetic levitation process by iterative feedback tuning," M.S. thesis, Swiss Federal Institute of Technology, Lausanne.

Gazdos, F., Dostal, P. & Marholt, F. 2011 "Robust control of unstable systems: algebraic approach using sensitivity function", *International Journal of Mathematical models and methods in Applied Sciences*, Issue 7, Volume 5.

Gazdos, F., Dostal, P. & Pelikan, R. 2009 "Polynomial approach to control system design for magnetic levitation system," *Cybernetic Letters*, December Issue, pp. 1-19.

Hagge, N. & Wagner, B. 2004, "Mapping reusable control components to java language constructs", *2nd IEEE International Conference on Industrial Informatics: Collaborative Automation - One Key for Intelligent Industrial Environments, INDIN'04, June 24, 2004 - June 26* Institute of Electrical and Electronics Engineers Computer Society, Berlin, Germany, pp. 108.

Han, S.I. & Lee, J.M. 2014. Recurrent fuzzy neural network backstepping control for the prescribed output tracking performance of nonlinear dynamic systems. *ISA transactions*, 53(1):33-43.

Hauser, J., Sastry, S & Kokotovic, P. 1992. "Nonlinear control via approximate input-output linearization: the ball and beam example," *IEEE Transactions on Automatic Control*, vol. 37, pp. 392-398.

Hedrick, J. K. and Girard, A. 2005 *Control of Nonlinear Dynamic System: Theory and Applications*. Columbia University, Fall 2005.

Humusoft Praha, CE 152 Educational Manual Magnetic Levitation Model. Pelikan, R. (2006).

Hurley, W. G. & Wolfle, W. H 1997. "Electromagnetic design of a magnetic suspension system," in *IEEE Trans. On Education*, vol. 40, no. 2, pp 124-30.

Hussain, T. & Frey, G. 2004, "Developing IEC 61499 compliant distributed systems with network enabled controllers", *2004 IEEE Conference on Robotics, Automation and Mechatronics, December 1, 2004 - December 3* Institute of Electrical and Electronics Engineers Inc, Singapore, Singapore, pp. 507.

Jain, S., Yuan, C. & Ferreira, P. 2002, "Embench: A rapid prototyping environment for numerical control systems", *ASME 2002 International Mechanical Engineering Congress and Exposition, IMECE2002, November 17, 2002 - November 22* American Society of Mechanical Engineers, New Orleans, LA, United states, pp. 7.

Jayawant, B. V. 1981. "Electromagnetic Levitation and Suspension Techniques," Edward Arnold Ltd., London.

Jiang, S. & Fang, H. 2013. Infinity static output feedback control for nonlinear networked control systems with time delays and packet dropouts. *ISA transactions*, 52(2):215-222.

Kashaninia, A. & Nikravesh, S.K. 2013. Modular design using feedback domination method for adaptive regulation of high-order lower-triangular nonlinear systems. *ISA transactions*, 52(2):223-230.

Keskar, P. Y. 1990. Structured approach in PLC (Programmable logic controller) programming for water/wastewater applications. *ISA transaction*, 29(2):47-56.

Khamesee, M. B., Kato, N., Nomura, Y. & Nakamura, T. 2000. "Design and control of a microbotic system using magnetic levitation," *EEE-ASME Transaction on Mechatronics*, Vol. 7, No. 1, pp 1-14.

Krener, A. & Respondek, W. 1985. "Nonlinear Observers with Linearizable Error Dynamics," *Siam J. Control and Optimization*, Vol. 23, No. 2, pp. 197-216.

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.

Lee, B. S. 2000. "Linear Switched Reluctance Machine Drives with Electromagnetic Levitation and Guidance Systems," Ph.D. Thesis, Virginia Tech.

Lewis, R. 2001. *Modeling Control Systems Using IEC 61499 - Applying Function Blocks to Distributed Systems*. London: The Institution of Electrical Engineers.

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.

Liu, Y., Slotine, J. & Barabasi, A, L. 2013. Observability of complex systems. *Proceeding of the National Academy of Science of the United States of America*, 110(7):2460-2465.

Ma, J. 2013, "The effect of the distance between the permanent magnets on the levitation force in hybrid magnetic levitation system", *2nd International Conference on Materials Science and Manufacturing, ICMSM 2013, March 29, 2013 - March 31* Trans Tech Publications Ltd, Zhangjia Jie, China, pp. 278.

Mařík, V., Vyatkin, V., Colombo, A.W. & SpringerLink 2007. *Holonic and multi-agent systems for manufacturing*. Berlin: Springer-Verlag.

Mizuya, T. & Miyazawa, I. 2003, "Representation and Implementation of Function Blocks by the Event Delegation Model", *The 29th Annual Conference of the IEEE Industrial Electronics Society, November 2, 2003 - November 6* Institute of Electrical and Electronics Engineers Computer Society, Roanoke, VA, United states, pp. 2522.

Munaro, C., Filho, M., Borges, R., Munareto, S. & Costa, W. 2002. "Modeling and Observer-Based Nonlinear Control of a Magnetic Levitation Systems," Proceeding of the 2002 IEEE International Conference on Control Application, pp. 162-167.

Ogata, K. 2002. Modern Control Engineering Fourth Edition. Upper Saddle River, New Jersey, Prentice Hall.

Olsen, S., Wang, J., Ramirez-Serrano, A. & Brennan, R.W. 2005, "Contingencies-based reconfiguration of distributed factory automation", *14th International Conference on Flexible Automation and Intelligent Manufacturing* Elsevier Ltd, , pp. 379.

Ouari, K., Rekioua, T. & Ouhrouche, M. 2014. Real time simulation of nonlinear generalized predictive control for wind energy conversion system with nonlinear observer. *ISA transactions*, 53(1):76-84.

Pang, C. 2012. *Model-driven development of distributed automation intelligence with IEC 61499*. Electrical and Electronic Engineering, University of Auckland, New Zealand.

Park, J. 2009. *Introduction of software engineering support for IEC 61499 FB: refactoring elements of control applications*. Software Engineering, University of Auckland, New Zealand.

Pelikan, R. 2006. "Magnetic Levitation – modeling, simulations and control," Dipl. Thesis, UTB, Zli.

Piyabongkarn, D. 2000. "Digital Control of a Magnetic Levitation System Through xPC Real-Time Operating System: Classical, Feedforward and Artificial Neural Network," Master's Thesis, Mechanical and Aerospace Engineering, UT Arlington.

Prokop, R., Volkova, N. & Prokopova, Z. 2001. "Tracking and Disturbance Attenuation for Unstable Systems: Algebraic Approach," in Recent Researches in Automatic Control – 13<sup>th</sup> WSEAS International Conference on Automatic Control, Modeling and Simulation (ACMOS '11), Lanzarote, Spai, pp. 57-62.

Rafee Nekoo, S. 2013. Nonlinear closed loop optimal control: a modified state-dependent Riccati equation. *ISA transactions*, 52(2):285-290.

Rogers, J. & Rabb, R. 2010, "Control theory in practice: Magnetic Levitation", *ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis, ESDA2010, July 12, 2010 - July 14* American Society of Mechanical Engineers, Istanbul, Turkey, pp. 823.

Rudra, S., Kumar Barai, R. & Maitra, M. 2014. Nonlinear state feedback controller design for underactuated mechanical system: A modified block backstepping approach. *ISA transactions*, Vol. 53, Issue 2, pp. 317-326.

Savran, A. 2013. Discrete state space modeling and control of nonlinear unknown systems. *ISA transactions*, 52(6):795-806.

Savran, A. & Kahraman, G. 2013. A fuzzy model based adaptive PID controller design for nonlinear and uncertain processes. *ISA transactions*, Vol. 53, Issue 2, pp. 280-288.

Scarlett, J., Brennan, R.W. & Norrie, D.H. 2003, "An interface to support real-time distributed control", *System Security and Assurance, October 5, 2003 - October 8* Institute of Electrical and Electronics Engineers Inc, Washington, DC, United states, pp. 636.

Schnakenbourg, C., Faure, J.-. & Lesage, J.-. 2002, "Towards IEC 61499 function blocks diagrams verification", *2002 IEEE International Conference on Systems, Man and Cybernetics, October 6, 2002 - October 9* Institute of Electrical and Electronics Engineers Inc, Yasmine Hammamet, Tunisia, pp. 210.

Schwab, C., Tangermann, M., Luder, A., Kalogeras, A. & Ferrarini, L. 2004, "Mapping of IEC 61499 function blocks to automation protocols within the TORERO approach", *2nd IEEE International Conference on Industrial Informatics: Collaborative Automation - One Key for Intelligent Industrial Environments, INDIN'04, June 24, 2004 - June 26* Institute of Electrical and Electronics Engineers Computer Society, Berlin, Germany, pp. 149.

Sintayehu, C. 2007. "Magnetic levitation on steel ball," Master's Thesis, Control Engineering, Addis Ababa University School of Graduate Studies.

Skogestad, S., Havre, K. & Larsson, T. 2002. "Control limitations for unstable plants," in Proc. 15<sup>th</sup> Triennial World Congress, Barcelona, pp. 485-490.

Slotine J. 1991. *Applied Nonlinear Control*, Prentice Hall, New Jersey.

Shakouri, P., Ordys, A. & Askari, M.R. 2012. Adaptive cruise control with stop&go function using the state-dependent nonlinear model predictive control approach. *ISA transactions*, 51(5):622-631.

Song, Z. & Sun, K. 2014. Adaptive backstepping sliding mode control with fuzzy monitoring strategy for a kind of mechanical system. *ISA transactions*, 53(1):125-133.

Strasser, T., Auinger, F. & Zoitl, A. 2004, "Development, implementation and use of an IEC 61499 function block library for embedded closed loop control", *2nd IEEE International Conference on Industrial Informatics: Collaborative Automation - One Key for Intelligent Industrial Environments, INDIN'04, June 24, 2004 - June 26*Institute of Electrical and Electronics Engineers Computer Society, Berlin, Germany, pp. 594.

Sun, Y.J., Wu, Y.B. & Wang, C.C. 2013. Robust stabilization for a class of nonlinear systems via a single input control applicable to chaotic systems and its circuit implementation. *Chaos (Woodbury, N. Y.)*, 23(2):023127.

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., Hori, T. and Wang, H, O. 2003. A Multiple Lyapunov Function Approach to Stabilization of Fuzzy Control Systems. *IEEE Trans. Fuzzy Systems*, Volume 11, No. 4, pp. 582-589.

Thramboulidis, K.C. 2003, "Towards an engineering tool for implementing reusable distributed control systems", *9th European Software Engineering Conference Held Jointly with 11th ACM SIGSOFT International Symposium on Foundations of Software Engineering, ESEC/FSE-11, September 1, 2003 - September 5*Association for Computing Machinery, Helsinki, Finland, pp. 351.

Tong, S., Wang, T., Li, Y. & Zhang, H. 2013. Adaptive Neural Network Output Feedback Control for Stochastic Nonlinear Systems with Unknown Dead-Zone and Unmodeled Dynamics. *IEEE transactions on cybernetics*, to be published.

Tsuchiya, *et al.*, 1999. "Development of a distributed process control programming tool for function block description," in *Emerging Technologies and Factory Automation, 1999*.

*Proceedings. ETFA '99. 1999 7<sup>th</sup> IEEE International Conference on*, 1999, pp. 11321-1325 vol. 2.

Valencia-Palomo, G. & Rossiter, J.A. 2011. Programmable logic controller implementation of an auto-tuned predictive control based on minimal plant information. *ISA transactions*, 50(1):92-100.

Valeriy, V. 2009. "The IEC 61499 standard and its semantics: Bridging the Gap between PLC Programming Languages and Distributed Systems," *IEEE Industrial Electronics Magazine.*, Vol. 3, pp. 40-48.

Vallance, P, J. 2001. "Digital control of levitation," Master's Thesis, Bradley Department of Electrical and Computer Engineering, Virginia Polytechnic Institute and State University.

Vrba, P. 2003, "JAVA-based agent platform evaluation", *First International Conference on Industrial Applications of Holonice and Multi-Agent Systems, HoloMAS 2003, September 1, 2003 - September 3*Springer Verlag, Prague, Czech republic, pp. 47.

Vyatkin, V. 2003. "Intelligent mechatronic components: control system engineering using an open distributed architecture.," presented at the Emerging Technologies and Factory Automation 2003 (ETFA '03), IEEE Conference.

Vyatkin, V. 2007. IEC 61499 function blocks for embedded and distributed control systems design. Research Triangle Park, NC: ISA-Instrumentation, Systems, and Automation Society, ISA/O3neida.

Vyatkin, V. 2012. *IEC 61499 function blocks for embedded and distributed control systems design*. 2nd ed. Research Triangle Park, NC: Isa.

Vyatkin, V. & Hanisch, H. 2001, "Application of visual specifications for verification of distributed controllers", *2001 IEEE International Conference on Systems, Man and Cybernetics, October 7, 2001 - October 10*Institute of Electrical and Electronics Engineers Inc, Tucson, AZ, United states, pp. 646.



- Vyatkin, V. & Hanisch, H. 2000. XI Workshop on Supervising and Diagnostics of Machining Systems, Mar 12 - Mar 17 2000. In *Modelling of IEC 61499 function blocks a clue to their verification*. Karpacz, Poland: Wydawnictwo Politechniki Wroclawskiej: 59-68.
- Wang, H., Chen, B., Liu, X., Liu, K. & Lin, C. 2013. Robust Adaptive Fuzzy Tracking Control for Pure-Feedback Stochastic Nonlinear Systems With Input Constraints. *IEEE transactions on cybernetics, ISA transactions*, 43(6):2093-2104 .
- Wang, J., Hu, Z. & Ye, Z. 2014. Static feedback stabilization of nonlinear systems with single sensor and single actuator. *ISA transactions*, 53(1):50-55.
- Wang, R., Li, J., Zhang, S., Gao, D. & Sun, H. 2013. Robust adaptive control for a class of uncertain nonlinear systems with time-varying delay. *TheScientificWorldJournal*, Vol. 2013, p.8.
- Wong, T. 1986. Design of a magnetic levitation system: an undergraduate project. *IEEE Transactions on Education*, 29, 196-200.
- Wu, J., Peng, H., Chen, Q. & Peng, X. 2014. Modeling and control approach to a distinctive quadrotor helicopter. *ISA transactions*, 53(1):173-185.
- Xia, F., Yin, H., Wang, Z. & Sun, Y. 2004, "Function block oriented architecture for open distributed automation", *WCICA 2004 - Fifth World Congress on Intelligent Control and Automation, Conference Proceedings, June 15, 2004 - June 19* Institute of Electrical and Electronics Engineers Inc, Hangzhou, China, pp. 2706.
- Yang, C. 2012. *Distributed automation of power systems with IEC61499: GOOSE messaging and design flow*. University of Auckland, New Zealand.
- Yang, C. & Vyatkin, V. 2009. "Automated Model Transformation between MATLAB Simulink/Stateflow and Function Blocks," in *INCOM*, Vol. 13, pp. 205-210.
- Yang, C. & Vyatkin, V. 2008. "Design and validation of distributed control with decentralized intelligence in process industries: A survey," in 6<sup>th</sup> IEEE International Conference on Industrial Informatics, 2008. *INDIN 2008.*, Daejeon, Korea, pp. 1395-1400.
- Yuan, C. & Ferreira, P. 2004, "An integrated environment for the design and control of deadlock-free flexible manufacturing cells", *2004 ASME International Mechanical Engineering*

Congress and Exposition, IMECE 2004, November 13, 2004 - November 19 American Society of Mechanical Engineers, Anaheim, CA, United states, pp. 471.

Yuan, C. & Ferreira, P. 2003, "An integrated rapid prototyping environment for reconfigurable manufacturing systems", *2003 ASME International Mechanical Engineering Congress, November 15, 2003 - November 21* American Society of Mechanical Engineers, Washington, DC., United states, pp. 737.

Zeng, S., Hu, H., Xu, L. & Li, G. 2012. Nonlinear adaptive PID control for greenhouse environment based on RBF network. *Sensors (Basel, Switzerland)*, 12(5):5328-5348.

Zha, W., Zhai, J. & Fei, S. 2013. Global output feedback control for a class of high-order feedforward nonlinear systems with input delay. *ISA transactions*, 52(4):494-500.

Zhabelova, G. 2009. *Investigation of intelligent smart grid automation implementation by means of IEC 61499 and IEC 61850*. Electrical and Electronic Engineering, University of Auckland.

Zhang, X., Brennan, R.W., Xu, Y. & Douglas, H.N. 2001, "Runtime adaptability of a concurrent function block model for a real-time holonic controller", *2001 IEEE International Conference on Systems, Man and Cybernetics, October 7, 2001 - October 10* Institute of Electrical and Electronics Engineers Inc, Tucson, AZ, United states, pp. 164.

Zhao, G., Sun, K. & Li, H. 2013. Tensor product model transformation based adaptive integral-sliding mode controller: equivalent control method. *TheScientificWorldJournal*, 2013:726963.

Zhou, Q., Shi, P., Liu, H. & Xu, S. 2012. Neural-network-based decentralized adaptive output-feedback control for large-scale stochastic nonlinear systems. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, 42(6):1608-1619.

Zoittl, R. Smodic, C. Sunder, and Grabmair, G. 2006. "Enhanced Real- Time Execution of Modular Control Software based on IEC 61499," IEEE International Conference on Robotics and Automation (ICRA), Orlando, USA.

Zurawski, R. 2006. *Integration technologies for industrial automated systems*. Boca Raton, FL: Taylor & Francis.