

The Islamic University of Gaza
Deanery of Graduate Studies
Faculty of Engineering
Electrical Engineering Department



الجامعة الإسلامية - غزة
عمادة الدراسات العليا
كلية الهندسة
قسم الهندسة الكهربائية

Master Thesis

MULTI-RATE RIPPLE-FREE DEADBEAT CONTROL FOR NON-LINEAR SYSTEMs

A Thesis Submitted To The Faculty Of Engineering.
In Partial Fulfillment of the Requirements For The
Degree of Master of Science in
Electrical Engineering
The Islamic University Of Gaza, Palestine

Prepared By:
Mohammed Elamassie

Advisor
Dr. Hatem Elaydi

2011



نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة الدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ محمد تيسير عوض الله العمصي لنيل درجة الماجستير في كلية الهندسة قسم الهندسة الكهربائية/أنظمة تحكم وموضوعها:

Multi-Rate Ripple-Free Deadbeat Control For Non-Linear Systems

وبعد المناقشة العلنية التي تمت اليوم السبت 05 جماد أول 1432هـ، الموافق 2011/04/09م الساعة الحادية عشرة صباحاً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

.....	مشرفاً ورئيساً	د. حاتم علي العايدي
.....	مناقشاً داخلياً	د. أسعد نمر أبو جاسر
.....	مناقشاً داخلياً	د. باسل محمود حمد

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية الهندسة/ قسم الهندسة الكهربائية/ أنظمة تحكم.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق،،،

عميد الدراسات العليا

د. زياد إبراهيم مقداد

ABSTRACT

Deadbeat control was investigated extensively in the last couple of decades, but a need still exists to offer a better methodology for both performance and robustness for linear and nonlinear systems. This research proposes a new design methodology for ripple-free deadbeat control for nonlinear systems in discrete-time. The proposed method combines two ripple-free deadbeat control laws. The new controller guarantees robustness and handles multi-rate systems. Multi-rate digital control is used when processing time is greater than controller updating time; thus, processing time can be decreased by increasing at least one of the following: state feedback sampling time; output feedback-sampling time, input sampling time, and/or decreasing controller-updating rate for some processes; therefore, the sampling time is not unique for the whole system. The new controller is applied on magnetic ball levitation CE 152 as a case study for nonlinear systems.

The deadbeat tracking formulation is based on a polynomial approach and a time domain approach; thus, the proposed method combines two deadbeat control laws. In this thesis, the time domain approach is used to ensure the local behavior of the nonlinear system, while the polynomial approach is used to provide deadbeat control to the linearized nonlinear system. Nonlinear system is linearized using feedback linearization, A second order linear model is used to approximate the nonlinear system based on two dominant poles; thus, the settling time which depend on the minimum order solution of Diophantine equations is minimized. Xilinx MATLAB toolbox is used to implement the new controller on real-time magnetic ball levitation. Sub controller, which depends on polynomial approach, is written in VHDL code, simulated and compared with original and approximated SIMULINK model using Xilinx DSP toolbox.

Simulation and real-time results shows that the output signal exactly tracks the input sinusoidal signal in short settling time. The time domain specification for the output signal, control signal, and error signal are computed and satisfied the requirement and constraints. A time delay is also presented and included in the model and the solution is based on two Diophantine equations that form the ripple-free deadbeat.

ملخص البحث

على الرغم من كل الجهود السابقة لتصميم نظام التحكم المرهق أو ما يمكن تسميته بنظام الضربة القاضية لأنظمة خطية وغير خطية في العقدين الأخيرين، فلا زلنا بحاجة لتصميم أنظمة تحكم بطرق جديدة لضمان الأداء الأفضل والقوة للأنظمة الخطية وغير خطية. هذا البحث يقدم نظام تحكم جديد لأنظمة غير خطية لها أكثر من معدل لأخذ العينات. وهو يقوم بالدمج بين نظامين يقومان سوياً بالتحكم بنظام غير خطي بسرعة فائقة وبثبات وقوة بحيث لا توجد نسبة خطأ في تتبع الإشارة المرجعية ولا يوجد اهتزاز في تتبعها. المتحكم ذو المعدلات المختلفة لأخذ العينات يستخدم لعدة أسباب منها أن زمن معالجة البيانات أكبر من زمن أخذ العينات؛ لذلك، يمكن تقليل زمن معالجة البيانات بزيادة واحد على الأقل من التالي: زمن أخذ عينات الإشارة الراجعة من متغيرات الحالة (state feedback sampling time)؛ زمن أخذ عينات الإشارة الراجعة من إشارة الخرج (output feedback-sampling time)؛ زمن أخذ عينات الإشارة الداخلة (input sampling time)؛ و/أو زيادة زمن معالجة البيانات؛ من أجل ذلك، تم أخذ العينات من أكثر من مدخل وبمعدلات مختلفة لأخذ العينات فمعدل أخذ العينات ليس واحد في كل النظام. تم تطبيق المتحكم الجديد على جهاز رفع الكرة مغناطيسياً CE 152 حيث أنه يعتبر مثال للأجهزة الغير خطية.

يتم إيجاد المتحكم المرهق لأنظمة خطية بطريقتين، إحداها هي طريقة المجال الزمني والأخرى باستخدام مجال المعادلات. طريقة المجال الزمني تعتمد على الحل الذي يستهلك طاقة أقل ويحل المشكلة باستخدام معادلات فضاء الحالة (state space). أما طريقة مجال المعادلات فإنها تعتمد على حل معادلة دايوفانتين (Diophantine) ويحل المشكلة باستخدام معادلة النقل (Transfer Function). أما الأنظمة غير الخطية يتم إيجاد نظام التحكم المرهق بطريقة المجال الزمني. في هذه الدراسة تم إيجاد نظام التحكم المرهق لأنظمة تحكم غير خطية لها معدلات مختلفة لأخذ العينات بالدمج بين طريقة المجال الزمني وطريقة مجال المعادلات حيث أن طريقة المجال الزمني تستخدم لضمان الإستجابة الجزئية والمحلية (local behavior) لأنظمة غير خطية حيث يتحول النظام الغير خطي إلى نظام خطي، بينما طريقة مجال المعادلات تستخدم لضمان الإستجابة التامة للنظام الغير خطي والذي تم تحويله لنظام خطي. الأنظمة الغير خطية يتم تحويلها إلى أنظمة خطية باستخدام الطريقة التي تعتمد على متغيرات الحالة وإشارة المخرج (feedback linearization). النظام الغير خطي سيتم عمل تقريب لادائه لنظام خطي من الدرجة الثانية بالإعتماد على الجذرين الأكثر تأثيراً بسلوك النظام (two dominant poles)؛ حيث أن زمن وصول إستجابة النظام لحالة الإستقرار (settling time) والتي تعتمد على حل الدرجة الأصغر لمعادلة دايوفانتين سيصبح أصغر. بيئة المحاكاة SIMULINK الموجودة في برنامج MATLAB تم استخدامها لعمل محاكاة لنظام رفع الكرة مغناطيسياً ثم استخدام Xilinx MATLAB toolbox لربط جهاز رفع الكرة مغناطيسياً وتطبيق نظام التحكم الجديد عليه والتحكم به عن طريق بيئة SIMULINK. الجزء الخاص بمجال المعادلات تم تحويله إلى لغة VHDL ومن ثم عمل مقارنة لكود VHDL مع نظام المتحكم الحقيقي باستخدام Xilinx DSP toolbox.

المحاكاة والتطبيق العملي للمتحكم الجديد أظهروا أن الإستجابة تتبع الإشارة المرجعية بدقة في زمن قصير جداً. الإستجابة والخصائص الخاصة بإشارة الإستجابة، وإشارة التحكم، وإشارة الخطأ حققوا المطلوب والتزموا بالقيود. التأخير الزمني تم التعامل معه أيضاً معتمدين على معادلتى دايوفانتين لإيجاد نظام التحكم المرهق الخالي من التموجات بمعدلات مختلفة لأخذ العينات.

DEDICATION

I dedicate this work to my Mother and my Father who encourage me to complete my studies, who taught me when I was child, who gave me their time, love, and attention.

I also dedicate this thesis to my beloved grandparents, my sisters, my wife, and my son: Taysier.

ACKNOWLEDGEMENT

At the beginning, I thank ALLAH for giving me the strength and health to let this work see the light.

I thank my supervisor Dr. Hatem A. Elaydi for his time, consideration, suggestions, ideas and advice during this thesis. Special thanks go to Dr. Basil Hamed and Dr. Assad Abu-Jasser - thesis examiners- for their patience, guidance, and generous supports during this research.

Many thanks go to my best friends for their help. I also thank my wife for her infinite support and patience. Words will not be enough to thank my family for their patience and encouragement during my thesis. Finally, thanks for everyone who has raised his hands and prayed ALLAH for my success.

TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION.....	1
1.1.	GENERAL INTRODUCTION.....	1
1.2.	CLOSED-LOOP CONTROL SYSTEM:.....	1
1.3.	DIGITAL CONTROLLER:.....	2
1.3.1.	<i>Features of Digital Controllers</i>	2
1.3.2.	<i>Digital controller requires:</i>	2
1.4.	PREVIOUS STUDIES:.....	3
1.5.	THESIS CONTRIBUTION:.....	4
1.6.	OUTLINE OF THE THESIS.....	4
CHAPTER 2	BACKGROUND.....	5
2.1.	MULTI-RATE DIGITAL CONTROL.....	5
2.2.	REASONS FOR USING MULTI-RATE DIGITAL CONTROL.....	5
2.3.	MULTI-RATE RIPPLE-FREE DEADBEAT CONTROL FOR NONLINEAR SYSTEMS.....	6
2.3.1.	<i>Multi-rate input controller for linear system (Hiroshi approach):</i>	6
2.3.2.	<i>Ripple-free deadbeat controller (R. PAZ approach)</i>	7
2.3.3.	<i>State and output feedbacks:</i>	7
2.3.4.	<i>Main objective</i>	8
2.4.	DEADBEAT CONTROLLER FOR LINEAR SYSTEMS (POLYNOMIAL APPROACH).....	8
2.5.	NONLINEAR SYSTEMS.....	10
2.6.	FEEDBACK LINEARIZATION.....	11
2.7.	SECOND ORDER APPROXIMATION.....	12
2.8.	DESIGNING STEPS TO DEADBEAT MAGNETIC BALL LEVITATION.....	15
2.9.	CONSTRAINTS.....	15
CHAPTER 3	MAGNETIC BALL LEVITATION CE 152.....	16
3.1.	INTRODUCTION TO MAGNETIC BALL LEVITATION CE152.....	16
3.2.	MODEL ANALYSIS.....	17
3.2.1.	<i>D/A converter</i>	18
3.2.2.	<i>Power amplifier:</i>	18
3.2.3.	<i>Ball and coil subsystem:</i>	19
3.2.4.	<i>Position sensor</i>	20
3.2.5.	<i>A/D converter</i>	21
3.2.6.	<i>Magnetic constant “k_c”</i>	21
3.3.	COMPLETE MODELING:.....	22
CHAPTER 4	FEEDBACK LINEARIZATNION.....	23
4.1.	NONLINEAR SYSTEMS.....	23
4.2.	TAYLOR SERIES.....	23
4.3.	STATE SPACE OF MAGNETIC BALL LEVITATION CE 152.....	25
4.4.	CONTROLLABILITY AND OBSERVABILITY TESTS:.....	30
4.5.	TRANSFER FUNCTION OF LINEARIZED MODEL.....	31
4.6.	SPECIFIED TRANSFER FUNCTION.....	31
4.7.	STATE FEEDBACK.....	32
4.8.	STATE FEEDBACK AND INTEGRAL CONTROLLER.....	33
4.9.	DIGITAL CONTROLLER FOR FEEDBACK LINEARIZATION.....	34
4.10.	PARAMETERS MODIFICATIONS FOR REAL TIME MAGNETIC BALL LEVITATION.....	35
4.11.	SIMPLIFICATION OF FEEDBACK LINEARIZATION OF REAL-TIME SYSTEM.....	37
4.12.	DIGITAL EQUATIONS OF FEEDBACK LINEARIZATION.....	39
4.12.1.	<i>Feedback Transfer function</i>	39
4.12.2.	<i>Integrator</i>	39
4.12.3.	<i>Saturation</i>	39

CHAPTER 5	METHODOLOGY AND APPROACH.....	40
5.1.	INTRODUCTION	40
5.2.	OBTAINING A MODEL	40
5.3.	DECREASING CONTROL SIGNAL USING THE FIRST APPROACH	43
5.4.	DECREASING CONTROL SIGNAL BY SECOND APPROACH	45
5.5.	MULTI-RATE RIPPLE-FREE DEADBEAT CONTROLLER FOR NONLINEAR SYSTEM	48
5.6.	CONVERTING REAL-TIME MODEL FROM BLOCKS TO EQUATIONS	53
5.7.	WRITING VHDL CODE FOR DEADBEAT CONTROLLER	59
CHAPTER 6	SIMULATION AND REAL-TIME APPLICATION	60
6.1.	IMPROVING DEADBEAT CONTROLLER FOR LINEAR SYSTEMS	60
6.2.	MAGNETIC BALL LEVITATION CE152 (NONLINEAR SYSTEM)	66
6.3.	DEADBEAT CONTROLLER FOR MAGNETIC BALL LEVITATION WITH FEEDBACK LINEARIZATION	69
6.4.	DEADBEAT CONTROLLER FOR REAL-TIME MAGLEV CE152.....	72
6.5.	VHDL CODE OF DEADBEAT CONTROLLER FOR REAL-TIME MAGNETIC BALL LEVITATION CE152.....	74
6.6.	THE XILINX DSP BLOCK SET	74
6.7.	OBSTACLES AND SOLUTIONS	75
6.8.	DESIGNING STEPS:	76
6.9.	ADC.....	76
6.10.	APPROXIMATED MODEL OF DEADBEAT CONTROLLER	77
6.11.	SIMULATION OF VHDL CODE	77
CHAPTER 7	CONCLUSION.....	80
APPENDIX A	VHDL CODE FOR DEADBEAT CONTROLLER	85
APPENDIX B	MATLAB M-FILE CODE.....	87
APPENDIX C	GRAPHICAL USER INTERFACE (GUI)	91
APPENDIX D	USED SOFTWARE AND HARD WARE.....	93

LIST OF FIGURES

Figure (1.1):	Closed loop control system.....	1
Figure (2.1):	Multi-rate control system.....	5
Figure (2.2):	Multi-rate control for deadbeat servomechanism	6
Figure (2.3):	Ripple-free deadbeat design based on internal model	7
Figure (2.4):	Integral control.....	7
Figure (2.5):	Multi-rate ripple-free deadbeat controller for nonlinear systems	8
Figure (2.6):	Block diagram representation of the state space equations	9
Figure (2.7):	Implementation of the RFDC for linear systems.	10
Figure (2.8):	RFDBC for linear systems with constraint on control magnitude.....	10
Figure (2.9):	Block diagram representation of system with feedback linearization	11
Figure (2.10):	Plant with state and output feedback with integral control.....	11
Figure (2.11):	Feedback linearization for nonlinear system with sampling time = T_1	12
Figure (2.12):	Second order approximation of maglev with feedback linearization	12
Figure (2.13):	Exact second order approximation of maglev CE152 with FBL.....	13
Figure (2.14):	Deadbeat controller for approximated model.....	13
Figure (2.15):	Deadbeat controller for nonlinear system.....	14
Figure (2.16):	VHDL code vs. SIMULINK block diagram.....	14
Figure (3.1):	CE152 magnetic ball levitation.....	17
Figure (3.2):	Principal scheme of the magnetic levitation model.	17
Figure (3.3):	D/A Converter.....	18
Figure (3.4):	The power amplifier and its internal structure.	18
Figure (3.5):	Free diagram of the ball and the forces.....	19
Figure (3.6):	Position sensor subsystem.....	20
Figure (3.7):	D/A converter.....	21
Figure (3.8):	The complete model of magnetic levitation CE152	22
Figure (4.1):	Taylor series expansion.	24
Figure (4.2):	Plant of magnetic Ball levitation CE152	27
Figure (4.3):	Plant of magnetic Ball levitation CE152 with ADC, and DAC.....	27
Figure (4.4):	Position of ball at equilibrium.....	29
Figure (4.5):	State space representation of a plant.....	32
Figure (4.6):	Plant with state feedback	33
Figure (4.7):	Plant with state feedback and integral control	33
Figure (4.8):	CE152 with feedback linearization and integral controller	34
Figure (4.9):	CE152 with digital feedback linearization and integral controller	35
Figure (4.10):	Feedback linearization of real-time magnetic ball levitation.	35
Figure (4.11):	Lowest and highest ball's positions.....	36
Figure (4.12):	Illustrative drawing of the need to add the feedback gain '0.2'	36
Figure (4.13):	First simplification of FBL of real-time maglev CE152.	37
Figure (4.14):	Second simplification of FBL of real-time maglev CE152.....	37
Figure (4.15):	Third simplification of FBL of real-time maglev CE152.....	38
Figure (4.16):	Simplification of FBL of real-time maglev CE152.....	38
Figure (4.17):	Feedback transfer function	39
Figure (4.18):	Integrator.....	39
Figure (4.19):	Saturation.....	39
Figure (5.1):	Block diagram of deadbeat controller.....	40
Figure (5.2):	CE152 with third FBL and its approximated model.....	51
Figure (5.3):	SIMULINK deadbeat controller for real-time maglev CE152	53

Figure (5.4):	Feedback linearization for real-time maglev CE152	54
Figure (5.5):	Integrator	55
Figure (5.6):	Feedback transfer function.....	55
Figure (5.7):	Feedback linearization of maglev CE152	56
Figure (5.8):	N_1 of deadbeat controller	56
Figure (5.9):	N_2 of deadbeat controller	57
Figure (5.10):	$\frac{1}{D_c}$ of deadbeat controller	58
Figure (5.11):	Deadbeat of maglev CE152	58
Figure (5.12):	Deadbeat of maglev CE152	59
Figure (6.1):	Deadbeat closed loop system.....	60
Figure (6.2):	Time response of minimum order solutions	62
Figure (6.3):	Time response of vector, which was produced by Paz.	63
Figure (6.4):	Time response of vector, which was produced by first approach.....	64
Figure (6.5):	Time response of vector, which was produced by second approach.	65
Figure (6.6):	Model of maglev CE152 with FBL.	66
Figure (6.7):	Step response of CE152 with FBL.....	66
Figure (6.8):	Zoom on to find overshoot and rising time.....	67
Figure (6.9):	CE152 with FBL and its second order approximation.....	67
Figure (6.10):	Step responses of CE152 with FBL and approximated model.	68
Figure (6.11):	CE152 with third FBL and its second order approximation.....	68
Figure (6.12):	Step responses of CE152 with third FBL and approximated model.	69
Figure (6.13):	CE152 with FBL with/without deadbeat controller.....	70
Figure (6.14):	Sinusoidal response of CE152 with FBL and with deadbeat controller.....	71
Figure (6.15):	Sinusoidal response of CE152 with FBL.....	71
Figure (6.16):	Sinusoidal response of CE152 with FBL and with/without deadbeat controller..	71
Figure (6.17):	FBL of real-time maglev CE152.	72
Figure (6.18):	Response of real-time CE152 with FBL.....	73
Figure (6.19):	Block diagram of deadbeat controller and FBL	73
Figure (6.20):	Step response of deadbeat controller with FBL for CE152.....	73
Figure (6.21):	Sinusoidal response of deadbeat controller with FBL for RT CE152	74
Figure (6.22):	Block diagram of analog to digital converter	76
Figure (6.23):	Block diagram of original deadbeat controller	77
Figure (6.24):	Block diagram of approximated model deadbeat controller.....	77
Figure (6.25):	Block diagram of subsystem Approx. DB , DB, Black box for VHDL	78
Figure (6.26):	Time response of VHDL code versus original controller.....	78
Figure (6.27):	Time response original controller versus its approximated model.....	79
Figure (6.28):	Time response of VHDL code versus approximated model.....	79

LIST OF TABLES

TABLE (4.1): PARAMETERS OF MAGNETIC BALL LEVITATION CE 152	28
TABLE (6.1): COMPARISON BETWEEN PAZ VECTOR, FIRST APPROACH, SECOND APPROACH	65

ABBREVIATIONS

AC	Alternating Current
ADC	Analog to Digital Converter
DAC	Digital to Analog Converter
DBC	Deadbeat Controller
FBL	Feedback Linearization
HDL	Hardware Description Language
Maglev	Magnetic Levitation
Maglev CE152	Magnetic Ball Levitation CE152
RFDBC	Ripple-Free Deadbeat Controller
RT	Real-Time
VHDL	Very high speed integrated Hardware Description Language

CHAPTER 1 INTRODUCTION

1.1. General Introduction

Control systems have played an increasingly important role in the development and advancement of modern civilization and technology [1]. Practically some types of control systems affect every aspect of our day-to-day activities. Control systems are found in abundance in all sectors of industry, such as quality control of manufactured products, automatic assembly line, machine-tool control, computer control and many others [2]. Today, almost all controllers are computer implemented meaning digital control [3]. Deadbeat controller is a type of digital controllers, which offers the fastest settling time. Therefore, deadbeat controller ensures that the error sequence vanishes at the sampling instants after a finite time. Plants and processes are typically nonlinear; the most typical nonlinearity is saturation. Since, computer implemented controllers are a standard configuration, a theory for discrete-time nonlinear systems is very important in particular for control design purposes. Indeed, we cannot use linear control theory in cases where: large dynamic range of process variables is possible, multiple operating points are required, the process is operating close to its limits, small actuators cause saturation, etc [4]. A control system is a device or set of devices –called the controller– that manage the behavior of other devices – called the plant–.[5].

1.2. Closed-loop control system:

In a closed-loop control system, a sensor monitors the output and feeds the data to a computer, which continuously adjusts the control output (system input) as necessary to minimize the error as shown in Fig. (1.1). (That is, to maintain the desired speed, desired position, and so on). Feedback allows the controller to dynamically compensate the disturbances. An ideal feedback control system cancels out all errors, effectively mitigating the effects of any forces that might or might not arise during operation and producing a response in the system that perfectly matches the user's wishes.

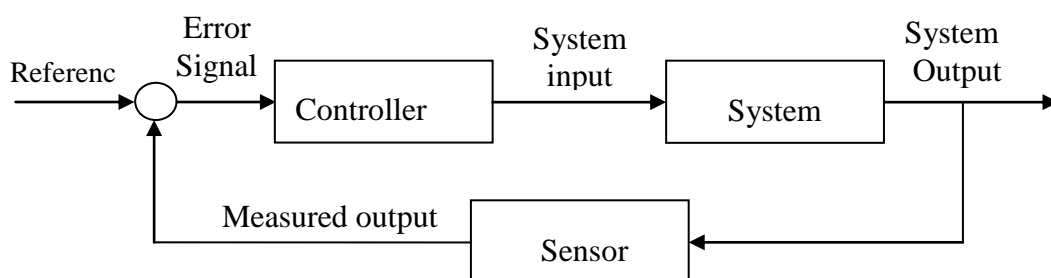


Figure (1.1): Closed loop control system

In reality, this cannot be achieved due to measurement of errors in the sensors, delays in the controller, and imperfections in the control input. The concept of the feedback loop: the sensed value is subtracted from the desired value to create the error signal, which is handled by the controller to produce the control signal, which can achieve the desired response [6].

1.3. Digital Controller:

Digital control is a branch of control theory that uses digital computers to act as system controllers. Depending on the requirements, a digital control system can be a microcontroller, DSP kit, FPGA kit, standard desktop computer and so on. Since a digital computer is a discrete system, the Laplace transform is replaced with the Z-transform. In addition, since a digital computer has finite precision, extra care is needed to ensure that the error in coefficients, A/D conversion, D/A conversion, etc. are not producing undesired or unplanned effects. For any digital controller, the output is a weighted sum of current and past input samples, as well as past output samples, this can be implemented by storing relevant values in any digital controller [3].

1.3.1. Features of Digital Controllers

- **Inexpensive**
- **Flexible:** easy to configure and reconfigure through software
- **Scalable:** programs can be scaled to the limits of the storage space without extra cost
- **Adaptable:** parameters of the program can be changed
- **Static operation:** digital computers are much less prone to environmental conditions than capacitors, inductors, etc[7].

1.3.2. Digital controller requires:

- **A/D converter:** converts analog inputs to machine readable format (digital)
- **D/A converter:** converts digital output to a form that can be input to a plant
- **Software program:** that relates the outputs to the inputs [3]

1.4. Previous Studies:

Multi-rate ripple-free deadbeat controller for nonlinear system can't see the light without the following researches:

- H. Elaydi and R. A. PAZ, (1998), proposed optimal ripple-free deadbeat controllers for systems with time delays. Matrix parameterization of the Diophantine equation approach was used to solve this problem. Based on this parameterization, LMI conditions were provided for optimal or constrained controllers with design quantities such as overshoot, undershoot, control amplitude, "slew rate" as well as for norm bounds such as ℓ_1, ℓ_2 and ℓ_∞ [8]. However, they treated linear systems only
- Dragan Nešić (1996), proposed ripple-free deadbeat control for polynomial systems of nonlinear input-output polynomial model. The proposed method dealt with multi-input multi-output systems. Mathematical tools, such as algebraic geometry, real algebraic geometry, symbolic computation and convex analysis were exploited. A number of analytic results were obtained utilizing computationally feasible controllability tests and design methodologies [4]. However, he didn't treat multi-rate problems and didn't use general rule such as Diophantine equations.
- L. Jetto and S. Longhi, (1999), proposed parameterized solution of the deadbeat ripple-free control problem for multi-rate sampled data systems. This paper was provided a parameterization of all causal feedback periodic controllers which guaranteed the deadbeat ripple-free behavior of the output of a linear time-invariant plant with a general multi-rate control scheme [9]. However, they treated multi-rate problems for linear systems only.
- H. Ito, (2001), improved performance of deadbeat servomechanism by means of multi-rate input control. A state-space approach to deadbeat servomechanism design was proposed using multi-rate input control. Multi-rate input mechanism yielded shorter settling time than single-rate control using the same frequency of sampling. However, multi-rate control often exhibited inter-sample ripple. Furthermore, the paper proposed a design method for multi-rate ripple-free deadbeat control which guaranteed robustness against continuous-time model uncertainty and disturbance [10]. However, the paper still didn't deal with nonlinear systems.

- R. A. PAZ, (2006), proposed a ripple free tracking with robustness. A hybrid two-degree-of freedom (2DOF) controller for the fixed-order constrained optimization problem addressing performance and robustness specifications was shown. The controller was given in terms of the solution of two Diophantine equations [11]. However, the nonlinear plants were not dealt with.
- M. E. Salgado and D. A. Oyarzun, (2007), presented two objective optimal multivariable ripple free deadbeat controls. A simple parameterization of all stabilizing ripple-free deadbeat controller of a given order was given. The free parameter was then optimized in the sense that a quadratic index was kept minimal [12]. However, the nonlinear and multi-rate were not approached.
- Al Batsh (2009), proposed multi-rate ripple-free deadbeat control. Two degree of freedom controller for the fixed-order constrained optimization problem addressing performance specifications utilizing the parameters of Diophantine equation to build a multi-rate ripple-free deadbeat control was presented. A combination between the concept of multi-rate and robust single rate was utilized. A time delay was also presented in simulation and was solved by using deadbeat controller based on solving Diophantine equation parameters [13]. However, he didn't treat nonlinear on multi-rate systems.

1.5. Thesis Contribution:

This thesis presents methodologies for designing internally stabilizing ripple-free deadbeat controllers to solve the tracking of an arbitrary reference signal and the attenuation of general disturbances for nonlinear systems. The deadbeat tracking formulation is based on Paz's results [11] and H. Ito's result [10], thus; the proposed controller combines two deadbeat controllers for linear systems to deadbeat nonlinear systems; thus, the new controller can be divided into two sub controllers: one of them will use polynomial approach and the other will use time domain approach.

1.6. Outline of the Thesis

This thesis is organized as follow: The second chapter introduces the forthcoming chapters and summarizes them, the third chapter presents the magnetic ball levitation CE152, the fourth chapter presents the feedback linearization of magnetic ball levitation CE152, the fifth chapter shows the methodology and approach, the sixth chapter shows simulations and results, and the final chapter concludes this thesis.

CHAPTER 2 BACKGROUND

This chapter gives background for the forthcoming chapters. Previous studies which mentioned in page 3 treated single-rate ripple-free deadbeat controller for linear systems, single rate ripple-free deadbeat controller for nonlinear systems without using general rule by dealing with polynomial systems of each nonlinear plant, and multi-rate ripple-free deadbeat controller for linear systems. Therefore, the previous studies did not deal with multi rate ripple-free deadbeat controller for nonlinear system; thus, I will start my work with Paz's result and apply Diophantine equations to the linearized plant using feedback linearization. Therefore, this chapter will cover briefly the multi-rate digital control, deadbeat controller for linear systems, nonlinear systems, feedback linearization, designing steps to deadbeat magnetic ball levitation CE152, and the necessary assumptions.

2.1. Multi-rate digital control

Multi-rate digital control systems are those, which use more than one sampling rate as shown in Fig.(2.1). If the controller updating rate $f_c=1/T_c$ is the same as the state feedback sampling rate $1/T_2$, output feedback sampling rate $1/T_3$, and input sampling rate $1/T_1$, then it's a single-rate system.

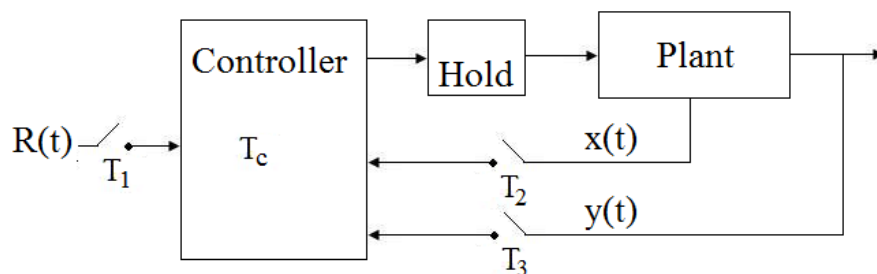


Figure (2.1): Multi-rate control system

2.2. Reasons for using multi-rate digital control

There are many practical reasons why multi-rate digital control systems might be used. For example:

- Required sampling rate is greater than the frequency of sampler; therefore; multi-rate technique can be used to map each input sample to K samples (i.e. increase the controller updating rate)

- Processing time is greater than controller updating time, therefore; we should decrease number of processes by increasing at least one of the following: state feedback sampling time; output feedback-sampling time, input sampling time, and/or decreasing controller-updating rate for some processes.

2.3. Multi-rate ripple-free deadbeat control for nonlinear systems

This approach here combines results developed on multi-rate ripple-free deadbeat [10], robust ripple-free control [11], and nonlinear control theory [19].

2.3.1. Multi-rate input controller for linear system (Hiroshi approach):

Deadbeat control can be achieved using state and output feedback with integral control. The two mappings ζ and ξ shown in Fig.(2.2) are linear operators maps each input sample to K samples; thus, the sampling rate will increase to $N \cdot K$ samples/sec instead of N samples/sec.

This structure will yield shorter settling time than single rate with sampling rate equal k samples/sec, but longer settling time than single rate with sampling rate equal $N \cdot K$ samples/ sec.

Ripples will appear between original samples due to mapped samples which do not match the original signal [11].

Therefore, this technique is good when you need sampling rate greater than the frequency of sampler since multi-rate controllers can achieve the required settling time with less frequent sampling of measurement.

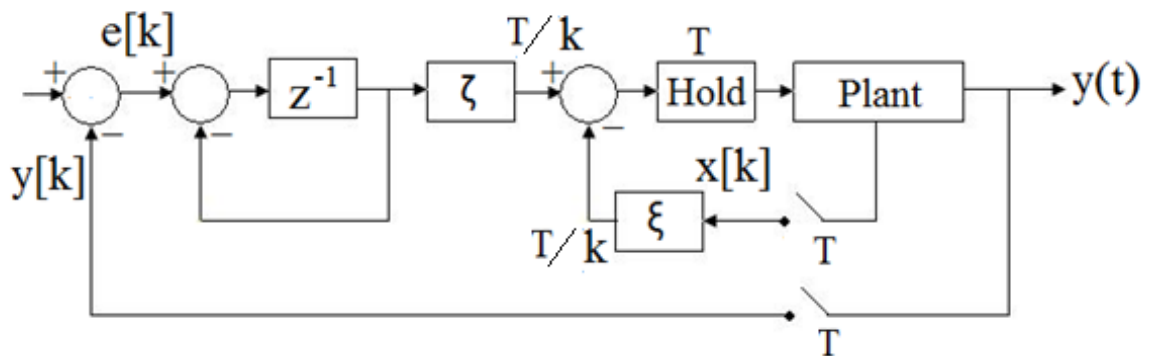


Figure (2.2): Multi-rate control for deadbeat servomechanism

2.3.2. Ripple-free deadbeat controller (R. PAZ approach)

A deadbeat controller design based on the internal model principle and ripple-free deadbeat design was presented by Paz; this technique gave a very good performance and robustness with very small settling time, but with a very high control signal.

Since, evaluating the three polynomials N_1 , N_2 , and D_c shown in Fig.(2.3) depend on the numerator and denominator of plant as well as denominator of reference signal, it can't be applied for nonlinear systems.

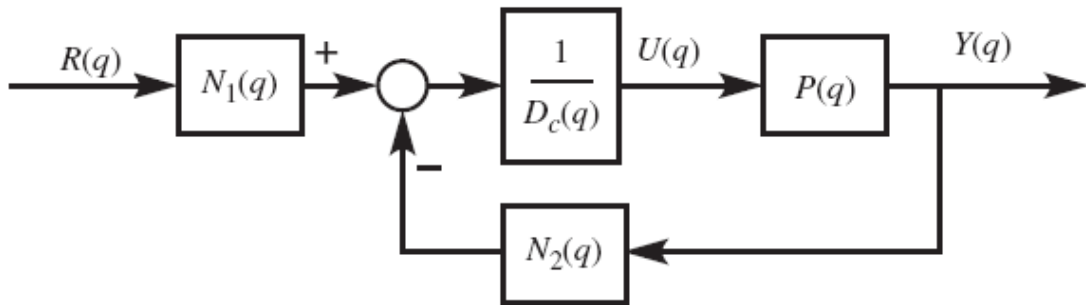


Figure (2.3): Ripple-free deadbeat design based on internal model

2.3.3. State and output feedbacks:

The basic philosophy of feedback linearization is to cancel the nonlinear terms of the system. Therefore, state and output feedbacks don't achieve deadbeat control the nonlinear system it can just linearize the nonlinear system and the state feedback control solves the local tracking problem [19].

Therefore, state and output feedbacks can be used to linearize and stabilize the nonlinear system and to make the response of the system closely equal the reference signal.

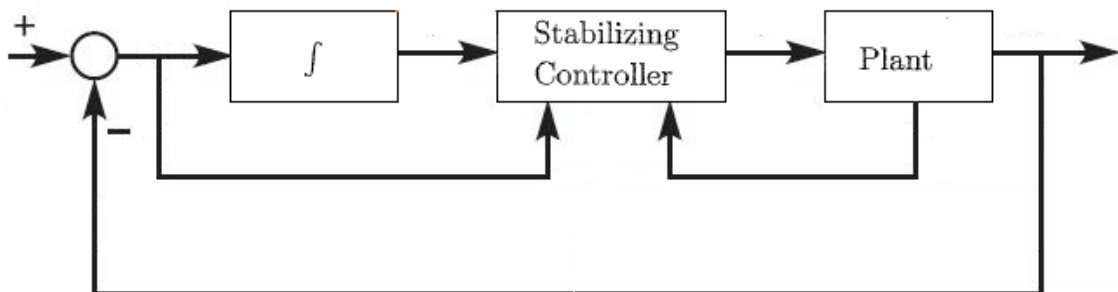


Figure (2.4): Integral control

2.3.4. Main objective

From all previous techniques, the ripple-free deadbeat controller for nonlinear system shown in Fig.(2.5) consist of the following:

- State and output feedbacks with integral controller will be used to linearize and stabilize nonlinear system with large sampling rate to make the response of nonlinear system closely equal the reference signal (Time-domain approach).
- Ripple-free deadbeat design based on the internal model principle will be utilized and applied to the linearized and stabilized nonlinear system with small sampling rate to make the response of the system exactly equal the reference signal and provide some robustness(Polynomial approach).

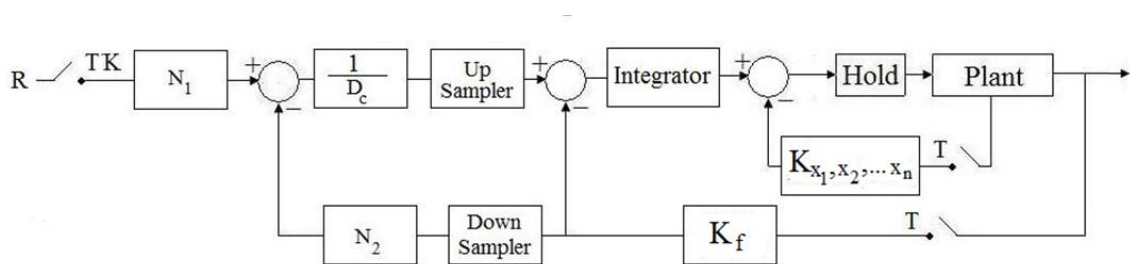


Figure (2.5): Multi-rate ripple-free deadbeat controller for nonlinear systems

Fig.(2.5) shows the multi-rate digital control for nonlinear system since the state feedback and output feedback have sampling rate $1/T$ samples/sec, while reference signal has sampling rate $1/(T*K)$ samples/sec and the controller updating rate of N_1, N_2, D_c is not the same as the controller updating rate of integrator and feedback gains.

2.4. Deadbeat controller for linear systems (Polynomial approach)

A linear system is a mathematical model of a system based on the use of a linear operator. Linear systems typically exhibit features and properties that are much simpler than nonlinear systems [14].

When we want to control the linear system, in general we use the Laplace transform (Z-Transform for digital systems) to represent the system, and when we want to examine the frequency characteristics of a system, we use the Fourier Transform. It is easy to evaluate the Laplace transform of first order differential equation, but it is difficult when the system has multiple first-order differential equations, so state-space approach has become popular [15].

In control engineering, a state space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations as shown in Fig. (2.6). The state space representation (also known as the "time-domain approach") provides a convenient and compact way to model and analyze systems with multiple inputs and outputs [16].

The most general state-space representation of a linear system is written in the following form:

$$\begin{bmatrix} \dot{x}(t) = A(t)x(t) + B(t)u(t) \\ y(t) = C(t)x(t) + D(t)u(t) \end{bmatrix} \text{-----(2.1)}$$

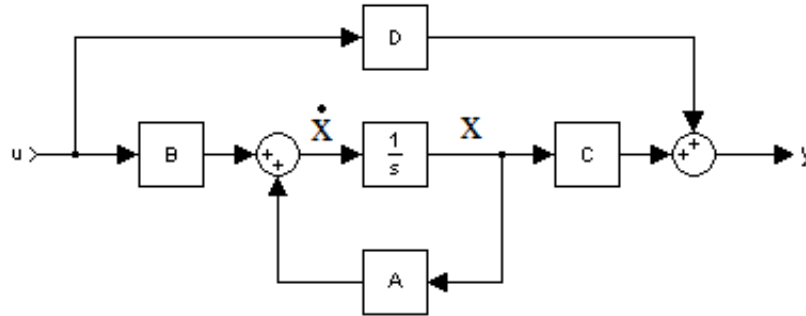


Figure (2.6): Block diagram representation of the state space equations

Were A(t), B(t), C(t) and D(t) are a function of time, In most models A,B,C and D are time invariant giving rise to the model

$$\begin{bmatrix} \dot{x}(t) = A x(t) + B u(t) \\ y(t) = C x(t) + D u(t) \end{bmatrix} \text{-----(2.2)}$$

This system can be converted to transfer function in the s-domain using Laplace transform. A discrete model can be obtained using the z-transform or modified z-transform for delayed systems. According to Paz [11] the deadbeat controller polynomials are obtained by solutions of two Diophantine equations:

$$N_p(q)N_1(q) + D_r(q)Q_1(q) = 1 \text{-----(2.3)}$$

$$N_p(q)N_2(q) + D_p(q)D_c(q) = 1 \text{-----(2.4)}$$

Since the Diophantine equation has an infinite number of solutions, we will seek specific solutions that provide desired transient behavior and robustness. In general, Diophantine equations have unique, minimum-order solutions, so there exist unique $N_{1min}(q)$, $Q_{1min}(q)$, $N_{2min}(q)$, and $D_{cmin}(q)$ where:

$$\text{Degree}(N_{1min}) < \text{Degree}(D_r) \text{-----(2.5)}$$

$$\text{Degree}(Q_{1min}) < \text{Degree}(N_p) \text{-----(2.6)}$$

$$\text{Degree}(N_{2min}) < \text{degree}(D_p) \text{-----(2.7)}$$

$$\text{Degree}(D_{cmin}) < \text{degree}(N_p) \text{-----(2.8)}$$

These solutions will be applied to the plant as shown in Fig. (2.7),

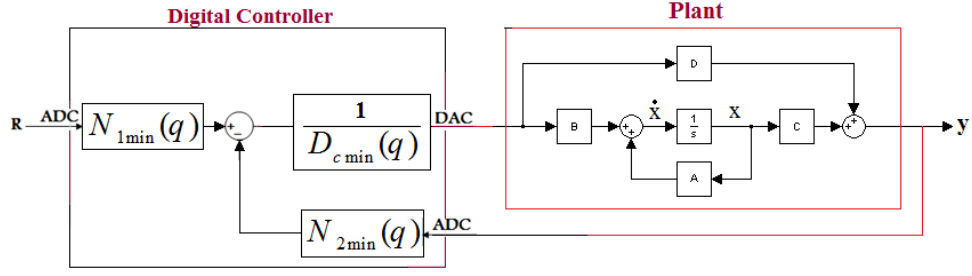


Figure (2.7): Implementation of the RFDC for linear systems.

When the control magnitude is constraint, Paz proposed a solution to minimize the magnitude of control signal according to the length of V_1 where V_1 is vector with unknown coefficients, Paz evaluates V_1 using the MATLAB built-in function “qp” which is a part of the optimization toolbox. Where:

$$N_1(q) = N_{1min}(q) - D_r(q)V_1(q) \text{-----(2.9)}$$

New N_1 will be applied to the system as shown in Fig. (2.8)

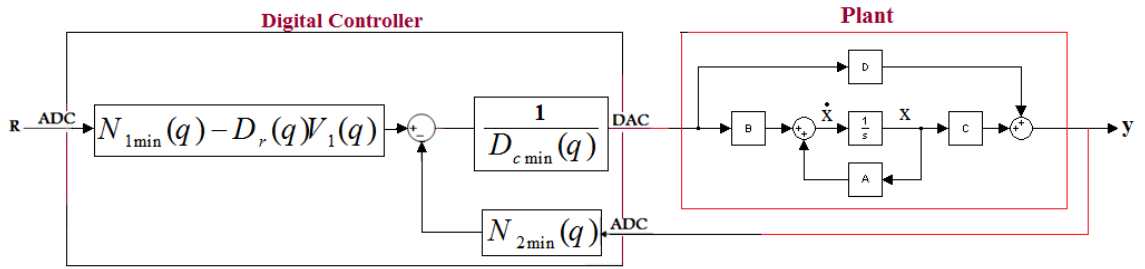


Figure (2.8): RFDC for linear systems with constraint on control magnitude.

Two steps will be used to evaluate V_1 , where the step response of system will be better, the new vector for the same settling time; will decrease the control magnitude, overshoot, and error signal.

2.5. Nonlinear systems

In mathematics, a nonlinear system is a system which does not satisfy the superposition principle, or whose output is not directly proportional to its input. Less technically, a nonlinear system is any problem where the variable(s) to be solved cannot be written as a linear combination of independent components such as squared terms in the following relation [17].

The motion equation of the ball of magnetic ball levitation CE152 is typically nonlinear that can be modeled such as:

$$m_k \ddot{x} = \frac{i^2 k_c}{(x-x_0)^2} - m_k g \text{-----(2.10)}$$

Where ‘i’ is the coil current, ‘k_c’ is coil constant, and ‘x₀’ is position offset.

Taylor series expansion will be used to linearize the nonlinear term in equation (2.10) by using the first two terms (constant and linear terms) of Taylor series then feedback linearization will be applied.

2.6. Feedback linearization

Feedback linearization shown in Fig.(2.9) is a popular approach to linearize nonlinear systems. Therefore, linear control techniques can be applied. Feedback linearization will be used because of the following reasons:

By using feedback linearization, the dynamic behavior of the system can be shaped (i.e. it is possible to assign the system eigenvalues to arbitrary values). System should be observable and controllable to get a feedback from states and to derive the state vector to final state.

Feedback linearization completely different from conventional linearization, because feedback linearization is achieved by exact state transformation and feedback, rather than by linear approximations of the dynamics.[18].

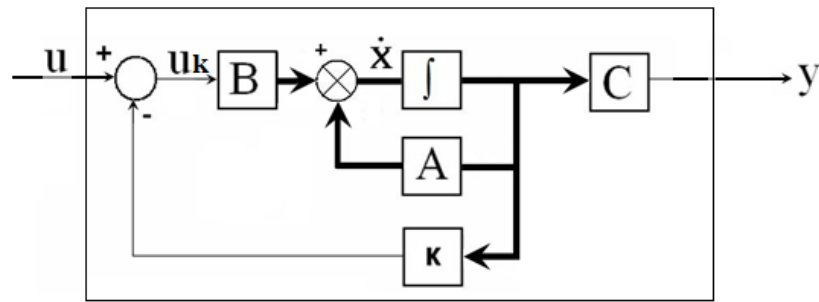


Figure (2.9): Block diagram representation of system with feedback linearization

A feedback path from the output will be added to form the error, e , which will be fed forward to the controlled plant via an integrator as shown in Fig. (2.10). The integrator increases the system type and reduces the error (the previous system with feedback linearization will be dealt as an open loop system).

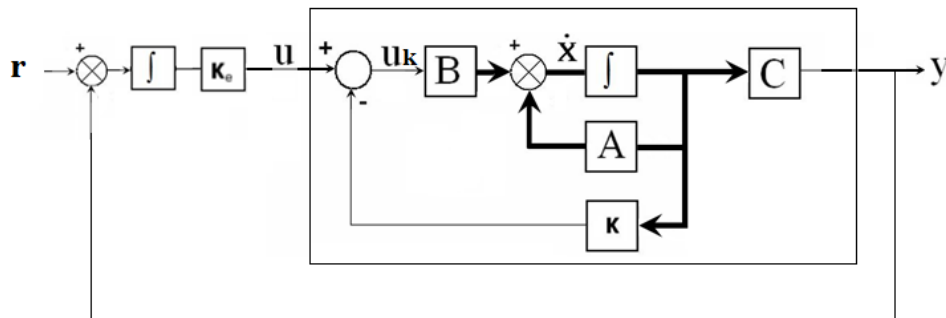


Figure (2.10): Plant with state and output feedback with integral control

Feedback linearization with integral controller [19] will be programmed using digital computer with sampling time = T_1 as shown in Fig. (2.11).

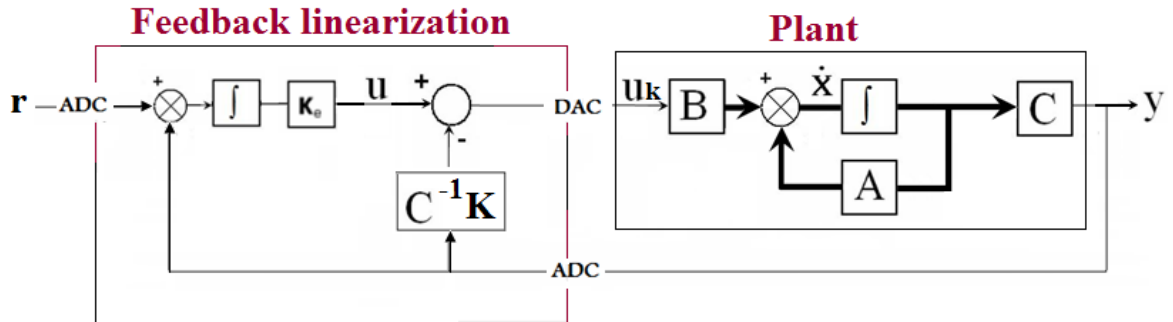


Figure (2.11): Feedback linearization for nonlinear system with sampling time = T_1

2.7. Second order approximation

Second order approximation for linearized model will be evaluated using two parameters –rising time ‘ t_r ’ (or settling time ‘ t_s ’) and overshoot- from step response in order to evaluate the deadbeat controller for linearized model with another sampling rate. The second order approximation will have the following form [6]:

$$\frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \text{-----(2.11)}$$

Where, ζ : Damping ratio and ω_n : natural frequency

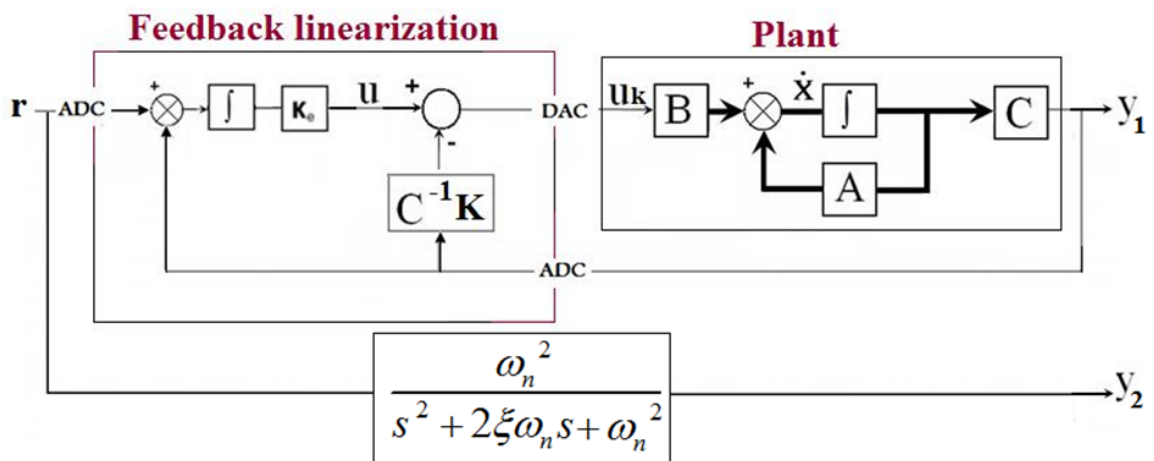


Figure (2.12): Second order approximation of maglev with feedback linearization

Tuning the parameters of second order system as shown in Fig. (2.12) until the two outputs Y_1 and Y_2 are the same. Then, ω_n and ξ will be ω_{n_new} and ξ_{new} respectively, and the output of approximated model will be equal to the output of linearized model as shown in Fig. (2.13).

$$\omega_{n_new} = \omega_n + \varepsilon_\omega \text{-----(2.15)}$$

$$\xi_{new} = \xi + \varepsilon_\xi \text{-----(2.16)}$$

Where, ε_ω and ε_ξ are small numbers.

The second order approximation will be as follow:

$$\frac{(\omega_n + \varepsilon_\omega)^2}{s^2 + 2(\xi + \varepsilon_\xi)(\omega_n + \varepsilon_\omega)s + (\omega_n + \varepsilon_\omega)^2} = \frac{\omega_{n_new}^2}{s^2 + 2\xi_{new}\omega_{n_new}s + \omega_{n_new}^2}$$

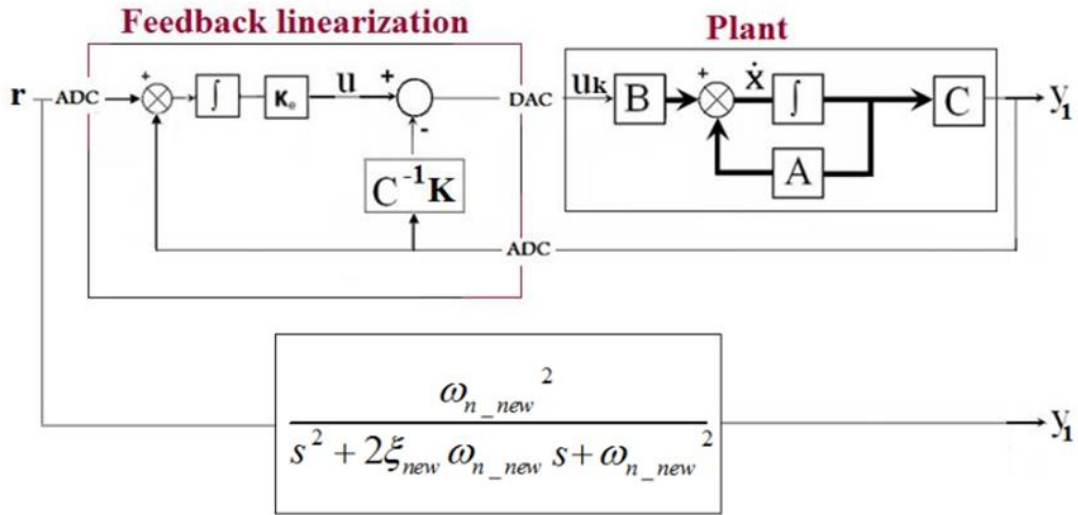


Figure (2.13): Exact second order approximation of maglev CE152 with FBL

Nonlinear plant has already linearized with feedback linearization using sampling time T_1 sec.; thus, the three blocks N_1 , N_2 , and D_c will be obtained with another sampling time T_2 sec. after evaluated the second order approximation to the linearized plant as shown in Fig.(2.14)

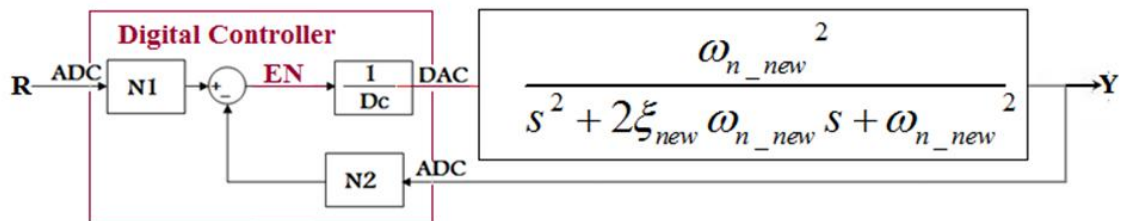


Figure (2.14): Deadbeat controller for approximated model

Deadbeat controller will be applied on real time magnetic ball levitation and N_1 , N_2 , and D_c will be tuned until reach the required response as shown in Fig.(2.15).

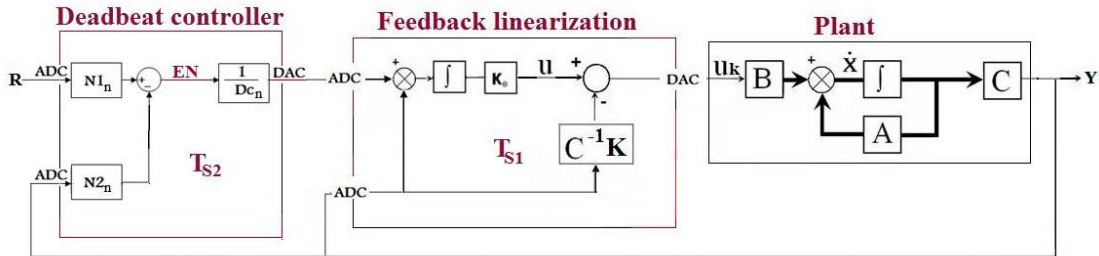


Figure (2.15): Deadbeat controller for nonlinear system

Finally, VHDL code for Deadbeat controller will be written and simulated using Xilinx DSP toolbox, in this toolbox you could define which FPGA kit will be used and many other properties.

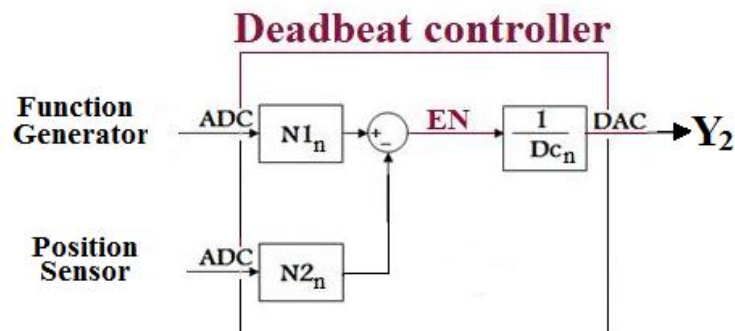
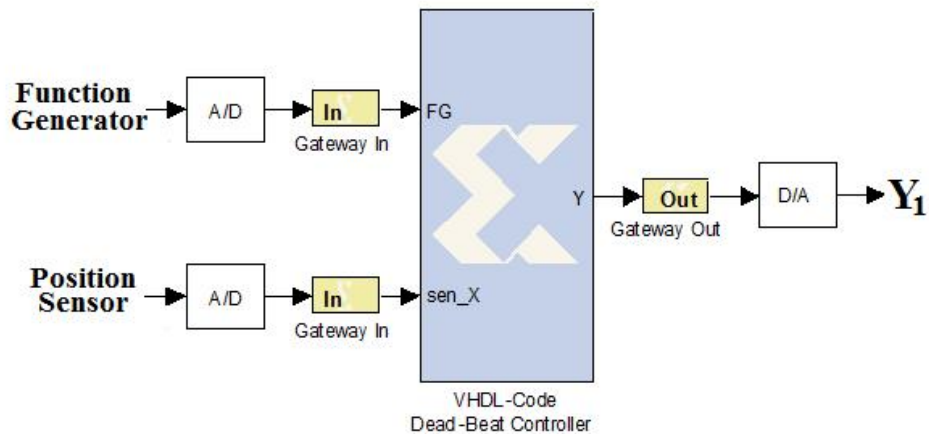


Figure (2.16): VHDL code vs. SIMULINK block diagram

2.8. Designing steps to deadbeat magnetic ball levitation

Magnetic ball levitation CE152 will be used as a case study for nonlinear system by the following steps

- Deriving the input/output relation of maglev sub-models (D/A converter, Power amplifier, ball & coil subsystem, Position sensor, A/D converter).
- Evaluating state space of magnetic ball levitation after linearization around midpoint.
- Applying Controllability and Observability tests to check if the following steps can be done.
- Evaluating feedback linearization with sampling time T_1 using MATLAB built-in function 'place', and applying it to magnetic ball levitation CE 152.
- Evaluating second order approximation to the step response of maglev with feedback linearization.
- Evaluating deadbeat controller for approximated model.
- Applying deadbeat controller on real time magnetic ball levitation CE152.
- Simulating VHDL code of deadbeat control for real time magnetic ball levitation using XILINX DSP toolbox and comparing it with the response of deadbeat block diagram.

2.9. Constraints

The following assumptions are necessary [11].

- **1:** *nonlinear system is controllable and observable*
Possibility of forcing the system into a particular state by using an appropriate control signal is required; thus, system should be controllable.
Possibility of reading all state variables is required in order to apply feedback linearization; thus, system should be observable.
- **2:** *denominator of reference signal and numerator of plant are coprime in discrete-time.*
Possibility of tracking reference signal requires no common factor between denominator of reference signal and numerator of plant to ensure that, there is no poles zeros cancellation; thus, denominator of reference signal and numerator of plant should be coprime.
- **3:** *there is no sinusoidal term in the reference signal with frequency that coincides with an integer multiple of the Nyquist frequency.*
Possibility of reconstruct the original continuous signal is required to compare between sensed and reference signals; thus, reference signal must not have frequency that coincides with an integer multiple of the Nyquist frequency.

CHAPTER 3 MAGNETIC BALL LEVITATION CE 152

Levitation (from Latin *levitas* "lightness") is the process by which an object is suspended by a physical force against gravity, in a stable position without solid physical contact. A number of different techniques have been developed to levitate matter, including the aerodynamic, magnetic, acoustic, electromagnetic, electrostatic, gas film, and optical levitation methods [20]. Magnetic levitation systems have many varied uses such as in frictionless bearings, high-speed maglev passenger trains, levitation of wind tunnel models, vibration isolation of sensitive machinery, levitation of molten metal in induction furnaces and levitation of metal slabs during manufacturing. These systems have nonlinear dynamics that are usually open loop unstable and, as a result, a high performance feedback controller is required to control the position of the levitated object. Due to inherent nonlinearities associated with electromechanical dynamics, the control problem is usually quite challenging to the control engineers, since a linear controller is valid only about a small region around a nominal operating point [21]. This chapter will talk about magnetic ball levitation CE152 as one of Magnetic levitation systems.

3.1. Introduction to magnetic ball levitation CE152

The Magnetic Levitation Apparatus shows control problems with nonlinear, unstable systems. The apparatus consists of a steel ball held in a magnetic field produced by a current-carrying coil. At equilibrium, the downward force on the ball due to gravity (its weight) is balanced by the upward magnetic force of attraction of the ball towards the coil. Any imbalance, the ball will move away from the set-point position. The basic control task is to control the vertical position of the freely levitating ball in the magnetic field of the coil. The Magnetic Levitation Apparatus is a nonlinear, dynamic system with one input (set point) and two outputs (ball position and coil current)[22].

The CE 152 Magnetic Levitation Model, shown in Fig(3.1) and its Diagram, shown in Fig.(3.2) is an unstable system designed for studying system dynamics and experimenting with number of different control algorithms[23].



Figure (3.1): CE152 magnetic ball levitation.

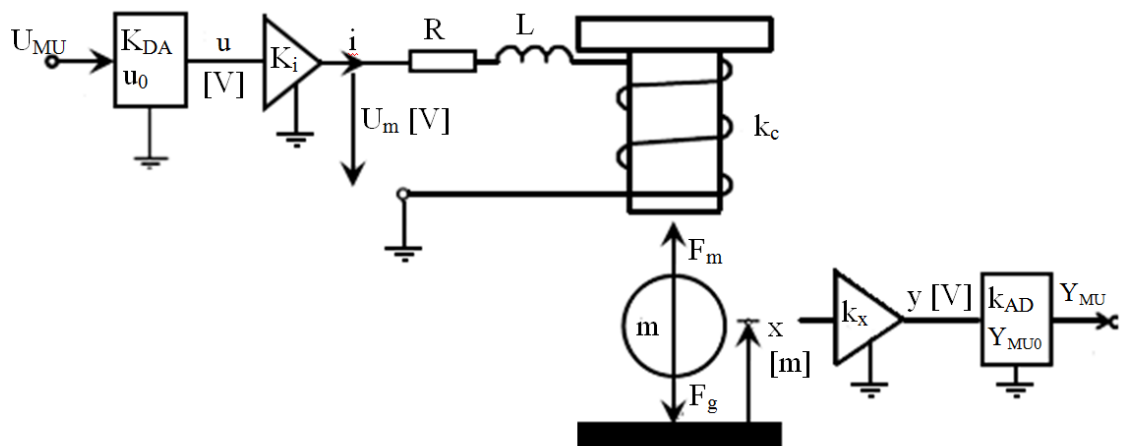


Figure (3.2): Principal scheme of the magnetic levitation model.

3.2. Model analysis

The CE152 model, shown in Fig. (3.2) consists of the following sub models [24]:

- D/A converter.
- Power amplifier.
- Ball & coil subsystem.
- Position sensor.
- A/D converter.

3.2.1. D/A converter

D/A Converter, shown in Fig.(3.3) has model output voltage 'u', The D/A converter input 'U_{MU}', The Digital to Analog converter gain 'K_{DA}', and The D/A converter offset U₀. The output is defined in eq.(3.1)

$$U = U_{MU} * K_{DA} + U_0 \text{-----(3.1)}$$

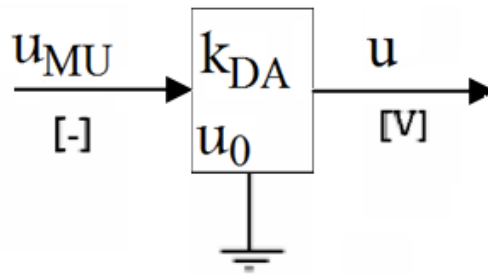
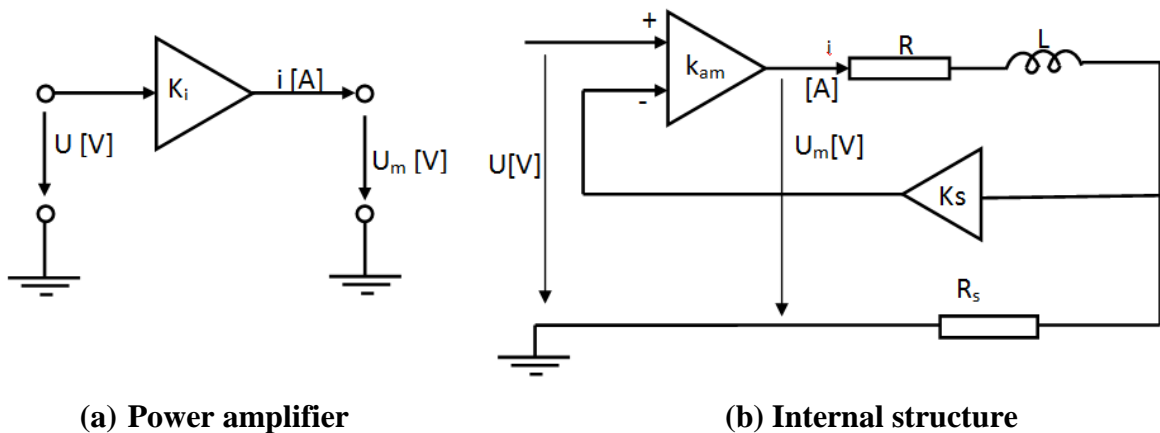


Figure (3.3): D/A Converter.

3.2.2. Power amplifier:

The power amplifier is designed as a source of constant current with the feedback current stabilization. As shown in Fig(3.4). Relation between input current to and output voltage from power amplifier will be found:



(a) Power amplifier

(b) Internal structure

Figure (3.4): The power amplifier and its internal structure.

From internal structure

$$u_m = iR + L \frac{di}{dt} + R_s i \text{-----(3.2)}$$

$$u_m = K_{am} (u - K_s (iR)) \text{-----(3.3)}$$

From (3.2) and (3.3)

$$\therefore iR + L \frac{di}{dt} + R_s i = K_{am} (u - K_s (iR_s)) \Rightarrow IR + LIS + R_s I = K_{am} U - K_{am} K_s R_s I$$

$$\Rightarrow \frac{I}{U} = \frac{K_{am}}{R} \left(\frac{1}{\frac{L}{R} S + 1 + \frac{R_s + K_{am} K_s R_s}{R}} \right)$$

if $R \gg R_s - K_{am} K_s R_s$ such as this system $\Rightarrow \frac{I}{U} = \frac{K_{am}}{R} \left(\frac{1}{\frac{L}{R} S + 1} \right)$ -----(3.4)

simplify the previous relation

$$\frac{I}{U} = K_i \frac{1}{T_a S + 1}$$
 -----(3.5)

where K_i is Gain , T_a is time constant

Equation (3.5) will be used in the next chapter to derive the state space of linearized model of magnetic ball levitation CE 152

3.2.3. Ball and coil subsystem:

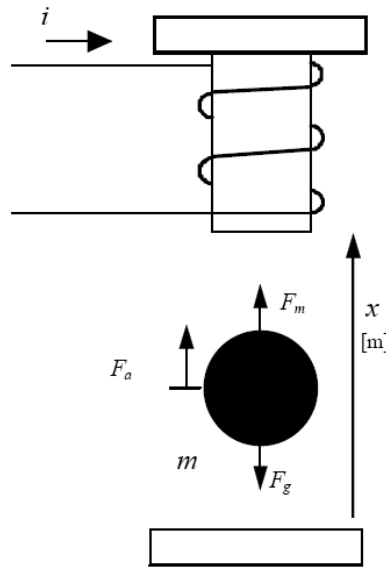


Figure (3.5): Free diagram of the ball and the forces.

The motion equation is based on the balance of all forces acting on the Ball. We have three forces: gravity force F_g , electromagnetic force F_m and the acceleration force F_a , as shown in Fig (3.5), equation of free body diagram will be derived where 'I' is the coil current, k_c is coil constant, x_0 is position offset, and K_{fv} is damping constant. According to Newton's second law of motion, the acceleration of an object as produced by a net force is directly proportional to the magnitude of the net force, in the same direction as the net force, and inversely proportional to the mass of the object [25].

The net force

$$F_a = F_m - F_g \text{ -----(3.6)}$$

Where;

Magnetic force

$$F_m = \frac{i^2 k_c}{(x-x_0)^2} \text{ -----(3.7)}$$

Gravitational force

$$F_g = m_k g \text{ -----(3.8)}$$

Acceleration force

$$F_a = m_k \ddot{x} \text{ -----(3.9)}$$

Substituting (3.7),(3.8), and (3.9) into (3.6)

$$m_k \ddot{x} = \frac{i^2 k_c}{(x-x_0)^2} - m_k g \text{ -----(3.10)}$$

Limits of the ball movements and ball damping is taken into account. So, to model the damping, the term k_{fv} is introduced into the equation

$$m_k \ddot{x} + k_{fv} \dot{x} = \frac{i^2 k_c}{(x-x_0)^2} - m_k g \text{ -----(3.11)}$$

Equation (3.11) will be used in the next chapter to derive the state space of linearized model of magnetic ball levitation CE 152

3.2.4. Position sensor.

The position sensor, shown in Fig.(3.6) which used to measure the ball position has model output voltage 'Y', The Ball position 'x', The position sensor gain 'K_x', and The position sensor offset Y₀. The output is defined in eq.(3.12)

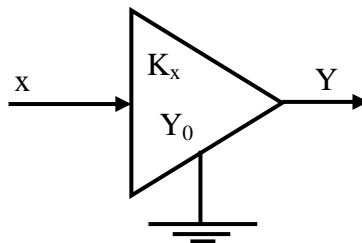


Figure (3.6): Position sensor subsystem.

$$Y = k_x x + Y_0 \text{ -----(3.12)}$$

3.2.5. A/D converter.

The A/D Converter, shown in Fig.(3.7) has model output voltage 'Y_{MU}', The A/D converter input 'Y', The analog to digital converter gain 'K_{AD}', and The A/D converter offset 'Y_{MU0}'.

The output is defined in eq.(3.13)

$$Y_{MU} = K_{AD} Y + Y_{MU0} \text{-----}(3.13)$$

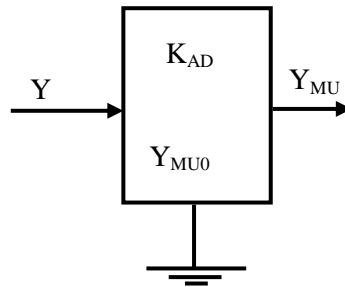


Figure (3.7): D/A converter

3.2.6. Magnetic constant "k_c"

$$\text{Coil energy } W_m = \frac{1}{2} Li^2 \text{-----}(3.14)$$

$$\text{Inductance } L = \frac{N^2}{R} \text{-----}(3.15)$$

$$\text{Resistance } R = \frac{l}{\mu A} \text{-----}(3.16)$$

substituting (3.16) into (3.15)

$$\therefore L = \frac{N^2}{R} = \frac{\mu AN^2}{l} \text{-----}(3.17)$$

substituting (3.17) into (3.14)

$$W_m = \frac{1}{2} \left(\frac{\mu AN^2}{l} \right) i^2 = \frac{\mu AN^2 i^2}{2l} \text{-----}(3.18)$$

$$\text{Force } F_m = \frac{dW_m}{dl} = -\frac{\mu AN^2 i^2}{2l^2} \text{-----}(3.19)$$

$$F_m = \frac{k_c i^2}{l^2} \text{-----}(3.20)$$

from (3.19) and (3.20)

$$\therefore k_c = \frac{\mu AN^2}{2}$$

3.3. Complete modeling:

The final block diagram of the magnetic levitation model CE 152 is given in SIMULINK model as shown in Fig.(3.8):

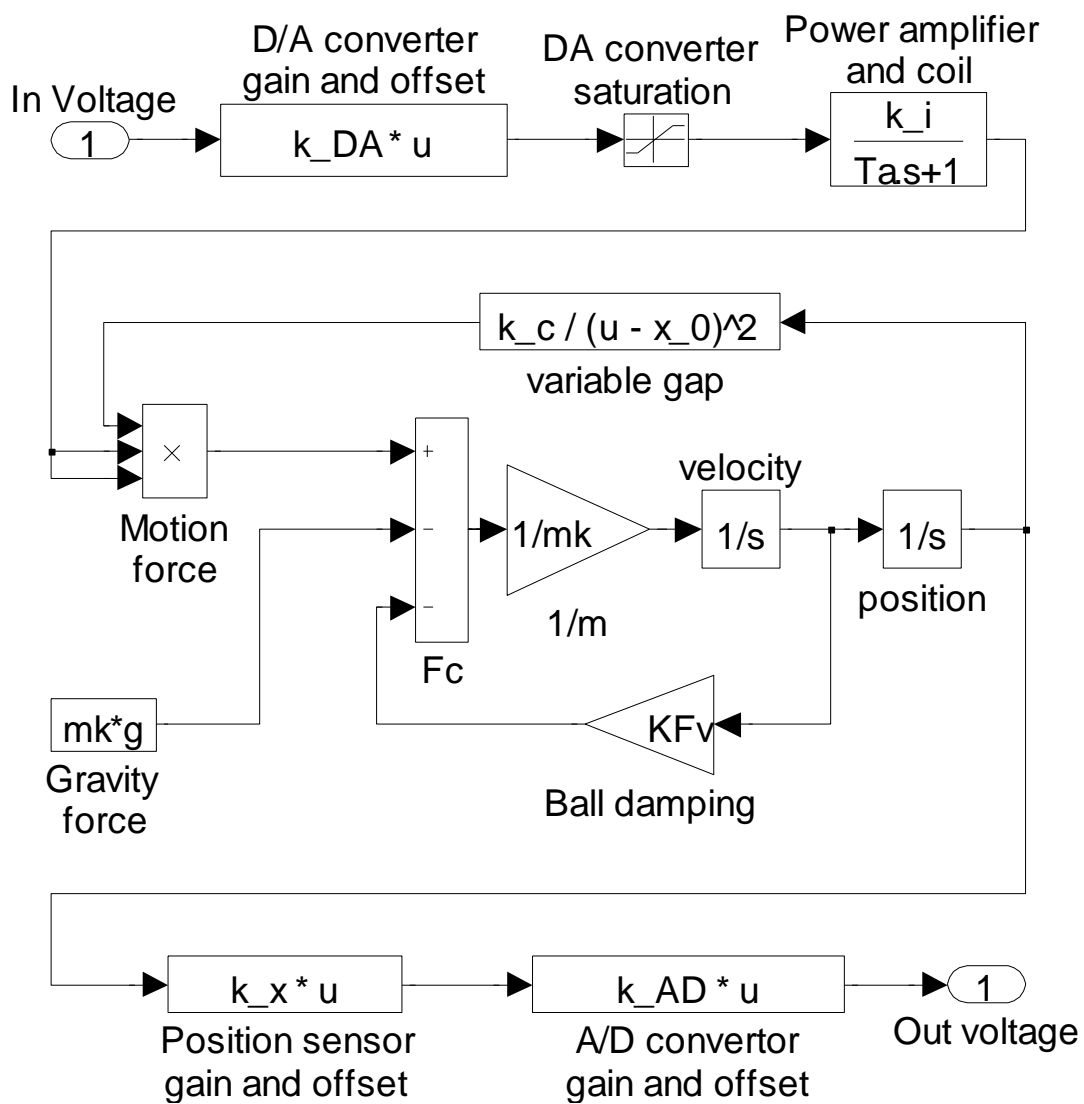


Figure (3.8): The complete model of magnetic levitation CE152

CHAPTER 4 FEEDBACK LINEARIZATION

Linearization refers to finding the linear approximation to a function at a given point. In the study of dynamical systems, linearization is a method for assessing the local stability of an equilibrium point of a system of nonlinear differential equations or discrete dynamical systems. This method is used in fields such as engineering.

Linearization makes it possible to use tools for studying linear systems to analyze the behavior of a nonlinear function near a given point [26].

4.1. Nonlinear systems

Nonlinear system representation means the characterization of nonlinear systems using nonlinear mathematical models. In fact, nonlinear models may be considered as a tool for explaining the nonlinear behavior patterns in terms of a set of easily understood elements [19].

In nature, most practical systems used for control are essentially nonlinear, and in many applications, particular in the area of chaos, it is the nonlinear rather than the linear characteristics that are most used. Signals found in the physical world are also far from conforming to linear models. Indeed, the complex structure of dynamic systems makes it almost impossible to use linear models to represent them accurately [27].

Nonlinear models are designed to provide a better mathematical way to characterize the inherent nonlinearity in real dynamic systems, although we may not be able to consider all their physical properties.

The linearization of a function is the first and second order terms of its Taylor series expansion around the point of interest.

4.2. Taylor series.

In mathematics, the Taylor series is a representation of a function as an infinite sum of terms calculated from the values of its derivatives at a single point. If the series is centered at zero, the series is also called a Maclaurin series (special case of Taylor's series). It is common practice to use a finite number of terms of the series to approximate a function.

The Taylor series of a real or complex function $f(x)$ that is infinitely differentiable in a neighborhood of a real or complex number a is the power series.

$$f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f^{(3)}(a)}{3!}(x-a)^3 + \dots$$

$$= \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n.$$

If the system has two variables, Taylor series can be written as :

$$f(x,y) = f(a,b) + \left(\frac{\partial f(a,b)}{\partial a} (x-a) + \frac{\partial f(a,b)}{\partial b} (y-b) \right) + \dots$$

$$= \sum_{n=0}^{\infty} \left(\frac{\partial^{(n)} f(a,b)}{\partial^{(n)} a} (x-a)^n + \frac{\partial^{(n)} f(a,b)}{\partial^{(n)} b} (y-b)^n \right)$$

example

If you don't have a calculator and you want to approximate $f(x) = \sqrt{4.001}$ let us use $a=4$

$$\therefore f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(4)}{n!} (4.001-4)^n =$$

$$f(4) + \frac{f'(4)}{1!}(4.001-4) + \frac{f''(4)}{2!}(4.001-4)^2 + \frac{f^{(3)}(4)}{3!}(4.001-4)^3 + \dots$$

$$= \sqrt{4} + \frac{1}{2 \cdot (4)^{\frac{1}{2}} \cdot 1!} (0.001) + \frac{-1}{4 \cdot (4)^{\frac{3}{2}} \cdot 2!} (0.001)^2 + \frac{3}{8 \cdot (4)^{\frac{5}{2}} \cdot 3!} (0.001)^3 + \dots$$

$$= 2 + \frac{1}{2 \cdot 2 \cdot 1} (1 \cdot 10^{-3}) + \frac{-1}{4 \cdot 8 \cdot 2} (1 \cdot 10^{-6}) + \frac{3}{8 \cdot 32 \cdot 6} (1 \cdot 10^{-9}) + \dots$$

$$= 2 + \frac{(1 \cdot 10^{-3})}{4} + \frac{-(1 \cdot 10^{-6})}{64} + \frac{(1 \cdot 10^{-9})}{512} + \dots$$

If I use just first two terms, the result will be $f(x) = 2 + 0.00025 = 2.0002500000$

and If I use a calculator, the result will be $f(x) = \sqrt{4.001} = 2.0002499844$

The two results are approximately the same, result of approximation will be better if I use extra terms.

In Figure (4.1) The exponential function (in blue), and the sum of the first $n+1$ terms of its Taylor series at 0 (in red)

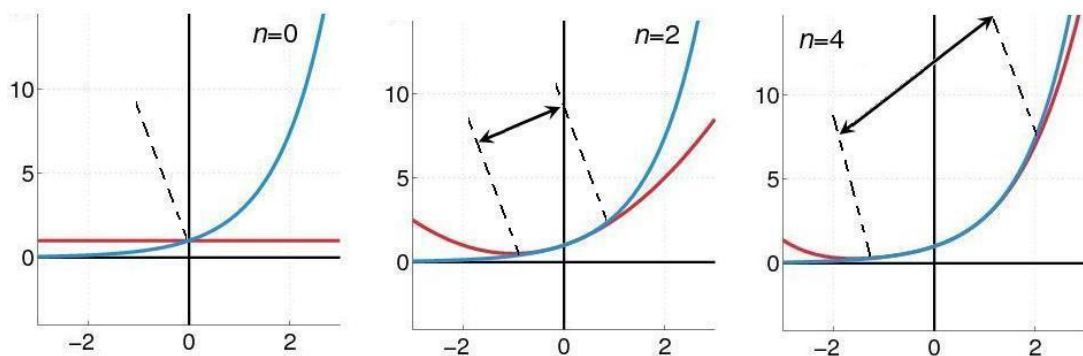


Figure (4.1): Taylor series expansion.

4.3. State space of magnetic ball levitation CE 152

The magnetic ball levitation CE152, shown in Fig.(3.1) and Fig.(3.2) is characterized by third order differential equation as shown in equation (3.11)

$$m_k \ddot{x} = \frac{i^2 k_c}{(x-x_0)^2} - m_k g - k_{fv} \dot{x}$$

In order to obtain a state variable model,

$$\text{let } x_1 = x \quad \text{-----(4.1)}$$

$$x_2 = \dot{x}_1 = \dot{x} \quad \text{-----(4.2)}$$

$$x_3 = i \quad \text{-----(4.3)}$$

Substituting (4.1),(4.2), and (4.3) into equation (3.11)

$$\Rightarrow \dot{x}_2 = \ddot{x} \Rightarrow \left(m_k \dot{x}_2 + k_{fv} x_2 = \frac{i^2 k_c}{(x_1 - x_0)^2} - m_k g \right) \text{-----(4.4)}$$

$$\begin{bmatrix} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{i^2 k_c}{m_k (x_1 - x_0)^2} - g - \frac{k_{fv} x_2}{m_k} \\ \dot{x}_3 = \dot{i} \end{bmatrix} \text{-----(4.5)}$$

From equation (3.5) $\Rightarrow (T_a s + 1)I = K_i U$

Where $i(t)$ can be defined such as:

$$T_a i \dot{(t)} + i(t) = K_i u(t) \Rightarrow i \dot{(t)} = \frac{K_i u(t) - i(t)}{T_a} \text{-----(4.6)}$$

from equation (4.4)

Using Taylor to linearize the term $\frac{i^2 k_c}{(x-x_0)^2}$ around $x=a$, and $i = b$.

$$f(x,i) = \frac{i^2 k_c}{(x-x_0)^2} = k_c \left[\frac{i}{(x-x_0)} \right]^2 \text{-----(4.7)}$$

Since, this term has two variables (position ‘vertical displacement’, and coil current), extended form of Taylor series with two variables will be used, thus; this series yields the first two terms such as:

$$f(x,i) \approx f(a,b) + \left(\frac{\partial f(a,b)}{\partial(a) * 1!} x(t) + \frac{\partial f(a,b)}{\partial(b) * 1!} i(t) \right) \text{-----(4.8)}$$

$$\therefore k_c \left[\frac{i}{(x-x_0)} \right]^2 = \frac{b^2 k_c}{(a-x_0)^2} + \left[\frac{-2 * k_c * b^2}{(a-x_0)^3} \right] x(t) + \left[\frac{2 * k_c * b}{(a-x_0)^2} \right] i(t) \text{-----(4.9)}$$

Substituting (4.9) into (4.4)

$$m_k \ddot{x} = \frac{b^2 k_c}{(a-x_0)^2} + \left[\frac{-2 * k_c * b^2}{(a-x_0)^3} \right] x(t) + \left[\frac{2 * k_c * b}{(a-x_0)^2} \right] i(t) - m_k g - k_{fv} \dot{x}$$

$$\therefore m_k \dot{x}_2 = \left[\frac{-2 * k_c * b^2}{(a-x_0)^3} \right] x_1 - k_{fv} x_2 + \left[\frac{2 * k_c * b}{(a-x_0)^2} \right] i(t) + \left(\frac{b^2 k_c}{(a-x_0)^2} - m_k g \right) \text{-----(4.10)}$$

When the ball is fixed at position (x_1) zero; then, the velocity (x_2), acceleration: the derivative of velocity (\dot{x}_2) and the coil current (i) are all equal zero.

Substituting x_1, x_2, \dot{x}_2 , and $i = 0$ into equation (3.6)

$$\therefore m_k(0) = \left[\frac{-2 * k_c * b^2}{(a-x_0)^3} \right] (0) - k_{fv} (0) + \left[\frac{2 * k_c * b}{(a-x_0)^2} \right] (0) + \left(\frac{b^2 k_c}{(a-x_0)^2} - m_k g \right)$$

$$\Rightarrow \left(\frac{b^2 k_c}{(a-x_0)^2} - m_k g \right) = 0 \text{-----(4.11)}$$

substituting (4.11) into (4.10)

$$\therefore m_k \dot{x}_2 = \left[\frac{-2 * k_c * b^2}{(a-x_0)^3} \right] x_1 - k_{fv} x_2 + \left[\frac{2 * k_c * b}{(a-x_0)^2} \right] i(t) \text{-----(4.12)}$$

from equation (4.12) and (4.6)

$$\dot{x}_2 = \left[\frac{-2 * k_c * b^2}{m_k (a-x_0)^3} \right] x_1 - \frac{k_{fv} x_2}{m_k} + \left[\frac{2 * k_c * b}{m_k (a-x_0)^2} \right] i(t)$$

$$\dot{x}_3 = \dot{i} = \frac{K_i u(t) - i(t)}{T_a}$$

$$\therefore \left[\begin{array}{l} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{-2 * k_c * b^2}{m_k (a-x_0)^3} x_1 - \frac{k_{fv}}{m_k} x_2 + \frac{2 * k_c * b}{m_k (a-x_0)^2} i(t) \\ \dot{x}_3 = \frac{di}{dt} i(t) \end{array} \right] \text{-----(4.13)}$$

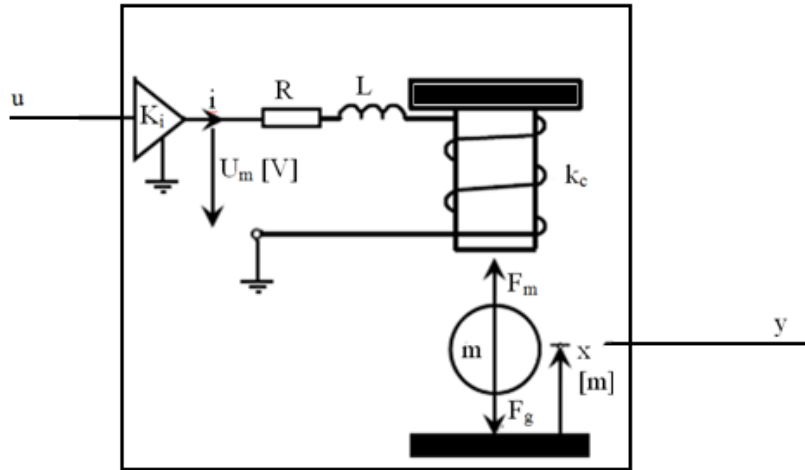


Figure (4.2): Plant of magnetic Ball levitation CE152

∴ The state space of linearized model shown in Fig.(4.2) around point (a,b) will be

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-2 * k_c * b^2}{m_k (a-x_0)^3} & \frac{-k_{fv}}{m_k} & \frac{2 * k_c * b}{m_k (a-x_0)^2} \\ 0 & 0 & -\frac{1}{T_a} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_i}{T_a} \end{bmatrix} u(t) \quad (4.14)$$

$$y = [1 \ 0 \ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.15)$$

Adding the gain of DAC, ADC, and sensor gain as shown in Fig(3.2), will result in the following state space model.

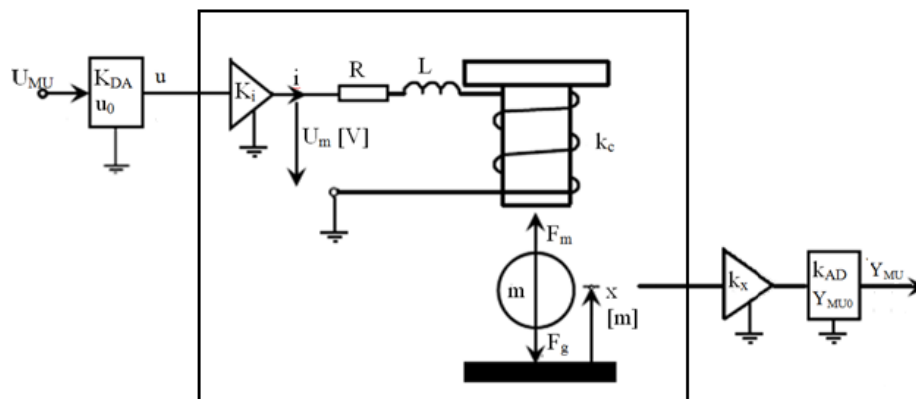


Figure (4.3): Plant of magnetic Ball levitation CE152 with ADC, and DAC

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{-2*k_c*b^2}{m_k(a-x_0)^3} & \frac{-k_{fv}}{m_k} & \frac{2*k_c*b}{m_k(a-x_0)^2} \\ 0 & 0 & -\frac{1}{T_a} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_i K_{DA}}{T_a} \end{bmatrix} U_{MU}(t) \quad (4.16)$$

$$Y_{MU} = [K_x K_{AD} \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.17)$$

Equation (4.16) and (4.17) will be used with Table (4.1) shown in the next page to find state space model of linearized magnetic ball levitation.

Table (4.1): Parameters of magnetic ball levitation CE 152 [22]

Parameter	Symbol	Value
ball diameter	Dk	12.7x10 ⁻³ m
ball mass	mk	0.0084 kg
distance from the ground and the edge of the magnetic coil	Td	0.019 m
distance of limits= 0.019 - Dk	L	0.0063 m
gravity acceleration constant	g	9.81 m.s ⁻²
maximum DA converter output voltage	U_DAm	5 V
coil resistance	Rc	3.5 Ω
coil inductance	Lc	30 x10 ⁻³ H
current sensor resistance	Rs	0.25 Ω
current sensor gain	Ks	13.33
power amplifier gain	K_am	100
maximum power amplifier output current	I_am	1.2 A
amplifier time constant= Lc/((Rc+Rs)+Rs*Ks*K_am)	Ta	1.8694 x10 ⁻⁵ s
amplifier gain= K_am / ((Rc+Rs)+Rs*Ks*K_am)	k_i	0.2967
viscose friction	KFv	0.02 N.s/m
converter gain	k_DA	10
Digital to Analog converter offset	u_0	0 V
Analog to Digital converter gain	k_AD	0.2
Analog to Digital converter offset	y_MU0	0 V
position sensor constant	k_x	797.4603
coil bias	x_0	8.26 x10 ⁻³ m
Aggregated coil constant	k_f	0.606 x10 ⁻⁶ N/V
coil constant =k_f/(k_i)^2	k_c	6.8823 x10 ⁻⁶ N/V

Fig. (4.4) shows all possible positions and the center position of magnetic ball levitation's ball

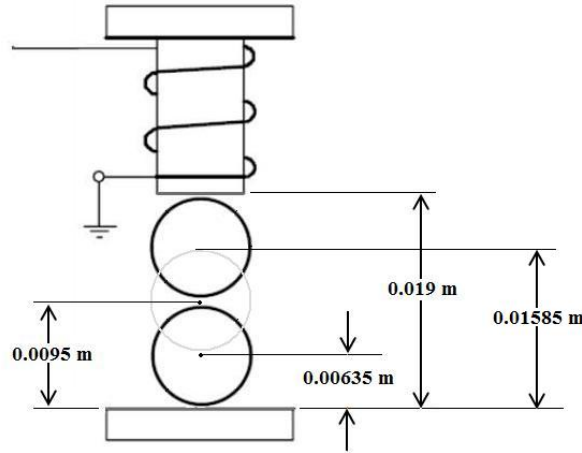


Figure (4.4): Position of ball at equilibrium

At equilibrium

- 1) Ball velocity, and acceleration of ball = 0 $\Rightarrow x_1 = \dot{x}_1 = x_2 = 0$
- 2) The derivative of current = 0
(let us use the center) $\Rightarrow x_1 = 0.0095m \therefore (a=0.0095)$

Substituting the position of the ball at center as shown in Fig.(4.3), and parameters as in Table.(4.1) in equation (4.11)

$$\therefore \left(\frac{b^2 k_c}{(a-x_0)^2} - m_k g \right) = 0 \Rightarrow \frac{b^2}{(a-x_0)^2} = \frac{m_k g}{k_c} \Rightarrow \frac{b}{(a-x_0)} = \pm \sqrt{\frac{m_k g}{k_c}}$$

$$\Rightarrow b = \pm (a-x_0) * \sqrt{\frac{m_k g}{k_c}} \text{-----(4.18)}$$

$$b = \pm (a-x_0) * \sqrt{\frac{m_k g}{k_c}} = \pm (0.0095-0.00826) * \sqrt{\frac{0.0084*9.81}{6.8823e-006}} =$$

$$(0.00759) (\sqrt{11973.32287}) = 0.13568 \text{ A} \therefore (b=0.13568)$$

To compute the input voltage at this point.
from equation (3.2)

$$\dot{x}_3 = \dot{i} = \frac{K_i u(t) - i(t)}{T_a} = 0 \Rightarrow K_i u(t) - i(t) = 0 \Rightarrow K_i u(t) = i(t)$$

$$\therefore u(t) = \frac{i(t)}{K_i} \text{-----(4.19)}$$

$$= \frac{0.13568}{0.2967} = 0.4573 \text{ V}$$

The resulting state space model of linearized magnetic ball levitation CE152 is

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -15821.62 & -2.381 & 144.596 \\ 0 & 0 & -53493.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 79357.013 \end{bmatrix} U_{MU}(t) \quad (4.20)$$

$$Y_{MU} = \begin{bmatrix} 159.49206 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (4.21)$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -15821.62 & -2.381 & 144.596 \\ 0 & 0 & -53493.1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ 79357.013 \end{bmatrix} \quad (4.22)$$

$$C = \begin{bmatrix} 159.49206 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \end{bmatrix} \quad (4.23)$$

4.4. Controllability and Observability tests:

Before, designing a feedback linearization controller, Controllability and Observability tests must be done.

Where

Controllability matrix

$$Q_c = \begin{bmatrix} B & AB & A^2B \end{bmatrix} \quad (4.24)$$

Observability matrix

$$O_v = \begin{bmatrix} C \\ CA \\ CA^2 \end{bmatrix} \quad (4.25)$$

Substituting (4.22) and (4.23) into (4.24) and (4.25)

$$\therefore Q_c = \begin{bmatrix} 0 & 0 & 1.1476e7 \\ 0 & 1.1476e7 & -6.139e11 \\ 7.9357e4 & -4.2454e9 & 2.270953e14 \end{bmatrix}$$

since Q_c has full rank, then system is controllable

$$O_v = \begin{bmatrix} 159.49206 & 0 & 0 \\ 0 & 159.49206 & 0 \\ -2523422.76 & -379.750594860 & 23061.913907760 \end{bmatrix}$$

since O_v has full rank, then system is observable

Thus, state feedback control is possible

4.5. Transfer function of linearized model

Transfer function $G(s)$ is defined such as: $G(s) = C(SI - A)^{-1} B$

$$= [159.49206 \quad 0 \quad 0] \left(\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -15821.62 & -2.381 & 144.596 \\ 0 & 0 & -53493.1 \end{bmatrix} \right)^{-1} \begin{bmatrix} 0 \\ 0 \\ 79357.013 \end{bmatrix}$$

$$G(s) = \frac{(1.83 * 10^9)}{s^3 + (5.349 * 10^4 s^2) + (1.432 * 10^5 s) + (8.464 * 10^8)} = \frac{N(s)}{D(s)} \text{-----(4.26)}$$

4.6. Specified transfer function.

The specification for the controller is to be able to position the ball at any arbitrary location in the magnetic field and to move the ball smoothly from one position to another. Thus, the specified transfer function can be obtained from the specifications such as: OS, of 5% and settling time, t_s , of 0.01 sec.

$$OS = 5\%; \text{-----(4.27)}$$

$$t_s = 0.01; \text{-----(4.28)}$$

To obtain a prototype, we need to translate the system specifications to ζ and ω_n such as;

∴

$$\zeta = \sqrt{\frac{\ln(OS)^2}{\ln(OS)^2 + \pi^2}} = \sqrt{\frac{\ln(0.05)^2}{\ln(0.05)^2 + \pi^2}} = 0.6901 \text{-----(4.29)}$$

$$\omega_n = \frac{\ln(0.5 \sqrt{1 - \zeta^2})}{\zeta t_s} = \frac{\ln(0.05 \sqrt{1 - 0.6901^2})}{0.6901 * 0.01} = 480.95 \text{-----(4.30)}$$

∴ The second order prototype is $\frac{\omega_n^2}{s^2 + 2*\zeta*\omega_n s + \omega_n^2}$ -----(4.31)

Substituting (4.31) and (4.32) into (4.33)

$$= \frac{(480.95)^2}{s^2 + (2*0.6901*480.95)s + (480.95)^2} = \frac{(2.313e5)}{s^2 + (663.8)s + (2.313e5)}$$

$$\frac{(2.313*10^5)}{s^2 + (663.8)s + (2.313*10^5)}$$
-----(4.32)

with characteristic equation $s^2 + (663.8)s + (2.313*10^5) = 0$ -----(4.33)

and pools $[(-331.91 + 348.07i), (-331.91 - 348.07i)]$

Since the linearized model of magnetic levitation CE152 is third order system 'equation(4.26)', we need to add a third pole to our prototype such as

Third pole = 100*real(first pole)=-33191

∴ Required Pools = $[(-331.91 + 348.07i), (-331.91 - 348.07i), (-33191)]$ ----- (4.34)

The specified Characteristic equation

$\alpha(s) = s^3 + (33854.82)s^2 + (22264166.593)s + (7677641650.843)$ ----- (4.35)

4.7. State feedback

To obtain a state feedback control law, we need to find a matrix K that equates the pools of magnetic ball levitation CE 152 with the pools of the prototype.

Given the system shown in Fig.(4.5),

$\dot{x} = Ax + Bu$ ----- (4.36)

$y = Cx$ ----- (4.37)

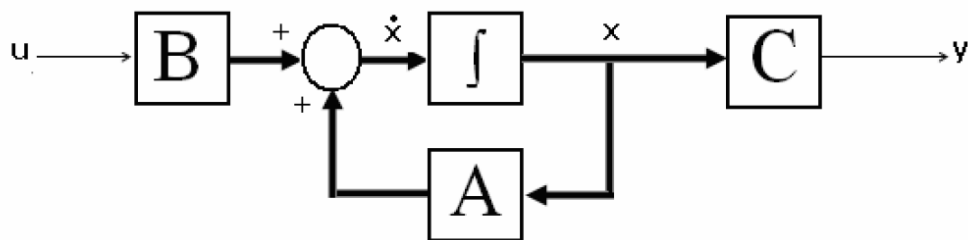


Figure (4.5): State space representation of a plant

We need to construct a state feedback control law as shown in Fig.(4.6)

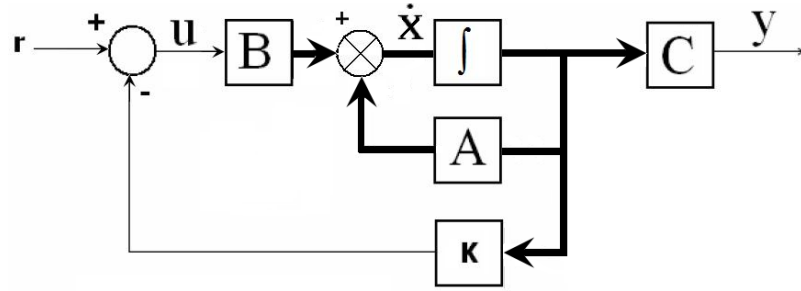


Figure (4.6): Plant with state feedback

In feedback control system, the output states are fed back by the feedback vector (K) to the summing junction. Then the state equations for the closed loop system can be written as:

$$\begin{aligned} \dot{x} &= Ax + Bu = Ax + B(r - Kx) = (A - BK)x + Br \text{-----(4.38)} \\ y &= Cx \end{aligned}$$

To obtain K equation, the specified characteristic equation $\alpha(s)$ and denominator of linearized maglev D(s) should be used, you could use the MATLAB built-in function 'place' to evaluate K.

Then,

$$K = [622.3322 \quad 1.9316 \quad -0.2475] = [k_1 \quad k_2 \quad k_3] \text{-----(4.39)}$$

$$k_3 = -0.2475, k_2 = 1.9316, k_1 = 622.3322 \text{-----(4.40)}$$

Adding the gain of position sensor k_x and the gain of ADC k_{AD} to the gain k_1

$$\therefore \text{new } k_1 = \frac{k_1}{k_x * k_{AD}} = \frac{622.3322}{797.4603 * 0.2} = 3.9019 \text{-----(4.41)}$$

4.8. State feedback and integral controller

When applying the feedback gain to the maglev CE152, the ball didn't settle which means the steady state error is very large; thus, a feedback path from the output has been added to form the error, e, which is fed forward to the controlled plant via an integrator as shown in Fig. (4.7). The integrator increases the system type and reduces the error [19].

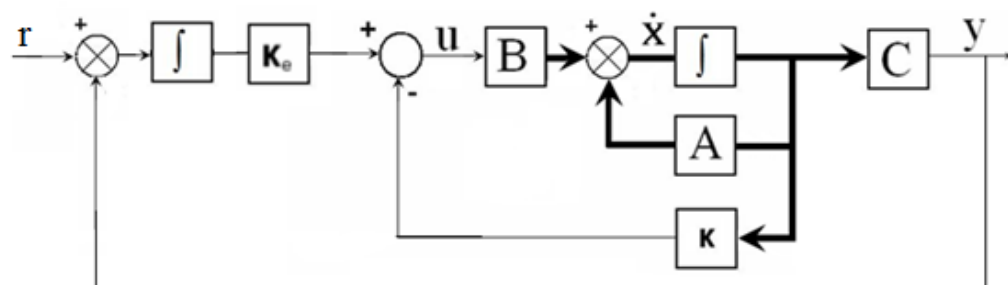


Figure (4.7): Plant with state feedback and integral control

K_4 will be obtained by trial and error, The resulting system is shown in SIMULINK formulation in Fig. (4.8).

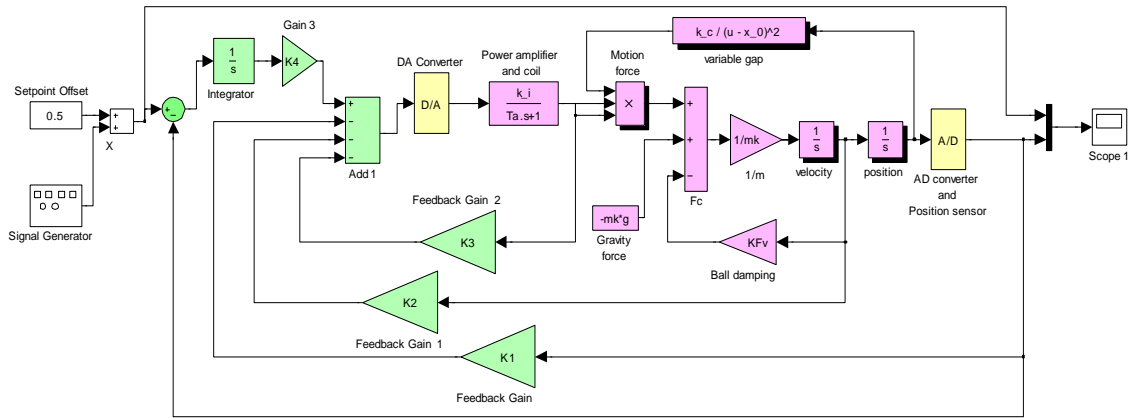


Figure (4.8): CE152 with feedback linearization and integral controller

4.9. Digital controller for feedback linearization

The selection of the sampling time plays a big role in digital control; however, this topic is beyond this research; thus, a sampling time $t_s=0.001$ sec was chosen. All feedback gains will be taken from the output, so, The gain of position sensor k_x and the gain of analog to digital converter k_{AD} should be applied.

In digital control, the velocity is obtained by taking the present value minus previous value of position divided by sampling time.

$$\text{so, multiplying gain } k_2 \text{ by } \frac{1}{k_x * k_{AD} * t_s} = \frac{1}{797.4603 * 0.2 * 0.001} = 6.2699 \text{-----(4.42)}$$

The MATLAB built-in function `c2d` is used to convert integrator from continuous to digital using first order hold. Since, when the integrator was converted from analog to digital using zero order hold, the system did not work well, after that I used first order hold since it's more accurate and the system worked well.

$$\text{c2d}\left(\text{tf}\left(1,[1 \ 0]\right),.001,'foh'\right) = \frac{0.0005 z + 0.0005}{z - 1} \text{-----(4.43)}$$

The resulting maglev system with digital feedback linearization and integral controller is shown in SIMULINK formulation in Fig.(4.9)

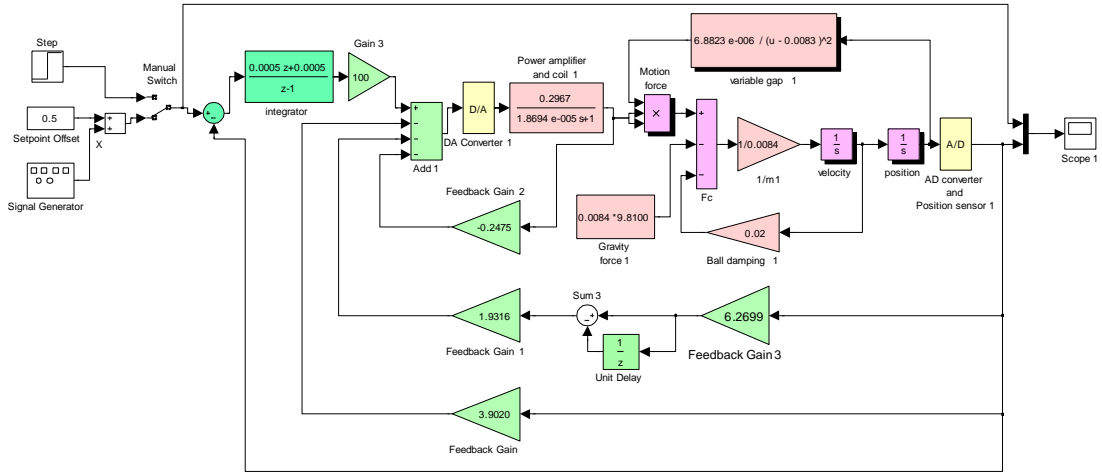


Figure (4.9): CE152 with digital feedback linearization and integral controller

4.10. Parameters modifications for real time magnetic ball levitation

When applying this controller to real time magnetic ball levitation CE152, the system did not work successfully. Modifications were made to the parameters of feedback linearization, as shown in Fig.(4.10) until the system worked well.

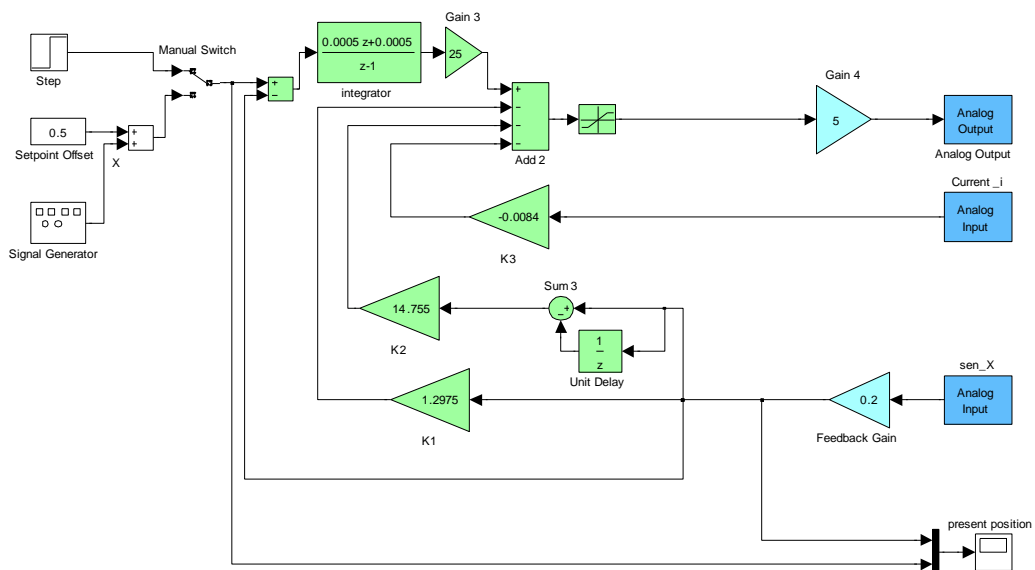


Figure (4.10): Feedback linearization of real-time magnetic ball levitation.

From Fig. (4.8) and (4.9) you can see that k_1 changed from 3.902 to 1.2975, k_2 changed from $(6.2966 \cdot 1.9316 = 12.1625)$ to 14.755, and There is no noticeable change when converting the value of k_3 .

Position sensor will read the position and convert it to volt. The maximum displacement from lowest to highest position is about 1cm as shown in Fig. (4.11). At lowest position '0 cm' the sensor will read 0V and from highest position '1 cm' the sensor will read 5V. To create an error signal, the sensed value (present ball's position) is subtracted from the desired value (required ball's position). Thus, the sensed value should be multiplied by '0.2' to convert it from voltage to position.

5 volts is required to be applied on the coil to make the ball suspended at the upper pole '1cm' . Therefore, the control value should be multiplied by '5' to convert it from position to voltage.

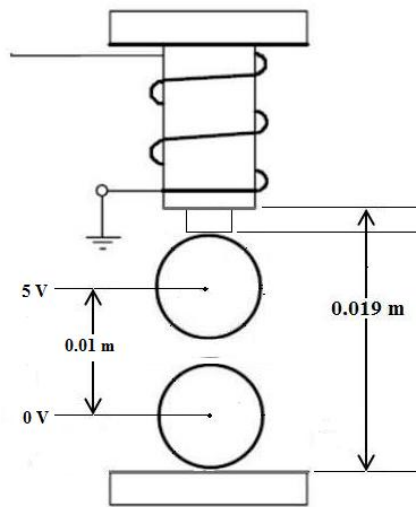


Figure (4.11): Lowest and highest ball's positions.

Example: The reference signal is constant and equal 0.5 ('required vertical displacement is 0.5cm')

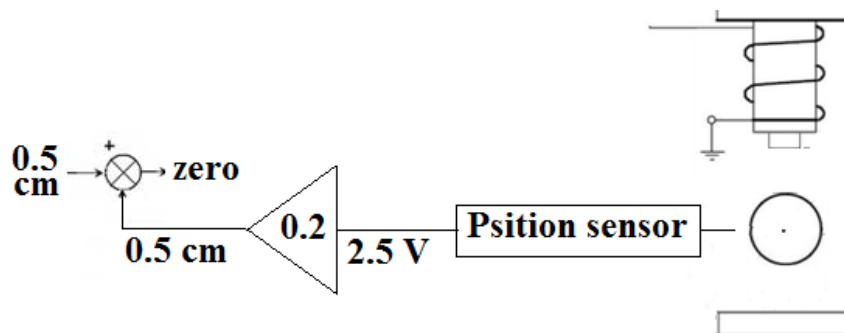


Figure (4.12): Illustrative drawing of the need to add the feedback gain '0.2'

4.11. Simplification of feedback linearization of real-time system

Remove Gain 3, Gain 4 and feedback Gain from Fig(4.9), and then make the required changes. The resulting system is shown in SIMULINK formulation in Fig.(4.13)

$$\text{Integrator} = \frac{0.0005 z + 0.0005}{z - 1} * 25 * 5 = \frac{0.0625 z + 0.0625}{z - 1} \text{-----(4.44)}$$

$$K_{3_{new}} = 5 * .0084 = 0.042 \text{-----(4.45)}$$

$$K_{1,2_{new}} = K_{1,2} * 5 * 0.2 = K_{1,2} \text{-----(4.46)}$$

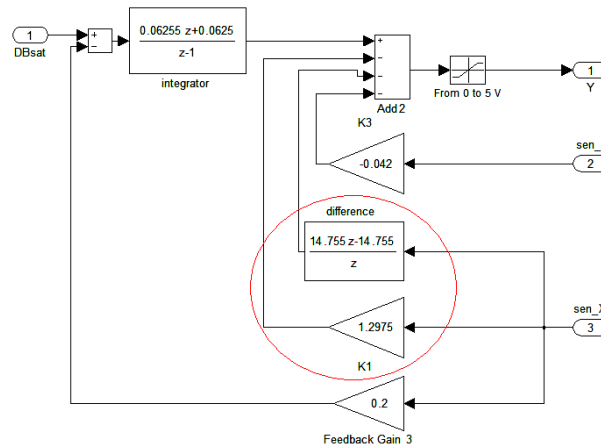


Figure (4.13): First simplification of FBL of real-time maglev CE152.

Current sensor smoothes the response of the system; thus, removing it to approximate the system and to be able to program the system in any programmable kit, combining the circled terms. The resulting system is shown in SIMULINK formulation in Fig.(4.14)

$$\text{Circled terms} = \frac{14.755 z - 14.755}{z} + 1.2975 = \frac{16.0525 z - 14.755}{z} \text{-----(4.47)}$$

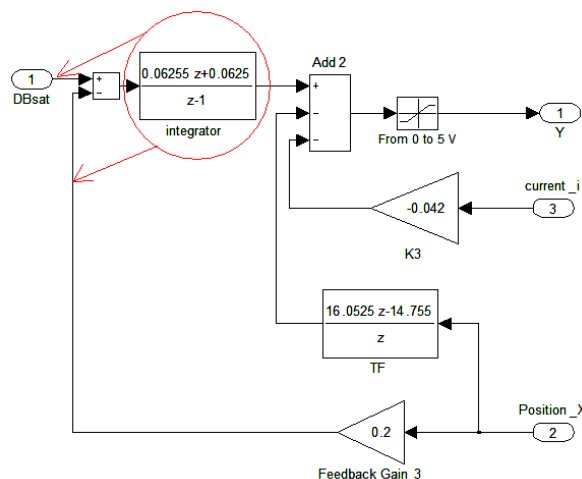


Figure (4.14): Second simplification of FBL of real-time maglev CE152.

The circled block can be moved to arrowed locations in order to simplify the system. The resulting system is shown in SIMULINK formulation in Fig.(4.15)

$$\text{Integrator} = 0.2 * \frac{0.0625 z + 0.0625}{z - 1} = \frac{0.0125 z + 0.0125}{z - 1} \text{-----(4.48)}$$

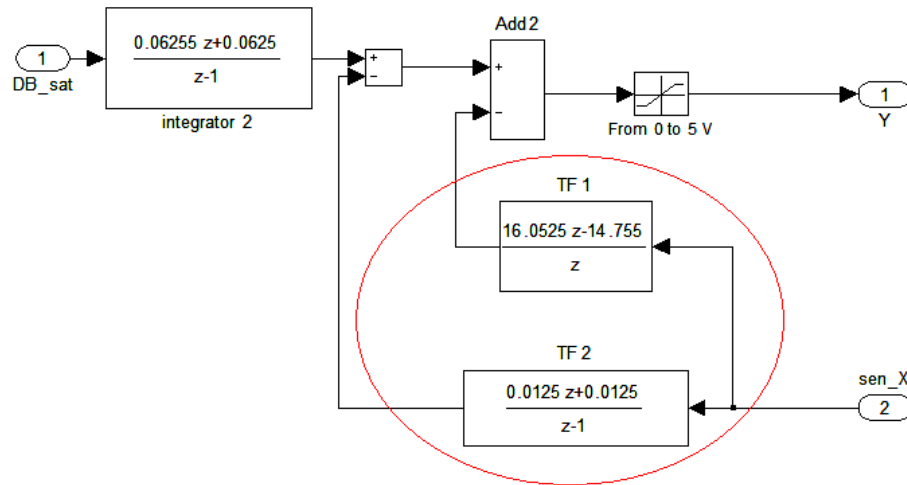


Figure (4.15): Third simplification of FBL of real-time maglev CE152.

The two circled blocks in Fig. (4.15) can be combined as shown in Fig(4.16)

$$\text{Circled terms} = \left(\frac{0.0125 z + 0.0125}{z - 1} \right) + \left(\frac{16.0525 z - 14.755}{z} \right) = \frac{16.065 z^2 - 30.795 z + 14.7550}{z^2 - z} \text{-----(4.49)}$$

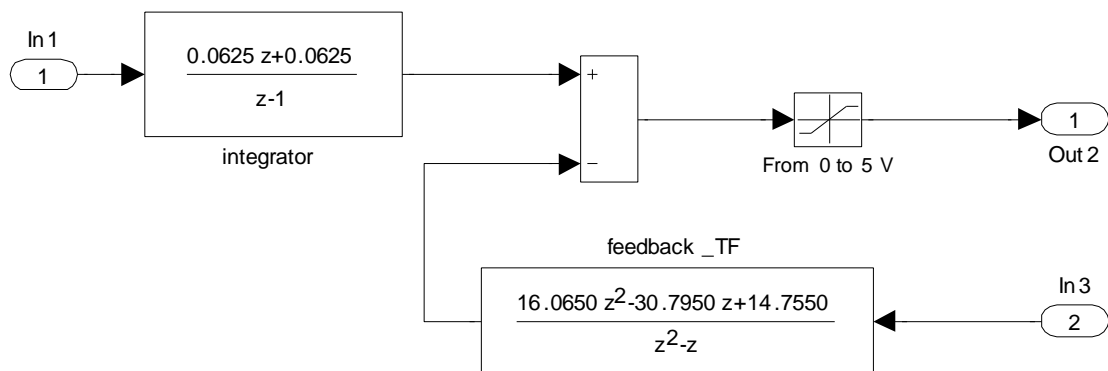


Figure (4.16): Simplification of FBL of real-time maglev CE152.

4.12. Digital equations of feedback linearization

Feedback linearization shown in Fig.(4.17) consists of three blocks: two transfer functions and saturation. The three blocks will be converted to equations that can be programmed using any programmable kit.

4.12.1. Feedback Transfer function

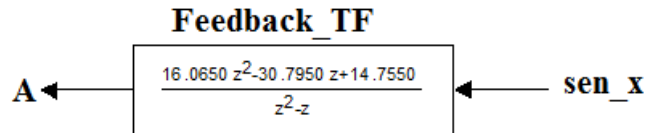


Figure (4.17): Feedback transfer function

$$\frac{A}{senX} = \frac{16.0650 Z^2 - 30.7950 Z + 14.7550}{Z^2 - Z} = \frac{16.0650 - 30.7950 Z^{-1} + 14.7550 Z^{-2}}{1 - Z^{-1}} \text{-----(4.50)}$$

$$\therefore A(n) = 16.0650 senX(n) - 30.7950 senX(n-1) + 14.7550 senX(n-2) + A(n-1) \text{-----(4.51)}$$

4.12.2. Integrator

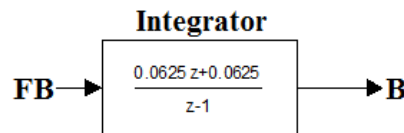


Figure (4.18): Integrator

$$\frac{B}{FB} = \frac{0.0625 Z + 0.0625}{Z - 1} = \frac{0.0625 + 0.0625 Z^{-1}}{1 - Z^{-1}} \text{-----(4.52)}$$

$$\therefore B(n) = 0.0625 FB(n) + 0.0625 FB(n-1) + B(n-1) \text{-----(4.53)}$$

4.12.3. Saturation

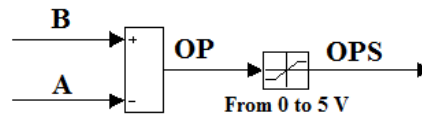


Figure (4.19): Saturation

From (4.51) and (4.53)

$$\therefore OP = B - A \text{-----(4.54)}$$

$$\text{if } (OP > 5) \text{ then } OP = 5; \text{ elseif } (OP < 0) \text{ then } OP = 0; \text{-----(4.55)}$$

OPS = OP;

Equations (4.51), (4.53) and (4.55) should be used to write VHDL code for feedback linearization

CHAPTER 5 METHODOLOGY AND APPROACH

This chapter covers the approached methodology used to develop the control law for linear systems, used to deadbeat nonlinear system using multi-rate deadbeat controller, used to write VHDL code for deadbeat controller of magnetic ball levitation.

5.1. Introduction

The problem of tracking a general reference signal in a ripple-free deadbeat fashion for nonlinear, SISO multi-rate systems is considered. We give a design procedure for a controller under which the output of the closed loop system exactly coincides with the reference signal after a fixed (finite) time. The design provided here allows for constraints on control magnitude as well as on many time domain properties such as overshoot, norms of control signal and error signal.

The Diophantine equation plays an important role in the design and synthesis of controllers in the frequency domain. The Diophantine equation has an infinite number of solutions that all provide an internally stabilizing controller. The parameterization of the Diophantine equation is based on obtaining a matrix equation with the two unknown expressed in matrix form

5.2. Obtaining a model

Fig. (5.1) shows the deadbeat controller for linear plant P which consist of the original plant to be controlled G_c and the modeled filter M_c

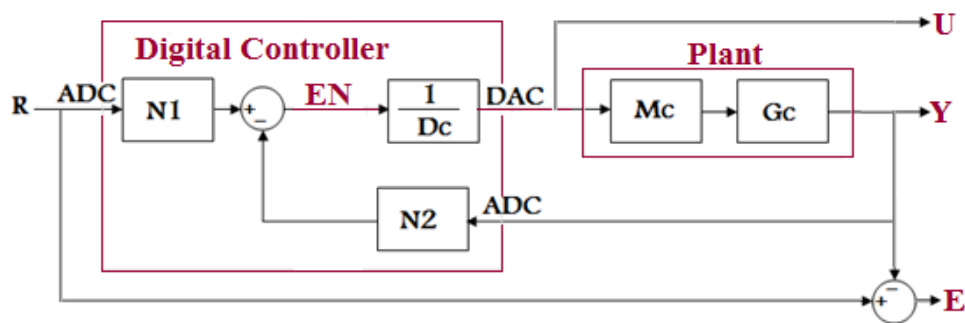


Figure (5.1): Block diagram of deadbeat controller

The following functions are written in q-domain

From figure (5.1)

$$P = Mc * Gc \text{-----}(5.1)$$

$$Y = \left(\frac{R * N_1 - Y * N_2}{D_c} \right) * P = \frac{R * N_1 * P}{D_c} - \frac{Y * N_2 * P}{D_c}$$

$$\therefore Y + \frac{Y * N_2 * P}{D_c} = \frac{R * N_1 * P}{D_c} \Rightarrow Y = \frac{R * N_1 * P}{D_c + N_2 * P} \text{-----}(5.2)$$

$$\therefore Y = \frac{\frac{numR}{denR} * N_1 * \frac{numP}{denP}}{D_c + N_2 * \frac{numP}{denP}} = \frac{numR * numP * N_1}{denR * denP * D_c + denR * numP * N_2}$$

$$Y = \frac{numR}{denR} \left(\frac{numP * N_1}{denP * D_c + numP * N_2} \right) \text{-----}(5.3)$$

From Equation number (5.3)

Y (Desired Response) will be equal Reference signal if and only if

$$numP * N_1 = denP * D_c + numP * N_2$$

and for fast response (deadbeat) all poles should be at Origin

$$\therefore denP * D_c + numP * N_2 = numP * N_1 = 1 \text{-----}(5.4)$$

$$\therefore denP * D_c + numP * N_2 = 1. \text{-----}(5.5)$$

$$\therefore numP * N_1 = 1. \text{-----}(5.6)$$

Transfer function between control and reference

$$U = \frac{Y}{P} \text{-----}(5.7)$$

substituting (5.2) into (5.6)

$$\Rightarrow U = \frac{R * N_1}{D_c + N_2 * P} = \frac{\frac{numR}{denR} * N_1}{D_c + N_2 * \frac{numP}{denP}} = \frac{denP * numR * N_1}{denP * denR * D_c + denR * numP * N_2} \text{-----}(5.8)$$

Transfer function between Error signal and reference

$$E = R - Y = R - \frac{R * N_1 * P}{D_c + N_2 * P} \Rightarrow$$

$$E = \left(\frac{(denP * D_c + N_2 * numP) * numR - (N_1 * numP) * numR}{denR * (denP * D_c + numP * N_2)} \right) \text{-----}(5.9)$$

Substituting (5.5) and (5.6) into (5.3),(5.8),and (5.9)

$$Y = \frac{\text{numR} \left(\frac{\text{numP} * N_1}{\text{denP} * D_c + \text{numP} * N_2} \right)}{\text{denR}} = \frac{\text{numR} \left(\frac{1}{1} \right)}{\text{denR}} = R \text{-----}(5.10)$$

$$U = \frac{\text{denP} * \text{numR} * N_1}{\text{den(R)} * (\text{denP} * D_c + \text{numP} * N_2)} = \frac{\text{denP} * \text{numR} * N_1}{\text{den(R)}} \text{-----}(5.11)$$

$$E = \left(\frac{(\text{denP} * D_c + N_2 * \text{numP}) * \text{numR} - (N_1 * \text{numP}) * \text{numR}}{\text{denR} * (\text{denP} * D_c + \text{numP} * N_2)} \right) = \text{Zero} \text{-----}(5.12)$$

The previous results are obtained by the default solution, which make the system to settle, and eliminate the error within smallest time. The control signal will depend only on denominator of plant, numerator and denominator of the reference signal, and the obtained polynomial N_1 , so if the infinity norm of control signal is larger than acceptable value for any programmable kit such as FPGA, DSP Kit, Microcontroller, and so on. We should decrease the maximum absolute value of control signal by evaluating another polynomial instead of N_1 , so let us called the previous N_1 as N_{1min} .

Zero term will be added to the equation (5.6)

$$\text{numP} * N_{1min} + (0) = 1 \Rightarrow \text{numP} * N_{1min} + (Q_1 * \text{denR}) = 1$$

$$\therefore \text{numP} * N_{1min} + Q_1 * \text{denR} - \text{numP} * \text{denR} * V_1 + \text{numP} * V_1 * \text{denR} = 1$$

$$\Rightarrow \text{numP} * (N_{1min} - \text{denR} * V_1) + \text{denR} (Q_1 + \text{numP} * V_1) = 1$$

Defining new N_1 , and Q_1

$$N_{1new} = N_{1min} - \text{denR} * V_1 \text{-----}(5.13)$$

$$Q_{1new} = Q_{1min} + \text{numP} * V_1 \text{-----}(5.14)$$

Substituting (5.13) into (5.3),(5.8), and(5.9) to find the new transfer functions from output, control, and error signals versus reference signal.

$$Y = \frac{\text{numR} \left(\frac{\text{numP} * (N_{1min} - \text{denR} * V_1)}{\text{denP} * D_c + \text{numP} * N_2} \right)}{\text{denR}} \Rightarrow \frac{\text{numR} \left(\frac{\text{numP} * N_{1new}}{\text{denP} * D_c + \text{numP} * N_2} \right)}{\text{denR}} \Rightarrow$$

$$Y = R \text{-----}(5.15)$$

$$U = \frac{\text{denP} * \text{numR} * (N_{1min} - \text{denR} * V_1)}{\text{den(R)} * (\text{denP} * D_c + \text{numP} * N_2)} = \frac{\text{denP} * \text{numR} * N_{1new}}{\text{den(R)}} \Rightarrow$$

$$U = R * \text{denP} (\text{numR} * N_{1min}) \text{-----}(5.16)$$

$$E = \left(\frac{(\text{denP} * \text{Dc} + \text{N}_2 * \text{numP}) * \text{numR} - \left((\text{N}_{1\text{min}} - \text{denR} * \text{V}_1 \right) * \text{numP} \right) * \text{numR}}{\text{denR} * (\text{denP} * \text{Dc} + \text{numP} * \text{N}_2)} \right)$$

$$= \left(\frac{(\text{denP} * \text{Dc} + \text{N}_2 * \text{numP}) * \text{numR} - (\text{N}_{1\text{new}} * \text{numP}) * \text{numR}}{\text{denR} * (\text{denP} * \text{Dc} + \text{numP} * \text{N}_2)} \right) \Rightarrow$$

$$E = \text{zero} \text{-----} (5.17)$$

The previous results make the system to settle, and eliminate the error. The control signal will depend only on denominator of plant, numerator and denominator of the reference signal, and the polynomial $N_{1\text{new}}$ that depend on the vector V_1 .

Paz evaluates V_1 that decrease the norm of control signal by quadratic programming.

In the next two sections, two approaches will be discussed to evaluate V_1 by another method.

5.3. Decreasing control signal using the first approach

Minimizing control signal by minimizing its numerator by two steps

V_1 will be obtained by making equation (5.13) equal zero; this will be solved by making all coefficient of each order equal zero.

$$N_1 = 0 \Rightarrow N_{1\text{min}} - \text{denR} * V_1 = 0 \text{-----} (5.18)$$

where $N_{1\text{min}}$ and denR are polynomial with known coefficients, while V_1 is vector with unknown coefficients

Steps:

- Choosing the length of V_1 then make a polynomial in q-domain $(a * q + b)$, $(a * q^2 + b * q + c)$, and so on. where a,b,c,... are unknowns.
- Substituting V_1 into (5.18) and solving the polynomial by making the coefficient of each order equal zero. Where $N_{1\text{min}}$ and D_r are polynomials with known coefficients. Such as:

$(n_1 z + n_2)$, $(n_1 z^2 + n_2 z + n_3)$, and so on. where n_1, n_2, n_3, \dots are knowns.

Then,

$$(N_{1\text{min}} - D_r * V_1 = 0)$$

$$= (n_1 * a) * q^4 + (n_2 * a + n_3 * b) * q^3 + (n_3 * a + n_4 * b + n_5 * c) * q^2 + (n_6 * b + n_7 * c - n_8) * q + (n_9 - n_{10} * c + n_{11} * d) = 0 \text{-----}(5.19)$$

$$\begin{bmatrix} n_1 * a = 0 \\ n_2 * a + n_3 * b = 0 \\ n_3 * a + n_4 * b + n_5 * c = 0 \\ n_6 * b + n_7 * c - n_8 = 0 \\ n_9 - n_{10} * c + n_{11} * d \end{bmatrix} \Rightarrow \begin{bmatrix} n_1 & 0 & 0 & 0 \\ n_2 & n_3 & 0 & 0 \\ n_3 & n_4 & n_5 & 0 \\ 0 & n_6 & n_7 & 0 \\ 0 & 0 & -n_{10} & n_{11} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ n_8 \\ -n_9 \end{bmatrix}$$

Using pseudo inverso to find the coefficients of V_1

$$\therefore \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \left(\begin{bmatrix} n_1 & 0 & 0 & 0 \\ n_2 & n_3 & 0 & 0 \\ n_3 & n_4 & n_5 & 0 \\ 0 & n_6 & n_7 & 0 \\ 0 & 0 & -n_{10} & n_{11} \end{bmatrix} \right)^\dagger \begin{bmatrix} 0 \\ 0 \\ 0 \\ n_8 \\ -n_9 \end{bmatrix} \text{-----}(5.20)$$

Example:

Here, we consider the system in q-domain

$$Pq = \frac{0.0015171 (1+110q) (1+3.265q) (1+0.4257q) (1+0.01981q)}{(1-2.166q + 1.181q^2)(1-1.972q + q^2)}$$

Discrete time reference signal in q-domain

$$Rq = \frac{0.020769 q (1+q)}{(1-1.972q + q^2)}$$

Minimum order equation of N_1

$$N_{1min} = -19.6779 * q + 2.8468$$

Computing the coefficients of V_1 with length equal 3

$$\text{Let } V_1 = a * q^2 + b * q + c \text{ where a,b,and c are unknown -----}(5.21)$$

$$\text{denR} = q^2 - 1.9723 * q + 1 \text{-----}(5.22)$$

Substituting (5.21), (5.22), N_{1min} into (5.18)

$$\therefore N_{1min} - \text{denR} * V_1 = (-a)q^4 + (1.9723 a - b)q^3 + \dots (-a + 1.9723b - c)q^2 + (-b + 1.9723c - 1.9678)q + (2.8468 - c) = 0 \text{-----}(5.23)$$

$$\therefore \begin{bmatrix} -a = 0 \\ 1.9723 a - b = 0 \\ -a + 1.9723b - c = 0 \\ -b + 1.9723c - 1.9678 = 0 \\ 2.8468 - c = 0 \end{bmatrix} \Rightarrow \begin{bmatrix} -1 & 0 & 0 \\ 1.9723 & -1 & 0 \\ -1 & 1.9723 & -1 \\ 0 & -1 & 1.9723 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.9678 \\ -2.8468 \end{bmatrix}$$

$$\therefore \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \left(\begin{bmatrix} -1 & 0 & 0 \\ 1.9723 & -1 & 0 \\ -1 & 1.9723 & -1 \\ 0 & -1 & 1.9723 \\ 0 & 0 & -1 \end{bmatrix} \right)^\dagger \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1.9678 \\ -2.8468 \end{bmatrix} = \begin{bmatrix} 0.5857 \\ 1.3719 \\ 1.9616 \end{bmatrix}$$

$$\therefore V_1 = \begin{bmatrix} 0.5857 \\ 1.3719 \\ 1.9616 \end{bmatrix}$$

Substituting $N_{1\min}$, denR, and V_1 into (2.9) to obtain new N_1 will increase the order of N_1 as shown in the following relation

$$\therefore N_1 = N_{1\min} - \text{denR} * V_1 = 0.8852 + 0.5292q + 0.1585q^2 - 0.2166q^3 - 0.5857q^4 \text{-----} (5.24)$$

The order of new N_1 is larger than the order of $N_{1\min}$ by three; therefore, the length of vector V_1 will increase the order of transfer functions from output, control, and error signals versus reference signal by three. Therefore, the settling time will be increased by three sampling times.

5.4. Decreasing control signal by second approach

Minimizing control signal by evaluating the new vector called “best V_1 ” by combining the vector which was obtained by quadratic programming (Paz method) with vector which was obtained by my first approach

Steps:

- Computing Vector V_1 by Paz method and infinity norm of control signal using this vector
- Computing Vector V_1 by my first approach and infinity norm of control signal using this vector
- Changing the elements of the vector which was obtained by my approach with elements of the vector which was obtained by Paz (element by element), and deciding which element of my vector will decrease the norm of control signal

when replacing with element of Paz vector to make a new vector called “ V_{new1} ”.

This is called “First step”

- Subtracting Vector which was obtained by Paz from vector V_{new1} to find the new vector and add the partial of this vector multi-times to V_{new1} and still adding while the norm of control signal is still decreasing. This is called “Second step”

The previous steps will be done as follow:

Paz Vector

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ \dots \ x_n],$$

Computing the norm of control signal using X

$$\text{norm}(U, \text{inf}) = \text{norm}x \text{-----} (5.25)$$

First approach Vector

$$V_1 = [v_1 \ v_2 \ v_3 \ v_4 \ \dots \ v_n],$$

Computing the norm of control signal using V_1

$$\text{norm}(U, \text{inf}) = \text{norm}v_1 \text{-----} (5.26)$$

First step:

Changing the first element of X with first element of V_1

$$V_{new1} = [v_1 \ x_2 \ x_3 \ x_4 \ \dots \ x_n],$$

Computing the norm of control signal using V_{new1}

$$\text{norm}(U, \text{inf}) = \text{norm}V_{new1} \text{-----} (5.27)$$

Changing the second element of X with second element of V_1

$$V_{new2} = [x_1 \ v_2 \ x_3 \ x_4 \ \dots \ x_n],$$

Computing the norm of control signal using V_{new2}

$$\text{norm}(U, \text{inf}) = \text{norm}V_{new2} \text{-----} (5.28)$$

Changing the third element of X with third element of V_1

$$V_{new3} = [x_1 \ x_2 \ v_3 \ x_4 \ \dots \ x_n],$$

Computing the norm of control signal using V_{new3}

$$\text{norm}(U, \text{inf}) = \text{norm}V_{new3} \text{-----} (5.29)$$

Changing the fourth element of X with fourth element of V₁

$$V_{new\ 4} = [x_1\ x_2\ x_3\ v_4\ \dots\dots\dots x_n],$$

Computing the norm of control signal using V_{new 4}

$$\text{norm}(U, \text{inf}) = \text{norm} V_{new\ 4} \text{-----}(5.30)$$

.

.

.

Changing the nth element of X with nth element of V₁

$$V_{new\ n} = [x_1\ x_2\ x_3\ x_4\ \dots\dots\dots v_n],$$

Computing the norm of control signal using V_{new n}

$$\text{norm}(U, \text{inf}) = \text{norm} V_{new\ n} \text{-----}(5.31)$$

Using elements which decrease the norm

Assume that $\text{norm} V_{new\ 2}$, $\text{norm} V_{new\ 4}$, and $\text{norm} V_{new\ n} < \text{norm} x$

$$\therefore \text{the new vector will be } V_{new} = [x_1\ v_2\ x_3\ v_4\ \dots\dots\dots v_n] \text{-----}(5.32)$$

The infinity norm obtained using V_{new} less than infinity norm obtained by using any of the previous vectors

Second step:

The difference between Vector V_{new} equation(5.32) and X equation (5.25) will be used to decrease the infinity norm of control signal by multi iterations as follow:

The difference between X and V_{new}

$$\begin{aligned} V_d &= V_{new} - X = [x_1\ v_2\ x_3\ v_4\ \dots v_n] - [x_1\ x_2\ x_3\ x_4\ \dots x_n] \\ &= [(x_1 - x_1)\ (v_2 - x_2)\ (x_3 - x_3)\ (v_4 - x_4)\ \dots\ (v_n - x_n)] \\ &= [0\ (v_2 - x_2)\ 0\ (v_4 - x_4)\ \dots\ (v_n - x_n)] \text{-----}(5.33) \end{aligned}$$

When adding V_d to X, the infinity norm of control signal is decreased; thus, the partial of V_d will be still added to X while the infinity norm of control signal is still decreasing as follow:

while new norm < previous norm

$$V_{new} = V_{new} + \frac{V_d}{L}, \text{ where } L \text{ is any positive decimal number} \text{-----}(5.34)$$

compute new infinity norm

end

$$V_{best} = V_{new}.$$

for example: the while loop was done 4 times and L=10 ⇒
the vector which I called V_{best} =

$$V_{best} = \left[x_1 \quad \left(v_2 + \frac{4v_2}{10} \right) \quad x_3 \quad \left(v_4 + \frac{4v_4}{10} \right) \quad \dots \quad \left(v_n + \frac{4v_n}{10} \right) \right] \text{-----}(5.35)$$

5.5. Multi-rate ripple-free deadbeat controller for nonlinear system

Magnetic ball Levitation CE152 was chosen as a case study since this apparatus is placed at IUG laboratory.

Ripple-free deadbeat control is desired to be used to meet the certain specifications such as: Finding second order approximation of maglev CE152 with feedback linearization using two parameters rising time (or settling time) and overshoot.

Equations (2.12), (2.13), and (2.14) will be used

$$\text{The second order approximation} \approx \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \text{-----}(5.36)$$

Converting the approximated model from analog to digital using required sampling rate in deadbeat Controller

The second order approximation in q-domain

$$\frac{Kq + Lq^2}{1 + Mq + Nq^2} \text{-----}(5.37)$$

Where K,L,M, and N are unknown and depend on damping ratio and natural frequency

The reference signal in q-domain

$$\text{Input signal } R(q) = \frac{O + Pq}{(1 - Qq + Rq^2)} \text{-----}(5.38)$$

in z-domain

$$\text{Input signal } R(q) = \frac{Oz^2 + Pz}{(z^2 - Qz + R)} \text{-----}(5.39)$$

Where O,P,Q, and R are unknown

Numerator and denominator of plant and reference signal in q-domain

$$\text{numP}(q) = Lq^2 + Qq, \text{denP}(q) = Nq^2 + Mq + 1, \text{denR}(q) = Rq^2 + Qq + 1 \text{-----}(5.40)$$

$$\text{numP}(z) = L + Qz, \text{denP}(z) = N + Mz + z^2, \text{denR}(z) = R + Qz + z^2 \text{-----}(5.41)$$

Evaluating the minimum order for N_1 and Q_1

$$\text{Order of vector } N_1 \text{ and } Q_1 = \max(\text{order}(\text{numP}), \text{order}(\text{denR})) - 1 = 1 \text{-----}(5.42)$$

Evaluating the minimum order for N_1 and Q_1

$$\text{Order of vector } N_2 \text{ and } D_c = \max(\text{order}(\text{numP}), \text{order}(\text{denP})) - 1 = 1 \text{-----}(5.43)$$

Defining two polynomials with order equal '1' with unknown coefficients and making the first Diophantine equation.

$$\text{let } N1 = a*q + b \text{ and } Q1 = c*q + d \text{ where } a, b, c, \text{ and } d \text{ are unknown-----}(5.44)$$

$$\therefore \text{First Diophantine equation } (Np * N1) + (Dr * Q1) = 1 \text{-----}(5.45)$$

Substituting (5.40) and (5.44) into (5.45)

$$\therefore (L * q^2 + K * q)(a*q + b) + (Rq^2 + Q*q + 1)(c*q + d) - 1 = 0 \text{-----}(5.46)$$

$$\Rightarrow (L * a + R * c) * q^3 + (K * a + L * b + Q * c + R * d) * q^2 + (K * b + c + Q * d) * q + (d - 1) = 0 \text{-----}(5.47)$$

All coefficients of equation (5.47) should be equal zero, to find all unknowns a, b, c and d,

$$\therefore \begin{bmatrix} L * a + R * c = 0 \\ K * a + L * b + Q * c + R * d = 0 \\ K * b + c + Q * d = 0 \\ d - 1 = 0 \end{bmatrix} \Rightarrow \begin{bmatrix} L * a + R * c = 0 \\ K * a + L * b + Q * c + R * d = 0 \\ K * b + c + Q * d = 0 \\ d = 1 \end{bmatrix} \text{-----}(5.48)$$

Using pseudo inverso to solve equation(5.48), in this special case we don't need to use pseudo inverso, in general I programmed an m-file to solve any equation either the matrix is square or not

$$\begin{bmatrix} L & 0 & R & 0 \\ K & L & Q & R \\ 0 & K & 1 & Q \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{-----}(5.49)$$

$$\Rightarrow \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} L & 0 & R & 0 \\ K & L & Q & R \\ 0 & K & 1 & Q \\ 0 & 0 & 0 & 1 \end{bmatrix}^\dagger * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{-----}(5.50)$$

$\therefore N1 = a*q + b, Q1 = c*q + d$, now a, b, c, and d are knowns

Defining two polynomials with order equal '1' with unknown coefficients and making the second Diophantine equation.

$$\text{let } N_2 = a*q+b \text{ and } D_c = c*q+d \text{ where } a, b, c, \text{ and } d \text{ are unknown} \text{-----}(5.51)$$

$$\therefore \text{Second Diophantine equation } (\text{numP}*N_2)+(\text{denP}*D_c)=1 \text{ -----}(5.52)$$

Substituting (5.40) and (5.51) into (5.52)

$$\therefore (L*q^2+Q*q)(a*q+b) + (N*q^2 +M*q+1)(c*q+d) - 1 = 0 \text{-----}(5.53)$$

$$\Rightarrow (L*a + N*c)*q^3 + (Q*a + L*b + M*c + N*d)*q^2 + (Q*b + c + M*d)*q + (d - 1) = 0 \text{-----}(5.54)$$

All coefficients of equation (5.47) should be equal zero, to find all unknowns a, b, c and d.

$$\therefore \begin{bmatrix} L*a + N*c = 0 \\ Q*a + L*b + M*c + N*d = 0 \\ Q*b + c + M*d = 0 \\ d - 1 = 0 \end{bmatrix} \Rightarrow \begin{bmatrix} L*a + N*c = 0 \\ Q*a + L*b + M*c + N*d = 0 \\ Q*b + c + M*d = 0 \\ d = 1 \end{bmatrix} \text{-----}(5.55)$$

Using pseudo inverso to solve equation(5.48)

$$\therefore \begin{bmatrix} L & 0 & N & 0 \\ Q & L & M & N \\ 0 & Q & 1 & M \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{-----}(5.56)$$

$$\therefore \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} L & 0 & N & 0 \\ Q & L & M & N \\ 0 & Q & 1 & M \\ 0 & 0 & 0 & 1 \end{bmatrix}^+ * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{-----}(5.57)$$

$$N_2 = a*q+b \text{ and } D_c = c*q+d \text{-----}(5.58)$$

These equations will be used in the next step to find deadbeat controller for approximated model of maglev CE152 with feedback linearization

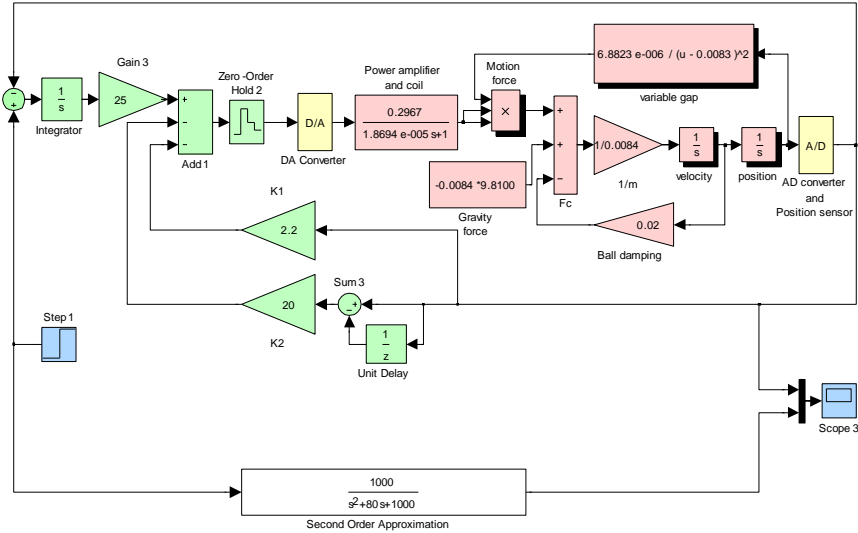


Figure (5.2): CE152 with third FBL and its approximated model.

Deadbeat controller for magnetic ball levitation with feedback linearization will be found by evaluating deadbeat controller for its approximated model, thus deadbeat controller will be found for $\frac{1000}{s^2+80s+1000}$ to follow reference signal $R(s)=\frac{6}{(s^2+4)}$.

Converting the plant and reference signal from s-domain to q-domain with sampling time = 0.1 sec.

$$P(q) = \frac{0.038652 q (1+0.7662q)}{(1-0.8564q) (1-0.5247q)} \text{-----(5.59)}$$

$$R(q) = \frac{0.0003q(1+q)}{(1-2q+q^2)} \text{-----(5.60)}$$

$$\therefore \text{numP}=[0 \quad 0.0387 \quad 0.0296], \text{denP}=[1 \quad -1.3811 \quad 0.4493]$$

Numerator and denominator of plant and reference signal in q-domain

$$\text{numP}(q)=0.0296*q^2+0.0387*q, \text{denP}=0.4493*q^2-1.3811*q+1 \text{-----(5.61)}$$

$$\text{denR}(q)=q^2-2*q+1 \text{-----(5.62)}$$

z-domain

$$\text{numP}(z)=0.0296+0.0387*z, \text{denP}=0.4493-1.3811z+z^2 \text{-----(5.63)}$$

$$\text{denR}(z)=1-2*z+z^2 \text{-----(5.64)}$$

Evaluating the minimum order for N_1 and Q_1 using equation (5.42)

$$\text{Order of vector } N_1 \text{ and } Q_1 = \max(1, 2) - 1 = 1 \text{-----(5.65)}$$

Evaluating the minimum order for N_1 and Q_1 using equation (5.43)

$$\text{Order of vector } N_2 \text{ and } D_c = \max(1, 2) - 1 = 1 \text{-----(5.66)}$$

Defining two polynomials with order equal '1' with unknown coefficients and making the first Diophantine equation using equation (5.44) and (5.45)

$$\therefore N_1 = a*q+b \text{ and } Q_1 = c*x+d \text{ where } a, b, c, \text{ and } d \text{ are unknowns-----(5.67)}$$

Substituting (5.61),(5.62) and (5.67) into (5.45) to make the first Diophantine equation

$$\text{First Diophantine equation } (N_p * N_1) + (D_r * Q_1) = 1$$

$$\therefore (N_p * N_1) + (D_r * Q_1) - 1 = 0$$

$$\therefore (0.0296*q^2 + 0.0387*q)(a*q+b) + (q^2 - 2*q + 1)(c*x+d) - 1 = 0$$

$$\therefore (0.0296*a+c)*q^3 + (0.0387*a + 0.0296*b - 2*c + d)*q^2 + \dots$$

$$(0.0387*b + c - 2*d)*q + (d - 1) = 0 \text{-----(5.68)}$$

Evaluating a, b, c, and d by making all coefficients of equation (5.68) equal zero to find all unknowns a, b, c and d using equations (5.47), (5.49) and (5.50).

$$\therefore \begin{bmatrix} 0.0296 & 0 & 1 & 0 \\ 0.0387 & 0.0296 & -2 & 1 \\ 0 & 0.0387 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0.0296 & 0 & 1 & 0 \\ 0.0387 & 0.0296 & -2 & 1 \\ 0 & 0.0387 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -21.0022 \\ 35.6409 \\ 0.6220 \\ 1 \end{bmatrix}$$

$$\therefore N_1 = 35.64 - 21q, Q_1 = 1 + 0.622q$$

Defining two polynomials with order equal '1' with unknown coefficients and making the second Diophantine equation using equation (5.51) and (5.52)

$$\therefore N_2 = a*q+b \text{ and } D_c = c*x+d \text{ where } a, b, c, \text{ and } d \text{ are unknowns-----(5.69)}$$

Substituting (5.61), and (5.69) into (5.52) to make the second Diophantine equation

Second Diophantine Equation $(\text{numP} \cdot N_2) + (\text{denP} \cdot D_c) = 1$

$$\therefore (\text{numP} \cdot N_2) + (\text{denP} \cdot D_c) - 1 = 0$$

$$(0.0296 \cdot a + 0.4493 \cdot c) \cdot q^3 + (0.0387 \cdot a + 0.0296 \cdot b - 1.3811 \cdot c + 0.4493 \cdot d) \cdot q^2 + \dots \\ (0.0387 \cdot b + c - 1.3811 \cdot d) \cdot q + (d - 1) = 0 \text{-----(5.70)}$$

Evaluating a, b, c, and d by making all coefficients of equation (5.70) equal zero to find all unknowns a, b, c and d using equations (5.54), (5.56) and (5.57).

$$\therefore \begin{bmatrix} 0.0296 & 0 & 0.4493 & 0 \\ 0.0387 & 0.0296 & -1.3811 & 0.4493 \\ 0 & 0.0387 & 1.0000 & -1.3811 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0.0296 & 0 & 0.4493 & 0 \\ 0.0387 & 0.0296 & -1.3811 & 0.4493 \\ 0 & 0.0387 & 1 & -1.3811 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -8.3666 \\ 21.4633 \\ 0.5515 \\ 1 \end{bmatrix}$$

$N_2 = 21.46 - 8.367 \cdot q$, $D_c = 1 + 0.5515 \cdot q$

$N_1 = 35.64 - 21 \cdot q$, $N_2 = 21.46 - 8.367 \cdot q$, and $D_c = 1 + 0.5515 \cdot q$ will be used in the following chapter to simulate the deadbeat controller for maglev CE152 with feedback linearization.

5.6. Converting real-time model from blocks to equations

Ripple-free deadbeat controller with feedback linearization for real-time magnetic ball levitation shown in Fig.(4.2) consists of two subsystems. The first subsystem is feedback linearization that consist of two transfer functions with sampling rate equal 0.001 sec and the second subsystem is deadbeat controller that consist of three transfer functions with sampling rate equal 0.01 sec.

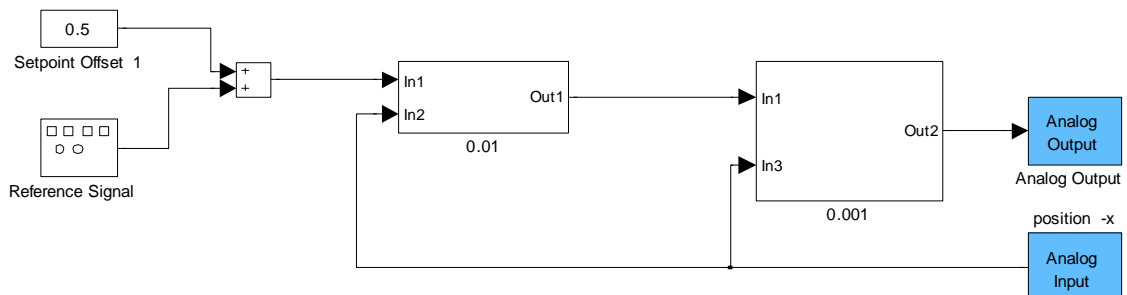


Figure (5.3): SIMULINK deadbeat controller for real-time maglev CE152

Converting real time model from blocks to equations is very important to program any circuit or controller on any programmable kit such as PIC, DSP, FPGA, and so on.

Some programmable kits don't deal with floating points; thus, fractional number will be removed with small very affecting on its performance by multiplying the numerator and denominator of Transfer function by large number 2^n before approximation, where n is positive integer number.

2^n : multiplying binary number by '2' n-times means shifting the binary number to the left n-bits

First subsystem: feed-Back linearization

Feedback linearization, shown in Fig.(5.3) consists of three blocks: integrator, feedback transfer function, and saturation

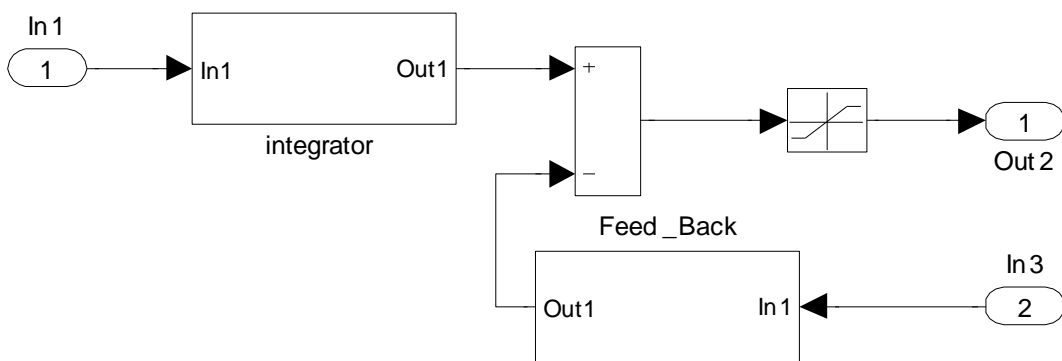


Figure (5.4): Feedback linearization for real-time maglev CE152

a) Integrator

Integrator of feedback linearization can be converted to equation and re-drawn by SIMULINK model as shown in Fig.(5.5)

$$\frac{OP}{IN} = \frac{0.0625 z + 0.0625}{z - 1} \text{-----(5.71)}$$

Multiplying numerator and denominator of equation (5.71) by 128

$$\begin{aligned} \therefore \frac{OP}{IN} &= \frac{8 z + 8}{128 z - 128} = \frac{8 + 8z^{-1}}{128 - 128 z^{-1}} \\ \therefore 128*OP &= 8*IP+8*IP1+128*OP1 \text{-----(5.72)} \end{aligned}$$

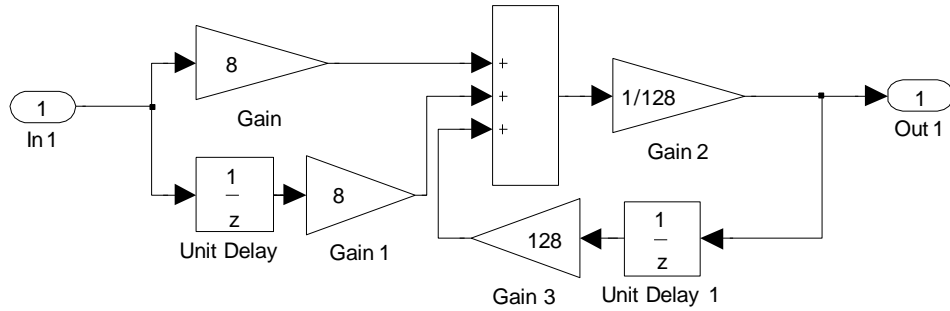


Figure (5.5): Integrator

All gains in the previous figure are not fractional number except the output gain, $\frac{1}{128} = \frac{1}{2^7} \Rightarrow$ shifting the output binary number to the right 7 bits.

b) Feedback Transfer Function

Feedback transfer function of feedback linearization can be converted to equation and re-drawn by SIMULINK model as shown in Fig.(5.6)

$$\frac{OP}{IP} = \frac{16.0650 - 30.7950 z^{-1} + 14.7550z^{-2}}{1 - z^{-1}} \text{-----(5.73)}$$

Multiplying numerator and denominator of equation (5.73) by 128

$$\frac{OP}{IP} \approx \frac{2056 - 3942z^{-1} + 1889z^{-2}}{128 - 128z^{-1}} \text{-----(5.74)}$$

$$\therefore 128 * OP = 2056*IP - 3942*IP1 + 1889*IP2 + 128* OP1 \text{-----(5.75)}$$

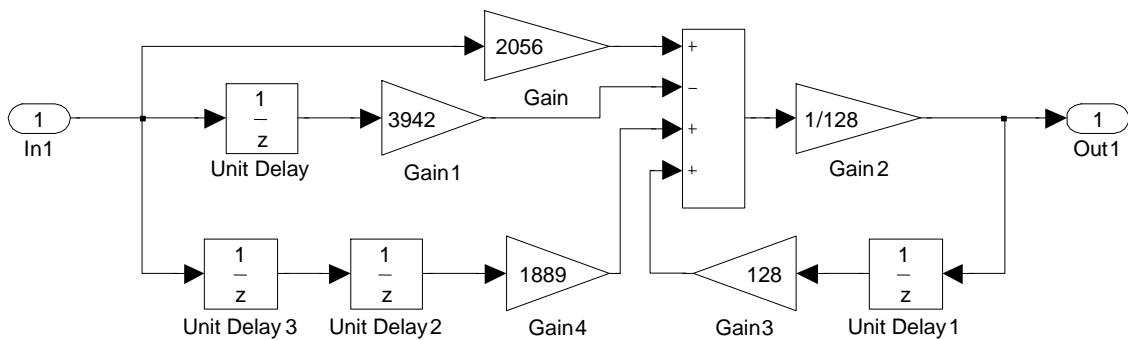


Figure (5.6): Feedback transfer function

All gains in the previous figure are not fractional number except the output gain, $\frac{1}{128} = \frac{1}{2^7} \Rightarrow$ shifting the output binary number to the right 7 bits.

All blocks of Feedback linearization

Feedback linearization of maglev CE152 can be re-drawn by SIMULINK model as shown in Fig.(5.6)

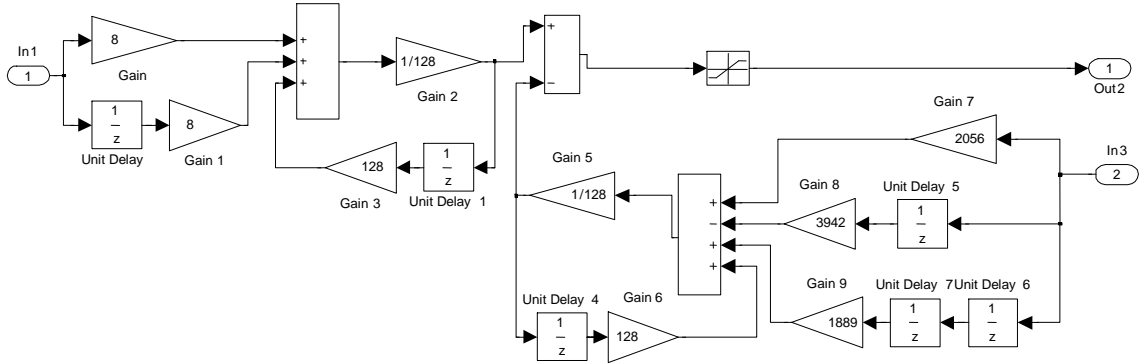


Figure (5.7): Feedback linearization of maglev CE152

Magnetic ball levitation CE152 worked successfully when applying the block diagram shown in Fig. (5.7); therefore, feedback linearization can be programmed on any programmable kit without using fractional number by programming the two equations (5.72) and (5.75).

Second subsystem: Deadbeat controller

Deadbeat controller consists of three blocks: N1, N2, and 1/Dc.

a) N1

N1 can be converted to equation and re-drawn by SIMULINK model as shown in Fig.(5.8)

$$\frac{OP}{IN} = \frac{26.1502 z - 15.4985}{z} \text{-----(5.76)}$$

Multiplying numerator and denominator of equation (5.76) by 128

$$\frac{OP}{IN} = \frac{3347.2256 z - 1983.808}{128 z} \approx \frac{3347 - 1894z^{-1}}{128}$$

$$\therefore 128 * OP = 3347 * IP - 1894 * IP1 \text{-----(5.77)}$$

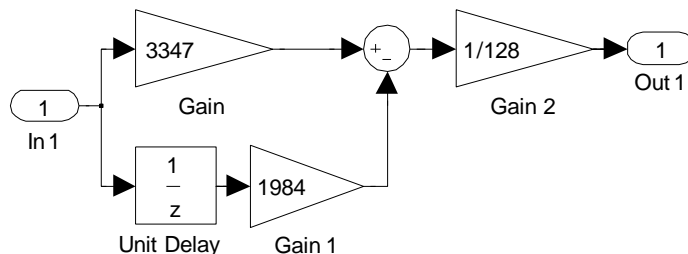


Figure (5.8): N1 of deadbeat controller

All gains in the previous figure are not fractional number except the output gain,

$$\frac{1}{128} = \frac{1}{2^7} \Rightarrow \text{shifting the output binary number to the right 7 bits.}$$

b) N_2

N_2 can be converted to equation and re-drawn by SIMULINK model as shown in Fig.(5.9)

$$\frac{OP}{IP} = \frac{3.4460 z^{-1} - 1.6353}{z} \text{-----(5.78)}$$

Multiplying numerator and denominator of equation (5.78) by 128

$$\begin{aligned} \therefore \frac{OP}{IP} &\approx \frac{441 z - 209}{128 z} = \frac{441 - 209z^{-1}}{128} \\ \therefore 128 * OP &= 441 * IP - 209 * IP1 \text{-----(5.79)} \end{aligned}$$

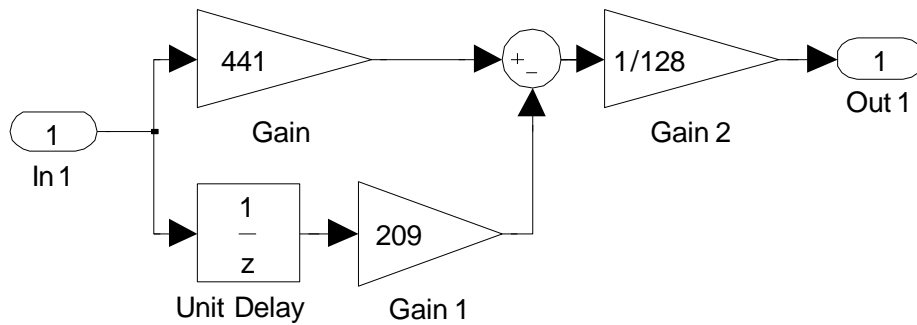


Figure (5.9): N_2 of deadbeat controller

All gains in the previous figure are not fractional number except the output gain,

$$\frac{1}{128} = \frac{1}{2^7} \Rightarrow \text{shifting the output binary number to the right 7 bits.}$$

c) $\frac{1}{D_c}$

$\frac{1}{D_c}$ can be converted to equation and re-drawn by SIMULINK model as shown

in Fig.(5.10)

$$\frac{OP}{IP} = \frac{Z}{Z + 0.60266} \text{-----(5.80)}$$

Multiplying numerator and denominator of equation (5.80) by 128

$$\begin{aligned} \frac{OP}{IP} &\approx \frac{128 Z}{128 Z + 77} = \frac{128}{128 + 77Z^{-1}} \\ 128 * OP &= 128 * IP - 77 * OP1 \text{-----(5.81)} \end{aligned}$$

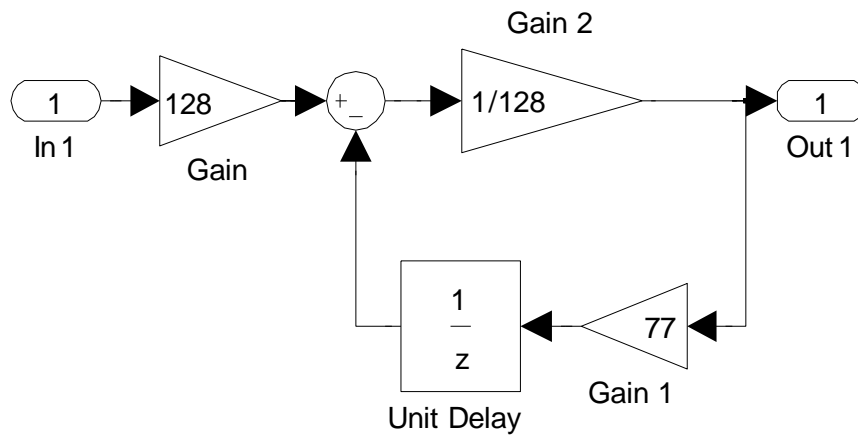


Figure (5.10): $\frac{1}{D_c}$ of deadbeat controller

Deadbeat model

Deadbeat controller of maglev CE152 can re-drawn by SIMULINK model as shown in Fig.(5.11)

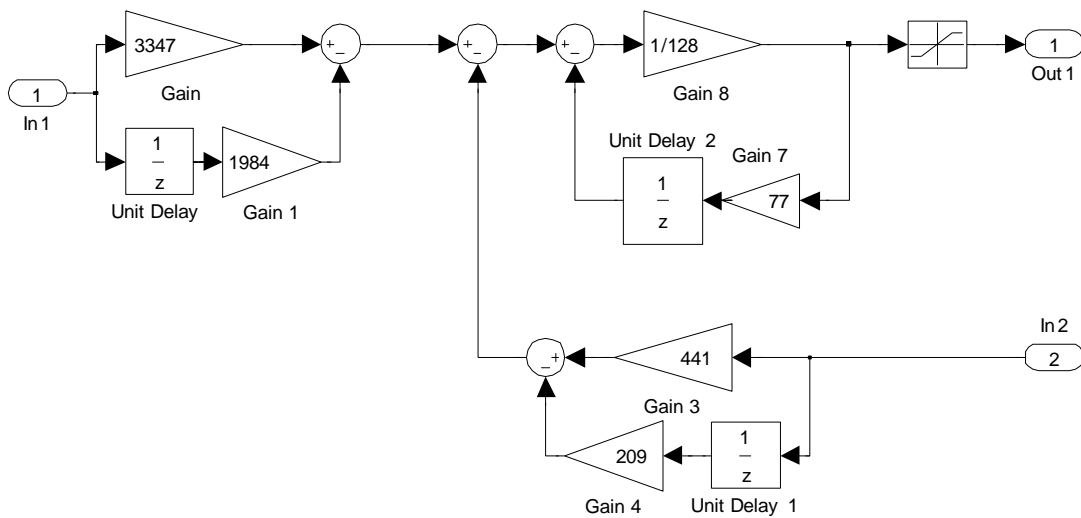


Figure (5.11): Deadbeat of maglev CE152

Therefore, deadbeat controller can be programmed on any programmable kit without using fractional number by programming the three equations (5.77),(5.79) and (5.80)

5.7. Writing VHDL code for deadbeat controller

In this section, VHDL code for deadbeat controller shown in Fig. (5.12) will be written using equations (5.77): $128 * OP = 3347 * IP - 1984 * IP1$, (5.79): $128 * OP = 441 * IP - 209 * IP1$, and (5.81): $128 * OP = 128 * IP - 77 * OP1$.

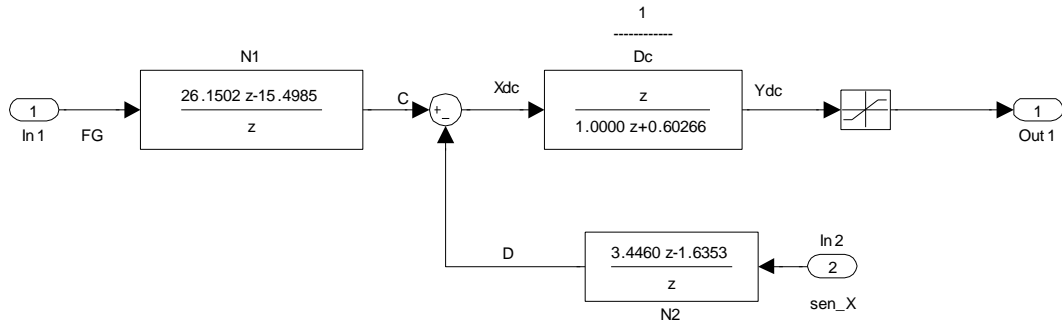


Figure (5.12): Deadbeat of maglev CE152

There are two important steps after evaluating the equations to be programmed

- Determining number of bits required for registers and signals in VHDL language.

Inputs and output are both 8 bits

The largest constant number (3347) requires 12 bits

the summation of multiplication terms for equation (5.72) and (5.75) are $4+2=6$ terms

1 bit for -ve or +ve sign and one more bit.

$\Rightarrow 20+6+1+1=28$ bits will be used

- Converting constant numbers of equations to be programmed from decimal to binary using suitable number of bits.

28 bits (used for registers and signals)- 8 bits (used for input and output signals)=20 bits

$$\left[\begin{array}{l} a_{3347} = "00000000110100010011"; \\ a_{1984} = "000000001111100000"; \\ a_{441} = "0000000000110111001"; \\ a_{209} = "0000000000011010001"; \\ a_{128} = "0000000000010000000"; \\ a_{77} = "0000000000001001101" \end{array} \right] \text{-----(5.82)}$$

You can see VHDL code for deadbeat controller in Appendix A

CHAPTER 6

SIMULATION AND REAL-TIME APPLICATION

This chapter consists of five sections: the first section discusses evaluating new vector that can reduce infinity norm of control signal, 2-norm of error signal and overshoot for linear systems and linearized nonlinear systems. Second section discusses evaluating second order approximation of linearized magnetic ball levitation. Third section discusses evaluating multi-rate ripple free deadbeat control for SIMULINK model of magnetic ball levitation CE152. Fourth section discusses evaluating multi-rate ripple free deadbeat control for real-time magnetic ball levitation CE152 using data acquisition card MF624 and real-time toolbox in SIMULINK environment. Fifth section discusses simulation of VHDL code using Xilinx DSP toolbox

6.1. Improving deadbeat controller for linear systems

In this section, improving deadbeat controller for linear system by evaluating the best vector which can reduce the maximum absolute value of control signal (required energy), reduce overshoot, and reduce infinity norm and H2-norm of the tracking error.

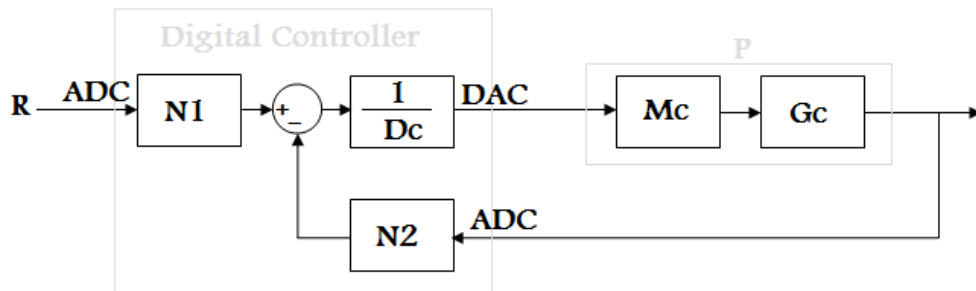


Figure (6.1): Deadbeat closed loop system

Example:

Given the system $G_c = \exp(-0.2*s) \frac{200}{s^2 - 2s + 2}$ with time delay = 0.2 sec. We wish to track the sinusoid $R(t) = \sin(2t + \pi/5)$ and minimizes the infinity norm of control energy.

Tracking model $M_c = \frac{100}{s^2 + 4}$ and the length of vector is 3 will be used.

The continuous tracking model filter, is a requirement that ensures the exact tracking of the reference signal $R(t)$ occurs with no ripple after the transient period[11]

$$M_c = \frac{100}{s^2 + 4} \text{ is chosen since Laplace transform of } \sin(2t) = \frac{2}{s^2 + 4}$$

If we choose the sampling time = 0.1 sec. then:

The reference signal and plant in q-domain will be,

$$R(q) = \frac{0.19935q(1-0.7573q)}{(1-1.96q+q^2)} \text{-----(6.1)}$$

$$P(q) = \frac{0.086608q(1+10.28q)(1+1.041q)(1+0.1054q)}{(1-2.166q+1.181q^2)(1-1.972q+q^2)} \text{-----(6.2)}$$

Computing the minimum order of N_1 and Q_1 using Equation (5.42)

Computing the minimum order solutions to the first Diophantine equations using equations (5.45) to (5.50), we obtain:

$$\left[\begin{array}{l} N_1 = 1.49 - 1.111q. \\ Q_1 = 1 + 1.831q + 1.211q^2 + 0.1085q^3. \end{array} \right] \text{-----(6.3)}$$

Computing the minimum order of N_2 and D_c using Equation (5.43)

Computing the minimum order solutions to the second Diophantine equations using equations (5.51) to (5.58), we obtain:

$$\left[\begin{array}{l} N_2 = 6.73 - 1.482q + 1.197q^2 - 3.434q^3 \\ D_c = 1 + 3.577q + 2.97q^2 + 0.2746q^3 \end{array} \right] \text{-----(6.4)}$$

After that,

Computing the transfer functions from output, control, and error signals versus reference signal using equations (5.3),(5,7), and (5.9) or using RFSys.m in Appendix B, we obtain:

$$T_{er} = 1 - 0.129q - 1.378q^2 - 0.435q^3 + 0.999q^4 + 0.109q^5. \text{-----(6.5)}$$

$$T_{ur} = 1.49 - 7.309q + 1.436q^2 - 1.41q^3 + 6.925q^4 - 1.357q^5. \text{-----(6.6)}$$

$$T_{yr} = 0.129q + 1.378q^2 + 0.435q^3 - 0.999q^4 - 0.109q^5. \text{-----(6.7)}$$

Note that: all Poles of transfer functions (6.5), (6.6) and (6.7) are at origin, due to the ripple free deadbeat property. Where all of them are polynomials with order equal 5, meaning that the system will settle down after 5 sampling times as shown in Fig. (6.2)

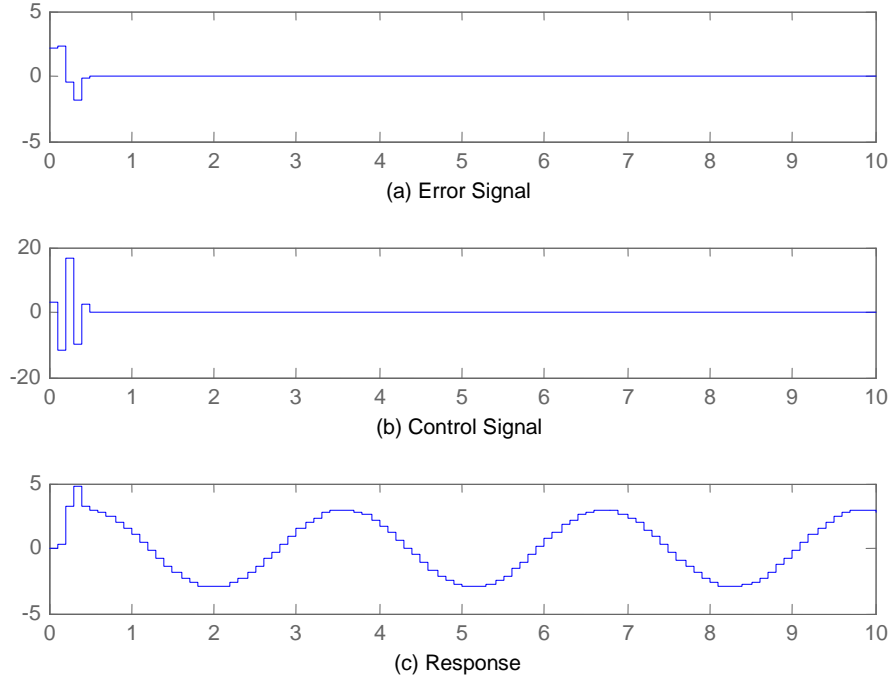


Figure (6.2): Time response of minimum order solutions

(a) Error signal (b) Control Signal (c) Response

Fig. (6.2) shows error, control, and output signals where all of them are polynomial with order equal 5 and since there exist time delay equal 2 sampling times, they have settled after 7 sampling times with overshoot = 61.53 %, settling time = 0.7 sec, $\|U\|_{\infty} = 16.6125$, $\|E\|_{\infty} = 2.2468$, and $\|E\|_2 = 3.6425$

- Decreasing control signal by vector which was produced by Paz using MATLAB built-in function 'qp' [11]. We obtain:

$$V_1 = [0.3082 \quad 0.7534 \quad 1.0847]$$

Computing the transfer functions from output, control, and error signals versus reference signal using equations (5.3), (5.7), and (5.9) or using RFsys.m in Appendix B, we obtain:

$$T_{er} = 1 - 0.035q - 0.424q^2 - 0.683q^3 - 0.379q^4 + 0.062q^5 + 0.45q^6 + 0.332q^7 + 0.03q^8 \dots \text{-----} (6.8)$$

$$T_{ue} = 0.405 - 1.425q + 1.645q^2 - 0.652q^3 + 0.155q^4 + 0.241q^5 - 1.225q^6 + 1.233q^7 - 0.376q^8 \dots \text{---} (6.9)$$

$$T_{yr} = 0.035q + 0.424q^2 + 0.683q^3 + 0.379q^4 - 0.062q^5 - 0.45q^6 - 0.332q^7 - 0.03q^8 \dots \text{-----} (6.10)$$

Note that: all Poles of transfer functions (6.8), (6.9) and (6.10) are at origin, due to the ripple free deadbeat property. Where all of them are polynomials, meaning that the system will settle down after 8 sampling times as shown in Fig. (6.3)

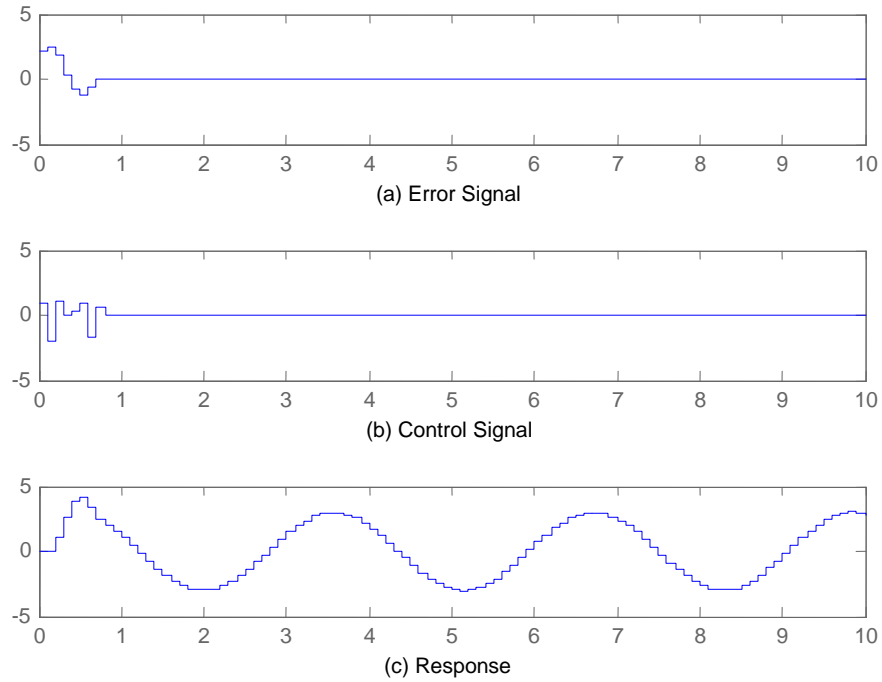


Figure (6.3): Time response of vector, which was produced by Paz.

(a) Error signal (b) Control Signal (c) Response

Fig. (6.3) shows error, control, and output signals where all of them are polynomial with order equal 8 and since there exist time delay equal 2 sampling times, they have settled after 10 sampling times with overshoot = 40.05%, settling time = 1 sec, $\|U\|_{\infty} = 2.0443$, $\|E\|_{\infty} = 2.4491$, and $\|E\|_2 = 4.0466$.

- Decreasing control signal by vector which was produced by first approach using equations (5.18), (5.19), and (5.20). We obtain:

$$V1 = [0.3097 \ 0.7339 \ 1.0674].$$

Computing the transfer functions from output, control, and error signals versus reference signal using equations (5.3), (5.7), and (5.9) or using RFSys.m in Appendix B, we obtain:

$$T_{er} = 1 - 0.037q - 0.44q^2 - 0.685q^3 - 0.345q^4 + 0.065q^5 + 0.431q^6 + 0.331q^7 + 0.03q^8 \text{-----(6.11)}$$

$$T_{ur} = 0.423 - 1.512q + 1.796q^2 - 0.71q^3 + 0.48q^4 - 1.366q^5 + 1.268q^6 - 0.378q^7 \text{-----(6.12)}$$

$$T_{yr} = 0.037q + 0.44q^2 + 0.685q^3 + 0.345q^4 - 0.065q^5 - 0.431q^6 - 0.331q^7 - 0.03q^8 \text{-----(6.13)}$$

Note that: all Poles of transfer functions (6.11), (6.12) and (6.13) are at origin, due to the ripple free deadbeat property. Where all of them are polynomials, meaning that the system will settle down after f sampling times as shown in Fig. (6.4)

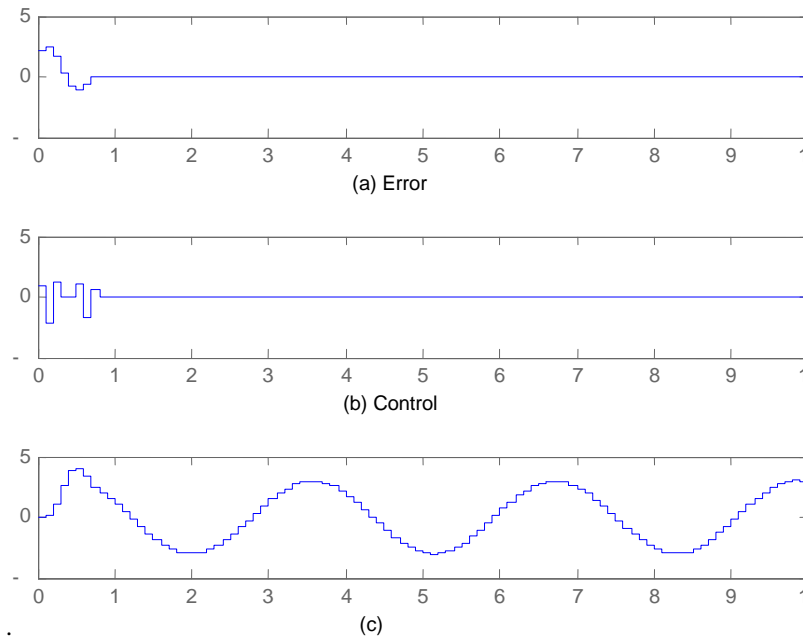


Figure (6.4): Time response of vector, which was produced by first approach.

(a) Error signal (b) Control Signal (c) Response

Fig. (6.4) shows error, control, and output signals where all of them are polynomial with order equal 8 and since there exist time delay equal 2 sampling times, they have settled after 10 sampling times with overshoot = 38.90%, settling time = 1 sec, $\|U\|_{\infty} = 2.1861$, $\|E\|_{\infty} = 2.4448$, and $\|E\|_2 = 4.0094$.

- Decreasing control signal by using second approach using equations (5.25) to (5.35): We obtain:

$$\text{bestV1}=[0.3082 \ 0.6752 \ 1.0847].$$

Computing the transfer functions from output, control, and error signals versus reference signal using equations (5.3),(5,7), and (5.9) or using RFsys.m in Appendix B, we obtain:

$$\text{Ter}=1-0.035q-0.43q^2-0.747q^3-0.314q^4+0.135q^5+0.385q^6+0.324q^7+0.03q^8 \text{-----}(6.14)$$

$$\text{Tur}=0.405-1.347q+1.166q^2+0.575q^3-1.532q^4+1.552q^5-1.772q^6+1.329q^7-0.376q^8 \text{.--}(6.15)$$

$$\text{Tyr}=0.035q+0.43q^2+0.747q^3+0.314q^4-0.135q^5-0.385q^6-0.324q^7-0.03q^8 \text{-----}(6.16)$$

Note that: all Poles of transfer functions (6.14), (6.15) and (6.16) are at origin, due to the ripple free deadbeat property. Where all of them are polynomials, meaning that the system will settle down after f sampling times as shown in Fig. (6.5)

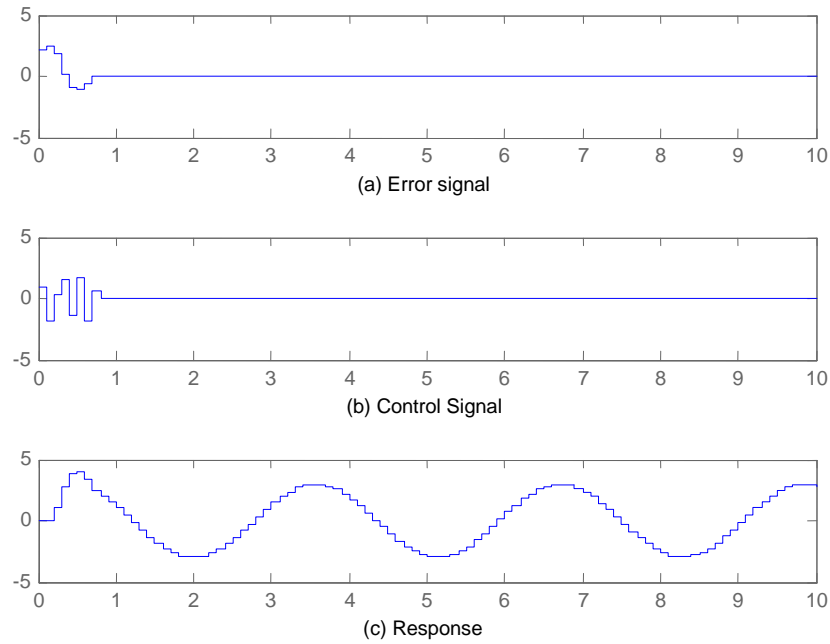


Figure (6.5): Time response of vector, which was produced by second approach.

(a) Error signal (b) Control Signal (c) Response

Fig. (6.5) shows error, control, and output signals where all of them are polynomial with order equal 8 and since there exist time delay equal 2 sampling times, they have settled after 10 sampling times with overshoot = 35.79%, settling time = 1 sec, $\|U\|_{\infty} = 1.8764$, $\|E\|_{\infty} = 2.4491$, and $\|E\|_2 = 4.0026$.

Results of step responses to the system using Paz vector, First approach, and second approach are summarized in Table (6.1)

Table(6.1): Comparison between Paz vector, first approach, second approach

Vector	Norm(u,inf)	Norm(E,inf)	Norm(E,2)	Overshoot	Settling time
Paz vector	2.0443	2.4491	4.0466	40.05%	1 sec
First app.	2.1861	2.4448	4.0094	38.90%	1 sec
Second app.	1.8764	2.4491	4.0026	35.79%	1 sec

For the same settling time, The controller design based on Second approach have minimum infinity norm of control signal, minimum 2-norm, and minimum overshoot.

Note that: Due to time delay which equal two sampling time, all systems have settled down after 7 sec.

6.2. Magnetic ball levitation CE152 (nonlinear system)

In this section, second order approximation of magnetic ball levitation CE152 will be computed after tacking two parameters –rising time and overshoot. After that, parameters of feedback linearization and second order approximation will be modified to make the coefficient of second order approximation simple and just integer numbers.

following the procedures covered in Chapter 4, Linearization for the magnetic ball levitation is obtained using feedback linearization as shown in Fig. (6.6)

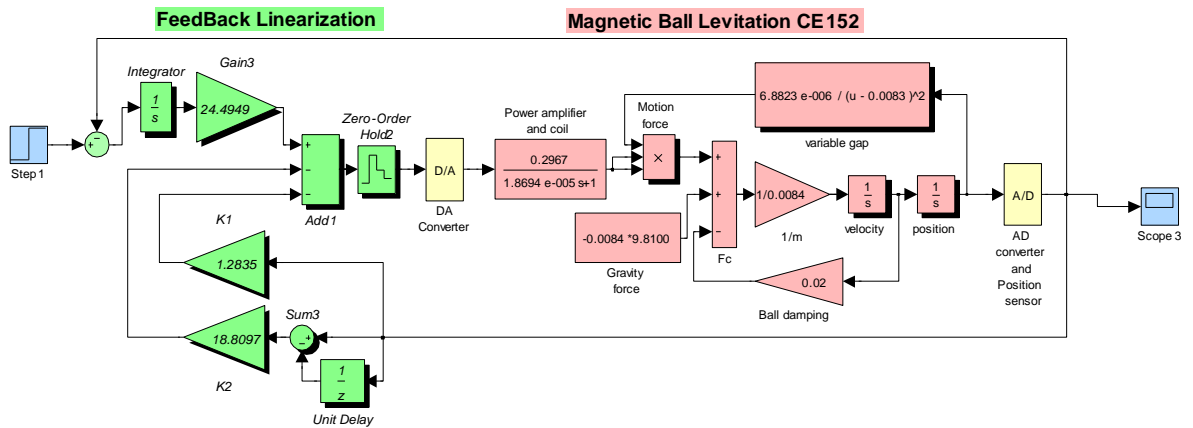


Figure (6.6): Model of maglev CE152 with FBL.

Finding the step response of magnetic Ball levitation with feedback linearization,

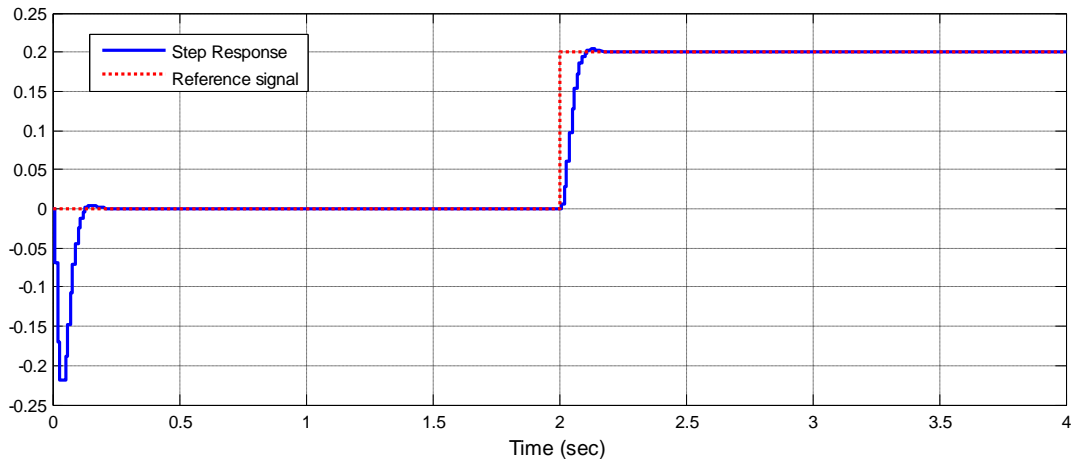


Figure (6.7): Step response of CE152 with FBL.

Overshoot = 1.65%, rising time = 0.07 sec, settling time = 0.1 sec, undershoot = 10%

Fig. (6.7) will be zoomed in to find the exact rising time and overshoot as shown in Fig. (6.8)

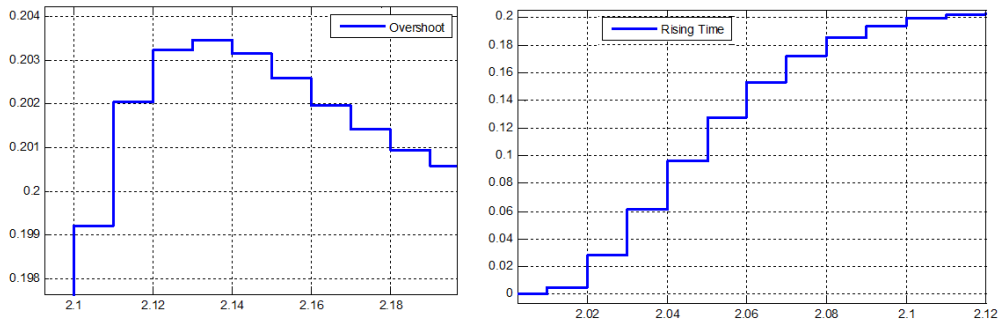


Figure (6.8): Zoom on to find overshoot and rising time

(a) overshoot

(b) Rising time

Obtaining the second order prototype based on the given specifications as covered in chapter two will yield:

$$P(s) = \frac{1583.187}{s^2 + 63.1933s + 1583.187} \text{-----(6.17)}$$

- Step response of linearized magnetic ball levitation and its approximated model to ensure that they are approximately the same

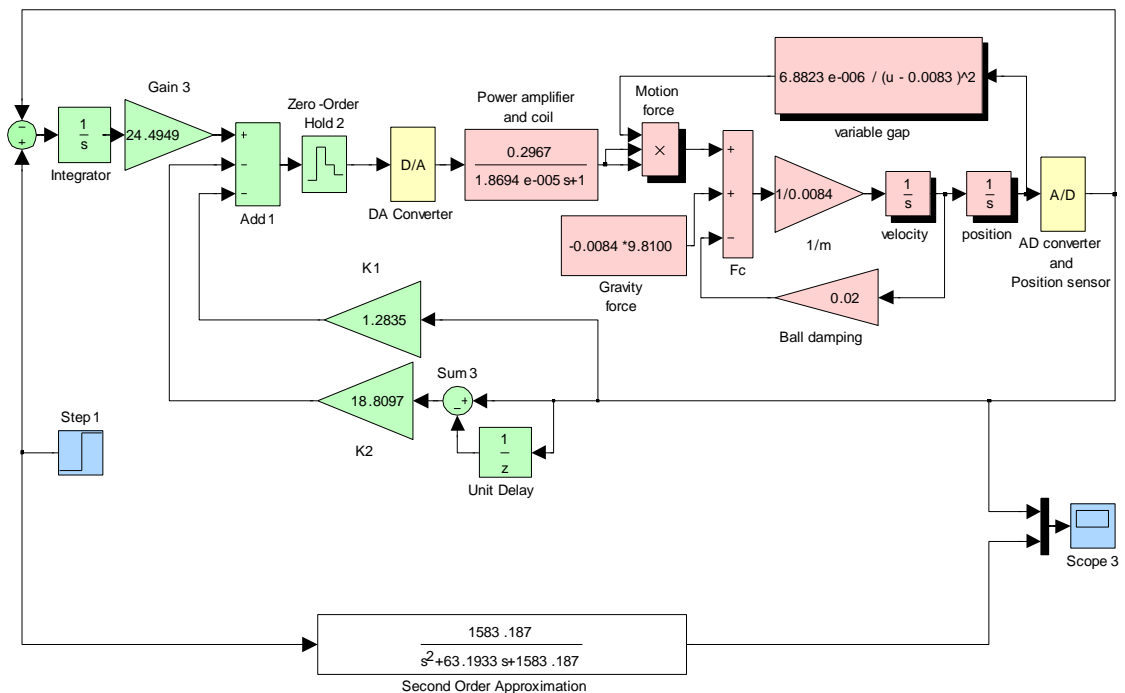


Figure (6.9): CE152 with FBL and its second order approximation.

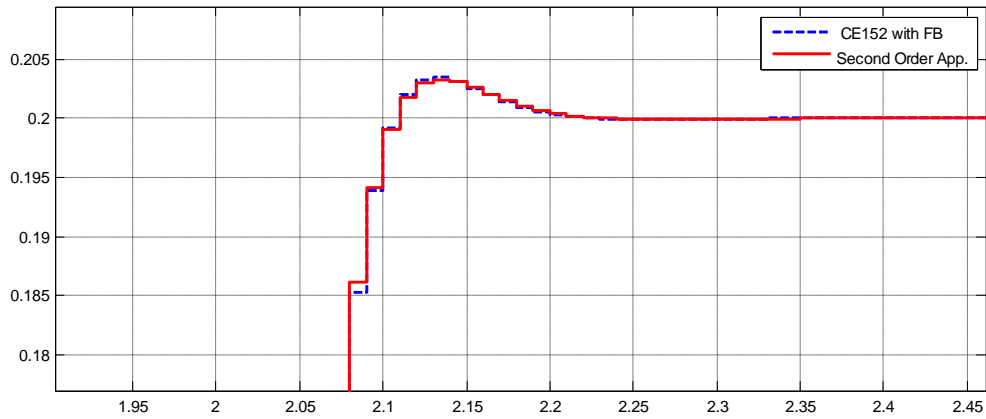


Figure (6.10): Step responses of CE152 with FBL and approximated model.

Fig. (6.10) shows that: the step response of approximated model and linearized model are the same.

The previous system has a very large overshoot, and deadbeat controller will contribute overshoot; moreover, it is not easy to deadbeat nonlinear system with very large overshoot; thus, another feedback linearization will be found as shown in Fig. (6.11)

Obtaining the second order prototype based on the given specifications as covered in chapter two will yield:

$$P(s) = \frac{1000}{s^2 + 80s + 1000} \text{-----(6.18)}$$

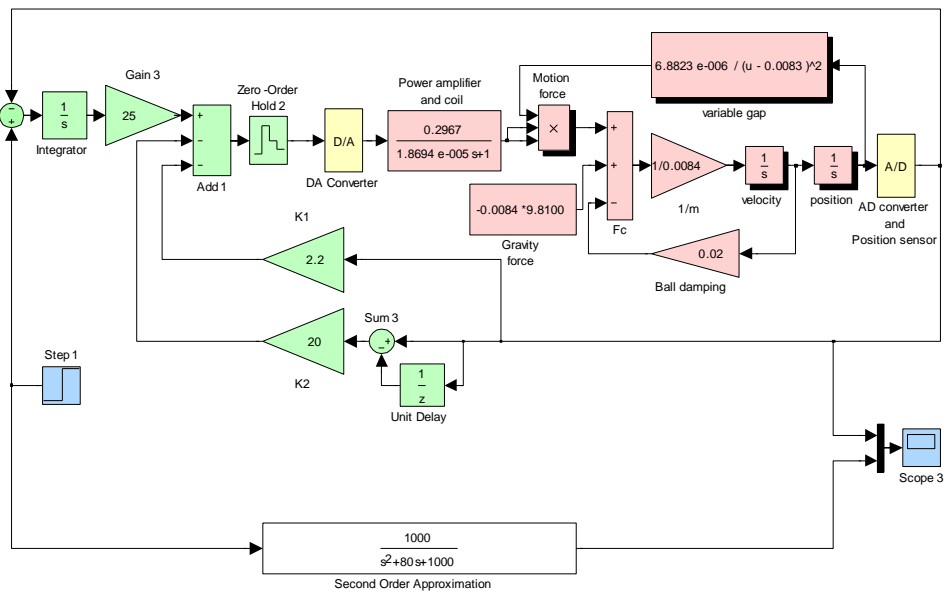


Figure (6.11): CE152 with third FBL and its second order approximation.

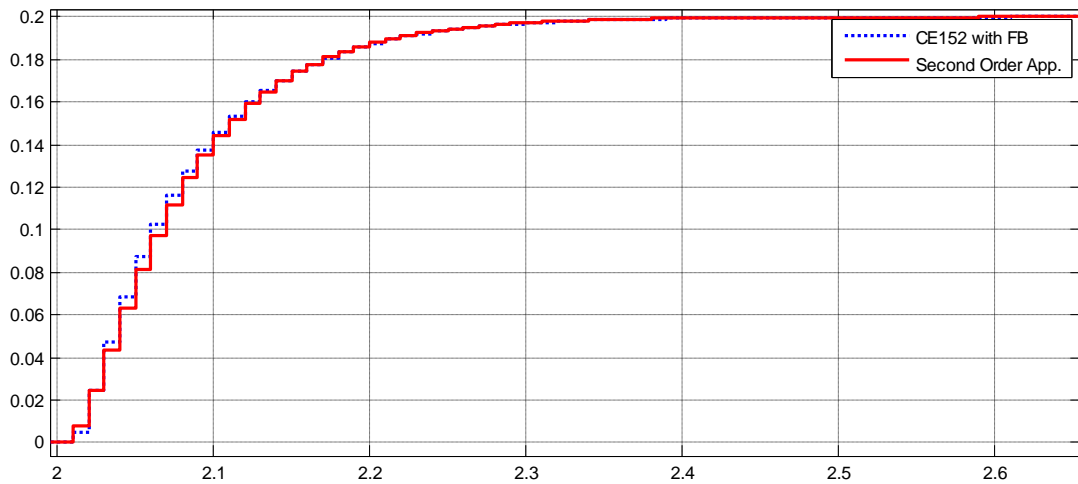


Figure (6.12): Step responses of CE152 with third FBL and approximated model.

Fig. (6.12) shows that the step response of linearized model and approximated model are the same and they do not have overshoot; thus, the plant $P(s) = \frac{1000}{s^2 + 80s + 1000}$ will be used to find deadbeat controller for linearized model of magnetic ball levitation.

If the first approximated model shown in Fig. (6.6) is used to find the deadbeat controller for magnetic ball levitation, the response will have a very large overshoot in real-time application; moreover, it is not easy to control the nonlinear system with very large overshoot.

6.3. Deadbeat controller for magnetic ball levitation with feedback linearization

In order to show the multi-rate system we select a sampling time for feedback linearization 1 ms, and another sampling time for deadbeat controller 10 ms.

Let us consider the system $\frac{1000}{s^2 + 80s + 1000}$, which has obtained in the previous section. We wish to track the sinusoid $r(t) = 3 \sin(2t)$.

Computing the minimum order solutions to the Diophantine equations systematically, following the procedure mentioned in section (5.5) we obtain

$$N_1 = 35.64 - 21q \text{-----(6.19)}$$

$$Q_1 = 1 + 0.622q \text{-----(6.20)}$$

$$N_2 = 21.46 - 8.367q \text{-----(6.21)}$$

$$D_c = 1 + 0.5515q \text{-----(6.22)}$$

N_1 , N_2 , and D_c will be applied to the magnetic ball levitation with feedback linearization as shown in Fig. (6.13)

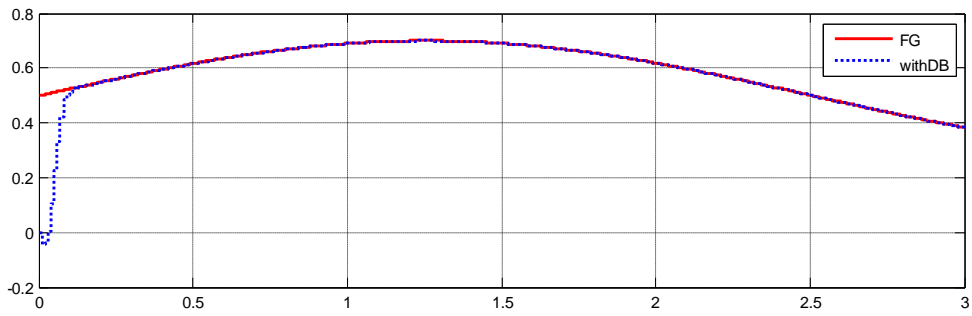


Figure (6.14): Sinusoidal response of CE152 with FBL and with deadbeat controller.

Fig.(6.15) shows the response of magnetic ball levitation with feedback linearization and with deadbeat controller, you can see that the response (dotted line) followed the reference signal (soled line) after finite time with steady state error equal zero.

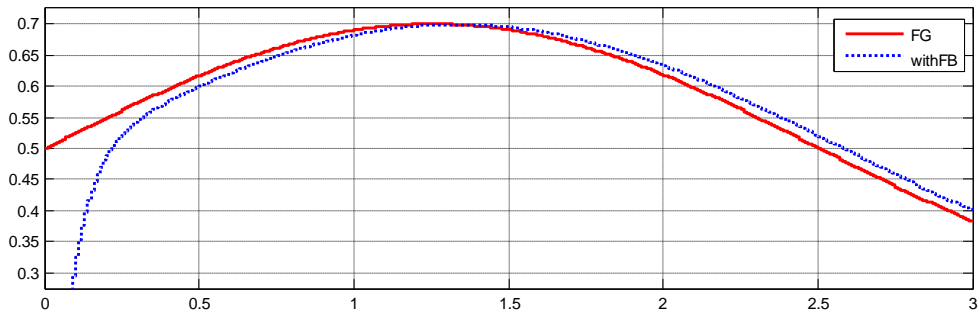


Figure (6.15): Sinusoidal response of CE152 with FBL.

Fig.(6.16) shows the response of magnetic ball levitation with feedback linearization without deadbeat controller, you can see that the response (dotted line followed the reference signal (soled line) with apparent steady state error.

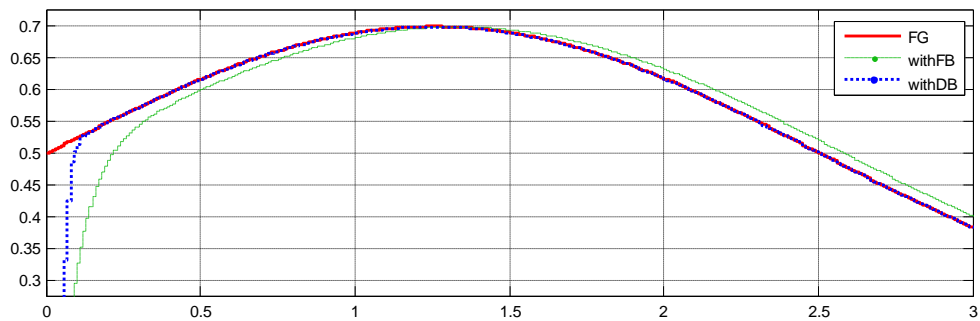


Figure (6.16): Sinusoidal response of CE152 with FBL and with/without deadbeat controller

Fig.(6.17) shows the response of magnetic ball levitation with feedback linearization without deadbeat controller and response of magnetic ball levitation with feedback linearization with deadbeat controller.

You can see that,

- the dotted line (Response of system with deadbeat Controller) can follow the solid line (Reference signal) better than the dashed line (Response of system without deadbeat Controller)
- Settling time when using deadbeat controller is smaller than settling time without using deadbeat controller.

6.4. Deadbeat controller for real-time maglev CE152

Second order approximation of real-time magnetic ball levitation CE 152 with feedback linearization shown in Fig. (6.18) will be found based on specifications of step response of real-time linearized model as covered in chapter two will yield:

$$P(s) = \frac{1237.3}{s^2 + 55s + 1237.3} \text{-----(6.23)}$$

Computing the minimum order solutions to the Diophantine equations systematically, following the procedure mentioned in section (5.5) we obtain

$$N1=26.05 - 15.44 q \text{-----(6.24)}$$

$$Q1=1 + 0.6605 q \text{-----(6.25)}$$

$$N2=17.13 - 8.117 q \text{-----(6.26)}$$

$$Dc=1 + 0.602 q \text{-----(6.27)}$$

After modifications the previous parameters will be

$$N1=26.1502 - 15.4985q \text{-----(6.28)}$$

$$N2=5(3.4460 - 1.6353q)=17.23 - 8.1765q \text{-----(6.29)}$$

$$Dc=1+0.60266q \text{-----(6.30)}$$

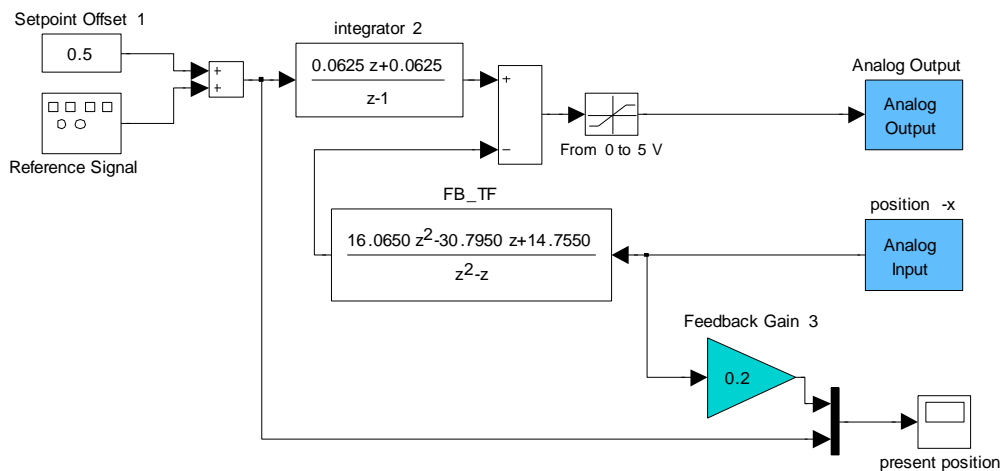


Figure (6.17): FBL of real-time maglev CE152.

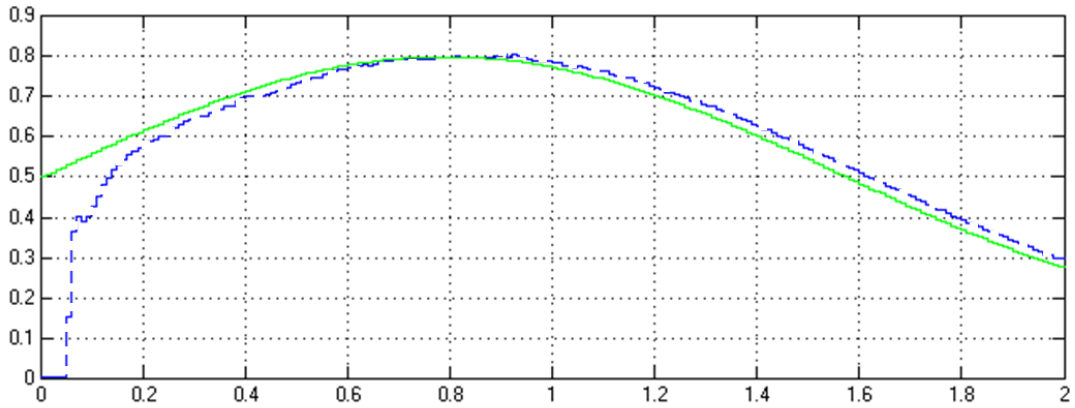


Figure (6.18): Response of real-time CE152 with FBL

Fig. (6.19) shows that: the response of real-time magnetic ball levitation without deadbeat controller (dashed line) can follow reference signal (solid line) with large steady state error

Applying N_1 , N_2 , and D_c to the real-time magnetic ball levitation CE152 as shown in Fig. (6.20)

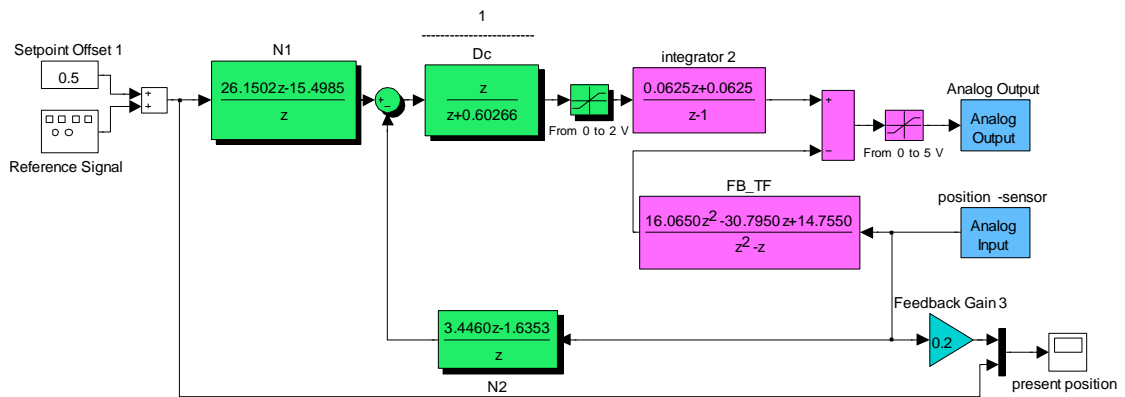


Figure (6.19): Block diagram of deadbeat controller and FBL

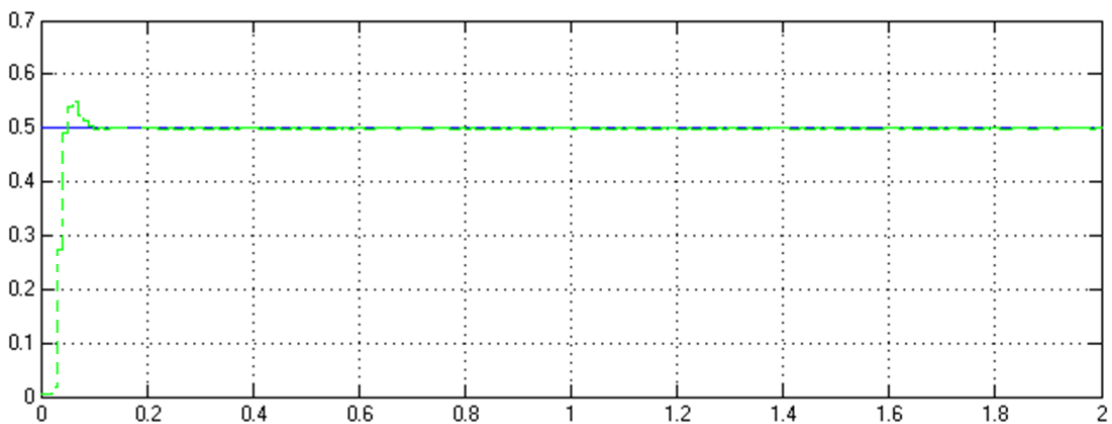


Figure (6.20): Step response of deadbeat controller with FBL for CE152

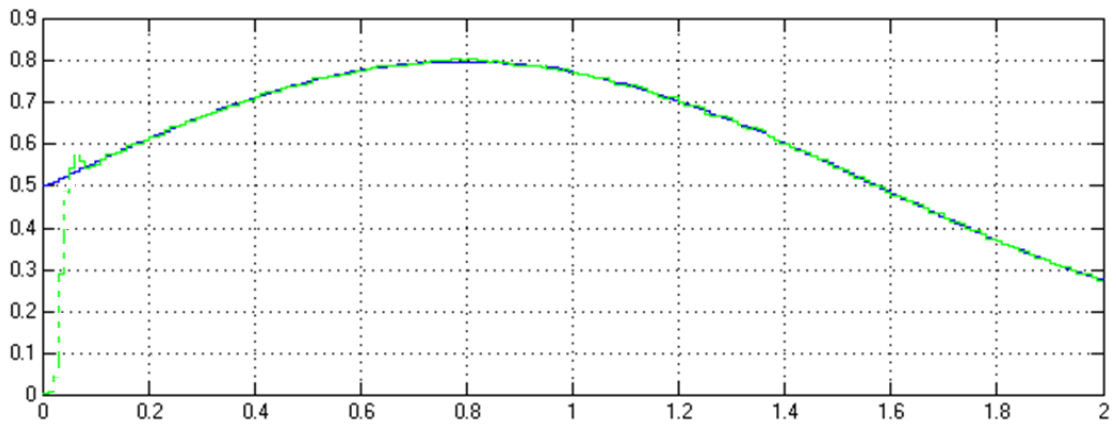


Figure (6.21): Sinusoidal response of deadbeat controller with FBL for RT CE152

Fig. (6.21) and (6.22) shows the response of real-time magnetic ball levitation CE152 with feedback linearization and with deadbeat controller, you can see that the response (dashed line) followed the reference signal (soled line) after finite time with steady state error equal zero.

6.5. VHDL code of deadbeat controller for real-time magnetic ball levitation CE152

In this section, VHDL code of deadbeat controller for real-time magnetic ball levitation will be simulated and compared with original blocks which has already deadbeat the system, and the approximated model which take into consideration the quantization error, number of input bits, and number of output bits.

DSP design tool from Xilinx will be used since this tool enables the use of the MathWorks SIMULINK environment for FPGA design.

Using DSP design tool to generate VHDL code is not a good technique for multi blocks model; since, large VHDL code will be generated. Therefore, VHDL code will be written for deadbeat controller [28], compiled using ISE Design Suite 10.1, and simulated using Xilinx Black Box to upload the VHDL of deadbeat controller for magnetic ball levitation with feedback linearization.

6.6. The Xilinx DSP Block Set

Over 90 DSP building blocks are provided in the Xilinx DSP blockset for SIMULINK. These blocks include the common DSP building blocks such as adders, multipliers and registers. In addition, the DSP tool included a set of complex DSP building blocks such as forward error correction blocks, FFTs, filters and memories [29]. Four blocks will be used in my project; these blocks will be described briefly:

- **Gateway in and Gateway out**

“Gateway In” and “Gateway Out” define the boundary of the FPGA from the SIMULINK simulation model. The Gateway In block converts the floating-point input to fixed-point. The Gateway out converts the FPGA outputs back to double precision block. Double-click on the block to bring up the properties editor.

Properties for gateway in:

Signed 2’s compliment,
Sample period = .001,
Number of bits = 8,
Binary point =0

- **Black box:**

Black box allows VHDL code to be imported into SIMULINK and co-simulated with either ModelSim or Xilinx ISE Simulator. VHDL code of deadbeat controller will be imported to this block and simulated.

Properties of Black Box:

Simulation mode: ISE simulator

- **System generator**

Once the design is completed, hardware implementation files can be generated using the Generate button available on the System Generator properties editor and this block can be used to simulate the already written hardware implementation file.

Properties of system generator:

Compilation: HDL netlist
Part: SPARTAN3A AND SPARTAN3AN XC3s700an -4 fg484
Hardware description language: HDL
FPGA clock period (ns): 20
Clock pin location: E12
SIMULINK system Period(sec): .001

6.7. Obstacles and solutions

Many obstacles have been encountered and solved during writing and simulating VHDL code. Some of these obstacles will be discussed here:

Any names for input and output ports can be used except the reserved names for VHDL commands. Thus, I chose ‘Clock’, ‘CLK50’, and many other names for FPGA clock. It’s ok when you synthesize this code using ISE Design Suite, and when you simulate the VHDL code using ModelSim or Xilinx ISE Simulator, but it’s not ok when you use DSP design tool on SIMULINK environment, since, you can’t build a square wave for clock with T_{high} or T_{low} smaller than sampling time. After studying some VHDL code that simulated using DSP design tool on SIMULINK environment, I found that you should define two ports ‘clk’ for clock and ‘ce’ for counter enable, then these

ports will be hidden and the DSP design tool will take the clock period from FPGA clock period property in Properties of system generator.

Any computational process will be registered in signal; Thus, number of bits for this signal should be determined depending on number of bits for coefficients, number of multiplication and summation terms, 1 bit for negative and positive sign, and one margin bit as mentioned in section (5.9). Just summation of equally number of bits in VHDL code can be done “ $z(n \text{ bits}) = x(n \text{ bits}) + y(n \text{ bits})$ ”, the result number of bits for multiplying terms equal summation of number of bits for each term “ $x(n_1 \text{ bits}) * y(n_2 \text{ bits}) = z(n_3 \text{ bits}) = w((n_1 + n_2) \text{ bits})$ ”.

6.8. Designing Steps:

Simulating VHDL code for constant input signal and comparing the results appeared on Display screens with original deadbeat controller will be done, then manual switch will be used to convert from constant input to sinusoid form with offset then simulating and comparing the results on different scopes will be done.

- VHDL code for deadbeat controller will be written using equations (5.77), (5.79), (5.81), and (5.83).
- VHDL code will be synthesized using ISE Design Suite .
- DSP design tool from Xilinx will be used to build the required block as shown in Fig. (6.26) to simulate VHDL code and compare it with original SIMULINK model for deadbeat controller.
- If the simulated results are not ok, the first two steps will be reused.

6.9. ADC

Analog to digital converter as shown in Fig. (6.23) will be used to convert unlimited number of levels from input signal to limited number of levels. Eight bits will be used; one of them for negative and positive sign and seven bits for output level.

Peak value of input signal =5 Volts; thus, saturation will be used to eliminate any input larger than 5 volts.

A/D output gain will be used to convert the peak of input signal from 5 to 127

Rounding function will be used to convert the infinity number of levels to limited number of levels by removing the floating point.

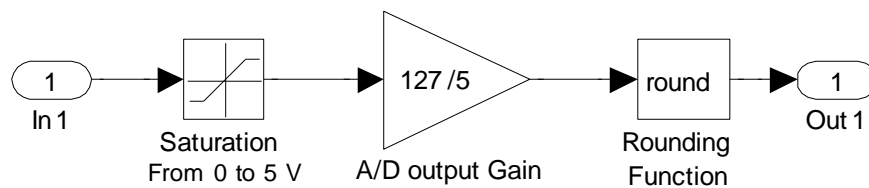


Figure (6.22): Block diagram of analog to digital converter

6.10. Approximated model of deadbeat controller

Comparing original deadbeat controller shown in Fig. (6.24) which uses unlimited number of levels for input signal and real number for coefficients with VHDL code that uses limited number of levels for input signal and just natural number for coefficients is not efficient. Thus, an approximated model for deadbeat controller will be made as shown in Fig. (6.25) using analog to digital converter shown in Fig. (6.23) and deadbeat controller shown in Fig. (5.11)

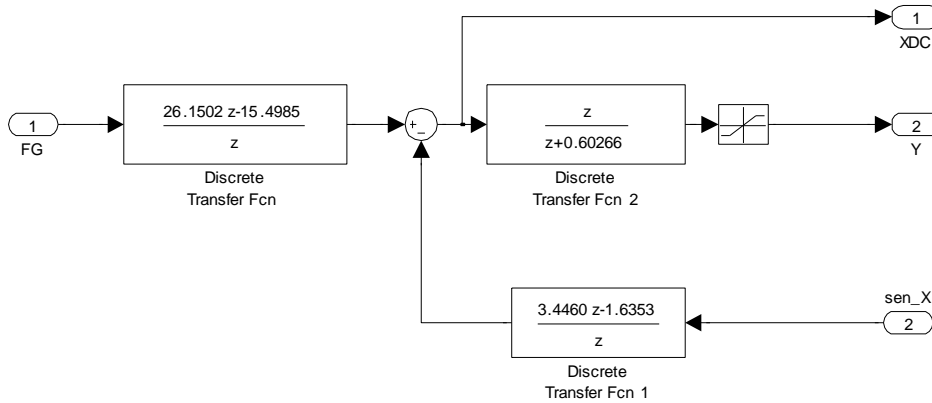


Figure (6.23): Block diagram of original deadbeat controller

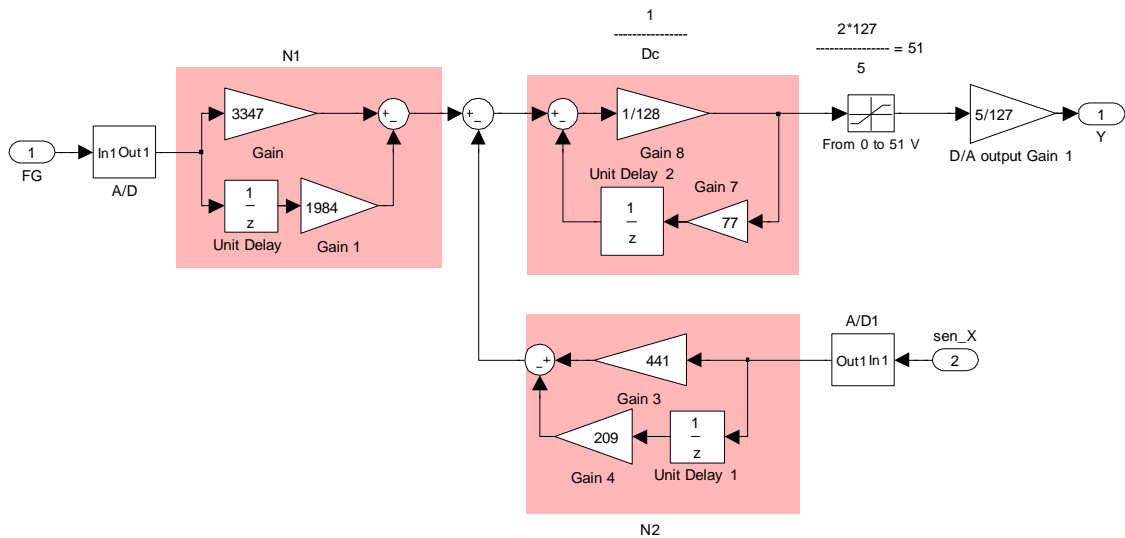


Figure (6.24): Block diagram of approximated model deadbeat controller

6.11. Simulation of VHDL code

Fig. (6.26) shows black box that imported VHDL code for deadbeat controller called “VHDL-Code Dead-Beat Controller”, subsystem contains deadbeat controller shown in Fig. (6.24) which called “DB”, and subsystem contains approximated model of deadbeat controller shown in Fig. (6.25)

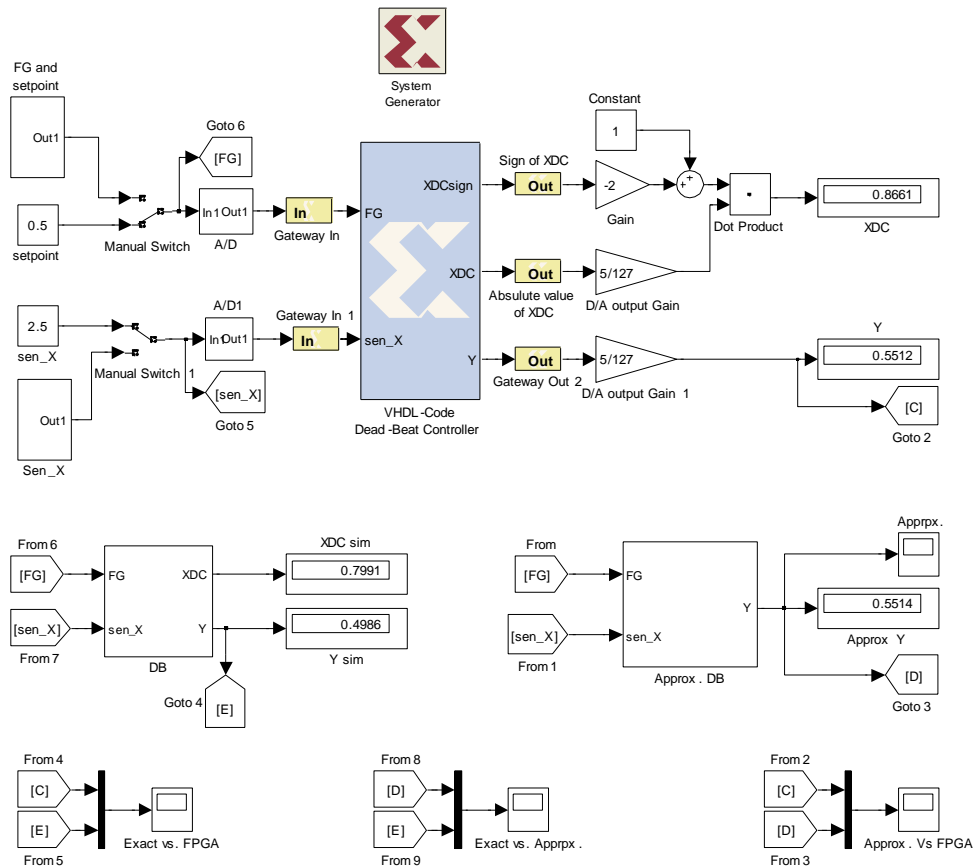


Figure (6.25): Block diagram of subsystem Approx. DB , DB, Black box for VHDL

Fig. (6.26) shows that the output of VHDL code almost equal the output of approximated model; thus, the VHDL code programmed successfully.

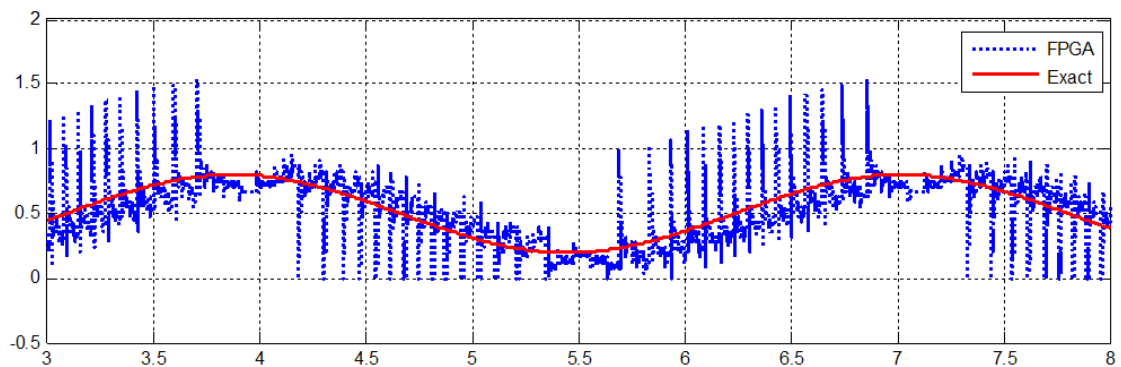


Figure (6.26): Time response of VHDL code versus original controller

Fig. (6.27) shows the response of VHDL code for deadbeat controller (dotted line) and step response of original deadbeat controller (solid line).

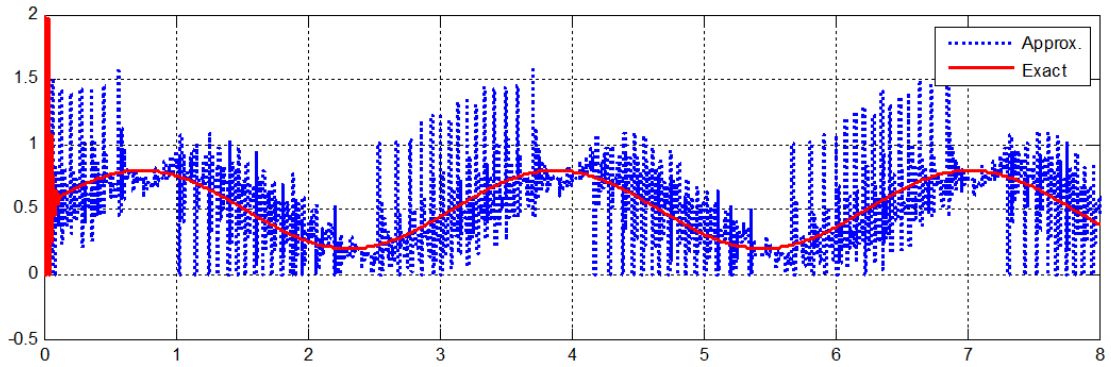


Figure (6.27): Time response original controller versus its approximated model

Fig. (6.28) shows the response of an approximated model of deadbeat controller (dotted line) and step response of original deadbeat controller (solid line).

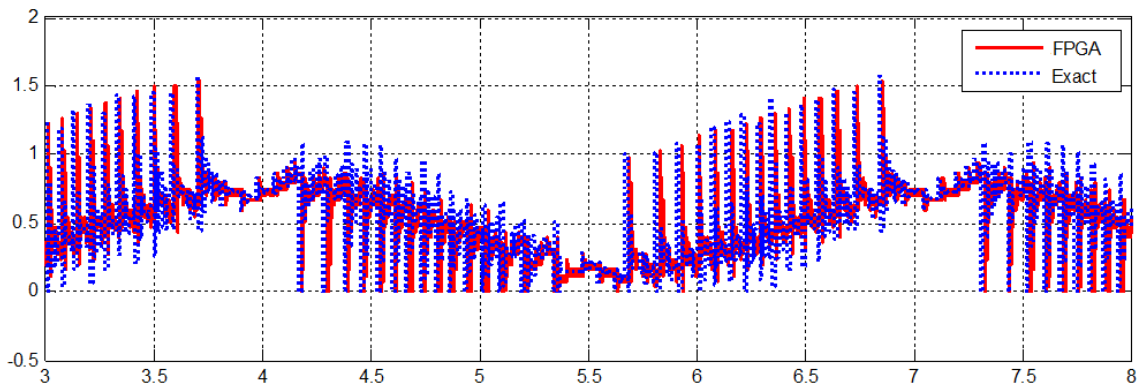


Figure (6.28): Time response of VHDL code versus approximated model.

Fig. (6.29) shows the response of an approximated model of deadbeat controller (dotted line) and step response of VHDL code for deadbeat controller (solid line).

Note:

- 1) The response of VHDL code exactly equals the response of approximated model, which means: deadbeat Controller is programmed successfully.
- 2) The response of VHDL and approximated model are not exactly equal the response of original model of deadbeat Controller, since the function of deadbeat controller is to follow the signal as fast as possible which make an overshoot that directly proportional to the difference between present and wanted values which means it's directly proportional to the quantization error, So to decrease this chattering we need to use extra number of bits for input signal.

CHAPTER 7 CONCLUSION

A new design methodology for ripple-free deadbeat control of nonlinear systems in discrete-time was proposed. This new control methodology combined two ripple-free deadbeat controllers to control nonlinear systems. The new control methodology guaranteed the robustness and handled multi-rate systems. The results shown that, the response of nonlinear system tracked the reference signal with zero steady state after very small rising time. This thesis minimized the settling time by using second order linear model to approximate the nonlinear system.

The settling time depends on the length of three polynomials N_1 , N_2 , and D_c . The length of these polynomials depends on the degree of numerator of plant, denominator of plant, and denominator of reference signal; thus, when the order of plant for linear systems and order of linearized plant for nonlinear systems greater than two, the second order approximation that was applied depended on the two dominant poles. This thesis proposed a hybrid two degree of freedom controller for the nonlinear optimization problem addressing performance and robustness specifications, utilizing the parameters of Diophantine equation to build a robust multi-rate ripple-free deadbeat control. A combination between the concept of multi-rate and robust single was proposed.

The proposed controller was applied using SIMULINK model of magnetic ball levitation CE 152 as a case study for nonlinear systems, simulation results shown that the controller performed fine with simulated plant but gave harmonic signal.

The proposed controller was also applied on real-time magnetic ball levitation CE 152 using real-time toolbox in MATLAB environment, the controller worked fine with real-time plant but still gave harmonic signal.

Multi-rate ripple-free deadbeat control problem was evaluated by two steps: full state feedback which depended on (Time-domain approach) and Diophantine equations which depended on (Polynomial approach). State and output feedbacks were used to linearize and stabilize the nonlinear system and to make the response of the nonlinear system closely track the reference signal. The Diophantine equations which depend on internal model principle were utilized and applied to the linearized and stabilized nonlinear system to make the response of the system exactly equal the reference signal and provide some robustness.

VHDL code for deadbeat control for magnetic ball levitation with feedback linearization was written and simulated using Xilinx toolbox and compared with the approximated model of SIMULINK original blocks.

Simulation and real-time results showed that the output signal exactly tracked the input sinusoidal signal in short settling time. The time domain specification for the output signal, control signal, and error signal were computed and satisfied the requirement and constraints. A time delay was also presented with simulation and was solved by using deadbeat controller based on solving Diophantine equation parameters.

Future research can be used to deadbeat the nonlinear system using another linearization technique. Moreover, the effect of the noise and output disturbances on the system can be studied; the effect of changing working points, changing the sampling time, and changing the frequency of input signal can be also studied.

REFERENCES

- [1] Hyo-Sung Ahn, Yang Quan Chen, and Kevin L. Moore (2007), “Iterative Learning Control: Brief Survey and Categorization,” IEEE Systems, Man, and Cybernetics Society, VOL. 37, NO. 6.
- [2] Finn Orfano, Why are Control Systems Used?, Edited & published by Rebecca Scudder on May 25, 2010, accessed on May 18, 2011, online: <http://www.brighthub.com/engineering/electrical/articles/72133.aspx>
- [3] Wikipedia, the free encyclopedia (modified on 16 April 2011 at 15:24), Digital control, accessed on May 18, 2011, online: http://en.wikipedia.org/wiki/Digital_control.
- [4] Dragan Nešić, (1996) “Dead-Beat Control for Polynomial Systems,” PhD thesis, Australian National University.
- [5] Dictionary and Encyclopedia Directory, Control system – Definition, accessed on May 18, 2011, online: http://www.wordiq.com/definition/Control_system
- [6] B. C. Kou and F. Golnaraghi (2003), *Automation Control Systems*, Wiley, Hoboken.
- [7] Tou J.T. (1959), *Digital and Sampled-Data Control Systems*. McGraw-Hill, New York, NY.
- [8] H. Elaydi and R. A. Paz (1998), “Optimal Ripple-Free Deadbeat Controllers for Systems with Time Delays,” Proceedings of the American Control Conference, Philadelphia, USA.
- [9] L. Jetto and S. Longhi (2002), “Parameterized solution of the deadbeat ripple-free control problem for multirate esampled-data systems,” Proceedings of the 38th IEEE Conference on Decision & Control, Phoenix, Arizona.

- [10] H. Ito, (2001), "Improving performance of deadbeat servomechanism by means of multirate input control," Thesis, Kyushu Institute of Technology, Japan, Accessed on Nov. 1, 2008, online: <http://search2.libi.kyutech.ac.jp/TR/pdf/csse14.pdf>
- [11] R. A. Paz, (2006), "Ripple-free tracking with robustness," International Journal of control, Vol. 79, No. 6.
- [12] M. E. Salgado and D. A. Oyarzun (2007), "Two objective optimal multivariable ripplefree deadbeat control," University Tecnica Federico Santa MariaValparaiso, Chile, Accessed on April 1, 2009, online: <http://www.hamilton.ie/d.o/docs/saloya07ijc.pdf>.
- [13] Fadi M. Albatsh (2009), Multirate Ripple-Free Deadbeat Control, Master thesis, Islamic University of Gaza.
- [14] Wikipedia, the free encyclopedia (modified on 16 May 2011 at 06:12), Linear systems, accessed on May 18, 2011, online: http://en.wikipedia.org/wiki/Linear_system
- [15] Wikibooks, open books for an open world, Control Systems/State-Space Equations, accessed on May 19, 2011, online: http://en.wikibooks.org/wiki/Control_Systems/State-Space_Equations
- [16] Wikipedia, the free encyclopedia (modified on 13 May 2011 at 11:31), State space (controls), accessed on May 19, 2011, online: [http://en.wikipedia.org/wiki/State_space_\(controls\)](http://en.wikipedia.org/wiki/State_space_(controls))
- [17] Wikipedia, the free encyclopedia(modified on 29 April 2011 at 21:11.), Nonlinear system, accessed on May 19, 2011, online: http://en.wikipedia.org/wiki/Nonlinear_system
- [18] J. K. Hedrick and A. Girard (2005), Control of Nonlinear Dynamic Systems: Theory and Applications, Chapter 8, and accessed on May 19, 2011, online: http://www.me.berkeley.edu/ME237/8_feedback_lin.pdf
- [19] H. Khalil, *Nonlinear Systems*, 3rd Edition , Prentice-Hall, Englewood Cliffs, NJ.
- [20] Wikipedia, the free encyclopedia (modified on 13 May 2011 at 07:51), Scientific Techniques of Levitation, accessed on May 19, 2011, online: <http://en.wikipedia.org/wiki/Levitation>.

- [21] Marcio S. de Queiroz and Darren M. Dawson (1996), " Nonlinear Control of Active Magnetic Bearings: A Backstepping Approach," IEEE transactions on control system technology, Vol. 4, NO 5.
- [22] HUMUSOFT, (1996-2008), CE 152 Magnetic Levitation Model, Technical Manual, Czech Republic.
- [23] F. Gazdoš , P. Dostál, and R. Pelikán (2009), "Polynomial approach to control system design for a magnetic levitation system," Cybernetic Letters: Informatics, Cybernetics, and Robotics, PP. 1-19,
- [24] Magnetic Levitation Model (CE 152), Advanced-models-laboratory, laboratories VSB - Technical University of Ostrava, accessed on May 19, 2011, online:
<http://www.352.vsb.cz/english/laboratories/advanced-models-laboratory/levitation/levitation.htm>
- [25] Khalid Abdelhafiz Ali, Mohammed Abdelati, Mohammed Hussein (Jan.2010), Modelling, Identification and Control of A "Magnetic Levitation CE152", Al-Aqsa University Journal (Natural Sciences Series), Vol.14, No.1, PP 42-68.
- [26] Wikipedia (modified on 28 April 2011 at 16:29.), the free encyclopedia, Feedback linearization, accessed on May 19, 2011, online:
http://en.wikipedia.org/wiki/Feedback_linearization
- [27] J F Coales (1957), "An introduction to the study of non-linear control systems," Journal of Scientific Instruments Vol. 34 No. 2.
- [28] Pong P. Chu (2008), *FPGA Prototyping By VHDL Examples*, Wiley-Interscience XILINX SPARTAN-3 version,
- [29] Xilinx (December 14, 2010), System Generator for DSP, UG639 (v 12.4) access on May 19, 2011, online:
http://www.xilinx.com/support/documentation/sw_manuals/xilinx12_4/sysgen_gs.pdf

Appendix A

VHDL code for deadbeat controller

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL; -- change unsigned to signed
----- Define input/output ports-----
-- The name of port for FPGA's internal clock should be 'clk' to be
used in simulation
-- You should define ce 'clock enable'.
entity DeadBeatXilinx is
    Port (clk      : in  STD_LOGIC;
          ce       : in  STD_LOGIC;
          FG       : in  STD_LOGIC_VECTOR (7 downto 0);
          sen_X    : in  STD_LOGIC_VECTOR (7 downto 0);
          XDCsign  : out STD_LOGIC;
          XDC      : out STD_LOGIC_VECTOR (7 downto 0);
          Y        : out STD_LOGIC_VECTOR (7 downto 0)
    );
end DeadBeatXilinx;
architecture Behavioral of DeadBeatXilinx is
-----Define signals-----
signal sen_X_buffer      : STD_LOGIC_VECTOR (7 downto 0) :=(others=>'0');
signal sen_Xl_buffer    : STD_LOGIC_VECTOR (7 downto 0) :=(others=>'0');
signal FG_buffer        : STD_LOGIC_VECTOR (7 downto 0) :=(others=>'0');
signal FG1_buffer       : STD_LOGIC_VECTOR (7 downto 0) :=(others=>'0');
signal Y_buffer         : STD_LOGIC_VECTOR (7 downto 0) :=(others=>'0');
signal counter          : STD_LOGIC_VECTOR (4 downto 0) :=(others=>'0');
signal counterCLK50     : STD_LOGIC_VECTOR (17 downto 0) :=(others=>'0');
signal XDC_buffer       : STD_LOGIC_VECTOR (20 downto 0) :=(others=>'0');
signal XDC_buffer2      : STD_LOGIC_VECTOR (8 downto 0) :=(others=>'0');
signal XDC_buffer3      : STD_LOGIC_VECTOR (20 downto 0) :=(others=>'0');
signal XDC_buffer_sign  : STD_LOGIC_VECTOR (8 downto 0) :=(others=>'0');
signal YDC_buffer       : STD_LOGIC_VECTOR (20 downto 0) :=(others=>'0');
signal YDCl_buffer      : STD_LOGIC_VECTOR (7 downto 0) :=(others=>'0');
-----Define constants-----
-- the binary number will be -ve when it's decimal larger than 5
-- 5*127*128/5=16256="01111111000000000000"
--
begin
----- Start Process-----
process (clk,ce,FG,sen_X)
----- Define constants-----
constant a3347 : STD_LOGIC_VECTOR (12 downto 0) :="0110100010011";
constant a1984 : STD_LOGIC_VECTOR (12 downto 0) :="00111111000000";
constant a441  : STD_LOGIC_VECTOR (12 downto 0) :="0000110111001";
constant a209  : STD_LOGIC_VECTOR (12 downto 0) :="0000011010001";
constant a128  : STD_LOGIC_VECTOR (12 downto 0) :="0000010000000";
constant a77   : STD_LOGIC_VECTOR (12 downto 0) :="0000001001101";
```

```

begin
  if (ce='1' and rising_edge(clk)) then
    -- CLK50 = 50 MHz=> to find 1ms sampling time we need
    -- 1m*50M=50,000 samples, 50000="001100001101010000"
    -- comment the following three lines for simulation and don't forget
    -- to comment last 'end if'
    -- counterCLK50<=counterCLK50+1;
    -- if counterCLK50 = "001100001101010000" then -- 50,000 cloks => 1
    ms
    -- counterCLK50<=(others=>'0');
    sen_X_buffer<=sen_X;
    FG_buffer<=FG;
    counter<=counter+1;
    if counter="01010" then
      counter<=(others=>'0');
      XDC_buffer<=a3347*FG_buffer-a1984*FG1_buffer-
      a441*sen_X_buffer+a209*sen_X1_buffer;
      XDC_buffer3<=XDC_buffer;
      sen_X1_buffer<=sen_X_buffer;
      FG1_buffer<=FG_buffer;
    end if;
    if XDC_buffer(14)='1' then
      XDC_buffer2(7 downto 0)<=XDC_buffer(14 downto 7);
      XDC_buffer_sign<="100000000"-XDC_buffer2;
      XDC<=XDC_buffer_sign(7 downto 0);
      XDCsign<='1';
      Y_buffer<=(others=>'0');
      YDC1_buffer<=(others=>'0');
    else
      XDC<=XDC_buffer(14 downto 7);
      XDCsign<='0';
      YDC_buffer<=XDC_buffer3-a77*YDC1_buffer;
      -- Apply Saturation ( 2*127/5=50.8 ~= 51 = "00110011")
      if XDC_buffer3(14)='1' then
        Y_buffer<=(others=>'0');
        YDC1_buffer<=(others=>'0');
      elsif YDC_buffer(14 downto 7)>"00110011" then
        Y_buffer<="00110011";
        YDC1_buffer<="00110011";
      else
        Y_buffer<=YDC_buffer(14 downto 7);
        YDC1_buffer<=YDC_buffer(14 downto 7);
      end if;
      if Y_buffer(7)='1' then
        Y<=(others=>'0');
      else
        Y<=Y_buffer;
      end if;
    end if;
  end if;
--end if;
end process;
end Behavioral;

```

Appendix B

MATLAB Code « Main code »

```

clear all; close all; clc
T=.0833; % Sampling time
td=.13; % Time delay
Amplitude=3; % Amplitude of Reference Signal
Omega=2; % radian freq. of Reference Signal
PhaseRad=pi/2; % phase shift in radian
Tfinal=10; % final simulation time
Tfinal=round(Tfinal/T)*T;
Mc=tf(100,[1 0 4]); % Model Filter
L=3; % Length of vector V1
Gc=tf(200,[1 -2 2],'ioDelay',td);
%-----
P_q=SQ(Mc*Gc,T); % plant
[nump denp]=tfdata(P_q,'v');
denp=removeZeros(denp);
OrderP=length(denp)-1;
%-----
% Reference signal
t=sym('t'); Rt=Amplitude*sin(Omega*t); Rc=laplace(Rt); Rc=sym2s(Rc);
R_q=SQ(Rc,T); % Reference signal in q-domain
%-----
% compute minimum order of N1Q1 and N2Dc then find N1, Q1, N2, and Dc
[N1Q1_Order N2Dc_Order]=Mini_Order(P_q,R_q);
[N1_q Q1_q N2_q Dc_q]=N1Q1N2DC(R_q,P_q,N1Q1_Order,N2Dc_Order);
N1_q_NC=N1_q;
N2_q_NC=N2_q;
Q1_q_NC=Q1_q;
Dc_q_NC=Dc_q;
%-----
% check if the Diophantine equations are OK or not
[DC1 DC2]=DioCheck(N1_q,N2_q,Q1_q,Dc_q,R_q,P_q);
%-----
% construct the system
[sys Ter Tur Tyr]=RFsys(N1_q,N2_q,Dc_q,P_q);
%-----
% Generate The Time Domain input signal
[u,t]=Generate_Signal('sin',Amplitude,Omega,PhaseRad,Tfinal,T);
%-----
%-----Minimizing Control signal
[x N1new]=performance2(N1_q,P_q,R_q,Gc,L);
N1_q=tf(N1new,1,T,'variable','q');
[sys2 Ter2 Tur2 Tyr2]=RFsys(N1_q,N2_q,Dc_q,P_q);
C=lsim(Tyr2,u,t);
CC=norm(C,inf);
C=lsim(Tur2,u,t);
Normx=norm(C,inf);
C=lsim(Ter2,u,t);
Errorx=norm(C(OrderP-1:end),inf);
%-----Minimizing control signal by my FIRST method
[V1 N1_new]=findV1(N1_q_NC,R_q,L);
[sys3 Ter3 Tur3 Tyr3]=RFsys(N1_new,N2_q,Dc_q,P_q);
%-----Minimizing control signal by my Combined method
[bestV1 N1_new]=bestV1(N1_q_NC,N2_q,Dc_q,P_q,R_q,x,V1,NormV1,Normx,u,t)
[sys4 Ter4 Tur4 Tyr4]=RFsys(N1_new,N2_q,Dc_q,P_q);

```

This code calls many functions, all of them is placed in Thesis CD

Best vector to minimize control signal

```

function [VVV
N1_new]=bestV1(N1_q_NC,N2_q,Dc_q,P_q,R_q,x,V1,NormV1,Normx,u,t)
% This function has been built by Mohammed T. A. Elamassie
% This function will find new vector instead of V1 to minimize
the control
% signla, this vector is compination vector between vector x
which found by
% quadratic approach and V1 which I found by my method
dif=0;
L=length(x);
[numN1 denN1 T]=tfdata(N1_q_NC,'v');
numN1=numN1(end:-1:1);
[numr denr]=tfdata(R_q,'v');
denr=denr(end:-1:1);
N1_sym=poly2sym(numN1);
Dr_sym=poly2sym(denr);
KK=0;
    xx=x;
    previous_Norm=min(NormV1,Normx);

for i=1:L
    x(i)=V1(i);
    V1_sym=poly2sym(x);
    N1_new=sym2poly(collect(N1_sym-Dr_sym*V1_sym));
    N1_new=N1_new(end:-1:1);
    N1_new=tf(N1_new,1,T,'variable','q');
    [sys3 Ter3 Tur3 Tyr3]=RFsys(N1_new,N2_q,Dc_q,P_q);
    C=lsim(Tur3,u,t);
    NormV1=norm(C,inf);
    if NormV1<previous_Norm
        KK=KK+1;
        VVV=x;
        previous_Norm=NormV1;
    end
    x=xx;
end
if KK==0
    if NormV1<Normx
        VVV=V1;
    else
        VVV=x;
    end
end
if KK>0
    dif=x-VVV;
end

```

```

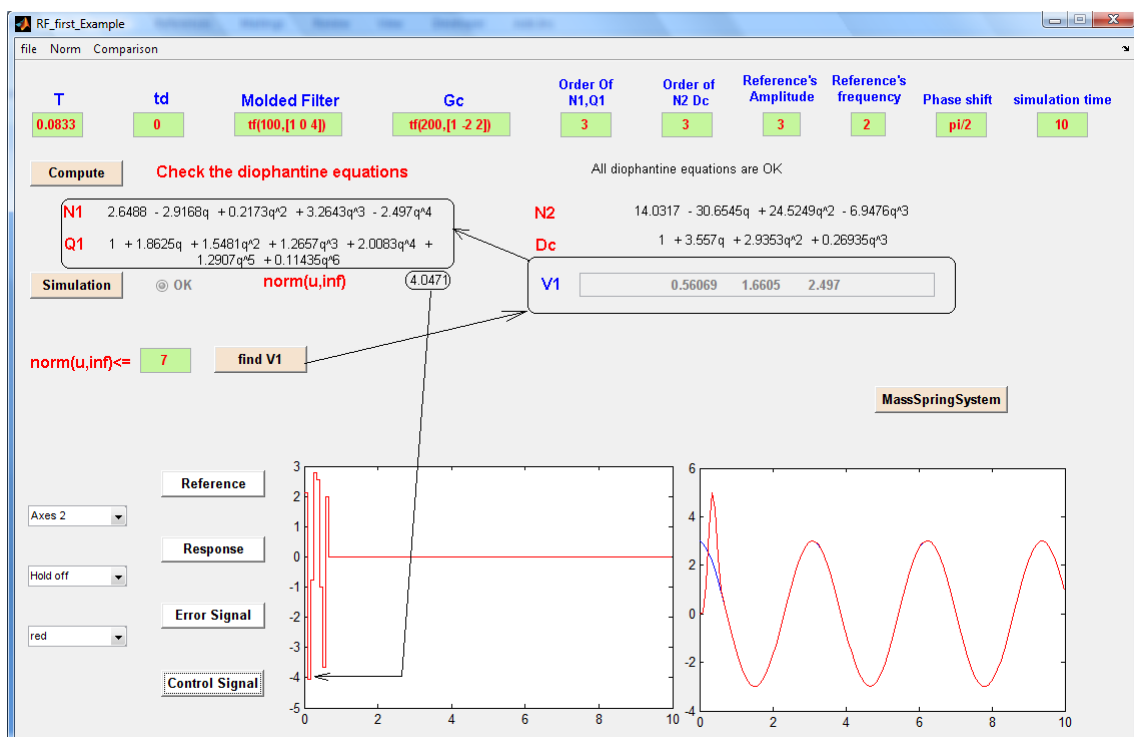
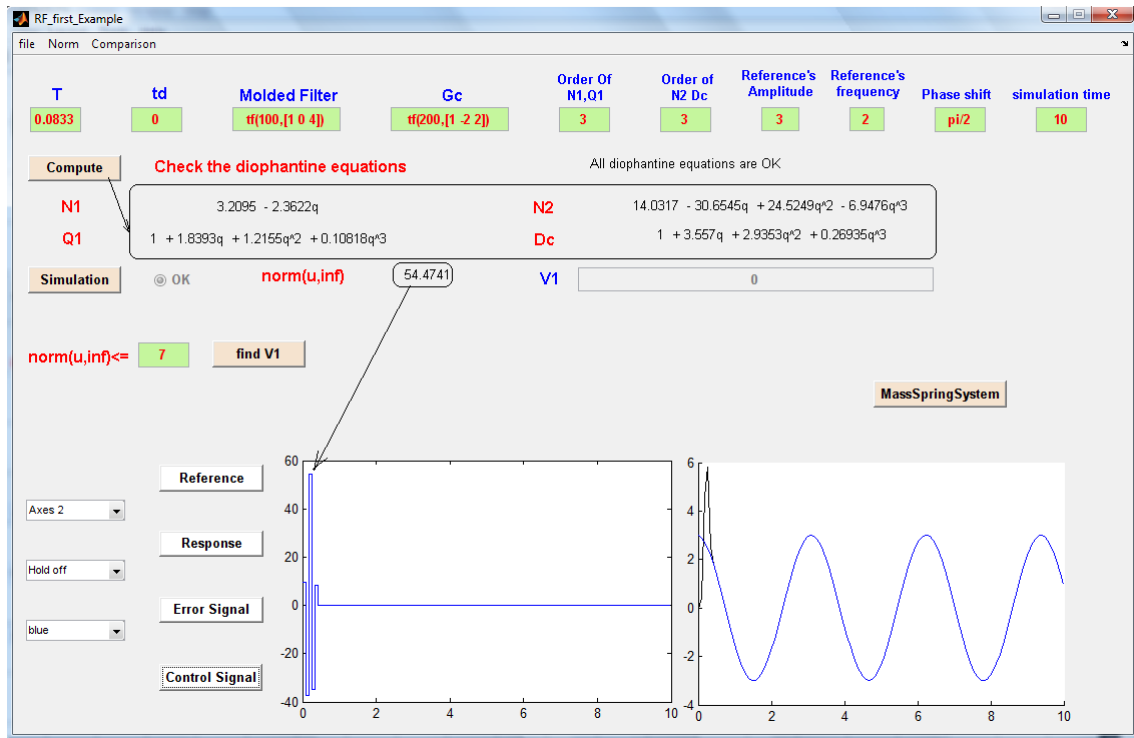
if sum(abs(dif))~=0
    for jj=1:20
        VVV=VVV-dif;
        V1_sym=poly2sym(VVV);
        N1_new=sym2poly(collect(N1_sym-Dr_sym*V1_sym));
        N1_new=N1_new(end:-1:1);
        N1_new=tf(N1_new,1,T,'variable','q');
        [sys3 Ter3 Tur3 Tyr3]=RFSys(N1_new,N2_q,Dc_q,P_q);
        C=lsim(Tur3,u,t);
        NormV1=norm(C,inf);
        if NormV1<previous_Norm
            previous_Norm=NormV1;
        else
            VVV=VVV+dif;
            break;
        end
    end
end
end
V1_sym=poly2sym(VVV);
N1_new=sym2poly(collect(N1_sym-Dr_sym*V1_sym));
N1_new=N1_new(end:-1:1);
N1_new=tf(N1_new,1,T,'variable','q');

```

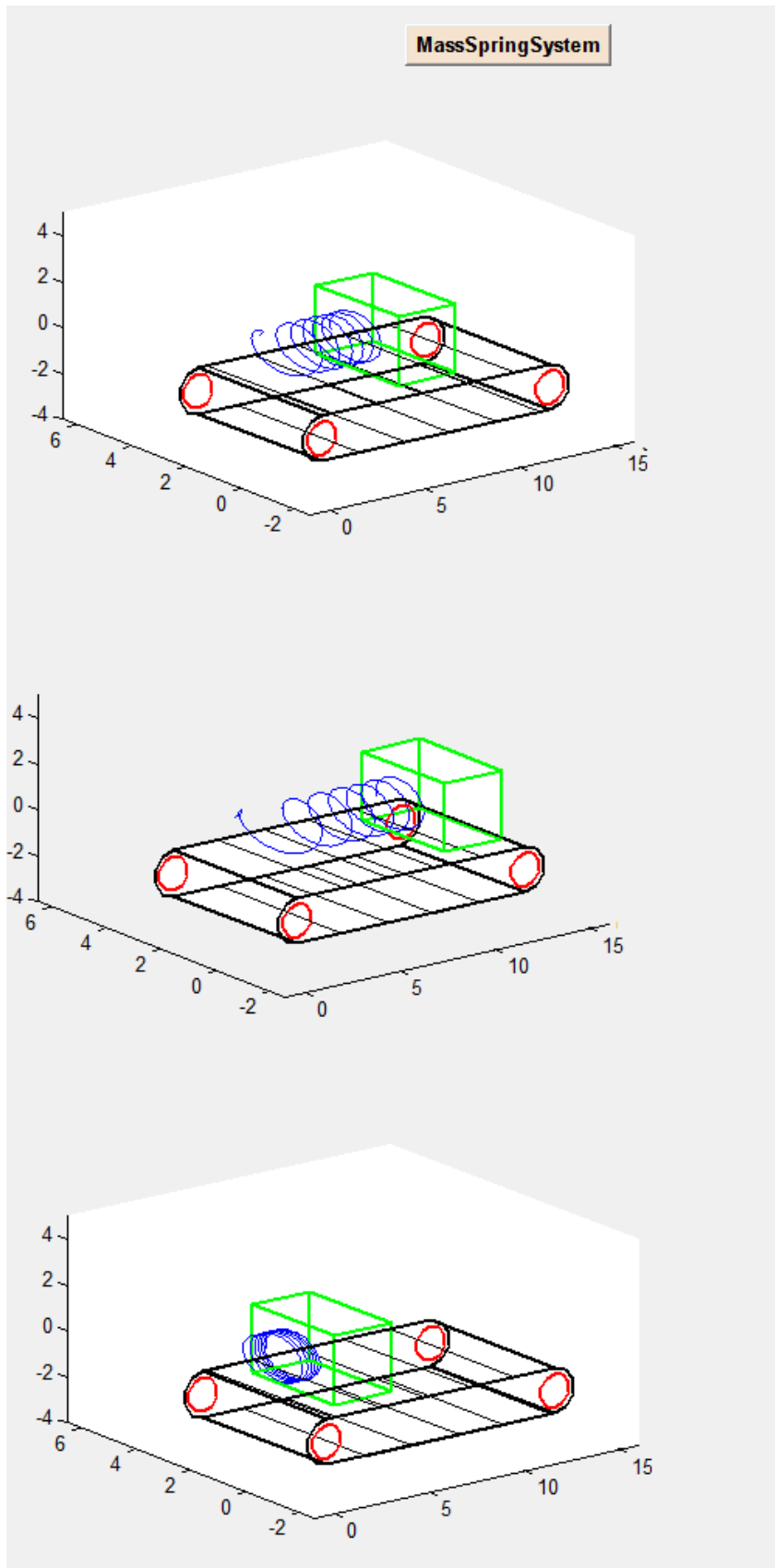

Appendix C

Graphical User Interface (GUI)

Evaluate parameters of Diophantine equation for a given system, minimize infinity norm of control signal, and plot the signals



Mass spring system over moving belt as an example of following sinusoid signal



Appendix D

Used Software and Hard ware

Software programs

- 1- Windows vista
- 2- MATLAB R2007b (matrix laboratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, and FORTRAN.

I used this program to simulate my works and to build Graphical User Interface

- 3- SIMULINK, developed by MathWorks, is a commercial tool for modeling, simulating and analyzing multi-domain dynamic systems. Its primary interface is a graphical block diagramming tool and a customizable set of block libraries. It offers tight integration with the rest of the MATLAB environment and can either drive MATLAB or be scripted from it. SIMULINK is widely used in control theory and digital signal processing for multi-domain simulation and design.

I used this tool to simulate my works and to run the Real-time magnetic ball levitation.

- 4- WebPACK_SFD_10.1 (Xilinx ISE 10.1)
Xilinx ISE is a software tool produced by Xilinx for synthesis and analysis of HDL designs, which enables the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. The Web Edition is a free version of Xilinx ISE that can be freely downloaded or delivered by mail. This edition provides synthesis and programming for a limited number of Xilinx devices.
I used this software computer program to synthesize and analyze The VHDL Code of Deed-Beat control for Real-Time Magnetic Ball Levitation.

- 5- Dsptools_SFD.tar (version 10.1) developed by Xilinx, is a tool for modeling, simulating and analyzing systems used in SIMULINK environment.
I used this software computer program to simulate The VHDL Code of Deed-Beat control for Real-Time Magnetic Ball Levitation.

Note: Dsptools_SFD.tar (version 10.1) is compatible with MATLAB R2007b

Hardware

- 1- Personal Computer
- 2- Magnetic Ball Levitation CE 152.
- 3- Multifunction Data Acquisition Card MF624.
- 4- Power supply.