Control of Linear and Nonlinear Systems Using Bit-Streams

CLAUDIO ARMANDO CAMASCA RAMIREZ

A THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE
DEGREE OF DOCTOR OF PHILOSOPHY IN ELECTRICAL AND ELECTRONIC ENGINEERING
THE UNIVERSITY OF AUCKLAND, OCTOBER 2012

SUPERVISORS: DR. AKSHYA SWAIN DR. NITISH PATEL

ABSTRACT

The implementation of controllers has evolved from its traditional analogue form into a more flexible digital form during past several decades due to the availability of low-cost high-performance digital platforms such as microprocessors and microcontrollers. The conventional approach of digital controller implementation in hardware is based on the assumption that analogue signals are converted into equivalent multi-bit digital format by multi-bit A/D converters. However, the multi-bit processing of signals requires larger processing circuits and routing areas and numerous Input/Output (I/O) pins for connection to external interfaces and therefore increases the complexity of the overall controller implementation. This dissertation investigates the feasibility of an alternate method of controller implementation based on 1-bit signal processing which may overcome the limitations of multi-bit processing. In 1-bit processing, the analogue or multi-bit signals are converted into a simple 1-bit stream by Delta-Sigma modulation at very high sampling frequencies. This can be implemented easily into FPGAs with bit-serial architectures and typically requires (1/nth) of the hardware required for the equivalent n-bit parallel design.

The focus of this research is on the implementation of controllers using bit-streams on a FPGA platform. Initially, the behaviour and performance of various functional elements such as analogue to bit-stream converter, bit-stream to analogue converter, bit-stream adder and scalar are studied using typical test signals and verified using both simulations and physical measurements. These functional elements are central to the successful implementation of controllers in bit-stream environment.

The effectiveness and feasibility of bit-streams in real time control is investigated by considering the design of Generalised Predictive Controller (GPC) for various types of linear and nonlinear systems. GPC belongs to the group of long range predictive controllers and inherits the advantages of adaptive control for its applicability in stochastic systems. The performance of bit-stream controller is verified using two distinctly different simulation engines, Matlab$^{\text{TM}}$and Very High Speed Integrated Circuits Hardware Description Language (VHDL), where the former serves as an ideal implementation platform. Firstly, bit-stream based GPC is designed for linear systems considering examples of a D.C. servo-mechanism and linear thermal system with time delay. Discrete parametric models are fitted to both these systems using the standard procedure of system identification and these are used to design the GPC. The optimum values of various parameters of the GPC such as control and prediction horizon, and control-weighting sequence are obtained using a genetic algorithm. The performance of bit-stream based GPC is compared with those obtained from an GPC implemented using Matlab$^{\text{TM}}$/Simulink$^{\text{TM}}$.

The feasibility of implementing controllers using bit-stream for unstable system is investigated considering the example of an unstable magnetic levitation system. The performance of various controllers such as feedback linearising controller, state space based model predictive controller, generalised predictive controller and lag-lead compensator are studied via simulations using Mentor Graphics' ModelSim$^{\text{TM}}$and compared to the results from an ideal MPC implemented using Matlab$^{\text{TM}}$/Simulink$^{\text{TM}}$. Finally, the GPC and lag-lead controller are implemented in bit-stream on an experimental prototype and their performance are compared to the ideal implementation and is found to be satisfactory.

A new method of designing Generalised Predictive Controller (GPC) for a wide class of nonlinear systems, which are represented by polynomial Nonlinear Auto Regressive Moving Average with eXogenous inputs (NARMAX) model, is proposed. At first, the NARMAX model is identified using a novel method of structure selection or which terms to include into the model, based on Evolutionary Programming. The proposed structure selection algorithm introduces an Internal Term Penalty (ITP) function to reject spurious terms and adopts a pruning strategy to remove insignificant terms from the model by assigning it a time-to-live parameter.

The design of GPC for nonlinear system is based on the output predictions from an extended model which consists of a linear controllable model (called the reference model) and a disturbance model. The disturbance model is represented by a polynomial NARMAX model and is estimated using evolutionary computation. Optimum values of some of the tuning parameters of the GPC such as control and prediction horizons are obtained using evolutionary programming. The nonlinear GPC is implemented in bit-streams and their performance is compared with the ideal implementation considering several simulated examples.

Dedication

My thesis is dedicated to the unforgettable memories of my grandfather.
Mi tesis esta dedicada a los inolvidables recuerdos de mi abuelo.

Armando Ramirez Aquije

Acknowledgements

From the inception of the simplest mathematical concept to the writing of these final words, it is inconceivable to imagine the completion of this thesis without the assistance of some truly inspiring individuals. I, therefore, would like to acknowledge the people that helped and inspired me during my long journey towards the completion of my doctorate.

I wish to express my foremost gratitude to Dr. Akshya Swain and Dr. Nitish Patel. Their trust, support and encouragement has forever shaped the way I approached my research. Thanks to them, I have learnt so much over the years and across the many aspects of control system theory and implementation. They have pushed me past further than I ever thought possible and until the very last day. I am truly thankful for everything they have done for me.

I am grateful also to have met my fellow research colleagues, Ravikesh Chandra, Nasser Giacaman, Wei-Tsun Sun, Shaun Dowler, Salim Namik, Hazim Namik, Ramin Vali and Andrew Austin. Their ability to engage in eloquent and entertaining conversations added enjoyment to being a researcher.

During the toughest of times I could always count on my dearest of friends. I would like to acknowledge the friendships of Simon Lee, Rajesh Bhana, Anthony Yung, Jeetesh Deva, Iain Lonie, Saurabh Rajvanshi and Jeremy Tam. Thank you for showing me life outside of engineering.

I am grateful for my sister Cynthia. Her company, support and love has always been nothing short of amazing. I am extremely proud to be her brother.

I would like to express my extreme and deepest gratitude my parents, Ada and Alberto Camasca. From the moment they taught me how to speak and count, I have been nothing but in debt to them. Their love, support and advice throughout the many years of my studies, have kept me focus during the most arduous and treacherous of roads. I can never repay what they have done for me and I will always remember their sacrifice whenever I read these words.

And to my Creator, thank you.

Claudio A. Camasca Ramirez
October 2012

CONTENTS

FIGURES

TABLES

CHAPTER 1

Introduction

1.1 Motivation and Rational

Computation is one of the traditions that gave rise to modern analogue electronics [1]. The origins of analogue computing can be traced back to the early 1930's with the mechanical differential analyser [2] which was able to resolve differential equations of up to sixth order. With the technological advancements which followed in the subsequent decades, and the widespread use of transistor technology, analogue circuitry became extremely common in the field of engineering. Moreover, as computation requirements increased, the demand for faster, smaller and more accurate devices increased. Amongst the most popular analogue components of these earlier times were the Operational Amplifiers K2-W, Fairchild $\mu A709$, LM101 and OP-07 to name a few [1]. Although the art of analogue circuit design has improved significantly, these circuits have some inherent problems. For example, uncertainty in measurement of values is a well documented effect and analogue circuitry is plagued by it. Some of these include inaccuracies in resistor values, unstable reference voltage and thermal effects [3]. Furthermore, analogue circuitry can be influenced by external radiation, electrostatic noise and other such types of environmental effects.

The technical limitations and imperfections of analogue circuits was an important issue in early years which was resolved due to advancement of digital technology. Many historians have argued that one of the most important driving forces behind technological advances is in the field of communications. A critical landmark that marked the commencement of modern times was the invention of the telephone in the late 1800s, which changed forever the way people interact and shared ideas with one another. The importance of this device is not only attributed to what it accomplished for society, but also for its technological wonder. For the first time in history, a human voice was electronically digitised and successfully decoded back into its original form. In essence, this was the first commercially available *analogue to digital converter*. As decades passed, and much more technological advancement were merged and discovered, the spectrum of applications for analogue to digital conversion, or ADC/DAC grew exponentially. Analogue to digital converters have evolved to such an unimaginable extend, that a whole new field was created, namely digital electronic systems.

The key characteristic of a digital system is that it only deals with two known possible states. There are many ways to describe these, such as TRUE/FALSE, 1/0, HIGH/LOW or 5.5V/-0.5V, but in essence these are chosen to represent one of the two cases. In this manner, any uncertainty is removed (or rounded). This is where digital systems take their advantage. A digital representation of an audio signal, for example, allows CD players to achieve virtually error free storage using optical disks [4].

A common practice in digital systems is to use binary encoding to interact between different signals. A binary line may consist, for example, of 4 unique lines each of which can represent a true or false value. Using this multi-bit type of encoding it is possible to represent 16 possible states. Arithmetic can be performed in binary-base much like in decimal system. The simple procedure described in this paragraph is the basis of the modern computer.

The advent of the microprocessor in 1970's, gave rise to a new area of control systems: *the digital control systems*. Digital control systems are being used in many applications such as machine tools, metal working processes, chemical processes, aircraft control and automobile traffic control and others [5, 6, 7, 8, 9]. Some of the main advantages of having a completely digital system include: improved measurement sensitivity; the use of digitally coded signals, digital sensors and transducers, and microprocessors; reduced sensitivity to signal noise; and the capability to easily reconfigure the control algorithms in software[10].

With the advancements of Very Large Scale Integration (VLSI) Technology and the widespread availability of low cost digital platforms such as microprocessors and microcontrollers, digital system implementation evolved into the more flexible digital form which is now used almost exclusively [11, 12, 13, 14, 15]. Traditionally, these control systems are implemented using microcontrollers, which perform the required control functions sequentially. However, as the complexity of control algorithms increases to achieve improved functionality and the time available to complete the calculations reduces to keep pace with ever-faster power electronic devices, microprocessor based solutions cannot execute the control program within the available time limit.

A potential solution to this performance limitation is the use of hardware based control systems, which can execute control algorithms extremely quickly because many operations are processed in parallel. This is attractive for use in control systems, but implementing hardware control systems is difficult. The designer must either manually translate the control algorithm into individual operations, which are then usually encoded using a Hardware Description Language (HDL), or employ an automated code generation tool. The code is then synthesised and used to program a Field Programmable Gate Array (FPGA).

An alternative method for implementing digital control systems is by using the bit-stream technique. The concept of 1-bit processing was originally proposed for reducing the silicon area occupied by complex VLSI-based systems[16]. The concept of Sigma-Delta, or 1-bit processing or bit-streams have existed since the middle of the century, however it is only in the last two decades that this method has become more attractive [17, 18, 19]. In contrast to the usual methods of implementing digital controllers, bit-stream control systems are developed using standard schematic editors provided by FPGA manufacturers. Provided that the desired control functions are available in the bit-stream library, the designer need not use HDL to develop the control system.

The application of bit-stream in controller synthesis is relatively new. In recent years, this has been applied to implement simple PI and PID controllers [17, 11, 20, 21, 22]. The use of bit-streams in more complex control algorithms is scarce. Therefore, the objective of this study is to investigate if bit-stream implementation can be considered as a possible alternative to implement advanced controllers due to various advantages this offer.

Amongst various types of controllers, the predictive control method has become one of the most popular control methods in industry and academia [23]. This controller has found a wide range of applications in the process, chemical, food processing and paper industries. Some of the most popular MPC algorithms that have found a wide acceptance in industry are Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC), Predictive Functional Control (PFC), Extended Prediction Self Adaptive Control (EPSAC), Extended Horizon Adaptive Control (EHAC) and Generalised Predictive Control (GPC) [23]. These controllers can be designed either using a discrete parametric model such as CARIMA, ARMAX *et cetera* or by using a state space model [24].

The predictive controller is capable of stable control of processes with variable parameters, with variable dead-time, and with a model order which changes instantaneously provided that the input/output data are sufficiently rich to allow reasonable plant identification[25]. It is effective with a plant which is simultaneously non-minimum phase and open-loop unstable and whose model is over-parameterised by the estimation scheme without special precautions being taken [25]. This controller is therefore being selected for investigating the feasibility of bit-stream implementation. Initially, the performance of a GPC implemented using bit-stream is evaluated considering examples of linear systems and linear systems with delay.

Generalised predictive control has been very successful when the plant is operating in the neighbourhood of the operating point. However, most industrial processes have inherent complex nonlinearities, and these can render the classical GPC algorithm impractical[26]. Thus there is a need to design predictive controllers for nonlinear systems. Many researchers have proposed several methods of designing GPC for nonlinear systems by extending the linear GPC[27, 28, 29, 30, 31, 32].

The first step in designing GPC for nonlinear systems is to select an appropriate model of the system. Although several possible representations of nonlinear systems have been proposed which include traditional functional series of Volterra and Wiener[33], Legendre polynomials[34], neural networks [35, 36] and weighted maps [37], the polynomial Nonlinear Auto Regressive Moving Average with eXogenous inputs (NARMAX) model proposed in [38, 39] provide a concise representation of a wide class of nonlinear systems and has attracted considerable interest during last three decades. Many types of nonlinear models such as Volterra, Hammerstein, Wiener can be considered as a special case of the polynomial NARMAX model [40, 26]. One distinct feature of this model is that it is linear-in-parameters and therefore the parameters associated with different terms can be estimated using simple least squares based algorithms.

However, one of the disadvantages of the polynomial NARMAX models is that the total number of terms increases rapidly with the increase in maximum lag of inputs, outputs, noise terms and degree of nonlinearity. Thus detecting the model structure or which terms to include into the model is extremely important. Amongst several approaches which have been proposed to address this structure selection problem, the Orthogonal Least Squares (OLS) algorithm with Error Reduction Ratio (ERR) introduced in [41] and later modified by Billings and co-workers [42, 43, 44] provides an efficient and elegant solution. The ERR criterion of the OLS algorithm essentially computes the significance of model terms based on their ability to explain the output variance. However, certain linear and nonlinear systems may produce erroneous results [45] as OLS-ERR may select spurious terms.

Parsimonious model structure selection is essentially a complex optimisation problem. With the arrival of faster computing power over the last few decades, Evolutionary Computation (EC) algorithms have become a viable solution for solving complex problems. Evolutionary computing offers insensitivity to mathematical concepts such as differentiability, continuity, optima traps *et cetera* [46]. However, EC algorithms are still in their infancy, especially in the system identification field. Arduous research is currently ongoing in the system identification community [47]. In the present study, a new intelligent structure selection method has been developed to obtain a parsimonious model of nonlinear systems which are represented by polynomial NARMAX model. The robustness of this algorithm is illustrated by successfully identifying three nonlinear systems. Generalised predictive controllers are designed for nonlinear systems using polynomial NARMAX model and implemented in bit-stream.

1.2 Objectives of the Thesis

Although considerable success has been achieved and reported in the use of bit-stream for signal processing applications [48, 49], its application for controller implementation in very limited. Moreover, due to only classical controllers such as P, PI and PID being implemented in bit-streams for various applications [17, 11, 49, 21, 50, 51, 52, 53], there remains several open problems and practical issues that need more attention. Therefore, the main objective of the thesis is to investigate if bit-streams can be considered as a viable alternative to multi-bit implementation of sophisticated controllers. In an attempt to address these issues, existing bit-stream theory, predictive control algorithms and system identification tools will be employed to achieve the following objectives:

- To study the various features of bit-stream functional elements and to test and implement bit-streams on real plants.

- Investigate the performance of generalised predictive controller using bit-streams for linear systems and systems with time delay.

- Investigate the performance of bit-streams for unstable systems.

- Develop new methods of designing generalised predictive controllers for nonlinear system represented by polynomial nonlinear auto regressive moving average with exogenous input (NARMAX) models.

- Investigate the feasibility of a new type of nonlinear generalised predictive controller using polynomial NARMAX representation in bit-streams.

1.3 Outline of the Thesis

The organisation of the thesis proceeds in accordance with the objectives mentioned above.

Following the introduction, Chapter 2 describes the various functional elements, such as bit-stream converters, adders, subtractors, multipliers, tapped delay *et cetera*, which are the basic building blocks of a bit-stream system.

Chapter 3 focuses on the design and implementation of generalised predictive controller for linear systems in bit-stream considering two benchmark examples. The general guidelines for successful implementation of bit-stream GPC are presented. The methods of obtaining optimal tuning parameters using genetic algorithm are discussed.

After designing and implanting GPC for stable linear systems, Chapter 4 designs and implements in bit-stream, various types of controllers such as feedback linearising controller, state space based model predictive controller and generalised predictive controller for an unstable magnetic levitation system.

The next objective of the research has been to design and implement nonlinear GPC for nonlinear systems. The first step of this design process is to obtain a nonlinear model of the system. In this study, a nonlinear system is modelled by a polynomial NARMAX model. Therefore, Chapter 5 proposes a new structure selection algorithm to select significant terms of the model using evolutionary programming.

In Chapter 6, a novel GPC is designed for a class of nonlinear systems which are represented by polynomial NARMAX model. This controller is designed and implemented in bit-stream considering several examples of complex nonlinear systems. The conclusions and future work are presented in Chapter 7.

CHAPTER 2

Bit-Streams - An Introduction

The traditional approach of digital signal processing is based on converting the analogue signals to digital using multi-bit A/D converters. All the computations in the digital processors are essentially executed using finite wordlength, typically 8-bit, 16-bit or 32-bit. Higher number of I/O pins are required to route these multi-bit signals and therefore consumes more silicon area. In order to reduce the silicon area and number of logic elements, researchers have proposed an alternate method of signal processing. This is called 1-bit or bit-stream processing which was initially proposed in [48]. This method offers significant advantages over multi-bit approach [54].

Generally, bit-stream technique has been exploited by many researchers in various fields including audio signal processing, neural networks [55, 56, 57, 58, 59], communication systems [48], power electronics [50, 60, 61], and nano-electronics [62, 63]. Concept of bit-stream and bit-stream based processing, which was initially restricted to signal processing application is now becoming popular in control applications. However, the bit-stream based system consists of several functional blocks and there interconnections. This chapter describes briefly about the key function blocks which are used in implementing controllers for both linear and nonlinear systems.

2.1 Analogue and Digital Conversion in Bit-stream

The theory of bit-streams has been in existence for many years, but only with recent technological advances in semiconductor devices has their use become widespread [64]. The introduction of bit-stream signal processing for control purposes is a relatively new concept [11]. This type of encoding may be found under several names: Bit-streams, Sigma-Delta conversion, $\Delta\Sigma$ modulation and 1-bit processing. To understand how bit-stream can be used to represent signals, it is important to first understand the relationship between analogue and digital domains.

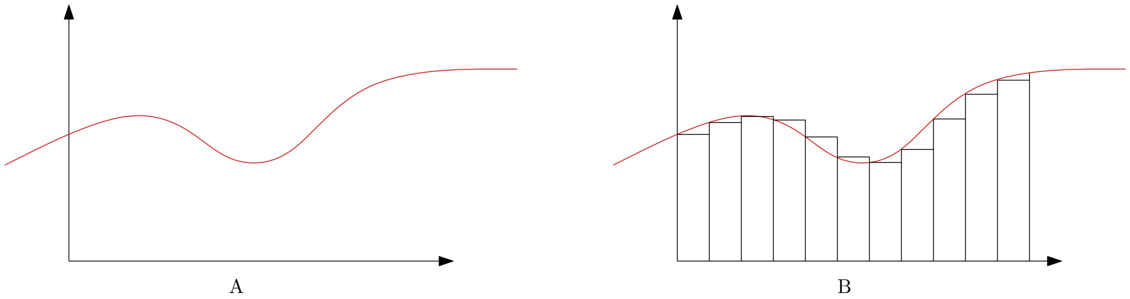


Figure 2.1: *A continuous signal (A) and a discrete signal (B).*

A continuous signal contains infinite information. In order to take advantage of modern computational power, a signal must first be discretised. This offers several advantages to a signal, such as [65]

- Reduced cost over analogue devices.
- Noise immunity.
- Flexibility in response to design changes.

The last item in the above list truly represents the main advantage of a discrete system. In complex systems, where analogue control is utilised, the adjustments of parameters may not be a trivial one. Such an adjustment may require the physical replacement of key components. Instead, by using digital circuitry the complete algorithms can be changed simply by re-configuring a device or via updating software.

However, analogue signal can never be avoided. Discretisation of signal will always involve Analogue to Digital (ADC) and Digital to Analogue (DAC) conversions. Therefore, DACs and ADCs are inherent components of any digital control system.

2.1.1 Signal Encoding

Consider a continuous time signal presented in Figure 2.1-A. A generic signal such as this one can be discretised by the use of a *zero-order sample and hold* and this yields the staircase signal shown in Figure 2.1-B. Each one of the particular stair *levels* is set to represent a specific digital number, which is read at each sample time t_s. For example, if the number of levels is 16, then a 4-bit word can be used for representation. That is, a 4-bit word is obtained at every sample time t_s.

Conventional digital systems have generally used this type of multi-bit lines to transfer control information by using multi-bit words that are output from ADCs. This typical approach is utilised by

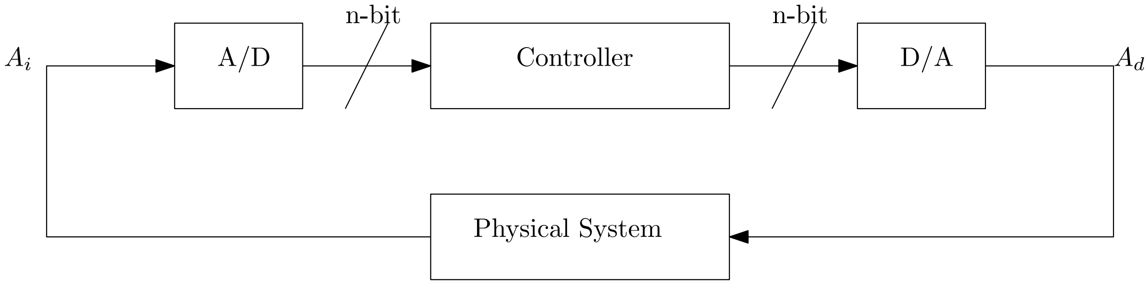


Figure 2.2: *Canonical control system using multi-bit encoding: A_d is an n-bit digital signal converted to analogue, and A_i is the analogue plant output to be digitised.*

many control schemes and it is shown in Figure 2.2. A major drawback of multi-bit systems is the requirement for conversion hardware. Such hardware suffers from decimation and interpolation limitations. The bit-resolution of such devices could be increased to reduce quantisation errors, but this causes the number of logic elements to increase dramatically. Another problem is the large amount of digital lines required to source the data out from a digital device and onto an analogue port. Moreover, these problems escalate severely with increasing complexity of the system.

However, it is possible to conceive that a digital system need not be of a base-*n*, but rather of a binary base. Looking at the example above, instead of using 4 individual lines to represent a word, a single line with a faster sampling rate can be used to push the bit information without packages by serialising and adjusting the binary weight of the stream. Moreover, this implies that the resolution of the value is now a factor of the sampling time. The intricacies of this method will be further described in this chapter.

2.2 The Concept of Bit-streams

To solve some of the problems inherent to multi-bit processing, a bit-stream signal paradigm using $\Delta\Sigma$ modulation has been proposed [49]. With the recent spread of hardware description language such as Verilog and VHDL and the development of advance compilers, the threshold of hardware design is becoming lower [66]. Essentially, the digital signal $\Delta\Sigma$ modulator will output a Pulse Width Density (PWD), which works similar to Pulse Width Modulation (PWM).

A bit-stream control system can be used for control purposes eliminating many unnecessary conversion. This idea is shown in Figure 2.3. Here, the component labelled $\Delta\Sigma$ represents an analogue to bit-stream modulator. On the controller side, a $\Delta\Sigma$ modulator is also included, however this is only an indication since the system itself will inherently contain bit-stream component and therefore no conversion should be needed. The $\Delta\Sigma$ modulator produces a PDM signal that can be fed into the physical system. The analogue system output is passed through a $\Delta\Sigma$ bit-

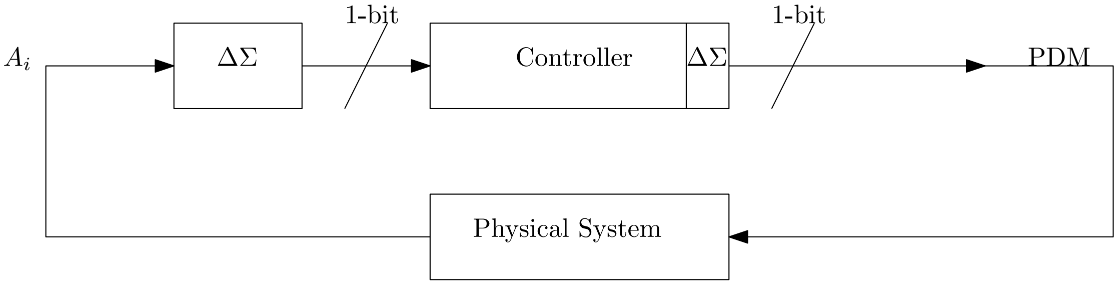


Figure 2.3: *Bit-stream encoding for control system: A_i is the analogue plant output to be digitised and PDM is the control action.*

stream converter which is then fed back into the controller. This type of set up provides many advantages over *n*-bit encoding. As mentioned earlier, an unnecessary encoding layer may be removed by the introduction of a $\Delta\Sigma$ modulator. These modulators can be used to directly output a bit-stream signal from an analogue signal. This enables the use of a single digital line to transfer all the encoding information. Moreover, it is more convenient to think of digital devices in terms of single-bit arithmetic, since these operate using binary systems.

The propose work expands the concept of bit-streams for control purposes by testing its effectiveness using real plants. This chapter will introduce the theoretical groundwork of bit-streams and many of the tools that are used in later chapter to create controllers.

2.2.1 Quanta

A bit-stream is a made up of collection of positive or negative quanta. Strictly, a bit-stream is a binary signal with logic levels ’1’ and ’0’, that can be used to represent any bipolar analogue level. Logic ’1’ present a positive quantum $+Q$ and logic ’0’ represents a negative quantum $-Q$. The injection of a quanta into a bit-stream will adjust the signal level in accordance to the quanta sign. Figure 2.4 shows bit-stream signal S and positive quantum impulse $+Q$ and negative quantum impulse $-Q$.

2.2.2 A Zero Magnitude Bit-Stream

The alternation between a positive and a negative quanta (i.e. logic ’1’ and logic ’0’) results in the cancelation of the over all incident quanta thus resulting in a zero value. Figure 2.5 contains two types of zero signal Z_1 and Z_2. It can be seen that a single or multiple bit delays does not affect the magnitude.

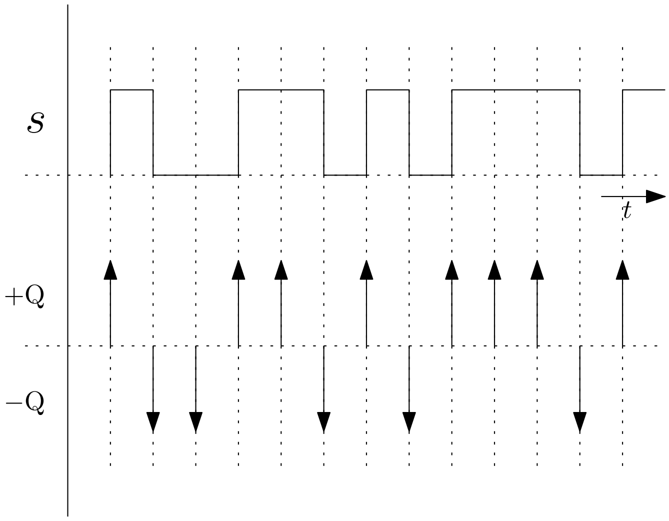


Figure 2.4: *A typical bit-stream signal.*

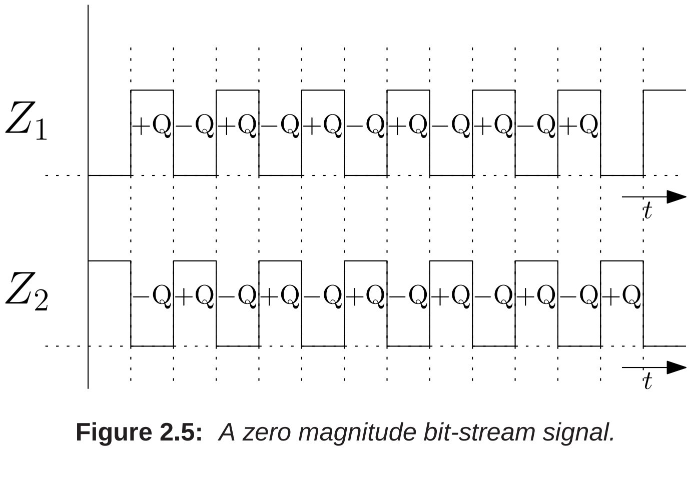

Z_1
+Q −Q +Q −Q +Q −Q +Q −Q +Q −Q +Q
t
Z_2
−Q +Q −Q +Q −Q +Q −Q +Q −Q +Q −Q +Q
t

Figure 2.5: *A zero magnitude bit-stream signal.*

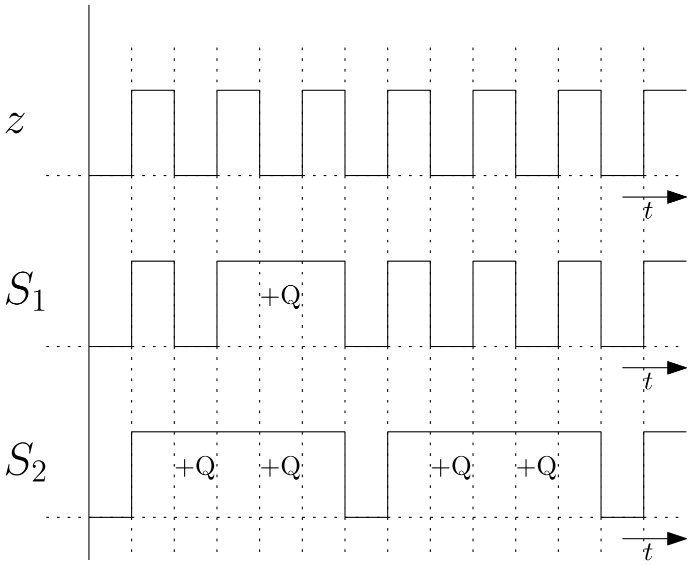


Figure 2.6: *Positive magnitude bit-stream signals S_1 and and S_2 synchronised with signal z.*

2.2.3 Non-Zero Magnitude Bit-Streams

Consider a zero level signal with some of its negative quanta inverted to positive. In this case, the overall net value of the bit-stream has changed positively compared to the original zero value signal. Figure 2.6 shows zero signal, z, a signal S_1 as a zero signal with one negative quantum being inverted and signal S_2 as a zero signal with four negative quanta being inverted. As the amount of negative inverted quanta increases, the magnitude of the overall bit-stream increases positively. Figure 2.7 shows the same concept, albeit with signals S_3 and S_4 as negatively valued bit-streams.

2.2.4 Multi-Bit Binary Word

The concept of a frame is introduced here. A frame contains the quanta summation from N past samples. This is shown in Figure 2.8 where a red box is shown to encompass the current quantum k as well as the seven previous quanta. This is known as an 8-bit frame. The frame captures five positive quanta and three negative quanta, thus the overall frame value is $5-3=2$. Since there are 8-bits in one frame, the resolution is $1/8$ and hence its overall value is $2/8$. In a bi-polar sense (four positive level and four negative levels) this number actually represents $+1/4$. If another signal were to contain six positive quanta, then similarly $6-2=4$ which equals $4/8$ or $+1/4$ in a bi-polar sense. Similarly, if the frame contained seven negative quanta, then $1-7=-6$, $-6/8$ or $-3/4$. It is worthwhile noting the relation between positive and negative quanta within a frame. If an 8-bit frame contains six positive quanta, then it must contain two negative quanta. Also, the number of possible states can be seen as $+Q\times 4, -Q\times 4$ plus one zeros state signal. If more resolution is required, then a bigger frame must be considered.

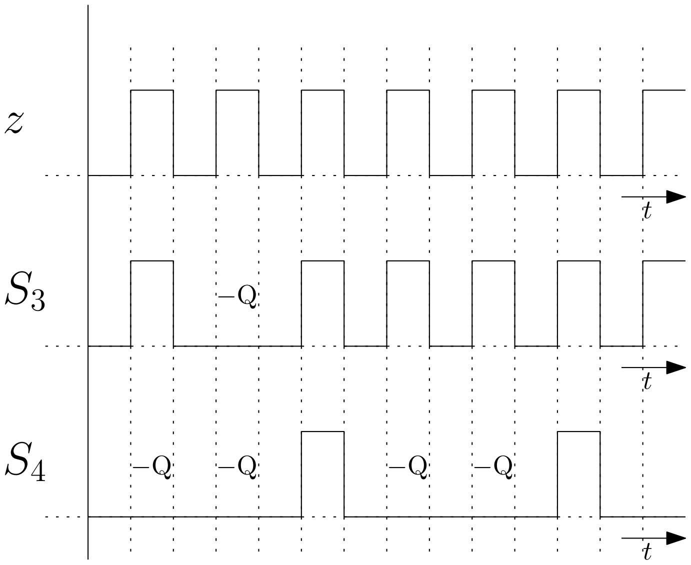


Figure 2.7: *Negative magnitude bit-stream signals S_1 and and S_2 synchronised with signal z*

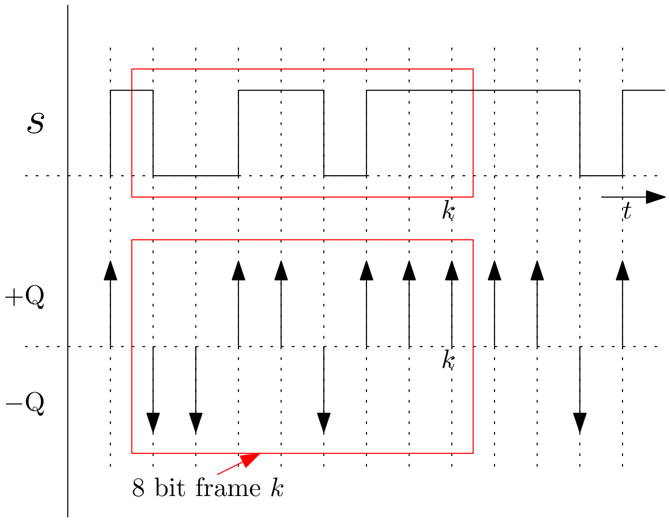


Figure 2.8: *An 8-bit frame encompassing a bit-stream signal.*

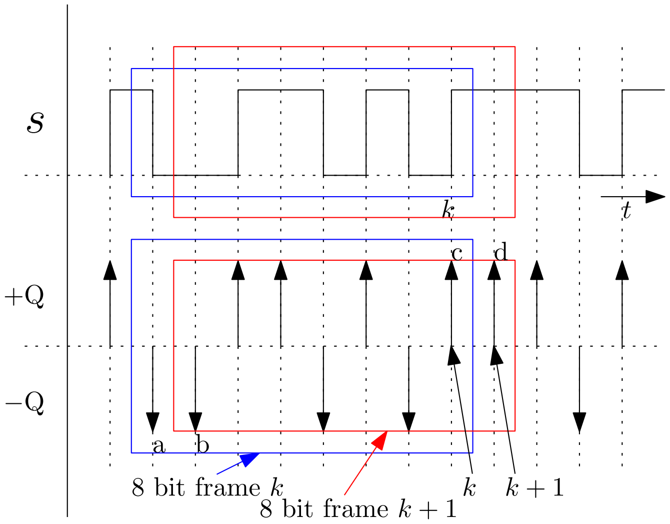


Figure 2.9: *Instantaneous value of a bit-stream signal: Blue frame shows frame at k, red frame shows frame at $k+1$.*

2.2.5 Instantaneous Value

The instantaneous value of a bit-stream can be seen from the extension of the frame concept introduced in §2.2.4 and is shown in Figure 2.8. By evaluating the quanta contained within each frame at every clock pulse it is possible to obtain the instantaneous bit-stream value at time k. This concept will be illustrated in Figure 2.9 using the same examples presented in Figure 2.4 and Figure 2.8. The amount of quanta within the blue 8 bit frame up to the current sample k has the bi-polar value of $+1/4$ (5 positive, 3 negative quanta). Moving the frame one sample forward to $k+1$ the quanta is now $+2/4$ (6 positive, 2 negative quanta).

The relationship to an analogue input can be seen in the following operation

$$V_i(k) = \frac{A(k) - B(k)}{N} \tag{2.1}$$

where $V_i(k)$ is the analogue value at instant k, $A(k)$ is the positive quanta, $B(k)$ is the negative quanta and N is the number of cycles of the bit clock.

2.2.6 Magnitude and Range

All components within the digital logic realm require a clock signal for synchronisation. Since a *zero* bit-stream is simply a square wave of frequency one half the bit rate, it is possible to derive this signal from the global clock. This is very beneficial since it will require minimal additional components.

A comparative operation can be performed on bit-streams to identify their sign of the incident

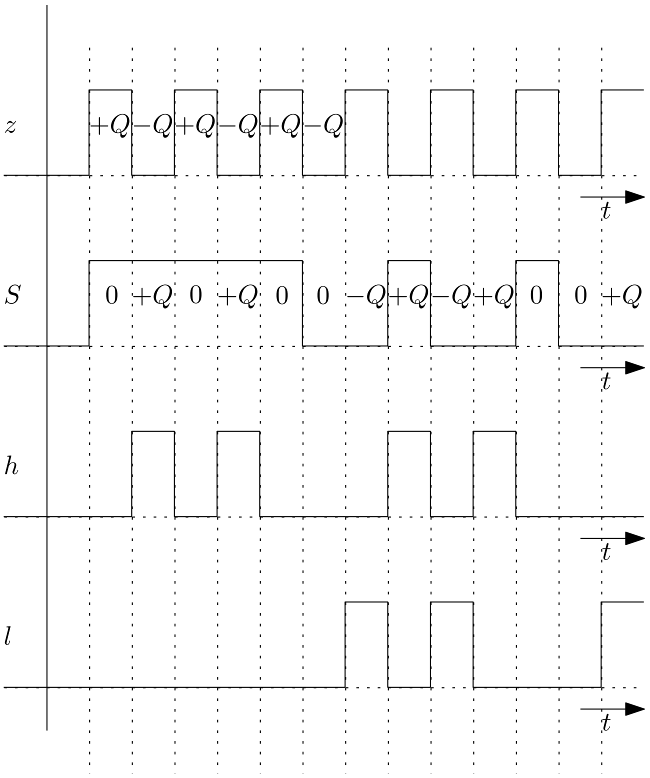


Figure 2.10: *Synchronised decoding for a bit-stream magnitude. Zero signal z is used to decode bit-stream S to obtain incident quanta h or ejected quanta l.*

quanta. Let S be the bit-stream under consideration and z denote a zero level bit-streams. Comparing all the states in S to those of z will tells us that S is also a zero level signal. Conversely, if the state of z is logic $'0'$ and S is at logic $'1'$, then S is positive by one quantum at that instant. Another way of saying this is that an additional positive quantum can only be inserted when z is at logic $'0'$. The incidence of an additional positive quantum is given by

$$h = S \wedge \bar{z} \tag{2.2}$$

where S is the bit-stream under consideration and z is the zero value signal. Is should be noted that h is active only when positive quanta in addition to those already in z are present in S. This is shown in Figure 2.10. Similarly if z is at logic '1' and S is at logic '0' then S is negative by one quantum at that instant. This can be represented by

$$l = \bar{S} \wedge z \tag{2.3}$$

The signals h and l are used for operation within functional elements, later described in §2.3. Lastly, it should be noted that these operations are shift independent, as the interchanging in logic of the l and h bit-streams will still produce a zero value.

Intuitively, a 4 bit word contains 16 different states. Using unipolar representation all the binary states from 0000_2 to 1111_2 can be mapped to a distinct value. Two's complement can be used for bipolar representation. This is where the most significant bit is used as a positive/negative (0/1) flag, 0001_2 to 0111_2 as positive and 1000_2 to 1111_2 as negative, while 0000_2 remains zero valued. The advantage of this representation is in addition and subtractions, since these operations require no adjustment from the normal procedures. It is worth noting that the effective 4 bit range has shift to include negative numbers. That is, a range of [0,15] for unipolar and [-8,7] for bipolar.

2.3 Bit-Stream Functional Elements

The aim of this thesis is to implement complex control algorithms using bit-streams. For this purpose, simple arithmetic operations must be performed using bit-stream. Therefore, the correct implementation of these low level operations are key for successful implementation of a digital controller.

2.3.1 Synchronisation

As with most digital components, synchronisation is extremely important. All functional elements must contain a synchronous clock, reset and bit-stream inputs and outputs. For the design of a functional element, it should be noted that all inputs and outputs must be of type bit-stream. Where appropriate, some elements may contain binary constants as for scaling purposes.

Implementation of these building blocks will be simulated in Mentor Graphics' ModelSim$^{\text{TM}}$. Upon successful simulation, the blocks are then downloaded into a DE2 Development and Education Board shown in Figure 2.11, using Quartus II 10.0. The DE2 Board contains a Cyclone II EP2C35F672C6 with EPCS16 16-Mbit serial configuration device FPGA. Some of the terminology for the design of bit-streams is shown below.

1. **ggclock**, is the fastest clock within the device.

2. **gclock**, is half of *ggclock* and used for the calculations of bit-stream operations.

3. **z**, is the clock rate of a zero valued bit-stream.

4. f_B, is the bit-stream frequency. This is also equal to *gclock*.

5. **init**, is a reset/enable trigger.

The following sections briefly describes some of the already existing bit-stream elements [17].

Figure 2.11: *The Altera DE2 development and education board.*

2.3.2 Analogue to Bit-Stream Conversion Theory

Let $V_i(t)$ be an analogue input and $V_i(k)$ be its bit-stream counterpart. The true value of $V_i(t)$ can be evaluated by integrating over a the NT_B interval as show below.

$$\int_{T-NT_B}^{T} V_i(t) = T_B \sum_{i=k-N}^{i=k} D(i) \tag{2.4}$$

where $D_N(k)$ is the value of the duty cycle at sample interval k and the resolution is give to be one part in N (usually taken to be $N = 2^R$ where R is the effective resolution). If the value for $V_i(t)$ is kept constant then V_a can be show to be

$$V_a = \frac{\sum_{k-N}^{k} D(i)}{N} \tag{2.5}$$

$D(i)$ can be written in terms of positive and negative quanta. If $a(i)$ represents the positive quanta and $b(i)$ the negative quanta, then

$$V_a(k) = \frac{\sum_{k-N}^{k} [a(i) - b(i)]}{N} \tag{2.6}$$

and let $\sum_{k-N}^{k} a(i)$ be $A_N(k)$, $\sum_{k-N}^{k} b(i)$ be $B_N(k)$ and $\sum_{k-N}^{k} D(i)$ by $[A_N(k) - B_N(k)]$ then

$$V_a(k) = \frac{A_N(k) - B_N(k)}{N} = \frac{D_N(k)}{N} \tag{2.7}$$

In summary, this expression represents the sampled analogue input signal $V_i(k)$, known as $V_a(k)$ in terms of positive and negative quanta with a resolution of one part in N. Note the following special cases:

- $V_a = 1$, the positive maximum occurs when $A_N = N$ and $B_N = 0$. In this case the bit-stream consists of only positive quanta.
- $V_a = 0$, the negative minimum occurs when $A_N = 0$ and $B_N = N$. In this case the bit-stream consists of only negative quanta.
- $V_a = 0$, the zero level signal occurs when $A_N = B_N$. In other words, the positive quanta equals the negative quanta. This should form a perfect square wave of half the bit-rate, i.e. $f_b/2$.

If all the quanta present in the bit-stream is included into a single expression, then

$$V_a(k) = \frac{[A_z - B_z + D_N(k)]}{N} \tag{2.8}$$

$$= \frac{[A_z + D_N(k)/2] - [B_z - D_N(k)/2]}{N} \tag{2.9}$$

where $A_z = B_z = N/2$ are the positive and negative quanta present in the zero valued bit-stream. The form of (2.9) is a more accurate representation of the dynamics. The increment of a single positive quanta necessarily means a decrement of a negative quanta.

2.3.3 Adder

The summation of two or more signals is the most common arithmetic operation. Its implementation can be naturally extended to the subtraction (or difference) operation as well. Conceptually, summation can be understood by examining Figure 2.12. Since the value of a bit-stream depends on the quanta present in the bit-stream, it implies that the addition of two or more bit-streams can be achieved by combining the quanta in these bit-streams. The traces in Figure 2.12 also show a box identifying a frame of some arbitrary length (this frame is shown here as a reference only). Trace S_1 has only one positive quanta in one frame and hence has a magnitude of +1. Note that this is only a conceptual diagram. Similarly trace S_2 has a value of +3.

To produce $S_3 = S_1 + S_2$, every quantum in S_1 and S_2 are combined and fed to the output. In the digital domain this is equivalent to a logical OR operation. Thus S_3 within the dimensions of an identical frame length will have one quanta from S_1 and 3 quanta from S_2. From (2.7) the

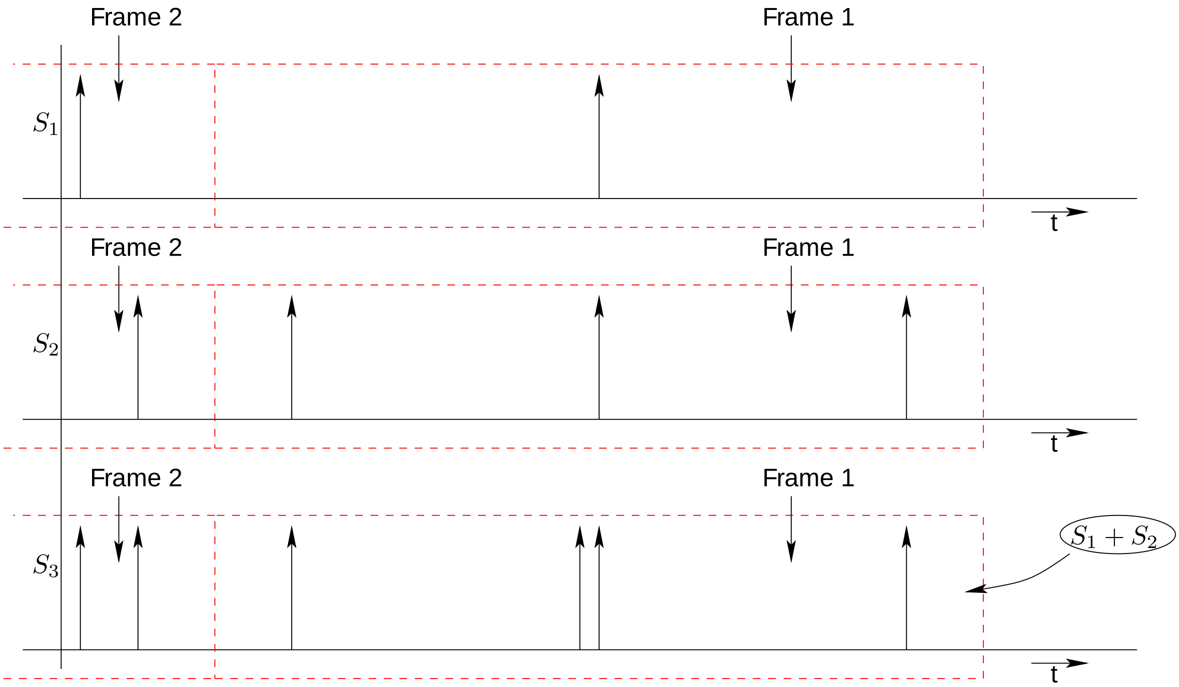


Figure 2.12: *Conceptual bit-stream summation.* $S_1 + S_2 = S_3$.

addition of two bit-stream signals, $V_1(k) = \frac{D_{N1}}{N}$ and $V_2(k) = \frac{D_{N2}}{N}$ can be expressed as follows.

$$V_3(k) = V_1(k) + V_2(k)$$
$$V_3(k) = \frac{(D_{N1} + D_{N2})}{N}$$

We can then write

$$\begin{aligned} V_3(k) &= \frac{(A_z - B_z + D_1 + D_2)}{N} \\ V_3(k) &= \frac{((A_z + D_1/2 + D_2/2) - (B_z - D_1/2 - D_2/2))}{N} \end{aligned} \tag{2.10}$$

Since the incidence of each quanta is indeterminate, when either one (or both) of the signals have a large magnitude, it is very likely that two additional quanta could arrive at the same time. Since only one quanta can be inserted into the output bit-stream, in a logical OR based adder, one of the two will be ’lost’. The loss of an occasional quanta may not be a big problem however at large signal magnitudes, the density of quanta increases and with it an increase in the chance of collisions and consequently the loss will also increase. An adder constructed with a simple OR gate may suffice at small signal magnitudes or when the density of additional quanta is sparse. However to construct a lossless adder some mechanism of buffering the additional and (instantaneous) excess quanta is necessary. A better design will cater to the full input range of

	z	S_1	S_2	Net Incident Quanta
1	0	0	0	0
2	0	0	1	+Q
3	0	1	0	+Q
4	0	1	1	+2Q
5	1	0	0	-2Q
6	1	0	1	-Q
7	1	1	0	-Q
8	1	1	1	0

Table 2.1: *Adder truth table.*

inputs equitably. Two techniques have been developed and they presented below.

The de-coupling of the zero valued bit-stream from the non-zero components is essential in all operations in this study. Equation (2.2) and (2.3) separates the non-zero components in any bit-stream. The de-coupled components are then summed and reinserted into a zero valued bit-stream, *zero*. In (2.10) the $(A_z + D_1/2 + D_2/2)$ term represents the insertion of all the positive quanta. Note that the $D_1/2$ and $D_2/2$ are aggregate differences over a frame size of length N. At any particular instant of time (bit-clock cycle) each term due to $V_1(k)$ will result in a single logic ’0’ or logic ’1’ and similarly a single logical quantity for $V_2(k)$. Thus the addition could easily implemented by a logical OR operation.

Type 1 Adder

This adder uses the expression presented in (2.10). To construct a lossless adder a mechanism of buffering quanta needs to be implemented. This can be achieved by some kind of an accumulator. The structure of this bit-stream adder can be split into two functional parts: an input section and an output section. The input section determines the number of quanta presented to the input while the output section deals with accumulation and the insertion of quanta into the output bit-stream. The input section is based on the truth table shown in Table 2.1.

The first column in this table is the state value of the *zero* reference, z, and columns two and three list the states of the two input bit-streams which are to be summed. The fourth column lists the incident quanta as computed from columns 1, 2 and 3. If the two inputs were zero valued (similar in shape and in phase with *zero*) then there are no quanta present at the input (Rows 1 and 8) and hence the input section feeds zero to the output section. If *zero* and S_1 are at logic ’0’ and S_2 is at logic ’1’ then S_2 is asserting one positive quantum. Similarly if *zero* is at logic ’1’

	z	$S1d$	$S2d$	Net Incident Quanta
1	0	0	0	0
2	0	0	1	+Q
3	0	1	0	+Q
4	0	1	1	+2Q
8	1	0	0	0
7	1	0	1	-Q
6	1	1	0	-Q
5	1	1	1	-2Q

Table 2.2: *Alternative adder truth table.*

while S_1 and S_2 are at logic '0' then two negative quanta have arrived at the input of the adder.

An alternative truth table is constructed on identifying changes from *zero*. The input bit-streams are first pre-processed by

$$S1d = S_1 \oplus z$$
$$S2d = S_2 \oplus z$$

The operation $S \oplus z$ is active high only when S and z are not equal. Thus the occurrence of a additional quanta (positive or negative) is flagged by $S \oplus z$ and the value of the quanta is obtained by $(S \oplus z) \wedge \overline{z}$. The corresponding truth table is shown in Table 2.2. Events similar in Table 2.1 and Table 2.2 have been identically numbered. The choice of the table depends on user preferences. The second form may consume a few more resources.

If there are no quanta present at the input, then the output section must produce a zero valued signal which really is *zero*. If a quantum is to be output then it waits for a suitable slot. If *zero* is at logic '0' then it is possible to assert only a positive quantum by inverting the state of *zero* (to logic '1') at that particular clock cycle. Similarly, if *zero* is at logic '1' then it is possible to assert only a negative quantum by inverting the state of *zero* (to a logic '1').

One of the design criteria is to ensure that bit-streams will originate from blocks not necessarily synchronised. Thus the incidence of a positive quantum may not be at a suitable time slot. With two or more inputs it is possible to have a situation where more than one quanta has to be output. Since it is impossible to insert two quanta, some mechanism for temporarily storing the unused quanta is necessary. There are at least two ways of implementing this. One uses a counter and the other could use a shift register. In this study a Sum-and-Accumulate, SAC, register has been used since it offers a higher storage capacity per FPGA resources used. The schematic for this adder is shown in Figure 2.13.

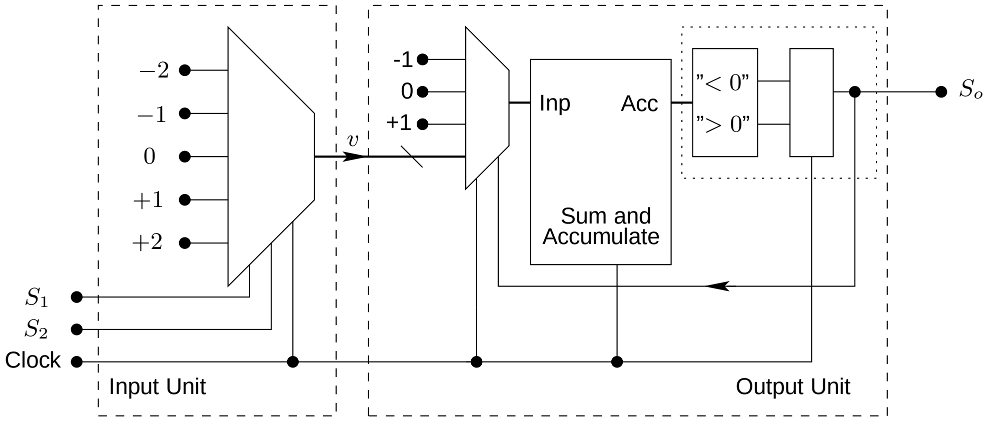


Figure 2.13: *Type 1 bit-stream adder*

The two dashed boxes conceptually isolate the input and the output sections. The thick solid line connecting the two sections corresponds to the fourth column in Table 2.1. The output section operates in two phases. In phase one, a quantum is inserted into the output *zero* if it is required as well as if it is possible. For example if the SAC is positive then there is at least one positive quanta which is buffered and it will be inserted into *zero* only if *zero* is at logic '0'. If a positive quanta is inserted into *zero* then the SAC is decremented and if a negative quanta is inserted then the SAC is incremented. If no quanta are output then there should be no change to the SAC. In the second phase, the output from the *Input Unit* is accumulated in the SAC. Note that the SAC updates its count twice in every bit-period. The VHDL code for the input decoder can be viewed in [17]. The output unit will be reused in other functional units and hence has been built as a separate entity which is included wherever it is required. This entity is called *out_gen_w* and can be viewed in [17]. Figure 2.14 is a simulation plot of the summation of two sinusoids, $S_1 = \frac{40}{128}\cos(200\pi t)$ and $S_2 = \frac{60}{128}\cos(200\pi t + \pi/2)$. Here the bit-clock, $f_b = 1$MHz and $N = 256$. The simulations have been normalised to range from -1 to +1. The theoretical peak value is ± 0.563 and the adder clearly satisfies the required operation.

Two important issues need to be noted. The Sum-and-Accumulate element will keep incrementing indefinitely if the rate at which quanta arrive is greater that the rate at which quanta are inserted into *zero*. For example, if we consider a frame size of N then a *zero* signal will have $N/2$ positive and $N/2$ negative quanta present. If, say, the total number of (additional) positive quanta in S_1 and S_2 is m where $m < N/2$, then each of these will be eventually inserted into the output bit-stream by inverting m negative quanta (in *zero*) into positive quanta. If $m = N/2$, then all the available negative quanta will be inverted and the resulting output bit-stream is made up of all logic '1' i.e. full scale, FS. Now if $m > N/2$ then only $N/2$ quanta can be inserted and hence the SAC will be left with $m - N/2$ quanta after one frame length. Note that the adder does not maintain any concept of frame length and the operates based only on a clock-by-clock basis. After several increments the SAC will reach its maximum capacity and overflow. For a 4 bit SAC an increment from 0111_H (+7) to 1000_H (-8) is not correct and hence overflow is not permitted and the SAC is implemented as a *saturating* sum-and-accumulate, SAT_SAC, element.

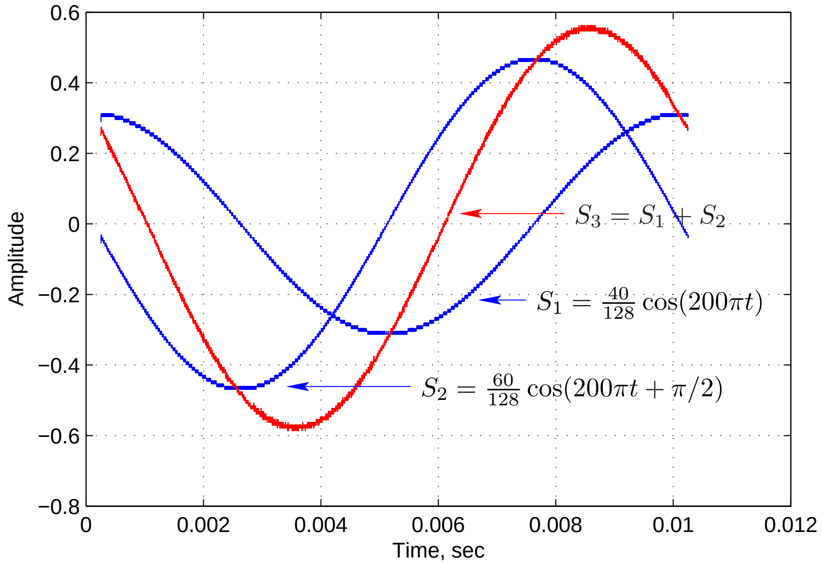
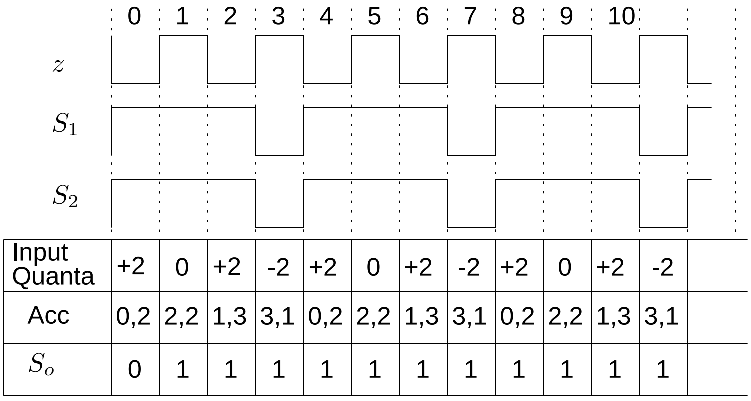


Figure 2.14: *Simulated bit-stream summation of two sinusoids.*



Input Quanta	+2	0	+2	-2	+2	0	+2	-2	+2	0	+2	-2	
Acc	0,2	2,2	1,3	3,1	0,2	2,2	1,3	3,1	0,2	2,2	1,3	3,1	
S_o	0	1	1	1	1	1	1	1	1	1	1	1	

Figure 2.15: *Limiting value accumulated in a Sum-and-Accumulate component.*

The second important issue addresses the width of the SAT_SAC. If S_1 is non-zero while S_2 is zero (and in phase with the internal *zero* reference, there are no co-incident quanta. If both S_1 and S_2 are non-zero and of small magnitude then the co-incident quanta will increment the SAC. Since there are plenty of *free quanta slots* the stored quanta are output fairly quickly. If S_1 is large and S_2 is small and $S_1 + S_2 <= N/2$ then the co-incident quanta have to wait a bit longer for a free slot. The worst case scenario is when $S_1 = S_2 = N/2$. Figure 2.15 shows one possibility. The row entitled *Input Quanta* follows Table 2.1. The *Acc* row is the output of the SAT_SAC. Since it is updated twice in each period two values are shown. The S_o row is the bit-stream output based on the value of the SAT_SAC at the previous cycle. If the SAT_SAC is positive and z is logic '0' then a positive quanta is inserted and hence *Acc* is decremented. If *Acc* is zero then the output bit-stream is exactly *zero*. The progression of SAC shows that the maximum value is 3. If z were to be delayed by one clock cycle, the maximum count in *Acc* would be 2.

Shown in Figure 2.16 are the simulation results of two adders with different SAT_SAC widths.

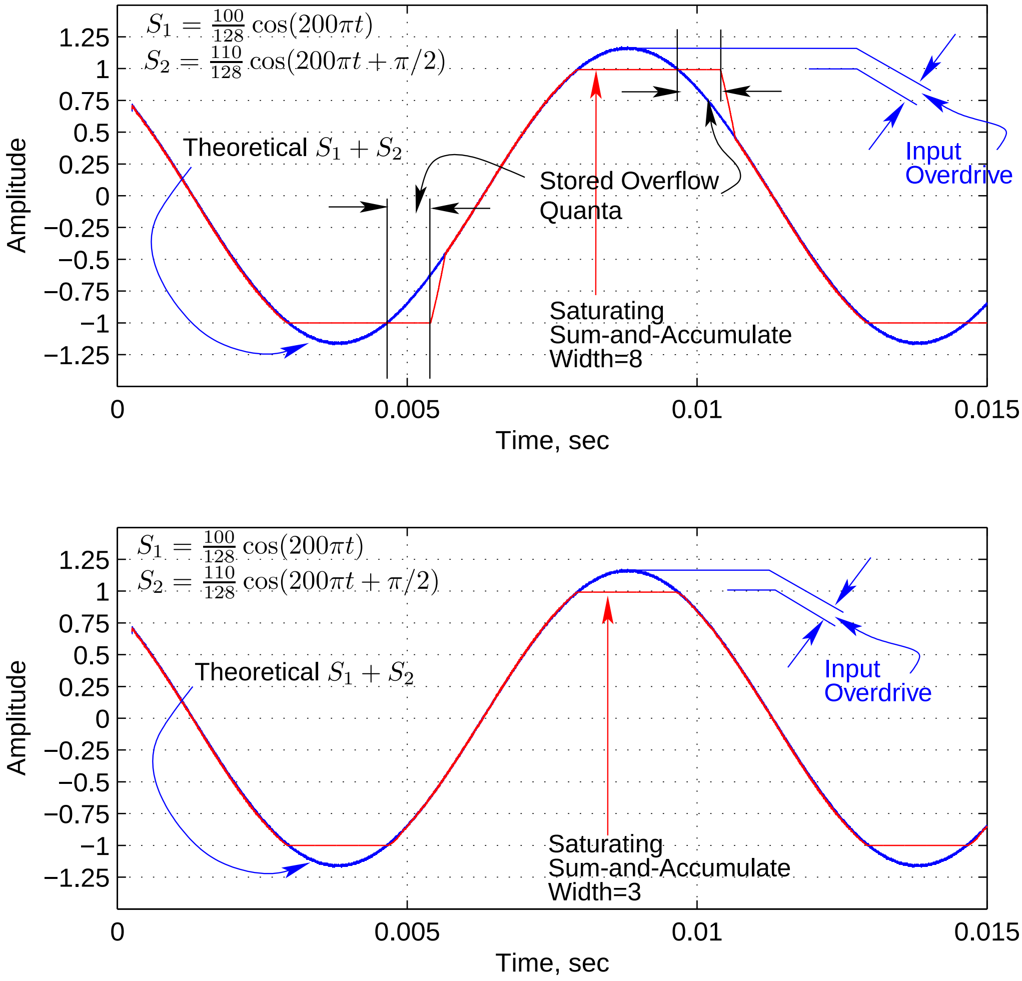


Figure 2.16: *Saturation effects of two different bit-stream adders.*

If $S_1 + S_2 > N$ then the overflow quanta will get accumulated in the SAT_SAC. The SAT_SAC with a bigger width will be able to store more quanta. As soon as the instantaneous value of $S_1 + S_2$ drops below N, more quanta slots will be available and hence the SAT_SAC will fill these up. Since every usable slot will be filled by these stored quanta, it will result in an extension of the flat-top. Figure 2.16 shows this. With the SAT_SAC width equal to 3 the flat-top width is the minimum possible.

The design choice of SAT_SAC width depends on the application. If one were sure that $S_1 + S_2$ never exceeds full-scale or if it does and clipping is acceptable then select the SAT_SAC width as 3 bits wide (-3 to +3). This reasoning is valid for systems where the bit-streams are evenly spread. In some cases it is possible for the bit-streams to get clumped. Alternatively, the behaviour of the system may result in short bursts of quanta. The $S_1 + S_2 < N/2$ condition may be violated at some instance in time but in the long term the condition may be satisfied. In this case it is likely to loose quanta if the width of the SAT_SAC is restricted to the evenly-spread optimum. It may then be necessary to increase the SAT_SAC width.

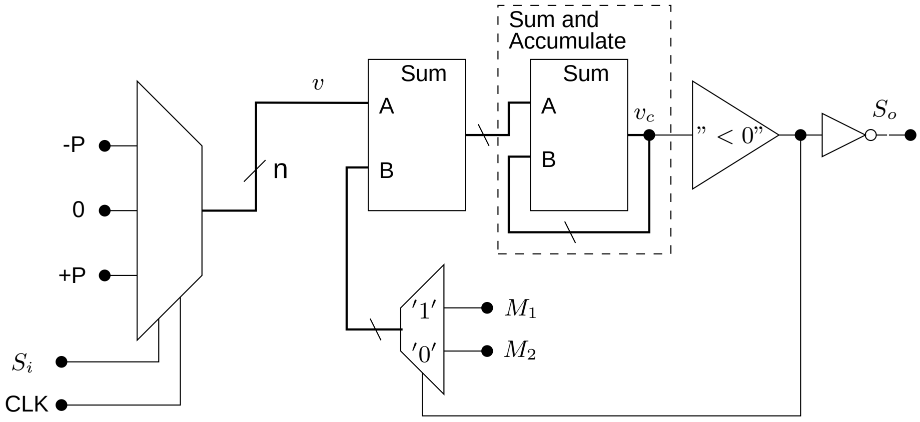


Figure 2.17: *Bit-Stream to bit-stream modulator generator.*

	z	S_i	Quanta
1	0	0	0
2	0	1	+P
3	1	0	-P
4	1	1	0

Table 2.3: *Bit-Stream to multi-bit truth table.*

Type 2 Adder

For more evenly spaced incoming input quanta, another type of adder can be implemented with no side effects.

The input bit-stream is converted into a multi-bit word by applying the truth-table shown in Table 2.3. Here the alternative decoding for, $Sd = S \oplus z$, has been adopted.

If the input is zero valued bit-stream and in-phase with the internal *zero* then there are no (additional) quanta in S, rows 1 and 3, and hence the truth-table *outputs* a multi-bit 0. If z = logic '0' and S = logic '1' (row 2) then the input is asserting a positive quanta and hence the truth-table outputs a $+P$. The same logic applies to the condition z = logic '1' and S = logic '0'.

Consider a bit-stream, $S_i(k)$, with value $D_i(k)/N$. Recall that $D_i(k)$ is the total difference between the positive and negative quanta. The inversion of a few positive quanta necessarily implies an inversion of the exact number of negative quanta. Thus over a frame length of N there are $N/2$ positive and $N/2$ negative quanta available to invert. A value of $+|D_i/N|$ then implies that $D_i/2$ negative quanta have been inverted into positive quanta.

Figure 2.18 plots the output of the sum-and-accumulate block with no feedback. With the appli-

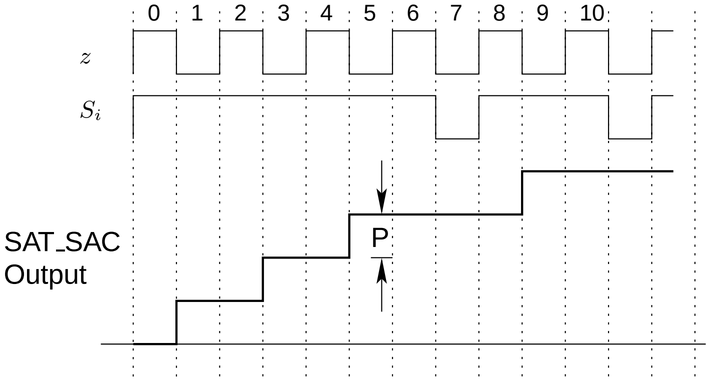


Figure 2.18: *Output of Sum-and-Accumulate without feedback.*

cation of S_i and Table 2.3, it can be seen that the SAT_SAC changes only when S_i and z are different. With an input of $+|D_i/N|$, there are $A_z + D_i/2$ positive quanta and $B_z - D_i/2$ negative quanta. In other words $D_i/2$ negative quanta are inverted. When these are decoded by the front end, in N clocks, the SAT_SAC will accumulate to

$$\frac{D_i}{2}.(+P) - 0.(-P) = \frac{D_i}{2}P \tag{2.11}$$

The $\frac{D_i}{2}.(+P)$ term corresponds to the $z = 0$ state and $0.(-P)$ corresponds to the $z = 1$ state. If the negative feedback were to be enabled then the quanta being fed back, M_1 and M_2, will neutralise this change. In steady state after N clocks, if the number of M_1 quanta is X then there must also be $N - X$ of the M_2 quanta. Thus

$$\frac{D_i}{2}P = XM_1 + (N - X)M_2 \tag{2.12}$$

If $M_1 - M_2 = \Delta M$ then

$$\begin{aligned} X(M_1 - M_2) + NM_2 &= \frac{D_i}{2}P \\ X\Delta M &= \frac{D_i}{2}P - NM_2 \\ X &= \frac{D_i}{2}\frac{P}{\Delta M} - \frac{NM_2}{\Delta M} \end{aligned} \tag{2.13}$$

This equation gives the relationship between the number of output quanta (of weights M_1 and M_2) which are required to match the input bit-stream (composed of quanta $+P$ and $-P$). In the $\frac{NM_2}{\Delta M}$ term, N, ΔM and M_2 are all constants and the term corresponds to an offset. Normally $M_1 = -M_2 = M$ and hence

$$X = \frac{D_i}{2}\frac{P}{\Delta M} + \frac{N}{2} \tag{2.14}$$

In the $\frac{D_i}{2}\frac{P}{\Delta M}$ term $\frac{P}{\Delta M}$ is a constant and the contribution to X is then proportional to $D_i/2$ with

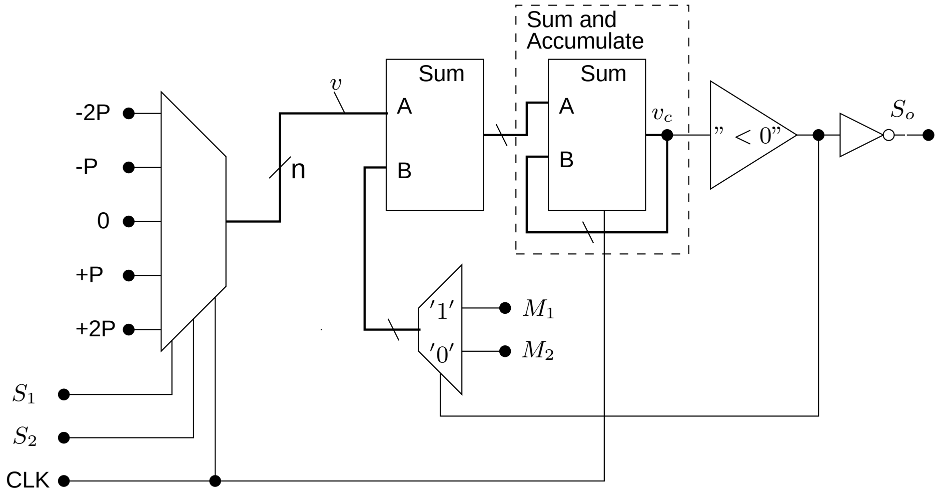


Figure 2.19: *Type 2 bit-stream adder.*

a gain of $\frac{P}{\Delta M}$. Since a unity gain is appropriate,

$$P = \Delta M = (M_1 - M_2) = 2M \tag{2.15}$$

The gain is set by the ratio, $P/2M$, and hence the individual values of P or M are not really important. Note that a non-unity gain may be appropriate for specific applications and coefficients to suit can be easily chosen. However, with hardware implementation being the primary focus, larger values of P and M will require wider word sizes and consequently the multiplexors, the Saturating Sum-and-Accumulate and the sign comparator all consume more hardware resources. The smallest suitable values for unity gain are $M = 1$ and $P = 2$. This selection results in

$$X = \frac{D_i}{2} + \frac{N}{2} \tag{2.16}$$

Where the input is a bit-stream of value D_i/N. A two input adder can now be easily constructed by extending the front end of the bit-stream generator in Figure 2.17. The fully functional 2 input bit-stream adder is shown in Figure 2.19.

(2.13) can also be expressed in terms of the output quanta difference, D_o. This form will be used later.

$$D_o = D_i\frac{P}{\Delta M} - N\left(\frac{2M_2}{\Delta M} + 1\right) \tag{2.17}$$

The truth table for the front end could be either one of Table 2.1 or Table 2.2. If

$$S1d = S_1 \oplus z$$
$$S2d = S_2 \oplus z$$

	z	$S1d$	$S2d$	Net Incident Quanta
1	0	0	0	0
2	0	0	1	+P
3	0	1	0	+P
4	0	1	1	+2P
8	1	0	0	0
7	1	0	1	-P
6	1	1	0	-P
5	1	1	1	-2P

Table 2.4: *Type 2 adder truth table.*

then Table 2.4 is a suitable truth table for Figure 2.19. The time domain response of the Type 2 Adder has no visually discernable differences when compared to the Type 1 Adder and hence a time domain plot has not been presented. However, a spectrum of the outputs of the two kinds of adders has been obtained and presented in Figure 2.20. Recall that Figure 2.14 is a plot of the sum of $S_1 = \frac{40}{128}\cos(200\pi t)$ and $S_2 = \frac{60}{128}\cos(200\pi t + \pi/2)$. The resulting sinusoid should have a amplitude of ≈ 0.563. Each of the components in Figure 2.20 has been normalised such that the fundamental component at 100Hz has a magnitude of unity. The magnitude axis has been deliberately scaled so as to make the smaller magnitudes visible. Consequently the fundamental frequency of 100Hz is off the scale. The spectrum shows that the odd harmonics are dominant. Also clearly seen is that the Type 1 adder produces a richer spectrum. The third harmonic is 8 times greater while the fifth and the seventh harmonics are 5 times and 4 times, respectively, greater than those produced by the Type 2. At larger signal magnitudes the difference between the two spectra diminish but nevertheless the harmonics produced by the Type 1 adder are always greater than those produced by the Type 2 Adder.

The issues with the buffering of the overflow quanta is similar to that of the Type 1 Adder. Widening the SAT_SAC will enable more of the overflow quanta to be stored. The absolute minimum width has been determined experimentally as 4 bit (-7 to +7) which is one bit larger than the Type 1 Adder.

Resource Usage

The resource usage of these two adders is listed in Table 2.5. This has been obtained by implementing each of the items in complete isolation and should be taken as indicative. The table shows that Type 1 tends to use more of the FPGAs resources. In practical use, some of the resources within a logic cell could be shared by neighbouring devices and hence the resource

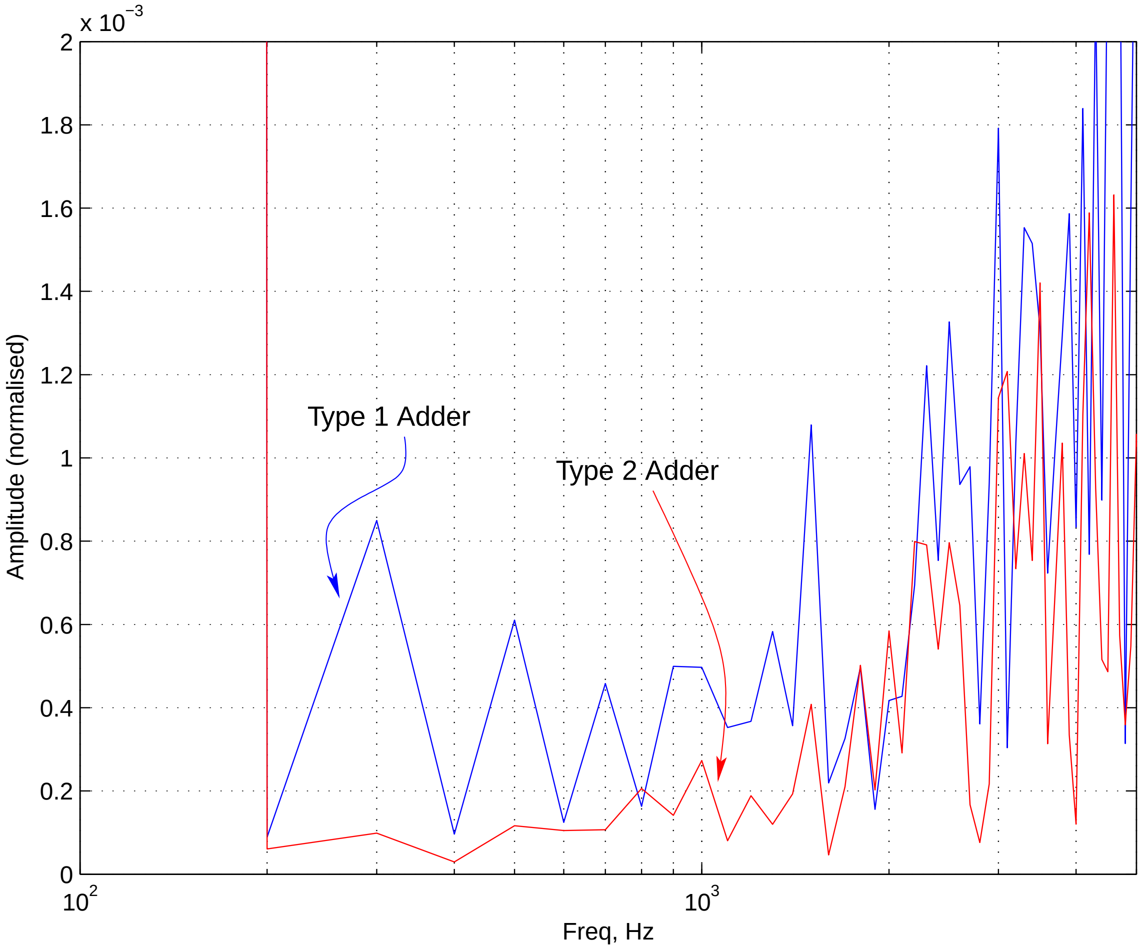


Figure 2.20: *Normalised spectra of type 1 and Type 2 bit-stream adder outputs.*

	Logic Cells	Registers	Carry Chains
Type 1	23	8	0
Type 2	14	8	0

Table 2.5: *Adder resource usage.*

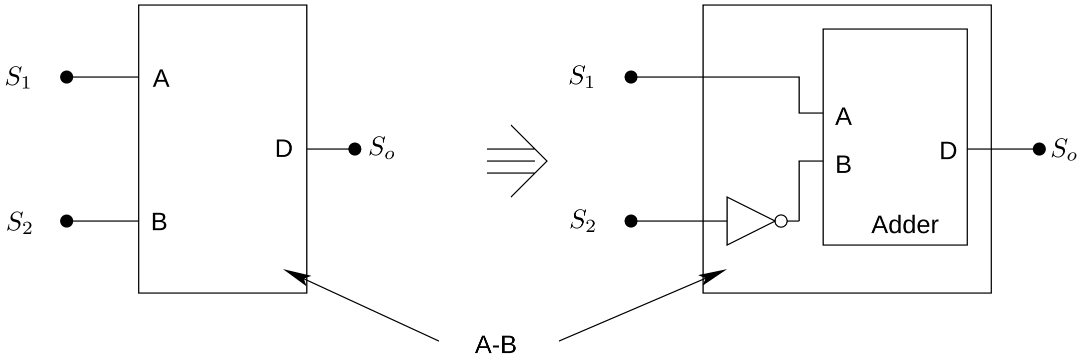


Figure 2.21: *Bit-Stream subtraction.*

usage figures change from application to application.

2.3.4 Subtractor

The subtraction operation is, conceptually, very similar to the addition operation. Particularly if subtraction is viewed as the addition of non-inverted and an inverted bit-stream. For example given two bit-streams, S_1 and S_2 and it is required to achieve $S_1 - S_2$ then every incident quanta in S_2 is inverted and then added to S_1. Figure 2.21 illustrates the concept.

The Type 1 and Type 2 Adders discussed earlier in §2.3.3 and §2.3.3 respectively can be easily modified to achieve bit-stream subtraction and will be discussed in the following sections.

Type 1 Subtractor

A schematic of the Type 1 Subtractor is shown in Figure 2.22. The only modification required is to the front end. All other issues regarding signs of the incident quanta or the combinations of past and current quanta are handled in exactly the same way as the adder. The truth table which achieves subtraction between two inputs is shown in Table 2.6. In particular, it is designed to help achieve $S_1 - S_2$. Column 1 is the *zero* reference, columns 2 and 3 are the S_1 and S_2 bit-stream inputs while the fourth column is required quanta combination for subtraction. If the incident quanta are identical then the difference between them is zero (rows 1, 4, 5 and 8). If z and S_1 are at logic '0' while S_2 is at logic '1' then a $+Q$ is present at the second input which

	z	S_1	S_2	Net Incident Quanta
1	0	0	0	0
2	0	0	1	-Q
3	0	1	0	+Q
4	0	1	1	0
5	1	0	0	0
6	1	0	1	-Q
7	1	1	0	+Q
8	1	1	1	0

Table 2.6: *Subtractor truth table.*

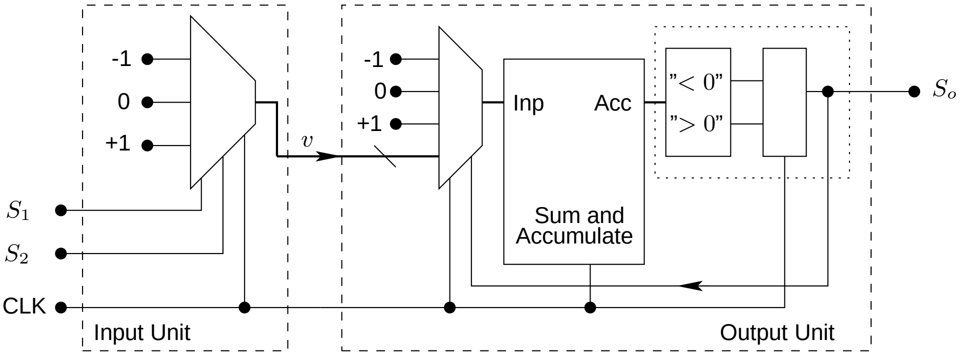


Figure 2.22: *Type 1 bit-stream subtractor.*

translates to combined value of $[(0) - (+Q)] = -Q$ (row 2). If z and S_1 are at logic '1' while S_2 is at logic '0' then a $-Q$ is present at the second input which translates to combined value of $[(0) - (-Q)] = +Q$ (row 7). The other conditions can be similarly analysed.

An alternative truth table for the front end is shown in Table 2.7. Here the bit-stream inputs are first pre-processed by

$$S1d = S_1 \oplus z$$
$$S2d = S_2 \oplus z$$

As with the adder, these pre-processed signals assert $S1d$ and $S2d$ only when they are different from the *zero* reference, z. The rows in this table are numbered to match the conditions in Table 2.6.

Simulation results of the subtraction between two sinusoids is shown in Figure 2.23. In this example, the bit-streams are identical to those used for the adder i.e. $S_1 = \frac{40}{128}\cos(200\pi t)$ and $S_2 = \frac{60}{128}\cos(200\pi t + \pi/2)$. The theoretical peak value is ± 0.563 and Figure 2.23 shows that the

	z	$S1d$	$S2d$	Net Incident Quanta
1	0	0	0	0
2	0	0	1	-Q
3	0	1	0	+Q
4	0	1	1	0
8	1	0	0	0
7	1	0	1	+Q
6	1	1	0	-Q
5	1	1	1	0

Table 2.7: *Alternate subtractor truth table.*

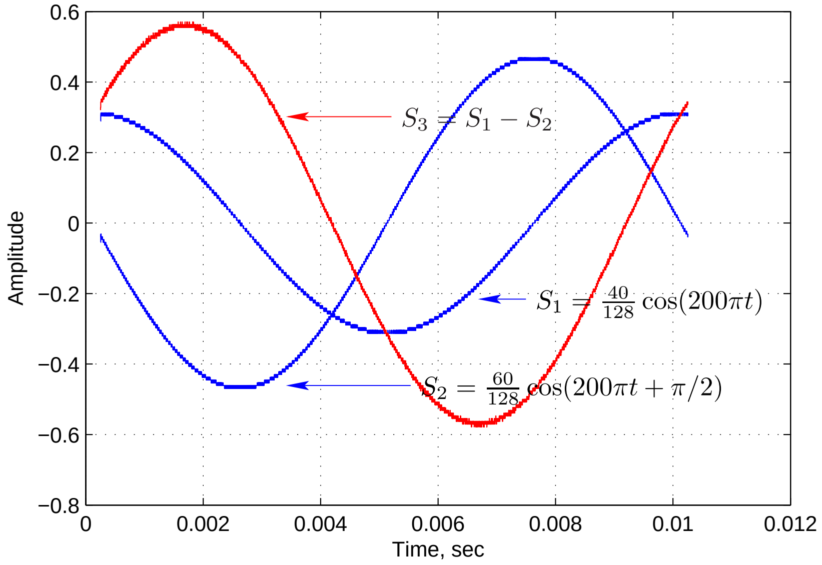


Figure 2.23: *Bit-stream subtraction of two sinusoids.*

subtractor achieves the desired operation.

Type 2 Subtractor

The Type 1 Subtractor also suffers from the *clumping* side-effect (as manifest in the Type 1 Adder). The impact of this clumping on the harmonics (shown later) is less severe than with Adder. Nevertheless a modification of the Type 2 Adder, for subtraction is easily possible and is shown in Figure 2.24. The front end of the Type 2 should satisfy truth tables similar to Table 2.6 or Table 2.7.

The equations governing the behaviour are identical to (2.11) to (2.16). Hence $P = 2$ and $M_1 =$

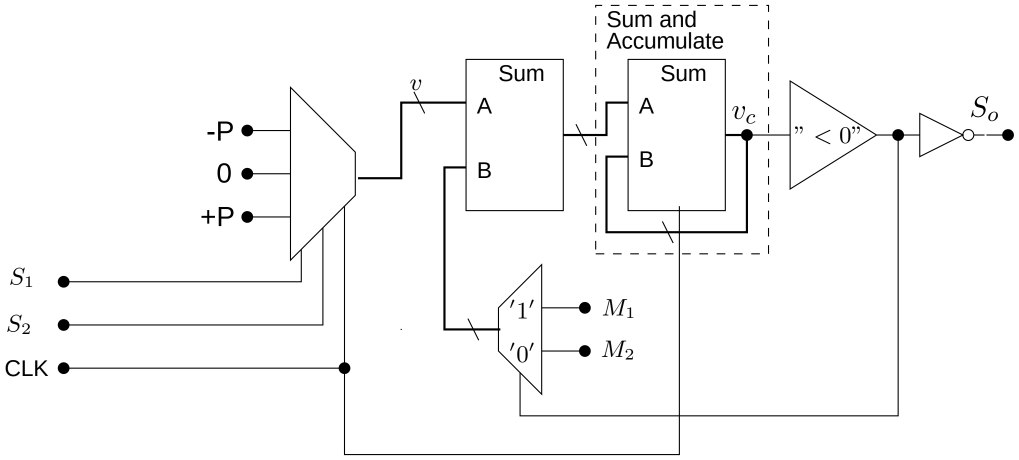


Figure 2.24: *Type 2 bit-stream subtraction.*

$-M_2 = 1$. The time domain behaviour of the two subtractors are not distinguishable and hence has not been reported. However, a spectral analysis is shown in Figure 2.25. The fundamental components (of both types)at 100Hz, as expected, have a magnitude of 0.563. The harmonics content is more uniform than the adder but with the odd harmonics being more dominant. For both the type 1 and the type 2, the harmonics up to one decade from the fundamental have magnitudes approximately 70dB to 80dB lower than the fundamental.

2.3.5 Scaling

The amplification or attenuation of a signal is frequently necessary when designing control systems. This scaling, preferably, should be frequency independent. In practice, this is difficult to ensure and frequency dependence is usually incorporated into system component models. A bit-stream can be scaled to have an effect similar to that of scaling an analogue signal. Recall (2.7) and (2.9) as a representation of an analogue signal $V_i(t)$ using bit-streams.

$$V_i(k) = \frac{D_N(k)}{N}$$
$$V_i(k) = \frac{(A_z - B_z + D_N(k))}{N}$$
$$V_i(k) = \frac{((A_z + D_N(k)/2) - (B_z - D_N(k)/2))}{N}$$

The magnitude of a bit-stream is proportional to the difference in the positive and negative quanta. If it is required to scale $V_1(k)$ by g such that

$$V_2(k) = gV_1(k)$$

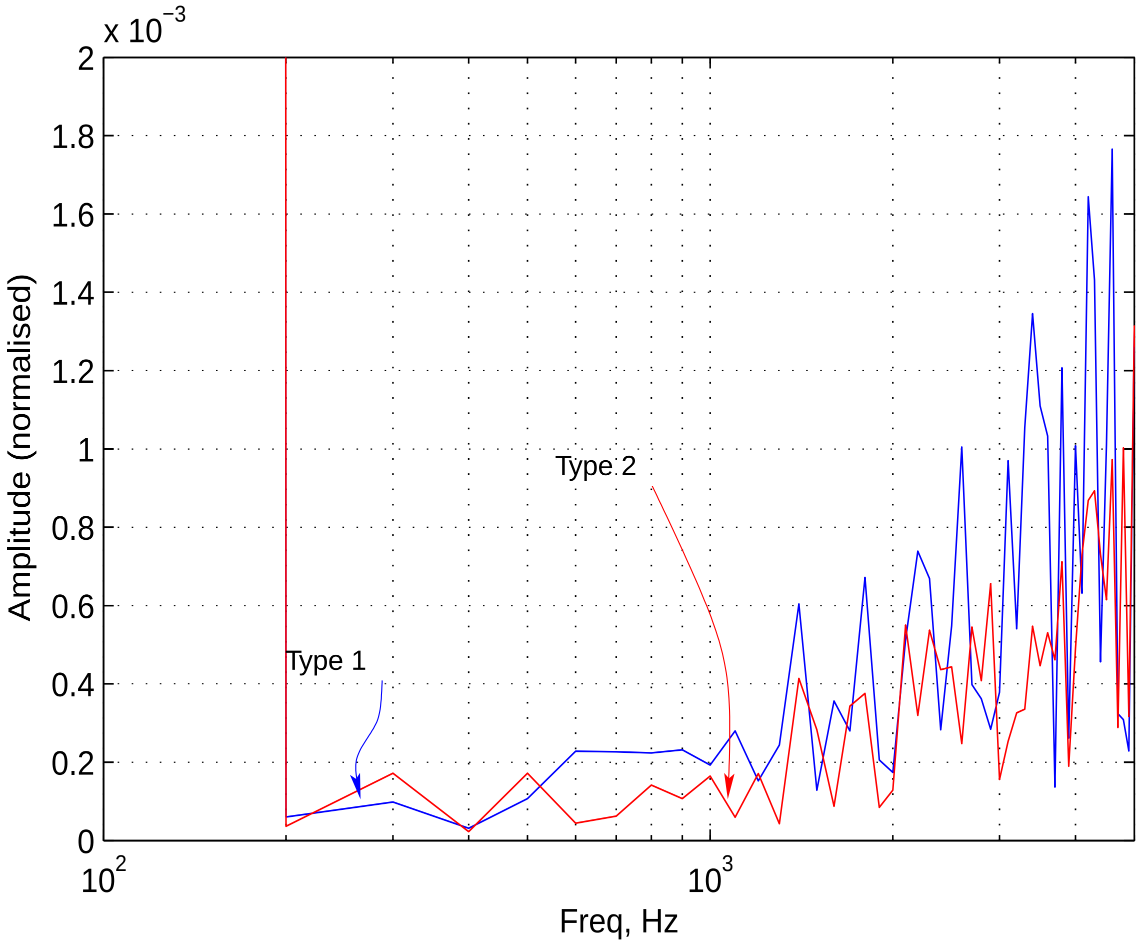


Figure 2.25: *Normalised spectra of Type 1 and Type 2 bit-stream subtractors outputs.*

Then

$$\begin{aligned} V_2(k) &= g\frac{D1_N(k)}{N} \\ V_2(k) &= \frac{(A_z - B_z + g\, D1_N(k))}{N} \\ V_2(k) &= \frac{((A_z + g\, D1_N(k)/2) - (B_z - g\, D1_N(k)/2))}{N} \end{aligned} \tag{2.18}$$

From (2.18), the magnitude can be modified by scaling the difference in quanta. If $g > 1$ then the difference is augmented which scales the bit-stream up and vice-versa when $1/g > 1$. Note that since it is implicit that the resolution is one part in N, the total number of quanta remains the same. In other words, $(A_z + g\, D1_N(k)/2) + (B_z - g\, D1_N(k)/2) = N$.

Scale-up

Figure 2.26 explains the scaling mechanism conceptually. The input bit-stream, S_i, is shown in the topmost trace. As was done in an earlier example, an arbitrary frame has been also shown using a dashed box. In the first frame one positive impulse (quantum) is shown. Note that the *zero* reference has been intentionally left out. This quantum can also be interpreted as the difference in the positive and negative quanta. To facilitate the explanation we will instead consider this single impulse to represent one quantum. Hence S_i has a value of 1. If another signal is 3 times greater in magnitude than S_i then it must have three times as many quanta. This can be achieved in two ways as shown in traces 2 and 3. If S_i is to be scaled up by a factor of $p = 3$ then for every incident additional quantum, $p - 1$ additional quanta are also inserted into the output bit-stream. Trace 2 shows this technique graphically. Since it is impossible to insert more than one quanta, the additional quanta have to be buffered. Alternatively, it is possible to spread the additional quanta over the entire frame. These two techniques will be discussed as Type 1 and Type 2 scalers.

Type 1 Scale Up

The schematic of the Type 1 scale-up element is shown in Figure 2.27. S_i is the input bit-stream, and the multi-bit scale-up value is presented at *Scale* input. The scaled output bit-stream is S_o. The schematic is shown to be partitioned into two functional sections: the input unit and a output unit. The functioning of the output unit has been discussed earlier. It is a generic device which maintains a record or a buffer of quanta yet to be output and inserts them into the *zero* reference appropriately. The input unit acts like a decoder and outputs a multi-bit word, v, in accordance with the state of the *zero* reference. The exact decoding logic, to produce $S_o = kS_i$, is shown

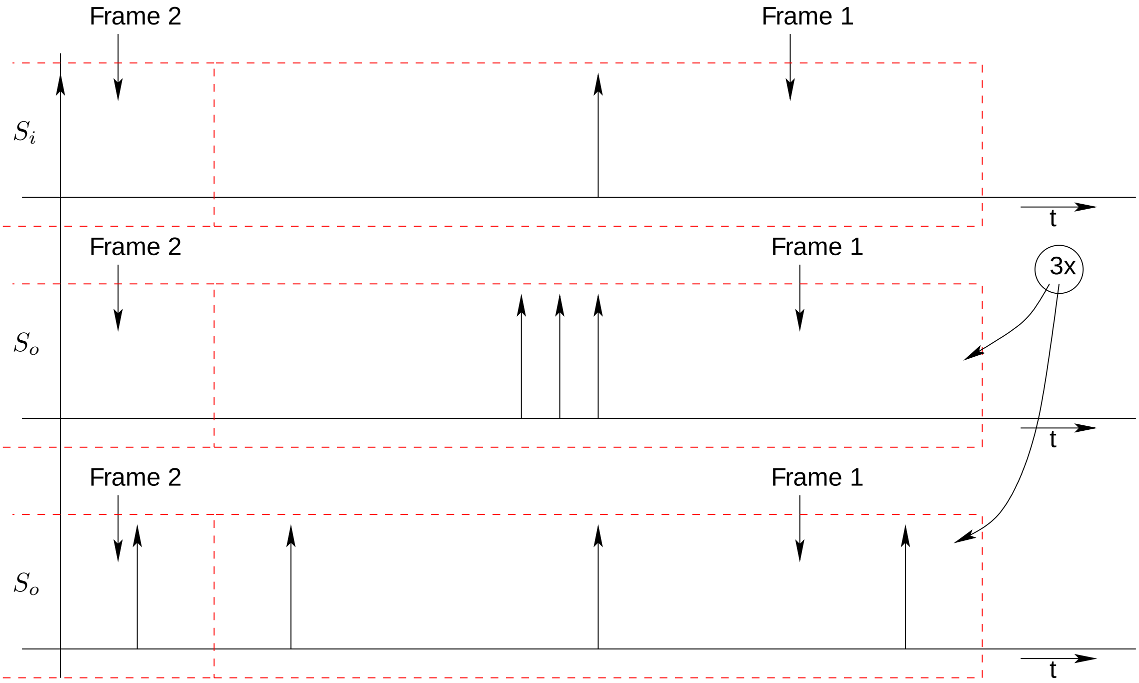


Figure 2.26: *Conceptual scaling of a bit-stream by 3.* $3S_i = S_o$

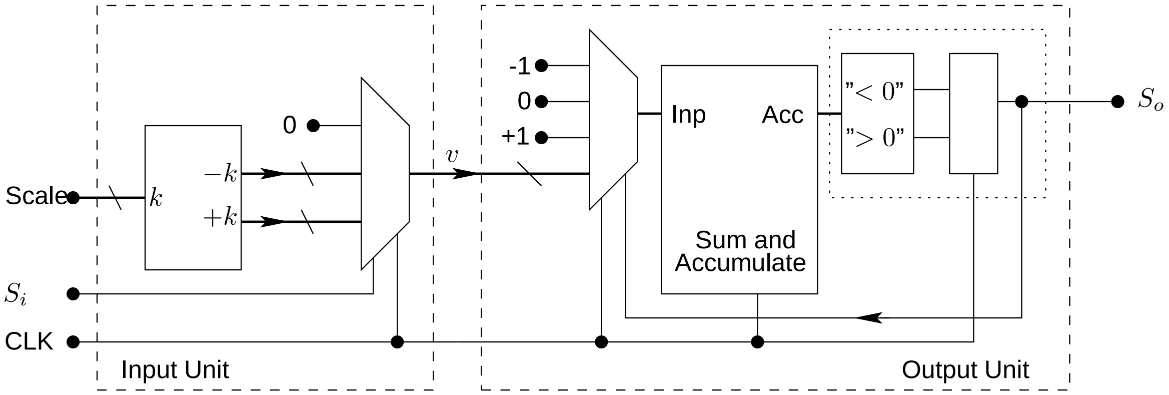


Figure 2.27: *Type 1 Scale Up for bit-streams.*

in Table 2.8. Here, the first column is the *zero* reference, the second is the input bit-stream, S_i, the third column is the output of the decoder, shown as v in Figure 2.27, and k is the amount of scaling.

A simulation of the time domain response of a scale-up element is shown in Figure 2.28. In this experiment $f_b = 1$MHz, $N = 256$, and the gain $k = 4$. The simulations have been normalised to range from -1 to +1. The input is a sinusoid with an amplitude of $30/128 = 0.2344$. Amplifying S_i by 4 gives $0.9375 = 120/128$. The simulations verify the expectations.

The impact of overdrive, or saturation on bit-stream adders has also been discussed earlier. In the case of a scale-up element the impact is similar. Figure 2.29 plots the behaviour of two bit-stream scale-up elements. Here the input is a sinusoid of amplitude $\frac{40}{128}$. This sinusoid is being

	z	S_i	v
1	0	0	0
2	0	1	k
3	1	0	-k
4	1	1	0

Table 2.8: *Scale Up decoder truth table.*

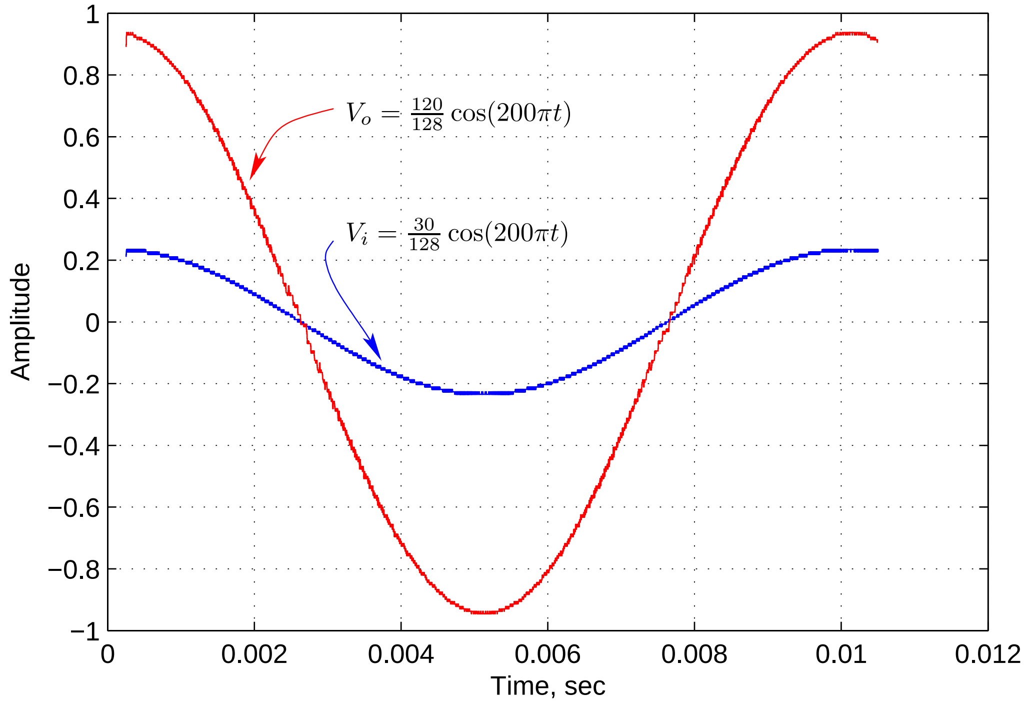


Figure 2.28: *Simulation of Type 1 Scale Up element.*

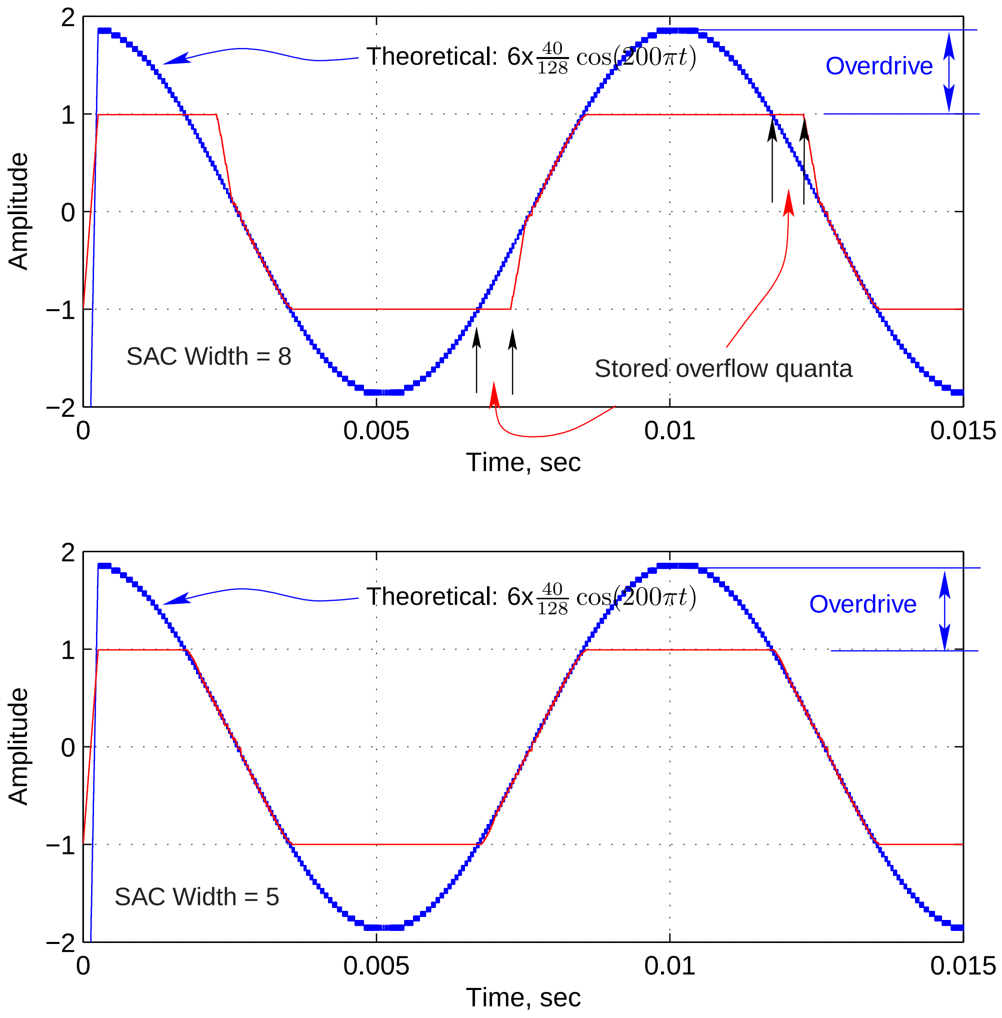


Figure 2.29: *Impact of SAC width on overdriven Type 1 Scale Up element.*

scaled up by 6. The top trace is the simulation of a scale-up element with its sum-and-accumulate element 8 bits wide. When the output reaches full scale there are no more free slots into which the stored quanta can be inserted and hence they keep accumulating in the SAC. A 8 bit SAC can accumulate (count) 128 negative quanta or 127 positive quanta. When the input decreases such that the output is now less than full scale, these stored quanta will find free slots. The output unit attempts to empty its buffer, aggressively, and hence each and every free slot will be populated with the stored quanta. The output will thus remain at full scale till each of the stored quanta has been output. This leads to an extension of the *flat top* which may not be acceptable in some applications.

The second trace plots the simulation results of a similar experiment but with a scale-up element built with a SAC 5 bits wide. This SAC can store only 16 negative quanta or 15 positive quanta and hence comes out of saturation much more quickly.

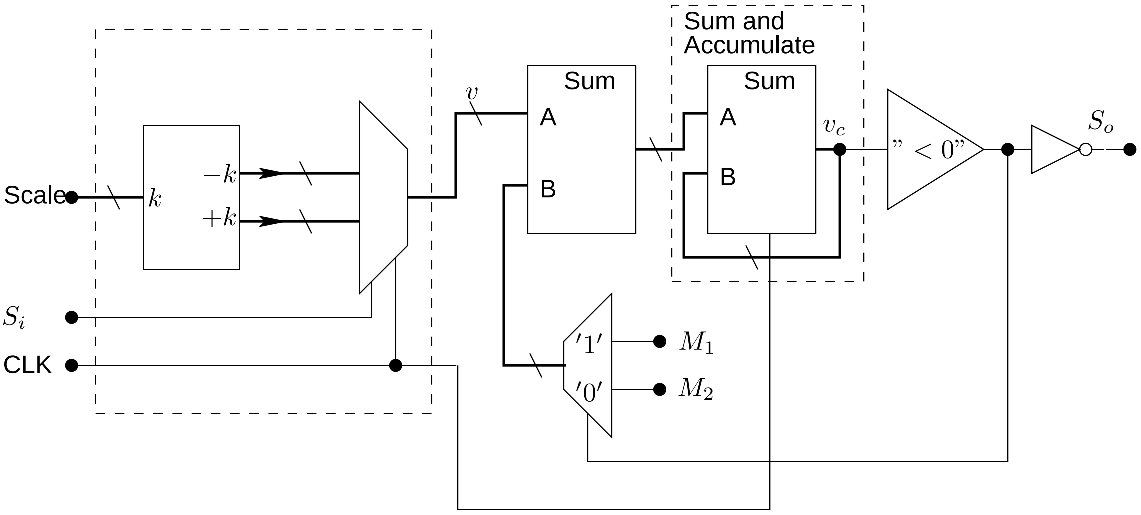


Figure 2.30: *Type 2 Scale Up for bit-streams.*

Type 2 Scale Up

The bit-stream to bit-stream modulator can easily be easily adapted to scale bit-streams. A schematic of this device is shown in Figure 2.30. This element is constructed with a front end which is identical to that of the Type 1 Scale Up element. Its decoding truth table is also identical (Table 2.8). The rest of the schematic is the feedback modular similar to those used in the other elements.

The analysis for the Type 2 Scale-up unit follows a similar line to that of the bit-stream to bit-stream modulator. The input bit-stream has a value of $D1_N/N$. Every quanta, positive and negative, results in k, or a $-k$ being deposited in the SAC. If the feedback were to be disabled the after N clocks the SAC will increment to

$$k(A_z + D1_N(k)/2) - k(B_z - D1_N(k)/2)$$

Since $A_z = B_z$, this reduces to

$$k \times D1_N(k)$$

When the feedback loop is closed, the inherent nature of the system will try to reduce the SAC to zero. If this is possible then assume it take X clocks of state M_1 and the remaining clocks, $N - X$ of state M_2 to reduce the SAC to zero.

$$k \times D1_N = XM_1 + (N - X)M_2$$

Re-arranging this results in

$$X = D1_N \frac{k}{\Delta M} - N \frac{M_2}{\Delta M} \tag{2.19}$$

Where $\Delta M = M_1 - M_2$. If $M_1 = -M_2 = 1$ then the total number of positive quanta is given by

$$X = D1_N \, \frac{k}{2} + \frac{N}{2} \tag{2.20}$$

If $S_i(k) = D_i/N$ is the input bit-stream to this scaler and the resulting output bit-stream is $S_o(k)$ then

$$S_o(k) = \frac{(A_z + D_o(k)/2) - (B_z - D_o(k)/2)}{N}$$

Then from (2.20)

$$S_o(k) = \frac{(X) - (N - X)}{N} \tag{2.21}$$

$$S_o(k) = \frac{(D_i \, \frac{k}{2} + \frac{N}{2}) - (N - (D_i \, \frac{k}{2} + \frac{N}{2})}{N} \tag{2.22}$$

$$S_o(k) = k\frac{D_i}{N} \tag{2.23}$$

The blue traces in Figure 2.33 shows the relationships for different $k/\Delta M$ ratios.

Scale down

Scaling a signal down is very similar to scaling a signal up in magnitude. In the case of scaling a signal up, the number of quanta was increased proportionally. If a bit-stream is to be attenuated then the number of additional quanta must be reduced by the amount the bit-stream is to be attenuated. Equation (2.18), repeated below clearly states this.

$$V_2(k) = \frac{((A_z + g \, D1_N(k)/2) - (B_z - g \, D1_N(k)/2))}{N}$$

If g is less than one then the number of additional quanta is reduced. This reduction can be achieve in several ways. Two techniques have been presented.

Type 1 Scale Down

The schematic for this techniques is shown in Figure 2.31. The front end is very similar to the other elements. The decoded output of the front end is counted in the SAC. Every additional incident quanta increments the SAC. Positive quanta cause positive increments and negative quanta lead to negative increments. Since the input is to be attenuated by k then the SAC count is compared to k. Affirmative comparisons lead to a quanta being output and the SAC being reset to zero.

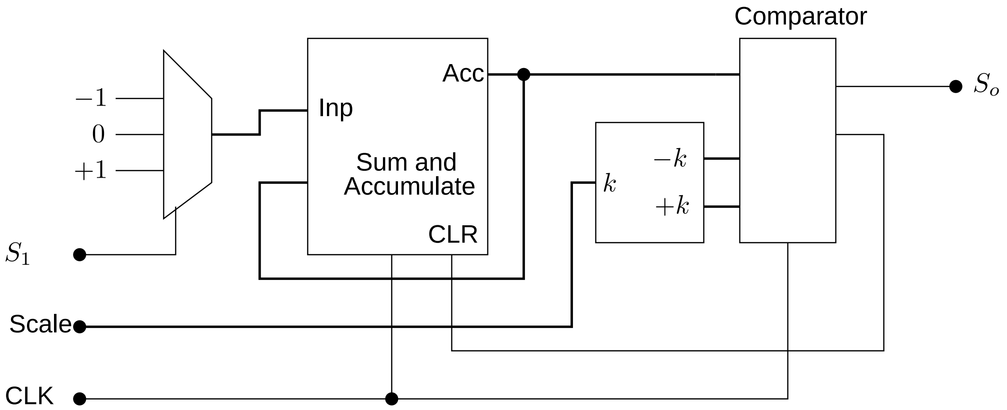


Figure 2.31: *Type 1 Scale Down element for bit-streams.*

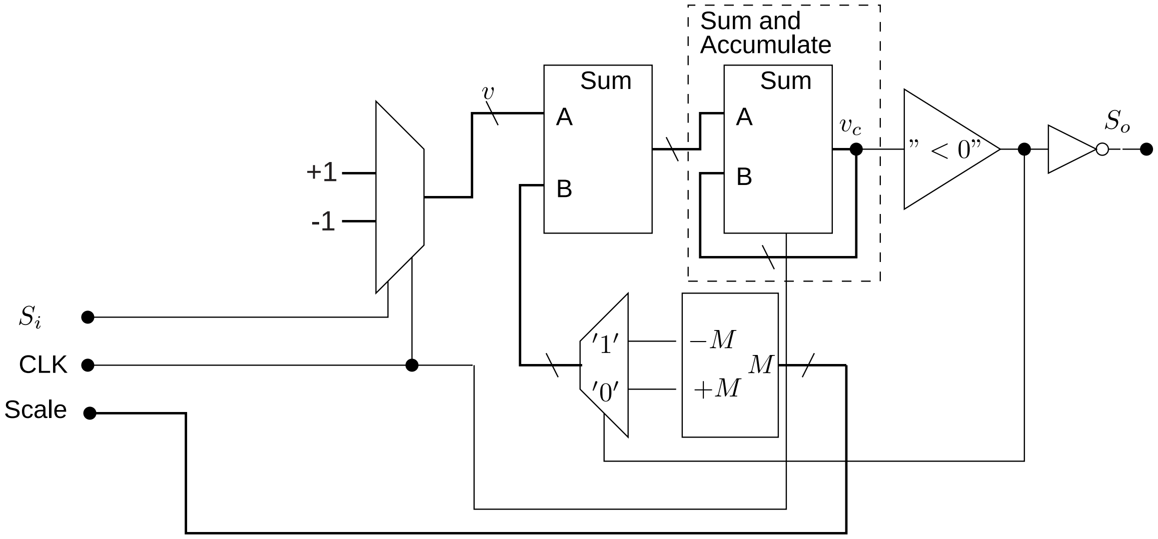


Figure 2.32: *Type 2 Scale Down element for bit-streams.*

Type 2 Scale Down

Examining (2.19), if $k = 1$ and $\Delta M > 1$ the (2.19) will attenuate the input bit-stream. The type 2 scale down element is shown in Figure 2.32. The red traces in Figure 2.33 shows the relationships for different $k/\Delta M$ ratios.

$$X = D1_N \frac{k}{\Delta M} - N \frac{M_2}{\Delta M}$$

Composite Scaling Element

Each of these scaling elements operate on integer scaling values. Fractional scaling is often required and the Type 2 scaling elements, Figure 2.30 and Figure 2.32, can be adapted to decrease the granularity in the scaling operation. Figure 2.34 shows the schematic which will achieve this.

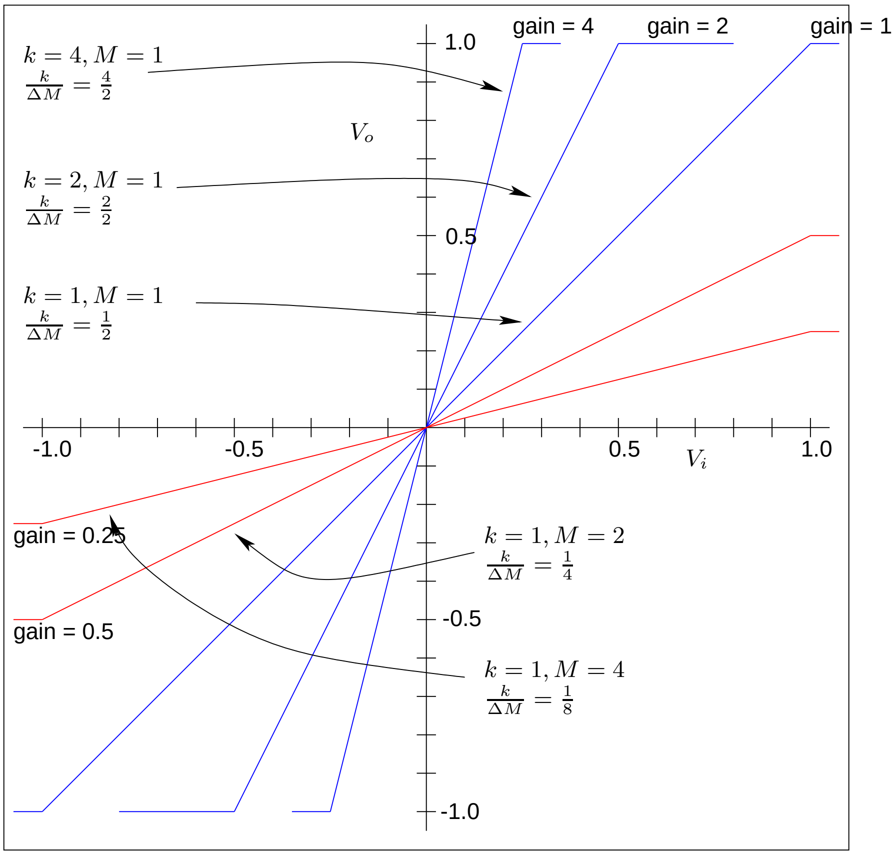


Figure 2.33: *Transfer function of Type 2 Scaling element.*

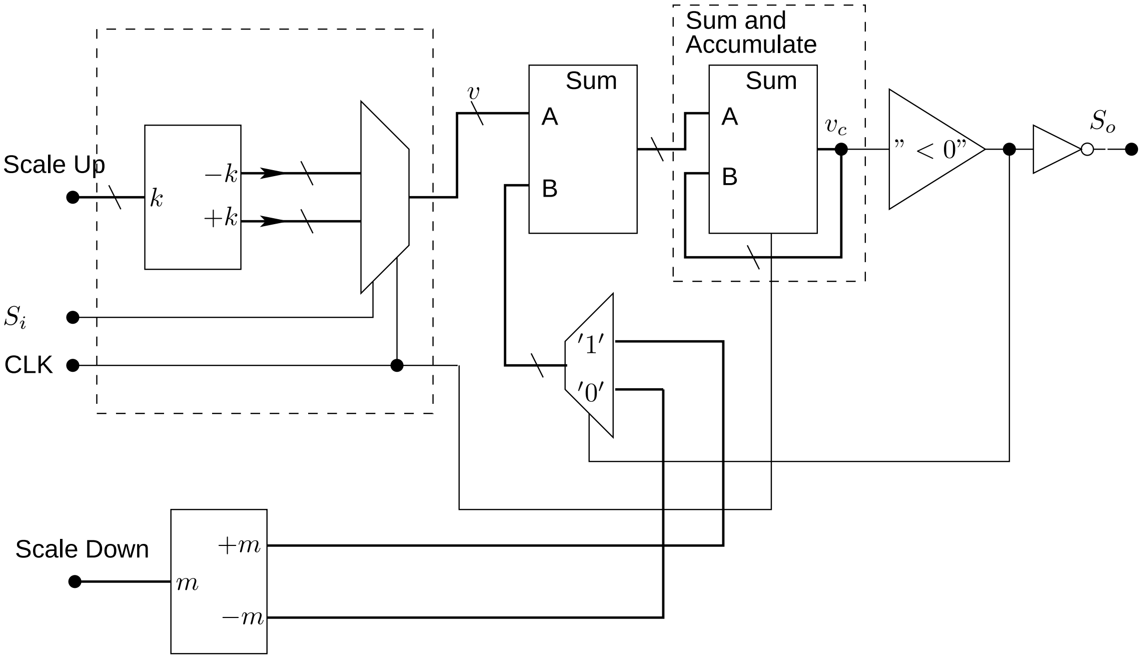


Figure 2.34: *Extended scaling element.*

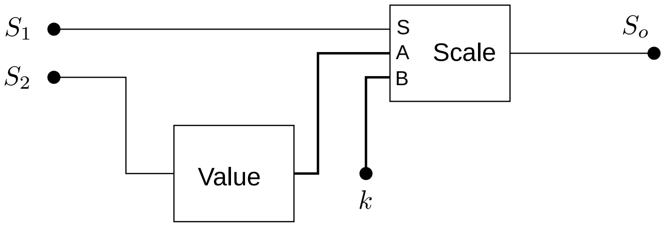


Figure 2.35: *A bit-stream multiplier.*

Here $M_1 = -M_2$ is always maintained. From (2.19) the expression for the number of output positive quanta produced by the schematic in Figure 2.34 is

$$X = D1_N \frac{k}{m} - N/2$$

The time domain behaviour is very similar to the other scaling elements and hence it has not been presented.

2.3.6 Multiplier

The value element and a scaler can be used to construct a bit-stream multiplier. The result of a multiplication requires twice the resolution of the operands. If the magnitude range of the inputs is known then, an alternative solution is to scale down the result to keep the product within the full scale limits. The composite scaler, with both a scale-up and a scale-down input is ideally suited for use in a multiplier. Figure 2.35 shows the schematic of the bit-stream multiplier with the operands, S_1 and S_2, and the resulting bit-stream S_3. The value of S_2 feeds into the scale-up terminal, labelled *A*, while the constant scale-down value is applied to the *B* terminal. This schematic has a transfer function given by

$$S_o = S_1 \frac{\text{value}(S_2)}{k}$$

Figure 2.36 shows the response to the multiplication of two sinusoids. Here the scale down constant is 16. By swapping the *A* and the *B* terminals a bit-stream divider is obtained.

$$S_o = S_1 \frac{k}{\text{value}(S_2)}$$

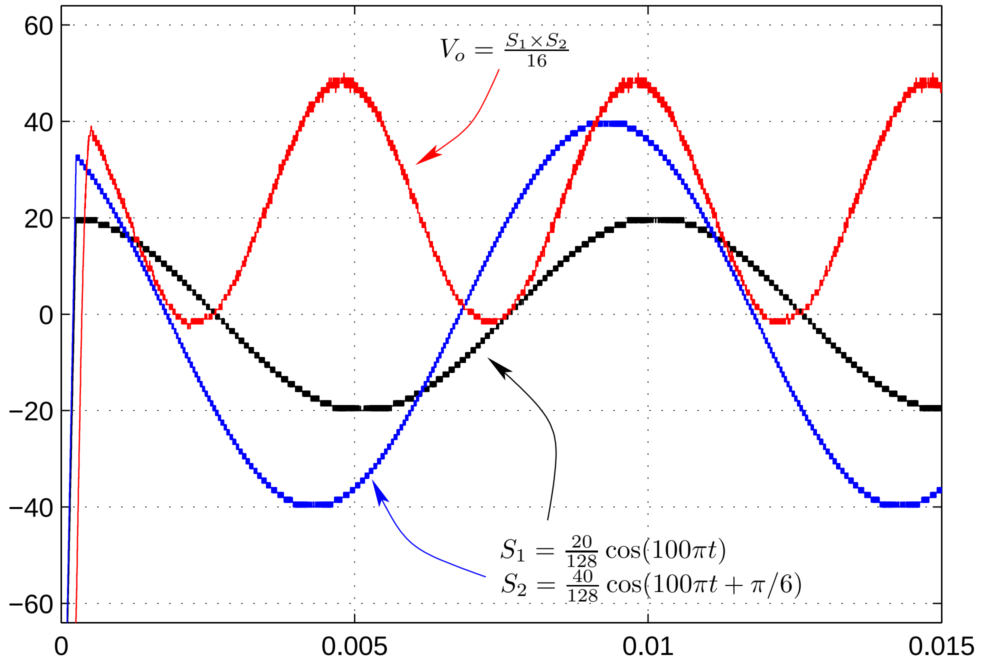


Figure 2.36: *Response of bit-stream multiplier.*

2.3.7 Tapped Delay

Figure 2.37 shows bit-streams S_i segmented into blocks that encapsulate quanta. Since each quantum arrives at a frequency f_b, it is trivial to see that it is possible to encapsulate sufficient quanta to represent a single operational sample time. It can be seen that these blocks of quanta may be stored and ejected at a later time as shown in bit-streams S_{i-1} and S_{i-2}.

Component Design

The basis of this component revolves around a bit-stream buffer. As each quantum is injected into this block, it is stored on a buffer of length x, where x is the discrete sampling time divided by bit-stream frequency f_b. This value must be calculated *a priori* and entered as a constant inside the delay block. A buffer of length x will perform a single discrete delay. If more delays are required, the buffer should be multiplied by the number of required delays. Figure 2.38 demonstrate this procedure when an incoming bit-stream y_{bs} enters the block and the buffer is *tapped* at certain points to output the delayed bit-streams $y_{bs}(q-x)$, $y_{bs}(q-2x)$ and $y_{bs}(q-3x)$.

To test the reliability of the tapped delay block, an input step function is applied. Figure 2.39 shows the bit-stream response for an input of 0.1. The bit-stream frequency f_b is 500kHz and the sampling time is $1ms$, thus making the buffer size 2×500 in size. The blue trace is the input, the green trace shows the signal which is delayed by one sample interval the red trace shows the signal delayed by two sample intervals. is the two sample delay.

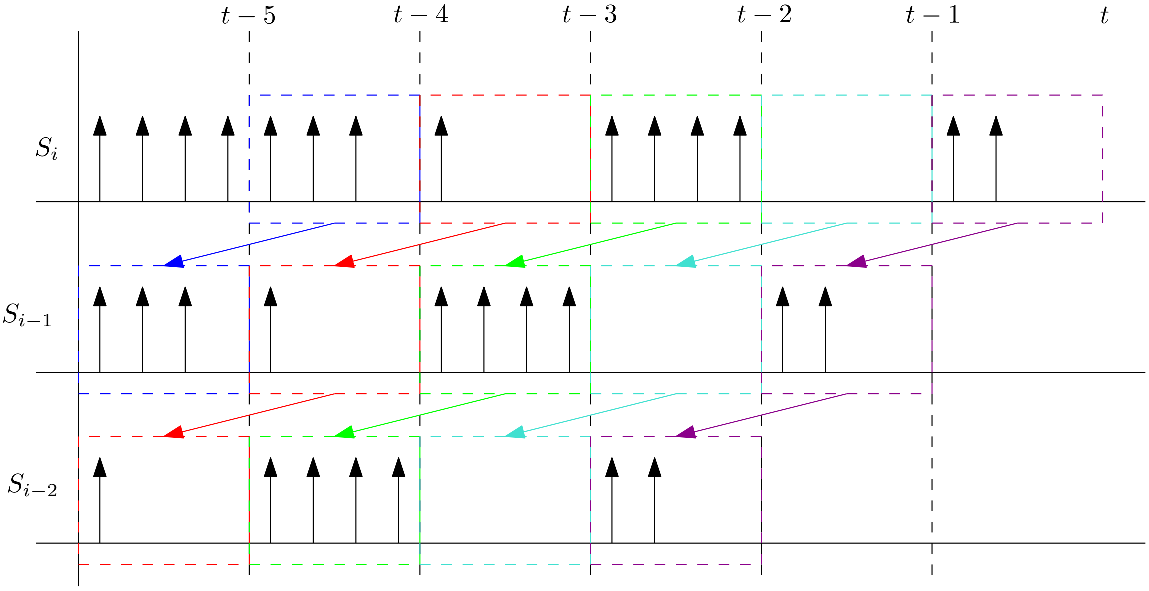


Figure 2.37: *Conceptual description of a Tapped Delay block for 1 and 2 sample delays.*

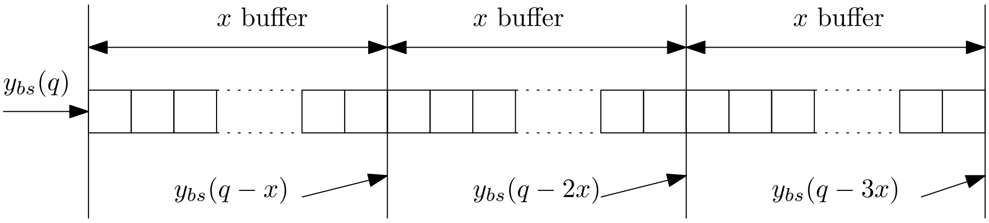


Figure 2.38: *Description of the buffer for a Tapped Delay block.*

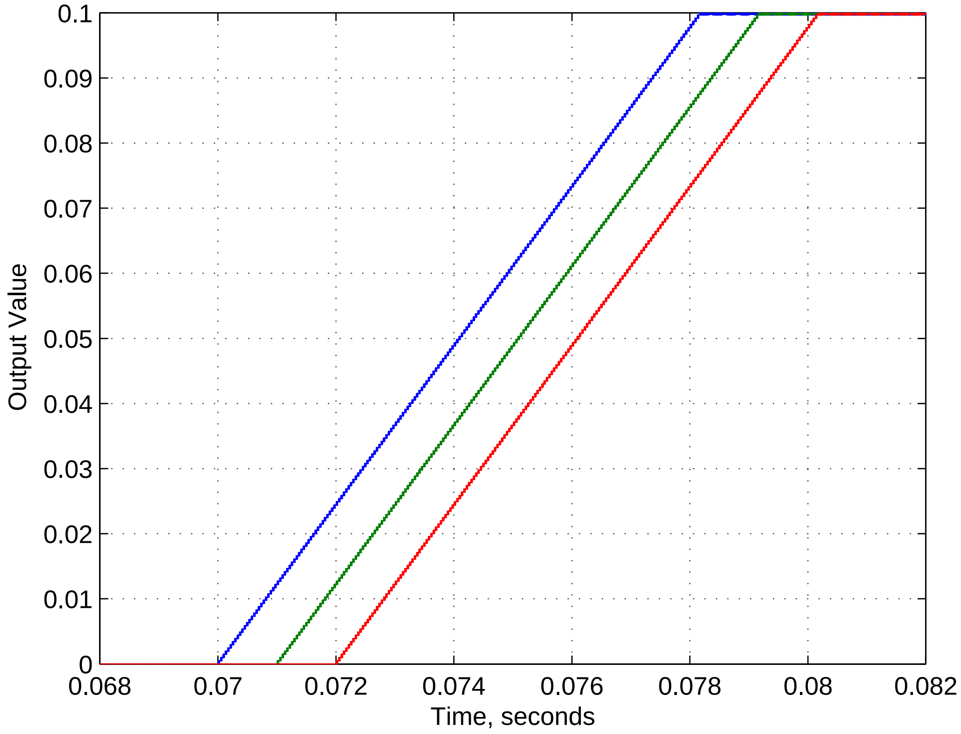


Figure 2.39: *Bit-stream Tapped Delay for* $t_s = 0.001sec$

To further test the response of this block a sinusoidal signal of amplitude 1 $1V_{pp}$ is used as an input. For this example, the sampling time is changed to $10ms$ whilst keeping f_b to 500kHz. This makes the size of the buffer equals to 2×5000 . The blue trace in Figure 2.40 shows the input sinusoid. The green and red trace of Figure 2.40 shows the delayed version of this signal.

2.4 Bit-stream Simulation of Linear and Nonlinear Systems

The focus of this section is to show the effectiveness of bit-streams in emulating the true behaviour of discrete systems. This section will use the tapped delay block described in $\S$2.3.7 extensively. A variety of systems will be tested ranging in complexity, from linear to nonlinear systems. These simulations are necessary to establish whether bit-stream can be considered as a possible alternative to implement controllers in future.

2.4.1 Guidelines for Bit-Stream Implementation

The following guidelines are to be followed to successfully implement bit-stream.

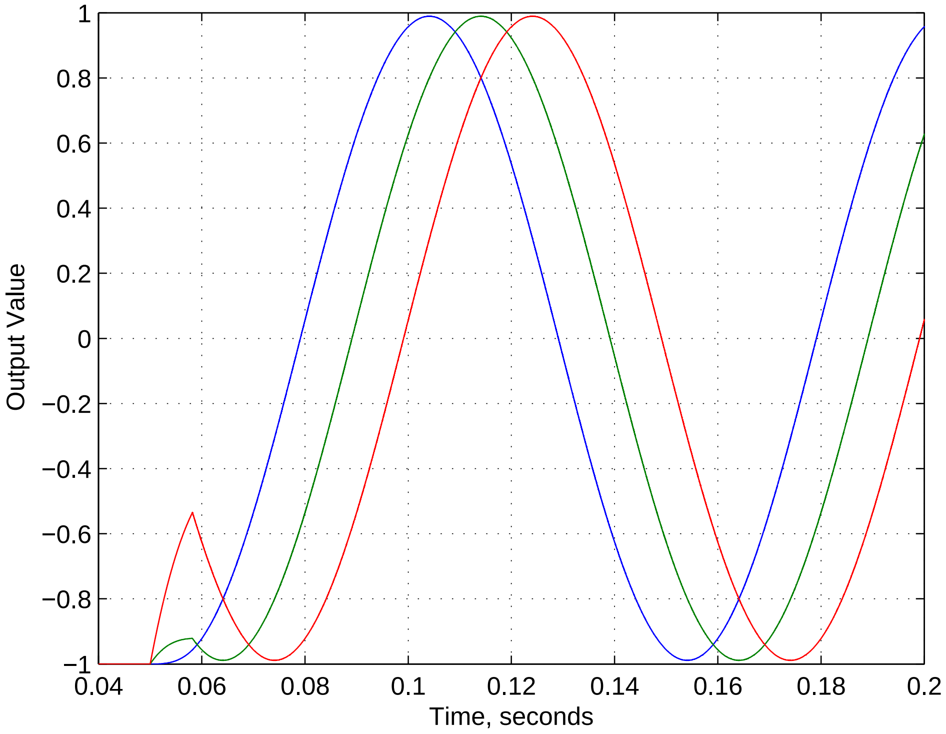


Figure 2.40: *Bit-stream Tapped Delay for* $t_s = 0.01sec$ *and sinusoidal input.*

- Various signals of the system which are used for control law calculations should be normalised to ensure that they lie within the range of the bit-stream

- The levels of control signals should be limited to avoid saturation of various bit-stream functional elements

- When more than one bit-stream signals are mathematically manipulated using bit-stream based adders/multipliers *et cetera*, the signal to noise ratio will deteriorate as a result of bit-clumping.

- The bit rate f_b of the bit-stream should be several times faster than the sampling frequency of the system f_s. Normally select $f_b = OSR \cdot 2 \cdot f_B$ where $f_B =$ bandwidth of the system. Typical OSR (over sampling ratio) values are 128, 256, 512 *et cetera*. This is necessary to achieve appropriate slew rate.

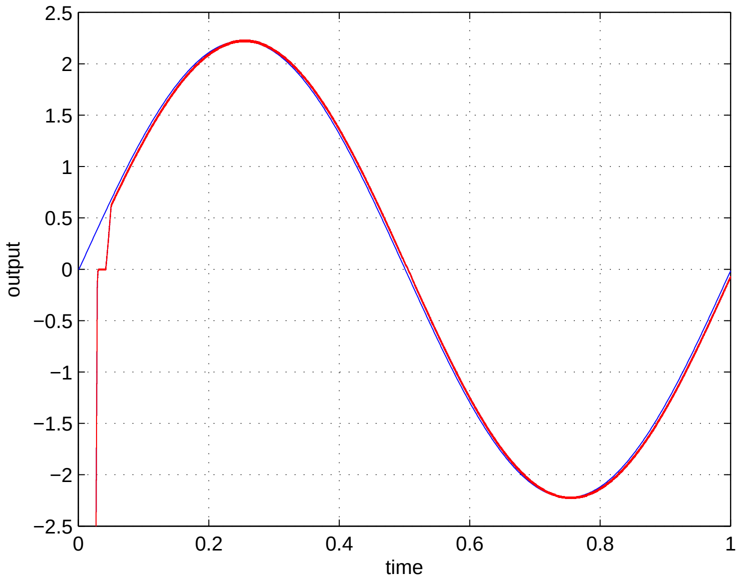


Figure 2.41: *Output response of a linear system implemented using multi-bit and bit-stream for a 1Hz sinusoidal input: Blue trace corresponds to multi-bit and the red trace corresponds to bit-streams.*

2.4.2 Examples

Initially a discrete first order system given by

$$y(k) = 0.1y(k-1) + 2.0u(k-1) \tag{2.24}$$

(2.24) is simulated using bit-stream by applying two sinusoidal inputs of unity amplitude of varying frequency; one with a frequency of 1Hz and another of 5Hz. The range of the bit-stream is set to $\pm$10 to prevent saturation. The implementation requires 2 bit-stream delay blocks to represent $y(k-1)$ and $u(k-1)$, 2 scaling blocks to represent the coefficients 0.1 and 2 and one summation block. The output response is compared to that obtained from multi-bit implementation and is shown in Figure 2.41 and Figure 2.42. From the figure, it is observed that the response of the system in bit-stream matched with that of multi-bit implementation.

To further illustrate that bit-stream can accurately emulate the response of a system, a discrete second order is considered. The system is described by

$$y(t) = 0.1y(t-1) - 0.5y(t-2) + 2.0u(t-1) - 0.1u(t-2) \tag{2.25}$$

The implementation requires 4 bit-stream delay blocks to represent $y(k-1)$, and $y(k-2)$, $u(k-1)$, $u(k-2)$ and 4 scaling blocks to represent the coefficients 0.1 ,-0.5, 2, -0.1 and three summation block. The output response of the system to the same sinusoidal inputs from the bit-stream is compared with that of the multi-bit and is shown in Figure 2.43 and Figure 2.44. After the short initial disruption which is caused by the bit-streams delays, the response from the multi-bit and

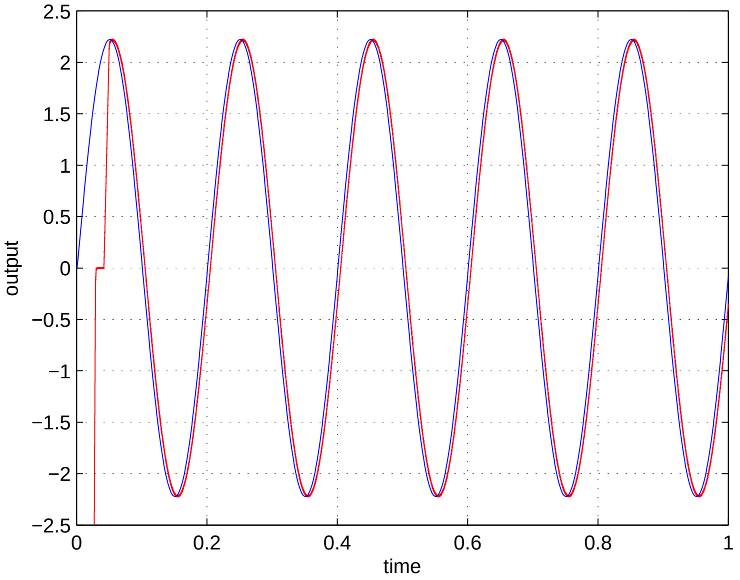


Figure 2.42: *Output response of a linear system implemented using multi-bit and bit-stream for a 5Hz sinusoidal input: Blue trace corresponds to multi-bit and the red trace corresponds to bit-streams.*

the bit-streams is virtually identical.

Finally a polynomial NARX model, which is a subset of NARMAX model is considered. The system is described by

$$y(k) = 0.1y(k-1) + u(k-1) - 0.1u^{(}k-1) - 0.5y(k-1)u(k-1) \tag{2.26}$$

The output response is computed by applying the same inputs in both bit-stream and multi-bit environment. The functional blocks used for this system include: 2 delays for $y(k-1)$ and $u(k-1)$ and 4 scale block for each parameter. The terms that contain bit-stream products are evaluated using the *multiplier* block. The performance of this block is understandably limited, since product terms escalate exponentially and may saturate from an early stage. Amongst the several adjustments that had to be made to compensate for this saturation, the most important is to include *pre-scaling* and *post-scaling* bit-stream blocks. Extreme care had to be taken not modify the internal mathematical formulations. This reduced the risk of saturation but also contributed to the noisy variations. The results of simulation are shown in Figure 2.45 and Figure 2.46. The response is shown to be noisy, which is an inherent property of bit-streams and its caused by the combination and large amalgamation of several functional blocks. Furthermore, as the complexity of the system increases, so does the sensitivity of the parameters. Though the response is noisy, the overall shape is maintained. In practice, bit-streams would be used to drive a continuous system, where small variations can be ignored.

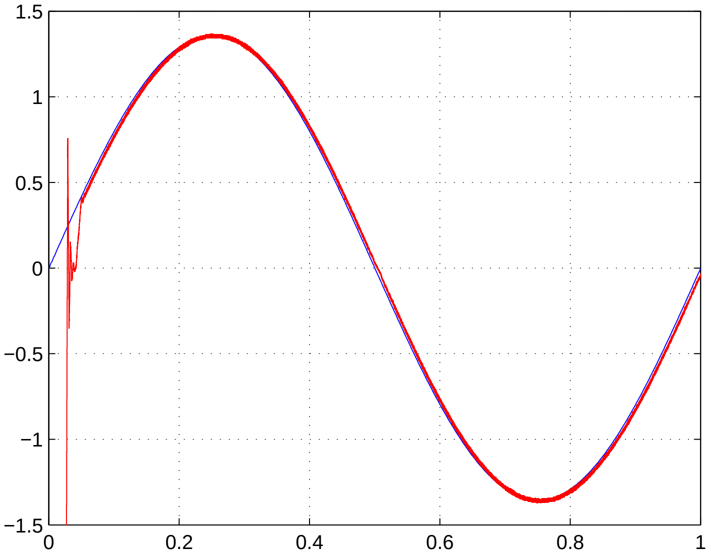


Figure 2.43: *Output response of a second order linear system implemented using multi-bit and bit-stream for a 1Hz sinusoidal input: Blue trace corresponds to multi-bit and the red trace corresponds to bit-streams.*

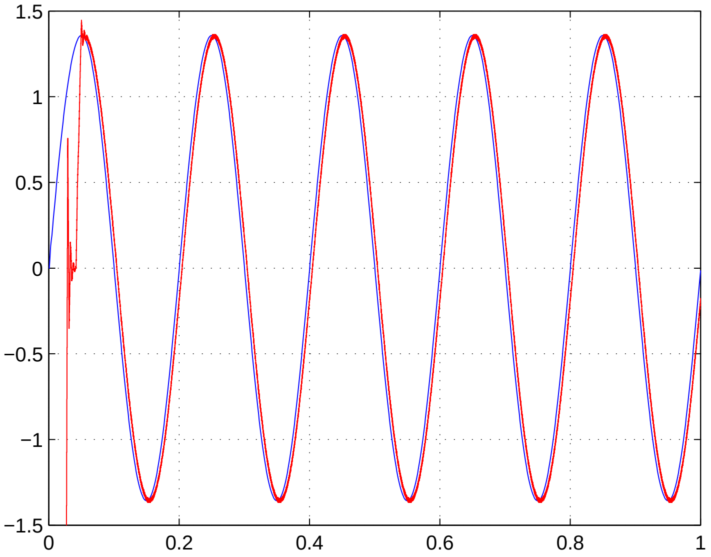


Figure 2.44: *Output response of a second order linear system implemented using multi-bit and bit-stream for a 5Hz sinusoidal input: Blue trace corresponds to multi-bit and the red trace corresponds to bit-streams.*

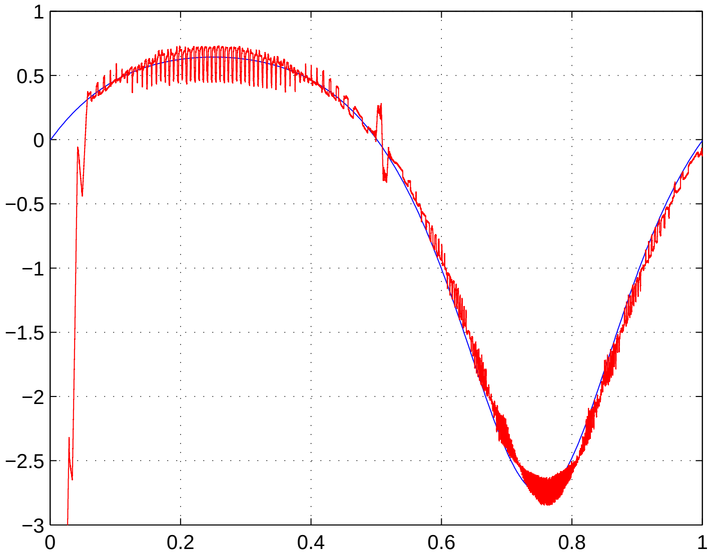


Figure 2.45: *Output response of a polynomial NARX system implemented using multi-bit and bit-stream for a 1Hz sinusoidal input: Blue trace corresponds to multi-bit and the red trace corresponds to bit-streams.*

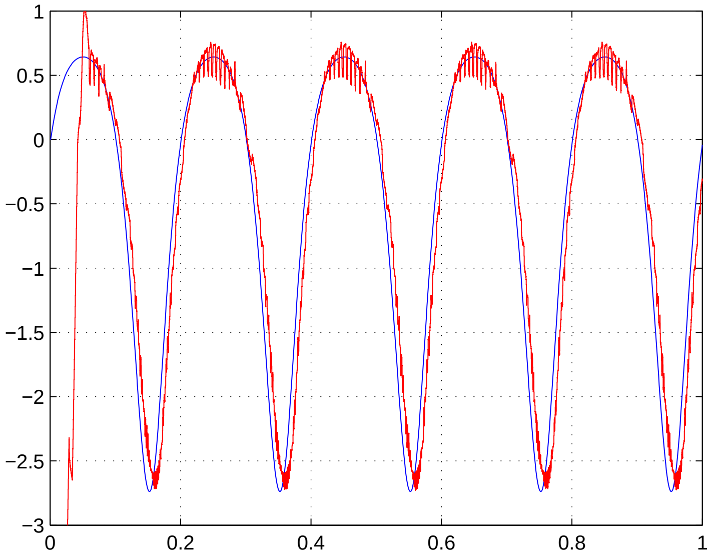


Figure 2.46: *Output response of a polynomial NARX system implemented using multi-bit and bit-stream for a 5Hz sinusoidal input: Blue trace corresponds to multi-bit and the red trace corresponds to bit-streams.*

2.5 Conclusions

This chapter presents an introduction to the use of bit-stream encoding for analogue signals. The concept of bit-stream is derived and presented from its fundamental building blocks by establishing the idea of *quantum* or *quanta*. This fundamental concept can be used to encode analogue signals into a single digital line, or bit-stream. One of the advantages of a bit-stream paradigm is its ability to perform mathematical operations based solely upon the manipulation of quanta. That is, functional blocks that operate only on the principle of bit-streams have been presented here. These operators include summation, substraction, scaling and multiplication. Their mathematical and physical proof has been included. Considering the overall theme of this thesis, the performance of bit-stream is illustrated by computing output response of both linear and non-linear systems. The implementation procedures are clearly outlined and the simulation results show satisfactory response for each of the systems compared to their multi-bit counterpart. This design methodology ensures predictable behaviour of the functional blocks for more complex arrangements. It is this combination of multiple functional elements that will serve as the basis for implementing new types of controllers in the next few chapters.

CHAPTER 3

Bitstream based Generalised Predictive Control for Linear Systems

3.1 Introduction

Digital control has gained popularity during the last several decades with most of the current control systems based on digital techniques which are implemented using microcontrollers and digital signal processors (DSPs). All these microcontrollers and DSPs are essentially serial, or multi-bit, in nature and are usually optimised for fixed width binary words ranging from 8 bits to 32 bits [59]. The choice of appropriate word size is decided by the required dynamic range, the signal-to-noise ratio, the complexity of the control algorithm and the cost of the associated development tool. Although controller implementation using serial processors do not pose significant difficulties for single-input-single-output (SISO) systems, the serial strategy offers considerable disadvantage while controlling complex multi-input-multi-output (MIMO) systems.

An alternate method of implementing digital controllers is based on the concept of bit-stream which has recently been proposed in [17]. The bit-streams are inherently parallel in nature. In this method, instead of using a microprocessor to implement the control functions, the controller is implemented in hardware using bit-streams inside programmable logic devices such as Field Programmable Gate Arrays (FPGAs). This technique differs from traditional digital implementations in that each signal is represented by a single bit wide digital signal instead of a multiple bit integer. Moreover, since all control elements are implemented in parallel, the addition of extra functionality to a given design will consume extra silicon area with little impact on the timing of the system unlike microcontroller based systems which execute control functions sequentially. This may exceed the available execution time as more functionality is added.

Recently, bit-stream based systems have been developed for various applications which include servo control [67], neural network applications [59], brushless D.C. motor control [21], variable frequency sine wave generation [60], control of induction motor [21] *et cetera*. The bit-stream based controllers designed to date are limited to proportional and integral controllers. The objective of the present study is to investigate whether the bit-stream principle can be used to implement more sophisticated control method such as generalised predictive control (GPC). The GPC has been developed extensively during past three decades and is considered as one of the most popular controller next to the Proportional-Integral-Derivative (PID) controllers by the industry due to numerous advantages this offer [25, 68, 69, 24]. In this chapter, the performance of a GPC has been investigated considering examples of simple linear system and a linear system with delay.

3.2 Generalised Predictive Control

The generalised predictive control algorithm proposed in [25] is briefly reviewed here for sake of completeness. The linearised model of most single-input-single-output (SISO) plants, around a particular operating point, can often be described by a Controlled Auto-Regressive Integrated Moving Average (CARIMA) model [25].

$$A(q^{-1})y(t) = B(q^{-1})q^{-d}u(t-1) + \frac{C(q^{-1})}{\Delta}e(t) \tag{3.1}$$

where Δ is the differencing operator $1 - q^{-1}$, d is the dead time of the system, $u(t)$ and $y(t)$ represent respectively the control and output sequences of the plant, and $e(t)$ is a zero mean white noise sequence. The polynomials A, B and C are given by:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + a_2q^{-2} + ... + a_{na}q^{-na} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + b_2q^{-2} + ... + b_{nb}q^{-nb} \\ C(q^{-1}) &= 1 + c_1q^{-1} + c_2q^{-2} + ... + c_{nc}q^{-nc} \end{aligned} \tag{3.2}$$

The Generalised Predictive Control (GPC) algorithm essentially applies a control sequence by minimising a cost function of the form:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} e^2(t+j|t) + \sum_{j=1}^{N_u} \lambda\Delta u^2(t+j|t) \tag{3.3}$$

where N_1 and N_2 are the minimum and maximum costing horizons, N_u is the control horizon with $N_1 \geq 1$, $N_2 \geq N_1$ and $1 \leq N_u \leq N_2$, λ is the positive weight coefficient and $e(t+j|t) = \hat{y}(t+j|t) - w(t+j)$ where $\hat{y}(t+j|t)$ is the optimum j step ahead prediction of the system output, given data up to time t and $w(t+j)$ is the future reference trajectory.

The objective of the predictive control is to obtain the future control input sequence $u(t)$,$u(t+1)$,. . . , such that the future plant output $y(t+j)$ successfully tracks $w(t+j)$. This is accomplished by minimising $J(N_1, N_2, N_u)$ of (3.3). The j-step ahead prediction of the output can be expressed as [25]

$$\hat{y}(t+j|t) = G_j \Delta u(t+j-d-1) + G'(q^{-1})\Delta u(t-1) + F_j(z^{-1})y(t) \tag{3.4}$$

where polynomials $G_j(q^{-1})$, $G'_j(q^{-1})$ and $F_j(q^{-1})$ are obtained recursively by solving the following Diophantine equations:

$$C(q^{-1}) = E(q^{-1})A(q^{-1})\Delta + q^{-j}F_j(q^{-1}) \tag{3.5}$$

$$B(q^{-1})E(q^{-1}) = G_j(q^{-1})C(q^{-1}) + q^{-j}G'(q^{-1}) \tag{3.6}$$

where the solutions for these polynomial equations (for $j = 1, \ldots, N$) are given by [70]

$$F(q^{-1}) = f_0^j + f_1^j q^{-1} + \ldots + f_{na}^j q^{-na}, \tag{3.7}$$

$$G(q^{-1}) = g_0^j + g_1^j q^{-1} + \ldots + g_{j-1}^j q^{-(j-1)}, \tag{3.8}$$

$$E(q^{-1}) = e_0^j + e_1^j q^{-1} + \ldots + e_{j-1}^j q^{-(j-1)}, \tag{3.9}$$

$$G'(q^{-1}) = g'^j_0 + g'^j_1 q^{-1} + \ldots + g'^j_{nb-1} q^{-(nb-1)} \tag{3.10}$$

The future output predictions when expressed in vector form becomes

$$\mathbf{\hat{y}_1} = \mathbf{G\Delta u} + \mathbf{F}(q^{-1})y(t) + \mathbf{G}'(z^{-1})\Delta u(t-1) \tag{3.11}$$

where,

$$\mathbf{\hat{y}_1} = \begin{bmatrix} \hat{y}(t+N_1|t) \\ \hat{y}(t+N_1+1|t) \\ \vdots \\ \hat{y}(t+N_2|t) \end{bmatrix} \tag{3.12}$$

$$\mathbf{\Delta u} = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N_u-1) \end{bmatrix} \tag{3.13}$$

$$\mathbf{G} = \begin{bmatrix} g_{N_1} & g_{N_1-1} & \cdots & g_{N_1-N_u+1} \\ g_{N_1+1} & g_{N_1} & \cdots & g_{N_1-N_u+2} \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2} & g_{N_2-1} & \cdots & g_{N_2-N_u+1} \end{bmatrix} \tag{3.14}$$

$$\mathbf{G}'(q^{-1}) = \begin{bmatrix} G'_{N_1}(q^{-1}) \\ G'_{N_1+1}(q^{-1}) \\ \vdots \\ G'_{N_2}(q^{-1}) \end{bmatrix} \tag{3.15}$$

$$\mathbf{F}(q^{-1}) = \begin{bmatrix} F_{N_1}(q^{-1}) \\ F_{N_1+1}(q^{-1}) \\ \vdots \\ F_{N_2}(q^{-1}) \end{bmatrix} \tag{3.16}$$

Note that the last two terms of (3.11) depend only on the previous states and therefore can be grouped into one term, denoted as $\mathbf{f}$. Therefore (3.11) can be expressed as:

$$\hat{\mathbf{y}}_\mathbf{1} = \mathbf{G}\boldsymbol{\Delta}\mathbf{u} + \mathbf{f} \tag{3.17}$$

The cost function of (3.17) therefore can be expressed as

$$J = (\mathbf{G}\boldsymbol{\Delta}\mathbf{u} + \mathbf{f} - \mathbf{w})^T(\mathbf{G}\boldsymbol{\Delta}\mathbf{u} + \mathbf{f} - \mathbf{w}) + \lambda\boldsymbol{\Delta}\mathbf{u}^T\boldsymbol{\Delta}\mathbf{u} \tag{3.18}$$

where $\mathbf{w} = [w(t+N_1), w(t+N_1+1), ..., w(t+N_2)]$. This equation can be re-written as shown in (3.19)

$$J = \frac{1}{2}\boldsymbol{\Delta}\mathbf{u}^T\mathbf{H}\boldsymbol{\Delta}\mathbf{u} + \mathbf{b}^T\boldsymbol{\Delta}\mathbf{u} + \mathbf{f_0} \tag{3.19}$$

where $\mathbf{H} = 2(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})$, $\mathbf{b}^T = 2(\mathbf{f} - \mathbf{w})^T\mathbf{G}$ and $\mathbf{f_0} = (\mathbf{f} - \mathbf{w})^T(\mathbf{f} - \mathbf{w})$. Assuming that there are no constraints on the control signals, the minimum value of J can be found by making the gradient of J equal to zero, which gives

$$\begin{aligned} \mathbf{u} &= -\mathbf{H}^{-1}\mathbf{b} \\ &= (\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I})^{-1}\mathbf{G}^T(\mathbf{w} - \mathbf{f}) \end{aligned} \tag{3.20}$$

Using only the first row of matrix $\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}$, the output control law becomes

$$u(t) = u(t-1) + \mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}(\mathbf{w} - \mathbf{f}) \tag{3.21}$$

3.3 Investigation Procedure

The feasibility and effectiveness of the bit-stream based GPC is illustrated by considering two examples. The first example considered in this study is a D.C. servo mechanism which represents a linear system and the second example is a thermal system which represents a linear system with time delay. The performance of the controller is investigated using the following procedure.

- Initially, the standard GPC is implemented in an idealised form in Matlab$^{\text{TM}}$using the mathematical model of the system and the tuning parameters are selected by trial and error.

- In the second stage, the bit-stream based GPC is implemented using Mentor Graphics' ModelSim$^{\text{TM}}$. The performance of the bit-stream based GPC wasis compared with that obtained from ideal GPC implemented in Matlab$^{\text{TM}}$.

- In the third stage, the bit-stream based GPC is implemented in the real system. Since the design of the GPC is based on a discrete parametric model of the system, discrete models are fitted to the system using standard procedure of system identification. Furthermore, the optimal values of some of the tuning parameters of GPC such as control horizon, prediction horizon and weighting factor are obtained using a genetic algorithm.

- After obtaining the model and the tuning parameters, experimental results are obtained by implementing the GPC in Simulink$^{\text{TM}}$/dSpace$^{\text{TM}}$and are compared with those of bit-stream based GPC.

3.4 Case Study 1: Design of GPC for Speed Control of a D.C. Motor

The dynamics of the D.C. motor considered in this study is described by [71]

$$\frac{di_a}{dt} = -\frac{R_a}{L_a} i_a(t) - \frac{K_b}{L_a}\omega(t) + v_{app}(t) \tag{3.22}$$

$$\frac{d\omega}{dt} = -\frac{1}{J} K_f \omega(t) + \frac{1}{J} K_m i_a(t) \tag{3.23}$$

where,

v_{app} is the input voltage applied to the system.

$i_a(t)$ is the armature current.

$\omega(t)$ represents the core angular velocity.

R_a represents the armature resistance.

L_a represents the inductance of the armature

K_b represents the back emf constant

K_f represents the field constant.

K_m represents the torque constant

J is the inertia of the motor.

The present simulation is carried out considering $R_a = 2\Omega$, $L_a = 0.5$H, $J = 0.02$kg.m^2, $K_m = 0.015$N.m/A, $K_b = 0.015$V.s/rad and $K_f = 0.2$N.m.s.

The discrete equivalent of the system at sampling frequency of 20Hz gives the following A and B polynomials of the CARIMA model of (3.1):

$$\begin{aligned} A(q^{-1}) &= 1 - 1.4252q^{-1} + 0.4966q^{-2} \\ B(q^{-1}) &= 0.0015 + 0.0012q^{-1} \\ C(q^{-1}) &= 1.0 \end{aligned}$$

The success of GPC is dependent on the careful selection of the control and prediction horizons, and control weighting sequence λ. Several methods to optimise these parameters can be found in relevant literature [68]. However, since the focus of the initial task is to affirm if GPC is a viable control mechanism for D.C. motor control, initial values for the GPC tuning algorithm are selected by trial and error until satisfactory results are obtained. In the present example, the prediction horizon N_2 and control horizon N_u is set to 5 and 2 respectively. Control-weighting sequence factor is set to $\lambda = 0.03$. The gain matrices K_r and K_{gpc} are computed. The matrix K_r is defined as follows:

$$K_R = \left(G^T G + \lambda I\right)^{-1} G^T \tag{3.24}$$

$$K_r = K_R[1, 1 \dots N] \tag{3.25}$$

$$K_r = \left[\begin{array}{ccccc} 0.0489 & 0.1567 & 0.2855 & 0.4156 & 0.5371 \end{array}\right] \tag{3.26}$$

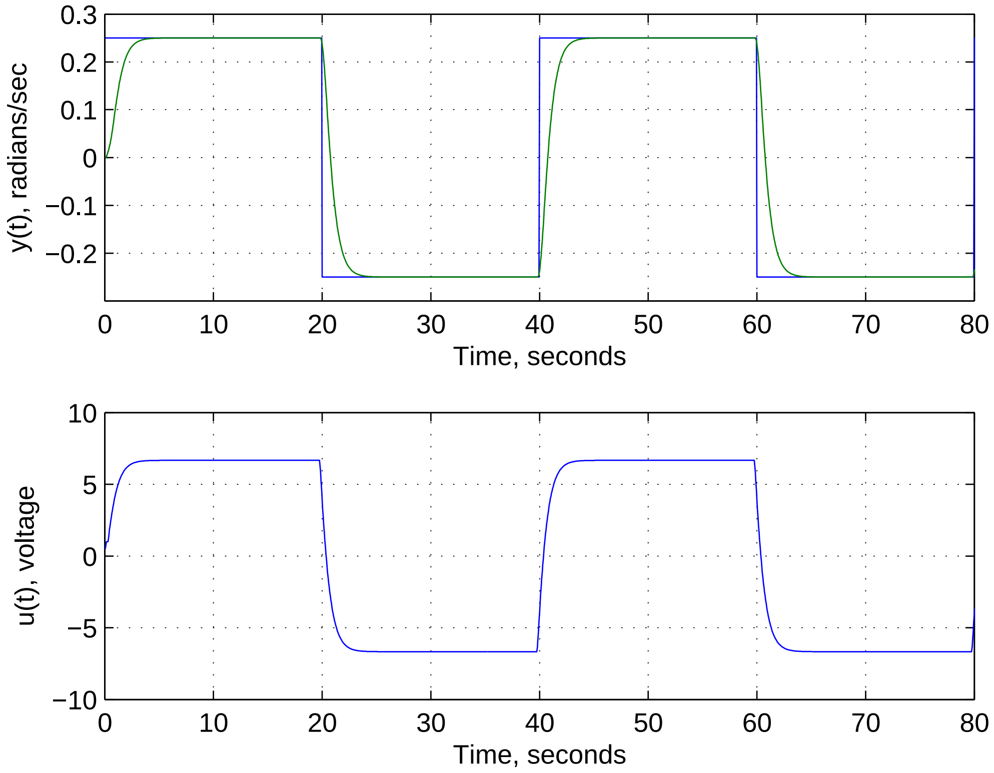


Figure 3.1: *Speed control of a D.C. servo mechanism using multi-bit GPC: Upper plot shows the output speed and the lower plot shows the control signal.*

The matrix K_{gpc} is of size $[N, n_a + 1]$ and it is defined as follows

$$G_x(i) = CONV\left[E(1\ldots i), B\right] \quad i \leq N \tag{3.27}$$

$$K_{gpc}(i,j) = \begin{cases} K_r(i)F(i,j), & i \leq N, j \leq n_a \\ K_r(i)G_x(i,i+1), & i \leq N, j = n_a + 1 \end{cases} \tag{3.28}$$

$$K_{gpc} = \begin{bmatrix} 0.1185 & -0.0939 & 0.0243 & 0.0001 \\ 0.6203 & -0.6523 & 0.1887 & 0.0004 \\ 1.5529 & -1.8289 & 0.5614 & 0.0013 \\ 2.8204 & -3.5274 & 1.1226 & 0.0027 \\ 4.2807 & -5.5535 & 1.8099 & 0.0043 \end{bmatrix} \tag{3.29}$$

where $CONV[x, y]$ is discrete time convolution of x on y.

Figure 3.1 shows the speed tracking performance of the D.C. motor when the reference speed is ± 0.25 rads/sec. This is an ideal result in the sense that it contains infinite resolution. This result is used to confirm that GPC is capable of controlling speed of a D.C. motor satisfactorily.

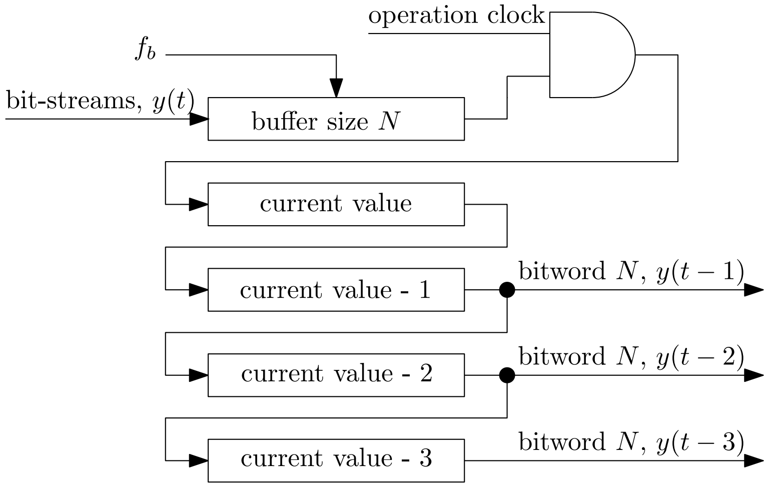


Figure 3.2: *Implementation of tapped delay block.*

To successfully implement a bit-stream GPC controller several issues must be addressed. One of the important blocks which is needed for implementing bit-stream GPC is the tapped delay block shown in Figure 3.2. This functional block is used to delay a bit-stream signal by a specific amount of time. In short, this is done by storing quanta into a memory buffer and releasing it after a specific amount of time. For example, we require a buffer size of 1000 bits to represent a delay $1ms$ for a bit-stream system which runs at a clock of $1\mu s$. For more information about this block refer to §2.3.7 of Chapter 2. An incoming signal is expressed in bit-streams at frequency f_b which is be buffered to a particular N-bit register. A snapshot of this buffered signal is triggered at every *operation clock* and moved to another register level. As this signal is shifted, subsequent registers are also shifted accordingly. Therefore, these registers will contain the delayed version of the signal required for a GPC implementation.

The schematic layout of the bit-stream based GPC is shown in Figure 3.3 which consists of several blocks, including a plant, 12-bit float-to-bit-stream converter, bit-stream-float conversion, a tapped delay and various arithmetic operations. The symbol $\bigotimes$ and $\bigoplus$ in Figure 3.3 refers to bit-stream (single bit) multiplication and addition respectably. Note, that the diagram describes only a single prediction window. This will be repeated for every window. The bit-stream GPC is implemented using $f_b = 5$MHz. The speed tracking performance is shown in Figure 3.4. From the results of Figure 3.4, it is obvious that the bit-stream controller exhibit very similar response to that obtained from Matlab$^{\text{TM}}$/Simulink$^{\text{TM}}$which demonstrate that the designed controller can successfully be implemented in bit-stream.

3.4.1 Bit-stream Based GPC for D.C. Motor: Experimental Setup

The experimental implementation of GPC using bit-stream requires several function blocks which are shown in Figure 3.5. Figure 3.6 shows the prototype of the D.C. motor used in this study. The

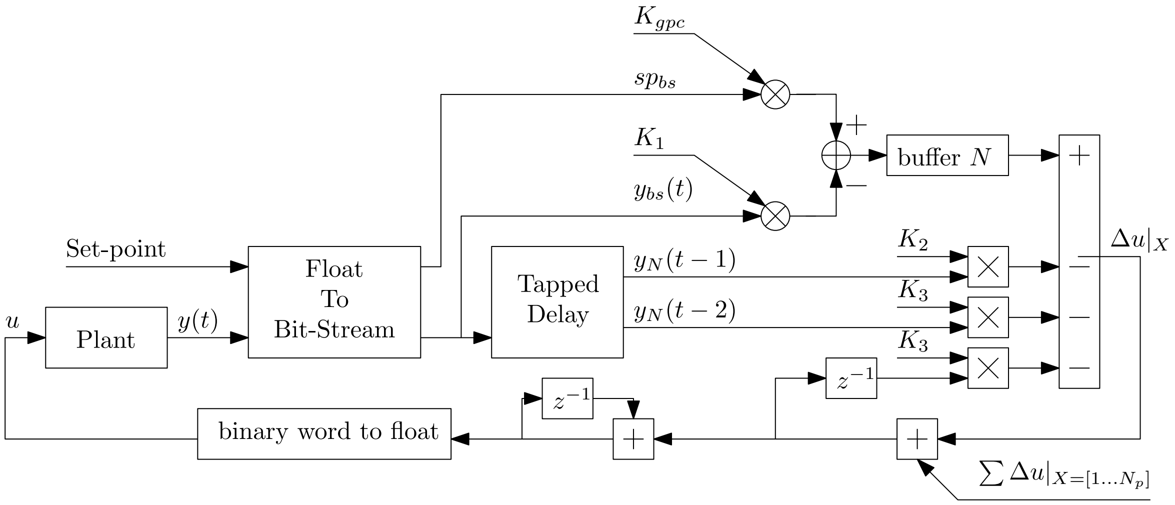


Figure 3.3: *Layout of bit-stream based GPC implementation using ModelSim^{TM}.*

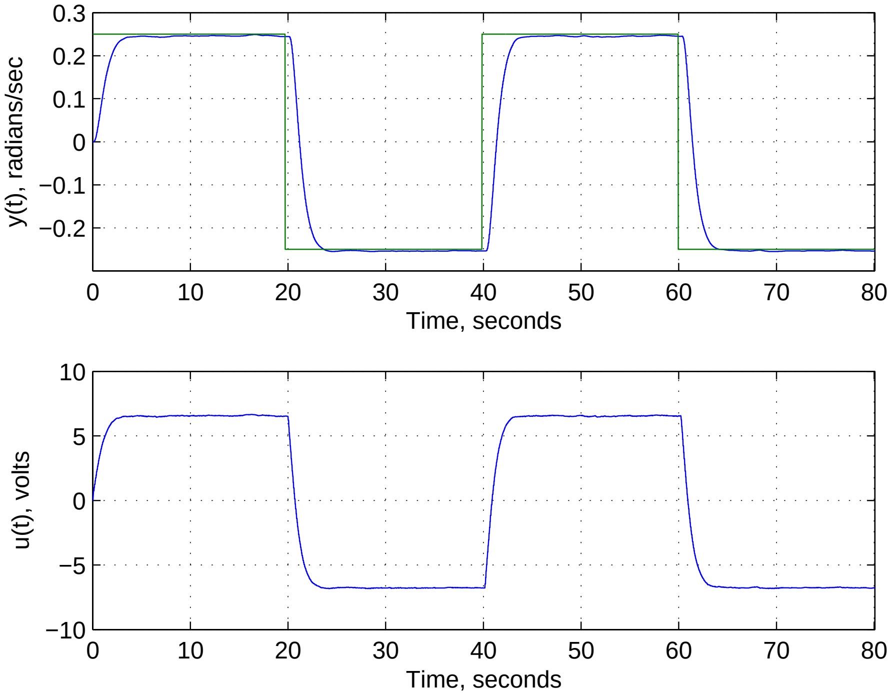


Figure 3.4: *Speed control of a D.C. motor using bit-stream based GPC: Upper plot shows the output speed and the lower plot shows the control signal.*

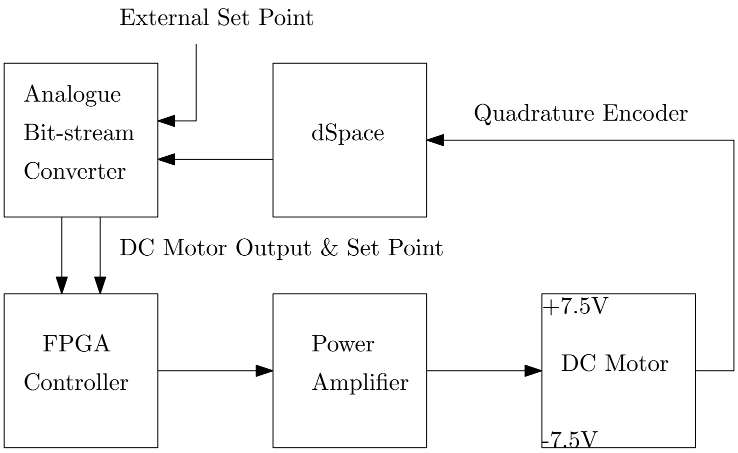


Figure 3.5: *Schematic diagram for D.C. servo mechanism experiment.*

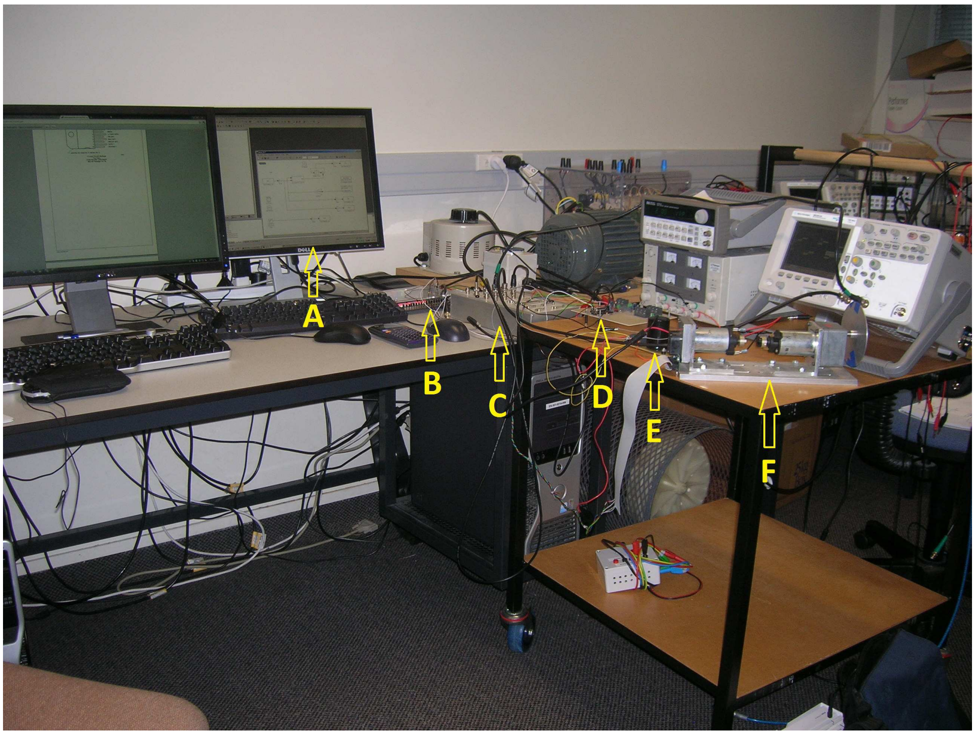


Figure 3.6: *Prototype of D.C. servo mechanism: (A) Computer interface with Simulink^{TM}, (B) DE2 FPGA, (C) dSpace I/O Ports, (D) bit-stream converter, (E) power amplifier and (F) D.C. motor plant.*

different functions of the blocks of Figure 3.5 are described below.

1. **FPGA Controller** - This is a DE2 Development Board which contains a Cyclone II EP2C35 FPGA device. This board is used to process all the control signals using bit-streams. This board has two 40-pin headers which is used to interact with the other devices. Logic levels for these pins are 0V for logic ’0’ and 3.3V for logic ’1’ and it is programmed using Quartus II software. The device is also being used to supply a clock signal and power to the Analogue-to-Bit-stream Converter. It is shown as item (B) in Figure 3.6.

2. **Analogue to Bit-stream Converter** - A Printed Circuit Board (PCB) is needed for converting analogue signals into bit-streams. This consists of a Cypress Semiconductor Programmable System on Chip (PSoC) Micro-controller CY8C26443-24PI which is programmed to take an input signal of ±5V and output a bit-stream with full scale at 5V, and a zero scale at -5V. This device requires three input lines: a clock (which runs at 4 times the bit-stream rate), 5V supply and ground. This microprocessor can output two separate bit-stream channels which will be used to encode the speed of the motor and the set point into bit-streams. This is shown as item (A) in Figure 3.7.

3. **dSpace** - A Simulink^{TM}/dSpace^{TM}interface is being used for performing the following tasks: 1) Generating the desired analogue set point, 2) Data acquisition, and 3) leveling the signal outputs to ensure the ±5V range is fully utilised. The dSpace board contains several analogue and digital I/O’s, as well as a quadrature encoder for the D.C. motor. Analogue ports have a ±10V range and digital ports take 0V to 3.3V for logic low and high respectively. The dSpace provides its own software called ControlDesktop, which gives various options for data acquisition such as data capturing, gain adjustment etc. This setup can be seen as item (A) and (C) in Figure 3.6.

4. **Power Amplifier** - This PCB contains two main ICs: A Programmable System on Chip (PSoC) Micro-controller CY8C26443-24PI and an LMD18200 H-Bridge. Since a bit-stream signal resembles a PWM signal, it can be used directly to drive the aforementioned H-Bridge. This can then be connected to the D.C. motor. The power amplification board can be seen as item (B) in Figure 3.7.

5. **D.C. Motor** - It is a general purpose 2000rpm, 12V, 16W motor (RS:715-106). The output of this motor is encoded via a HEDS-5500 quadrature encoder, and this is supplied directly to the dSpace for decoding as shown in Figure 3.7-(C).

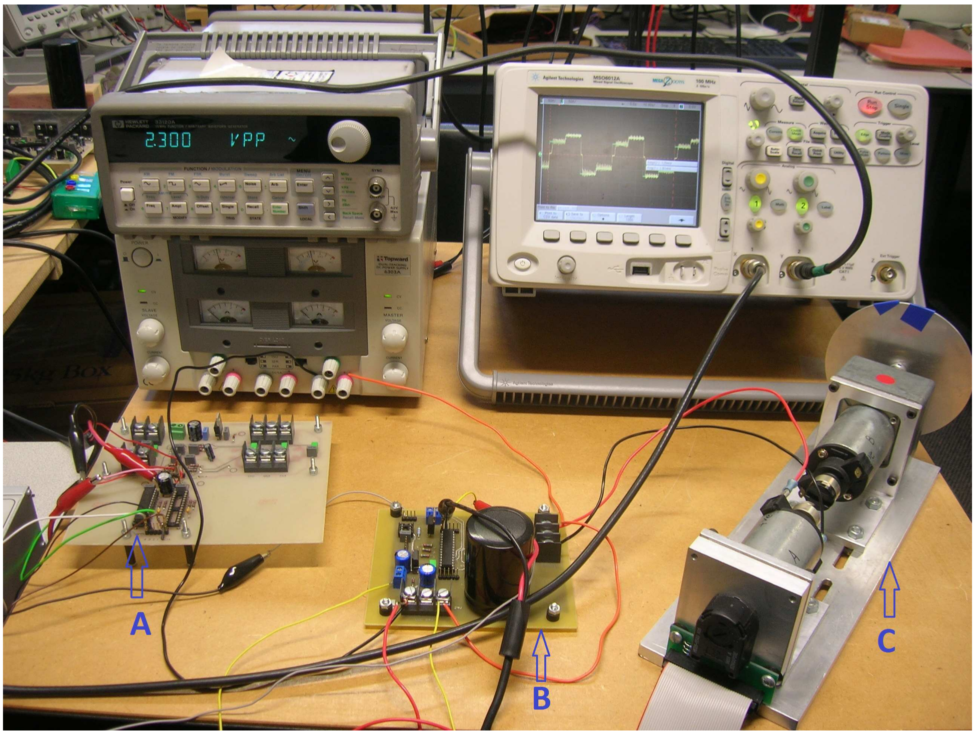


Figure 3.7: *Detailed view of D.C. motor prototype: (A) bit-stream converter, (B) power amplifier and (C) a D.C. motor.*

3.4.2 Step 1: Discrete Parametric Modelling of D.C. Motor

Since the design of the GPC is based on a discrete time model of the system, the first step is to obtain a discrete parametric model of the system from input-output data. To identify the D.C. motor, it is excited by a PRBS signal of magnitude ± 1.5V which is generated using a 9-bit shift register. The input and output data are collected at a sampling frequency of 200Hz by exciting the system for 10 seconds. The discrete parametric model is fitted using the orthogonal least squares algorithm with Error Reduction Ratio test [41].

In order to fit the discrete time model, 1700 data points are used for identification and 300 data points for validation. Figure 3.8 shows the One Step Ahead predicted output of the model. To further validate the model, the Model Predicted Output (MPO) over the remaining test set is computed and compared with the original data and it is shown in Figure 3.9. The MPO is in good agreement with the measured output and this serves as a simple and effective way of testing the validity of the model [72]. The majority of the prediction error terms often have a very low ERR value. If these terms are deleted the error may become a correlated sequence instead of white noise. This will cause bias in the model parameters. Fortunately, this can be readily detected by

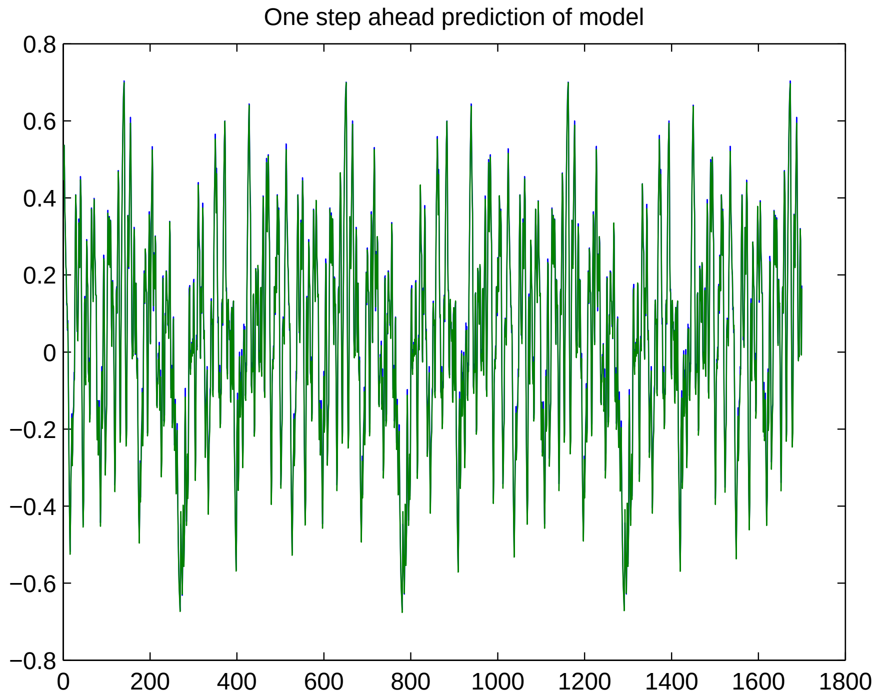


Figure 3.8: *One step ahead prediction for the identification of a D.C. servo mechanism.*

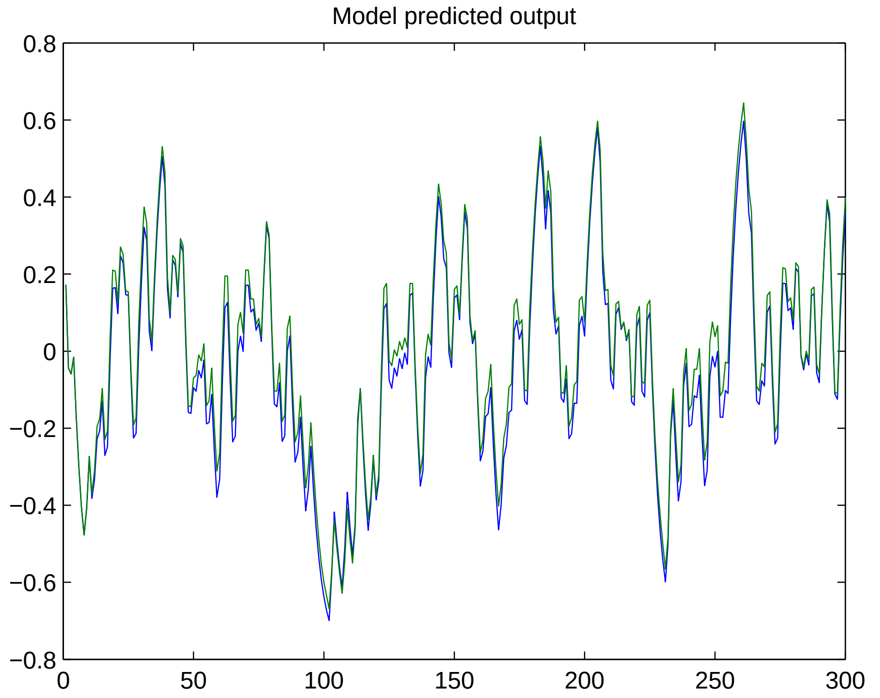


Figure 3.9: *Model predicted output for the identification of a D.C. servo mechanism.*

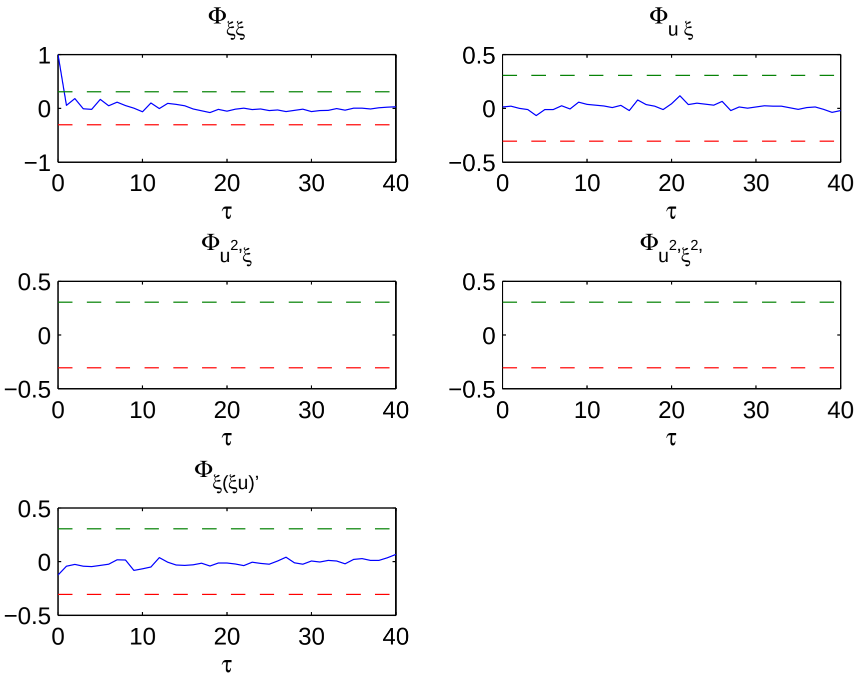


Figure 3.10: *Correlation plots for the identification a D.C. servo mechanism*

studying the relationships shown below.

$$\phi_{\xi\xi}(\tau) = \delta(\tau) \tag{3.30}$$

$$\phi_{u\xi}(\tau) = 0 \quad \forall\,\tau \tag{3.31}$$

$$\phi_{u^{2'}\xi^2}(\tau) = 0 \quad \forall\,\tau \tag{3.32}$$

$$\phi_{u^{2'}\xi}(\tau) = 0 \quad \forall\,\tau \tag{3.33}$$

$$\phi_{\xi\xi u}(\tau) = \tau \geq 0 \tag{3.34}$$

where $\xi(t)$ represents an estimate of the prediction error sequence and the $'$ indicates that the mean has been removed from a signal. The model validity test of (3.30) to (3.34) often indicates which, if any, terms have been omitted from the model. These terms can then be forced into the model to rectify any discrepancies. Figure 3.10 shows correlation plots for the D.C. motor plant. The results for the identification of the D.C. motor are given by

$$y(k) = 0.7803y(k-1) - 0.014262y(k-4) + 0.046456u(k-1) + 0.069196u(k-2) \tag{3.35}$$

3.4.3 Step 2: Optimisation of Gains for Generalised Predictive Controller for D.C. Motor

The success of GPC is critically dependent on some of the parameters such as prediction and control horizons and control weighting sequence. The optimisation of every parameter is essential, however the optimality is not a single point, but an optimality surface where the user can select different trade-offs for a desired operating point [73]. The present study uses genetic algorithm to optimize these parameters.

Genetic Algorithm (GA) is a stochastic global adaptive search optimisation technique based on the mechanisms of natural selection [74]. The algorithm aims to find the fittest individual from a set of candidate solution known as population. The main advantages of GA as a multi-objective optimisation algorithm are: it does not requires *a priori* knowledge about the relative importance of the objectives, and, there is a set of acceptable trade-off optimal solutions [73].

The GA was first proposed and analysed by Holland [75]. The classical definition of a GA consist of three distinctive sections: binary coding, proportional selection and crossover reproduction. From these features, the heavy reliance on crossover is what makes the GA distinctive over other intelligent search algorithms. Many variation of the classical GA algorithm have been developed using problem orientated approaches, but the underlying description proposed by Holland remains the same [76].

GAs are defined by two main operators, namely *crossover* and *mutation*. These are well known and described throughout the literature. More information on this can be found in Chapter 5 and [76, 77].

Previously published computer simulations have demonstrated the effectiveness of parameter tuning using GAs for minimisation of an objective function[73]. The selected performance index to be minimised is based on the following objectives:

- Minimise the rise time, t_r.
- Minimise the maximum overshoot, t_{OS}.
- Minimise the settling time, t_s.

Several existing tuning guidelines for GPC algorithms exist [78]. The GA commences with the selection of two candidates based on their fitness. In order for all the candidates to have a chance to partake in crossover, the selection is random, but the chance of each candidate being selected, is proportional to their fitness. the crossover probability lies within 90% and 95%. This is to allow some results to be passed unmodified. The crossover operator will be performed on the two selected candidates and this will produce a new offspring. A mutation rate is then defined and kept relatively low, between 1% and 10%. A zero mutation rate will cause the algorithm to converge to the nearest local minima and a high mutation rate will make the algorithm to behave like random search. The steps of GA are depicted as follows [73]:

I. **Start** - Generate random population of n chromosomes.

II. **Fitness** - Evaluate the fitness of $f(x)$ of each chromosome x in the population.

III. **New Population** - Create a new population by repeating following steps until the new population is complete

  III-1. **Selection** - Select two parent chromosomes from a population according to their fitness (the better their fitness, the higher their chance of being selected).

  III-2. **Crossover** - With a crossover probability, the parents form a new offspring (children). If no crossover is performed, offspring is an exact copy of the parents.

  III-3. **Mutation** - With a mutation probability, mutate new offspring at each position in the chromosome.

  III-4. **Accepting** - Insert the new offspring in a new population.

IV. **Replace** - Use the newly generated population for the next run of the algorithm.

V. **Test** - If the end condition is satisfied, **stop**, and return the best solution in current population.

VI. **Loop** - Go to step 2.

Using the algorithm outlined above, the three GPC parameters which are optimised are the prediction horizon N_2, the control horizon N_U and the control-weighting sequence λ. A population size of 10 is generated using binary encoding. Convergence of the GA algorithm is monitored by the minimisation of the cost function $J = \sum [w_1 t_r + w_2 t_s + w_3 t_{OS}]^{-1}$, where w_1, w_2 and w_3 are weight factor for each of the parameter. Crossover and mutation rates are set to 95% and 5% respectively. This algorithm is run through 50 generations.

The convergence of the fitness function is shown in Figure 3.11 as a function of generations. The values associated with the y-axis are relative to the operation and do not explicitly represent any value in particular. The final output values are shown in Table 3.1.

The parameters obtained via GA for the GPC algorithm are used to achieve successful tracking

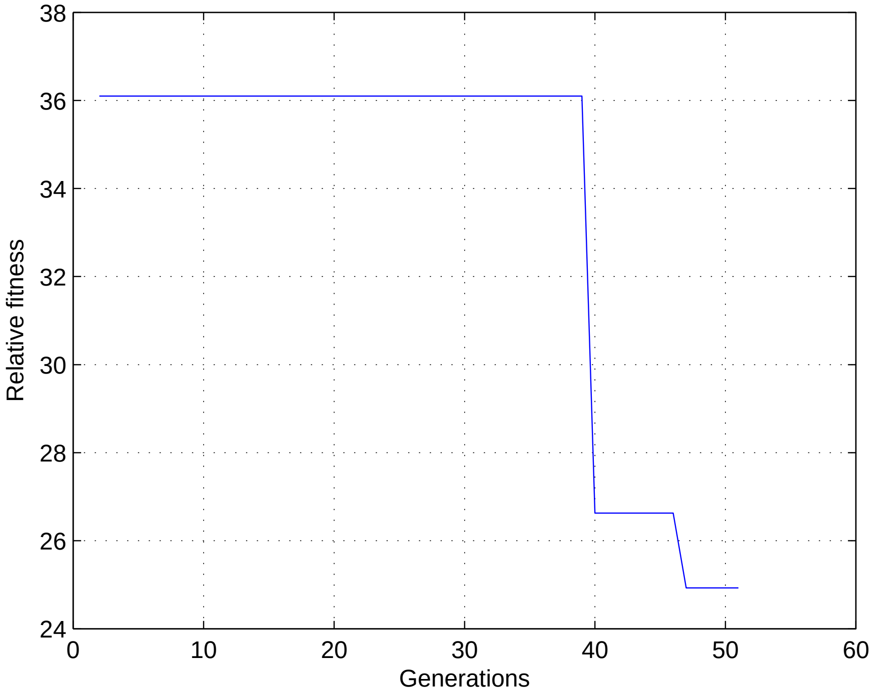


Figure 3.11: *Fitness trajectory for the optimisation of D.C. servo mechanism GPC parameters.*

GPC Parameter	Optimised Value
N_2	3
N_u	1
λ	8.64

Table 3.1: *Optimised GPC parameters obtained via genetic algorithms for D.C. Servo mechanism.*

when the reference speed is 3 rads/sec. The results show a rise time of 0.335 seconds and no steady state error, and are presented in Figure 3.12. In the present study, the performance of a the D.C. servo mechanism is tested using a stair case input as a reference signal. The results of simulation for a multi-bit GPC implemented in Matlab$^{\text{TM}}$is shown in Figure 3.13. From the figure it is obvious that the GPC could successfully track the reference.

3.4.4 Step 3: Experimental Results

Initially, the standard GPC is implemented using Simulink$^{\text{TM}}$/dSpace$^{\text{TM}}$on a D.C. motor plant through analogue ports. Then, the same D.C. motor is controlled using GPC control law encoded via bit-streams. Figure 3.14 show the response of an actual D.C. motor plant using GPC. The encoder produces 500 counts per revolution. Using this information, it is possible to display everything in terms of radians/second.

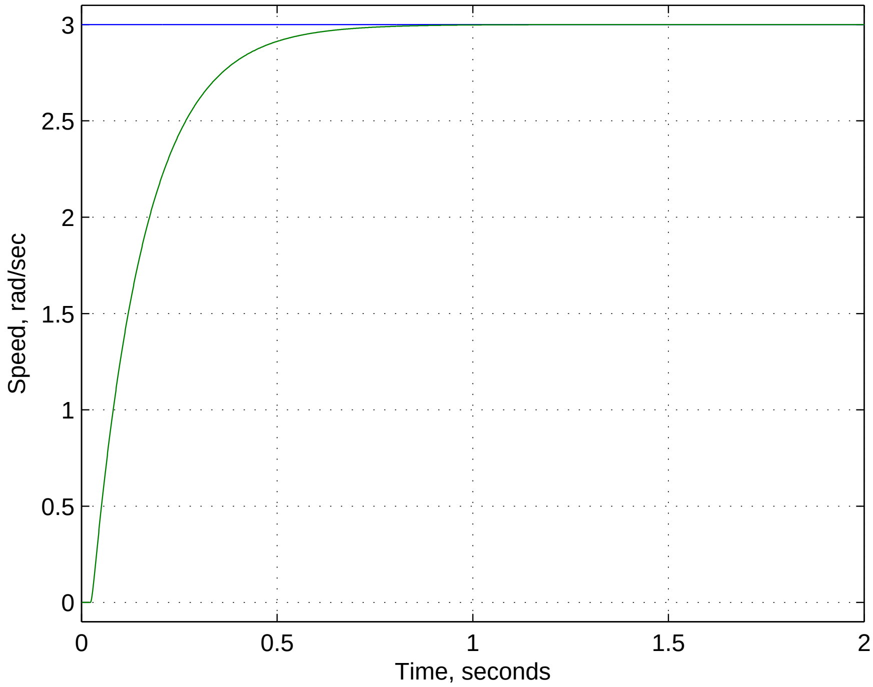


Figure 3.12: *Speed tracking performance of a D.C. servo mechanism using GPC with optimised parameters obtained via genetic algorithms.*

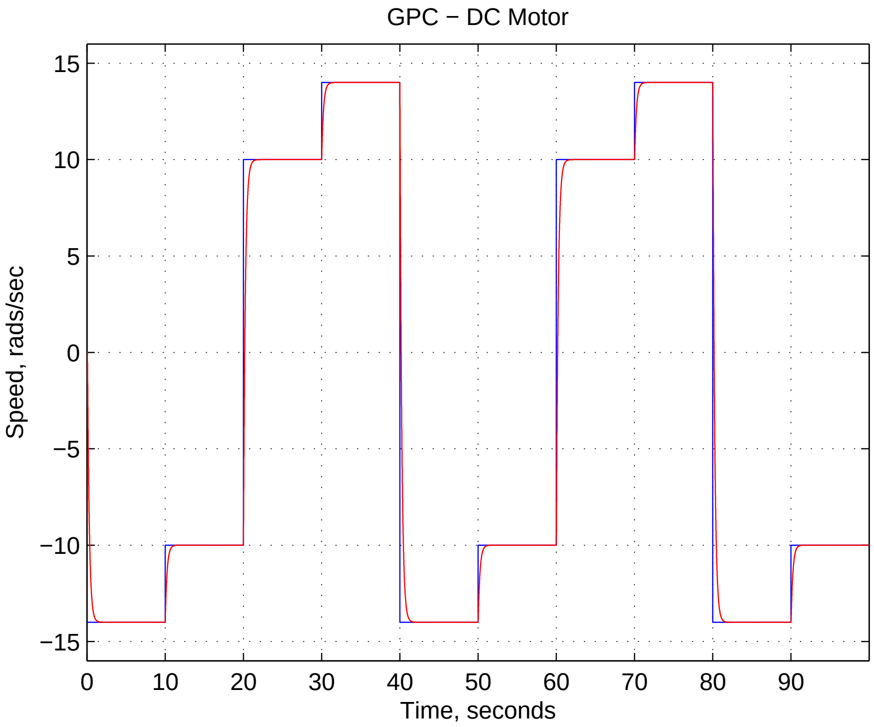


Figure 3.13: *Simulation results of speed control of D.C. motor using multi-bit GPC.*

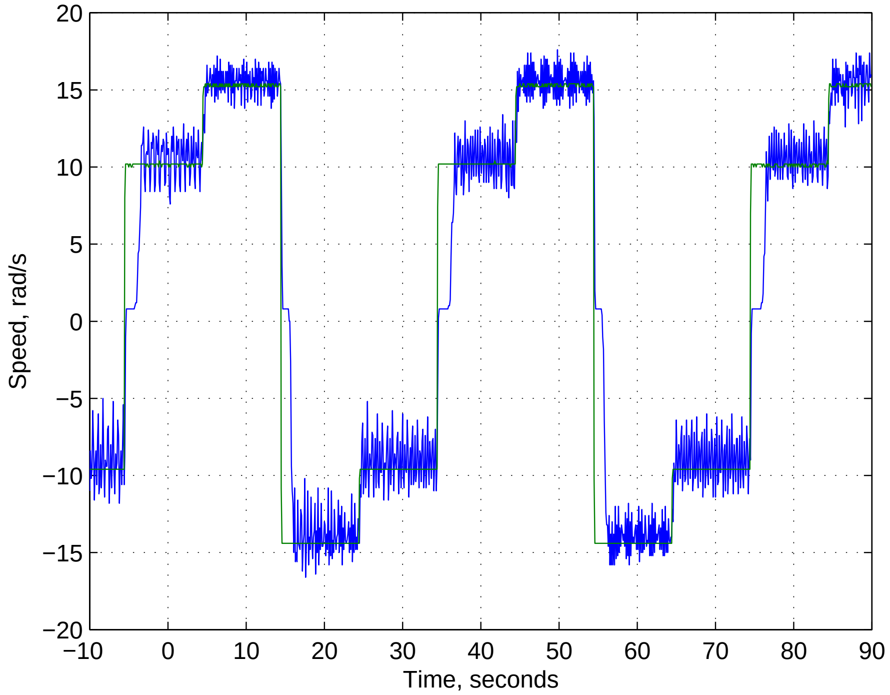


Figure 3.14: *Experimental results for speed control of D.C. motor using multi-bit GPC.*

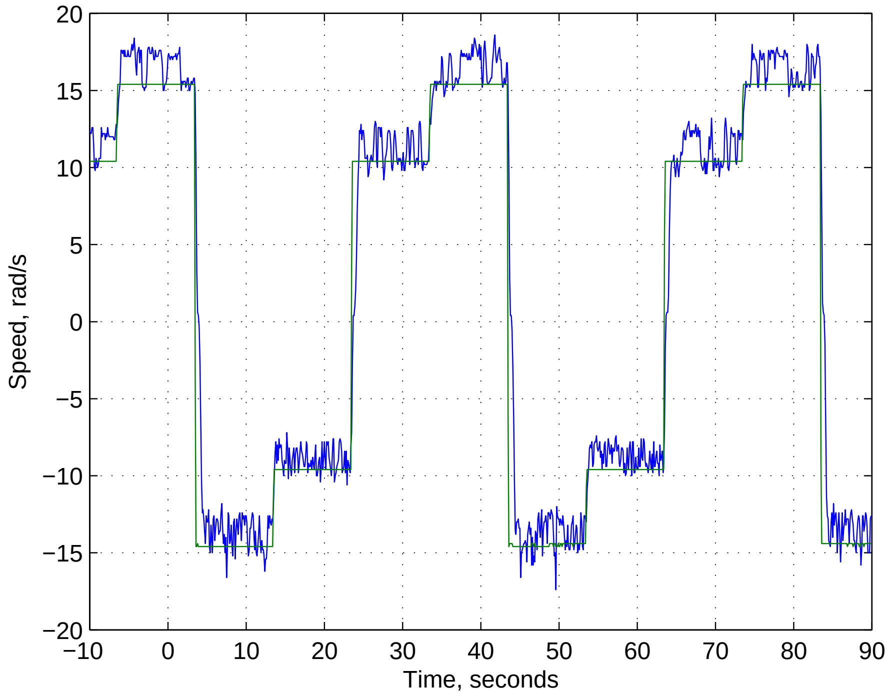


Figure 3.15: *Experimental results for speed control of D.C. motor using bit-stream GPC.*

From the results it is observed that the results of bit-stream GPC is comparable with those obtained from the multi-bit implementation.

3.5 Case Study 2: Design of GPC for Temperature Control of a Thermal System

The next example to be considered is a thermal system, which is essentially a linear system with transport lag. The dynamics of the thermal system considered in this study is given by [17, 79]

$$G_{th} = e^{-sT_{lag}} \frac{1.04}{\left(1+\frac{s}{3.14}\right)\left(1+\frac{s}{5.96}\right)} \tag{3.36}$$

where $T_{lag} = 0.18s$. Although the system is inherently nonlinear this model has been obtained at a specific operating point. The output of the plant is temperature, which is translated through a thermocouple into a voltage.

The CARIMA representation of this plant at sampling frequency of 56Hz is given by

$$A(q^{-1}) = 1 - 1.8433q^{-1} + 0.8489q^{-2} \tag{3.37}$$
$$B(q^{-1}) = 0.0030 + 0.00281q^{-1} \tag{3.38}$$
$$C(1^{-1}) = 1 \tag{3.39}$$

To better understand the dynamics of the plant, it is important to study its physical layout. It is understood that the output of the plant is in degrees Celsius, however measurements are taken through a thermistor with a range of ±15V, which can span through a range from ambient temperature to 80^0C. The nominal operating point of the system as recommended by the manufacturer is 35^0C.

Initially, GPC is designed using the model given by (3.37), (3.38), (3.39). Following a similar procedure as presented in §3.3, the GPC for the thermal system is implemented using both Matlab$^{\text{TM}}$/Simulink$^{\text{TM}}$and ModelSim$^{\text{TM}}$. The simulation wasis carried out considering the prediction horizon $N_2 = 4$, control horizon $N_u = 2$ and the weighting parameter $\lambda = 8$. The gains K_r and K_{gpc} for the GPC are given by

$$K_r = \begin{bmatrix} 0.0004 & 0.0014 & 0.0030 & 0.0051 \end{bmatrix} \tag{3.40}$$

$$K_{gpc} = \begin{bmatrix} 0.0011 & -0.0010 & 0.0003 & 0.0000 \\ 0.0076 & -0.0096 & 0.0034 & 0.0000 \\ 0.0257 & -0.0365 & 0.0138 & 0.0000 \\ 0.0617 & -0.0935 & 0.0368 & 0.0001 \end{bmatrix} \tag{3.41}$$

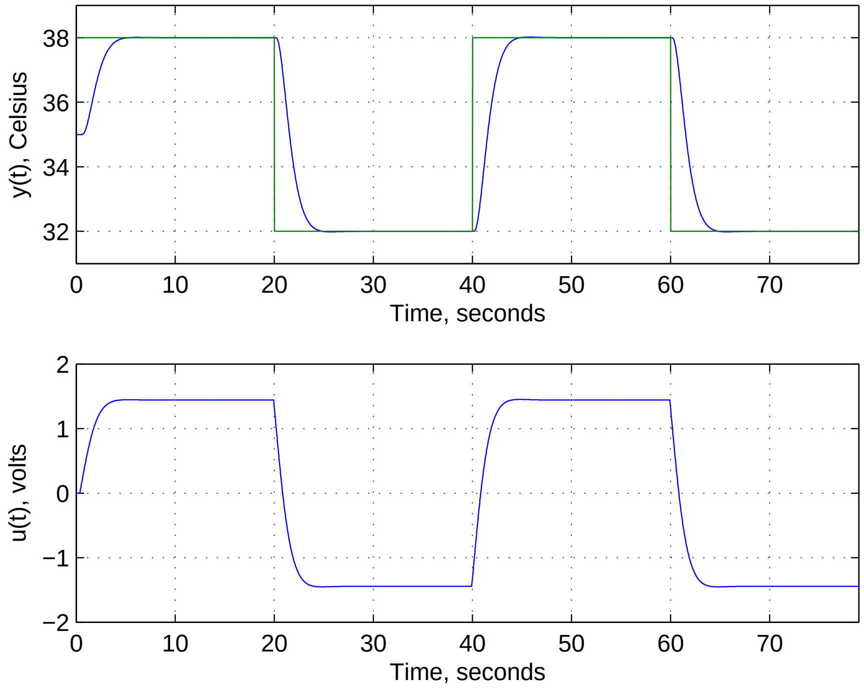


Figure 3.16: *Simulation results for temperature tracking of a thermal system using multi-bit GPC: Upper plot shows the plant output and the lower plot shows the control law.*

The simulated results of for multi-bit and bit-stream implementation are shown respectively in Figure 3.16 and Figure 3.17. From the result it is found that the performance of both the controllers are satisfactory.

3.5.1 Bit-Stream Based GPC for Thermal System: Experimental Setup

Figure 3.18 shows the schematic of experimental implementation and the prototype is shown in Figure 3.19 and Figure 3.20. Note that this system is a linear system with transport lag or delay. Therefore, before designing the GPC it is essential to determine the delay associated with the system.

3.5.2 Step 1: Discrete Parametric Modelling of Thermal System

One of the key properties of this system is that it contains transport lag. By adjusting the position of the thermocouple, it is possible either to increase or decrease this lag. Figure 3.21 shows the response of three settings of the thermal plant to a step function. The blue trace shows the minimum delay equals to 0.05s, the green trace shows the medium level of delay equals to 0.125s and the red trace show the maximum delay of 0.18s. The values presented here are most

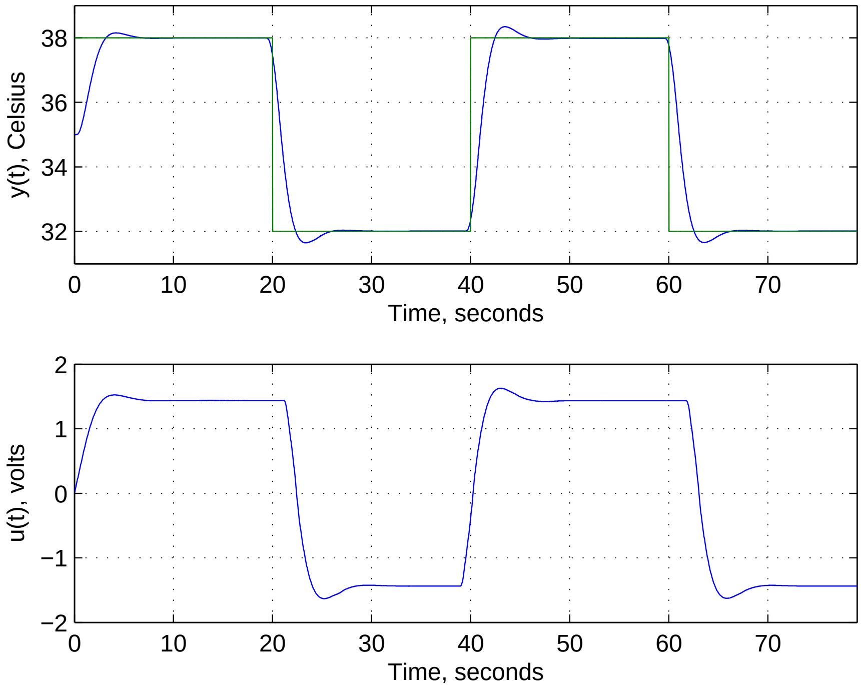


Figure 3.17: *Simulation results for temperature tracking of a thermal system using bit-stream GPC: Upper plot shows the plant output and the lower plot shows the control law.*

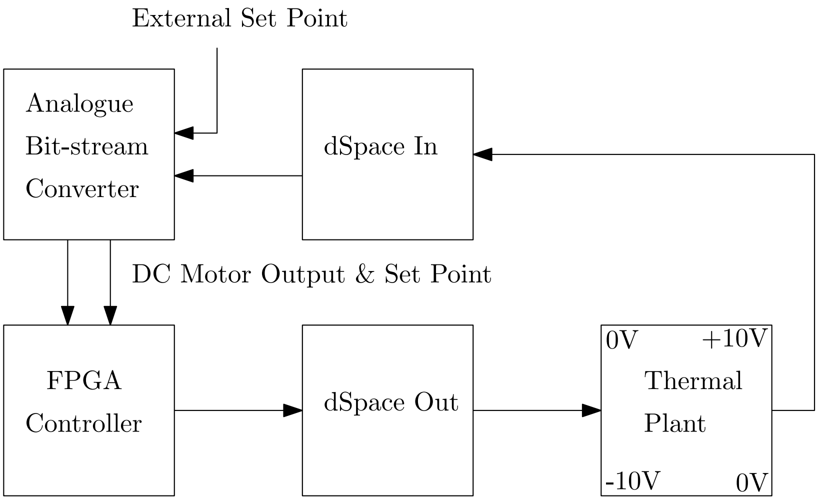


Figure 3.18: *Schematic diagram for thermal plant experiment.*

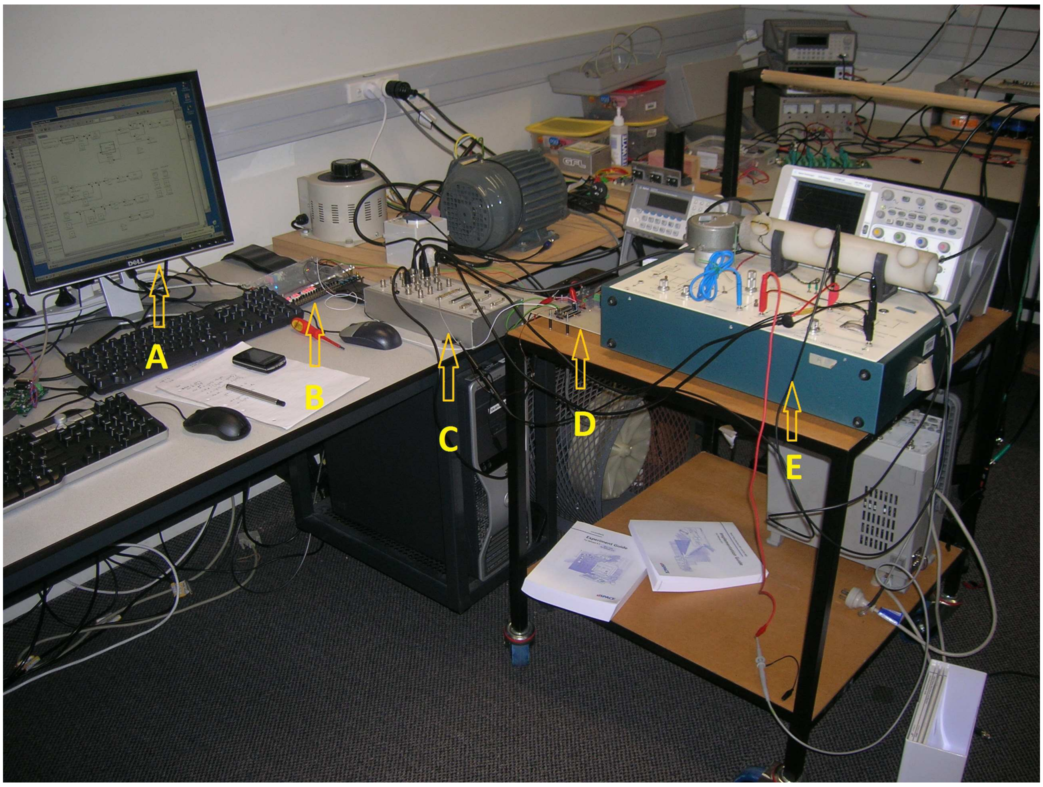


Figure 3.19: *Experimental Prototype of thermal system. (A) Computer interface with Simulink^{TM}, (B) DE2 FPGA, (C) dSpace I/O Ports, (D) bit-stream converter and (E) thermal plant.*

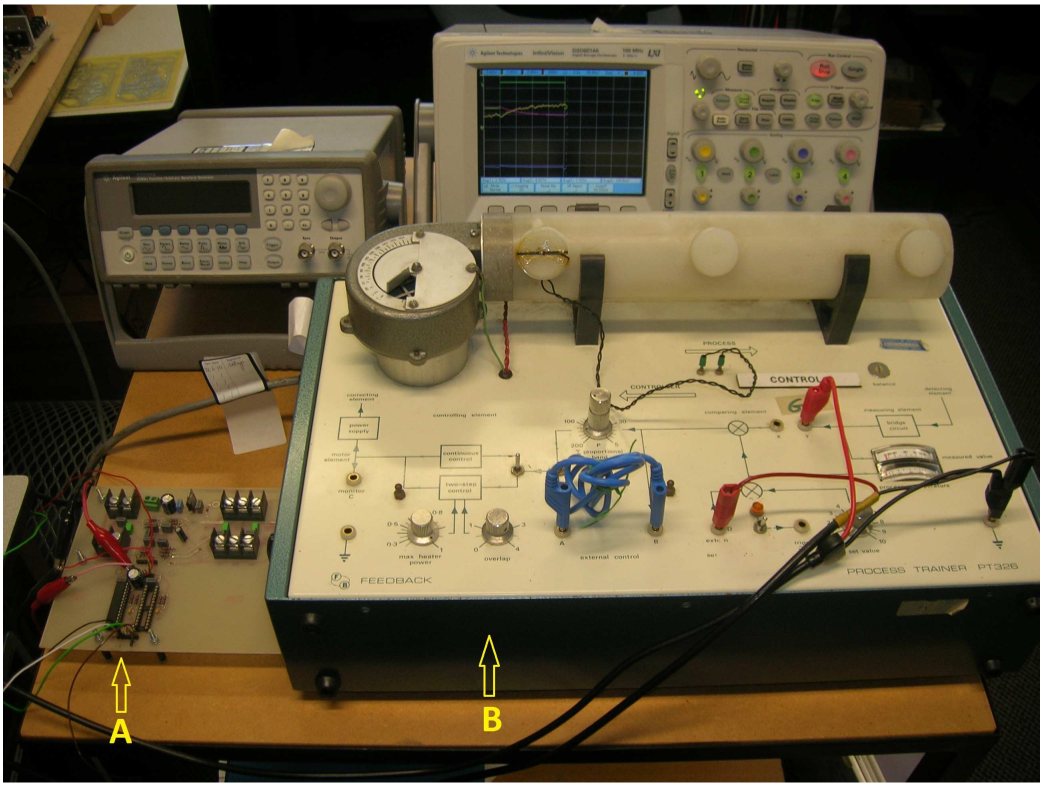


Figure 3.20: *Detailed view of thermal system: (A) bit-stream converter and (B) thermal plant.*

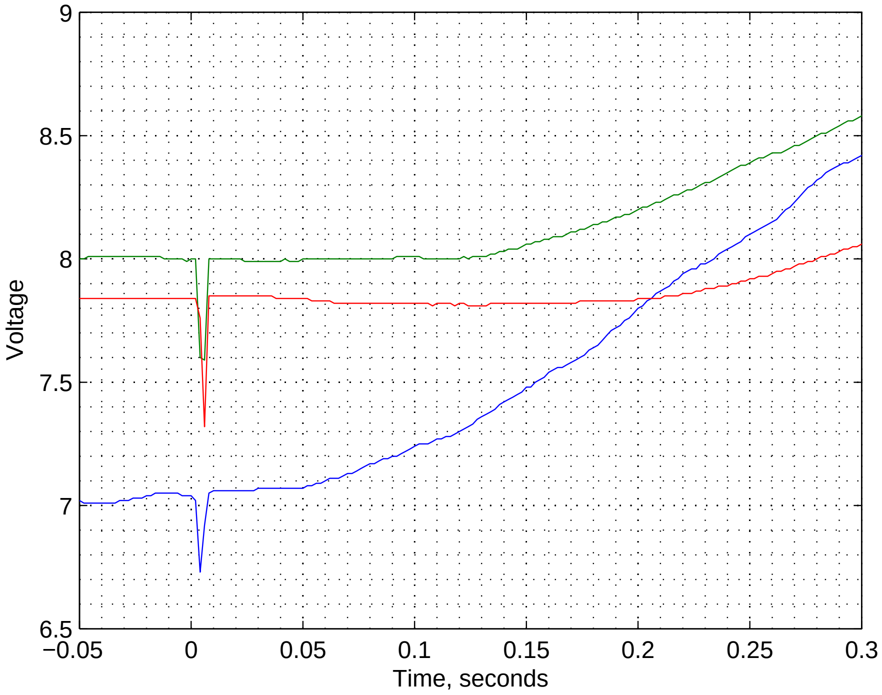


Figure 3.21: *Transport lag for thermal system: Minimum delay in blue, medium delay in green, maximum delay in red.*

important, since it will give an indication of the maximum lag of the discrete model of the plant which is identified next.

The thermal plant is identified using a linear model and validated through a series of tests to ensure that the system dynamics has been successfully captured. The system has been excited through a 9-bit PRBS signal with an upper level of $7V$ and a lower level of $3V$. This procedure is done at a sample frequency of 20Hz for 100 seconds, meaning 2000 data points are collected.

The system is identified using Orthogonal Least Squares (OLS) and the structure is selected using Error Reduction Ratio (ERR) as proposed by [80]. Figure 3.22 shows the One Step Ahead Prediction(OSA) of the model for the *medium delay*. Similar to the analysis performed on the D.C. motor plant, Figure 3.23 shows the model MPO for the thermal plant. Figure 3.24 shows correlation plots for the thermal plant under *medium delay*. The results for the identification

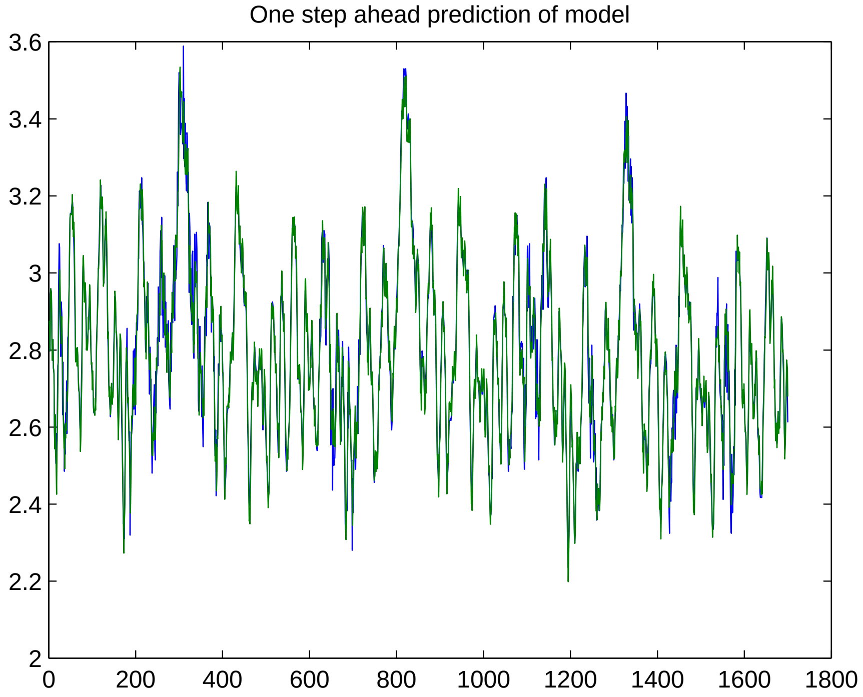


Figure 3.22: *One step ahead prediction of a thermal plant set to medium delay.*

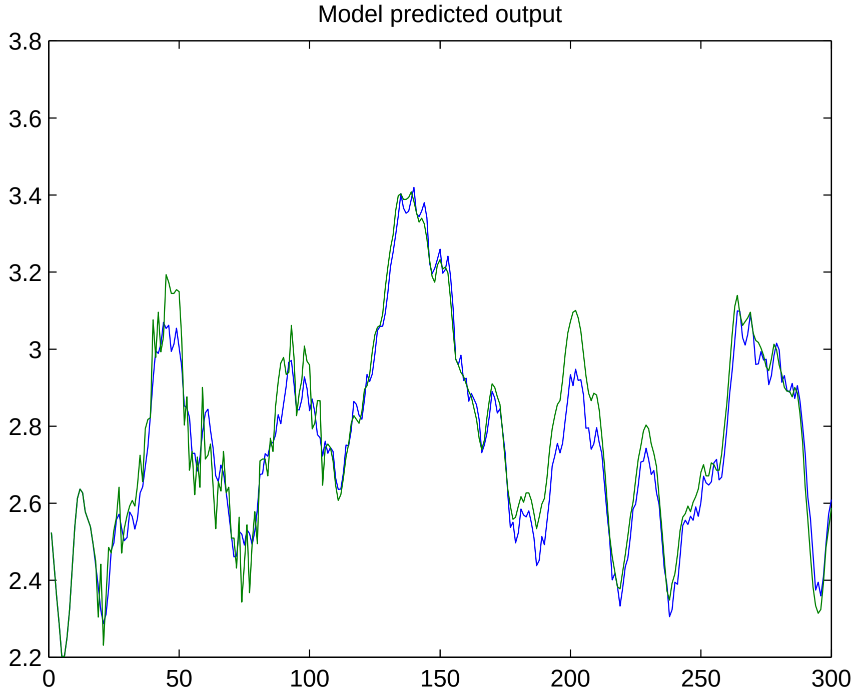


Figure 3.23: *Model predictive output of a thermal plant set to medium delay.*

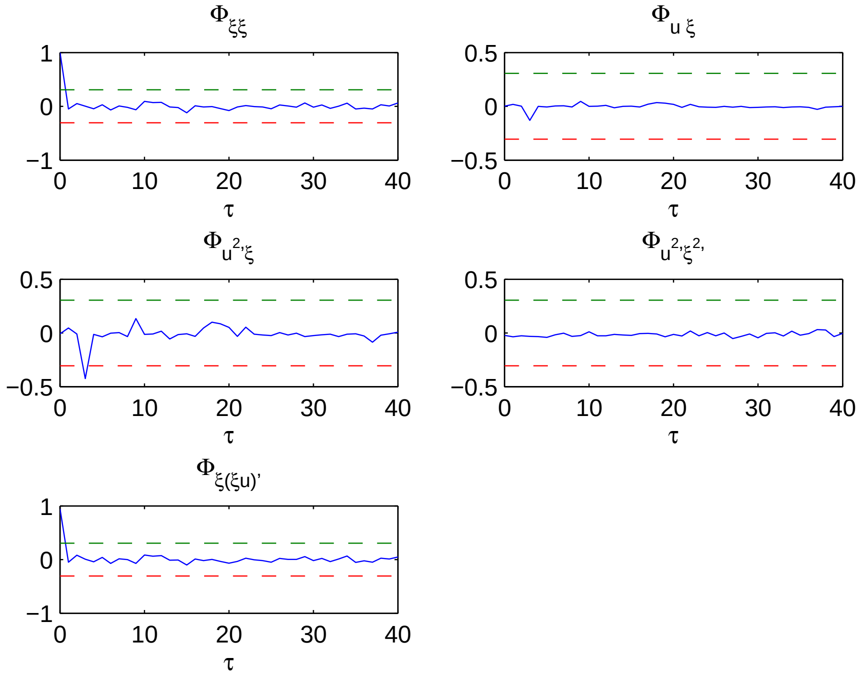


Figure 3.24: *Correlation plots of a thermal plant set to medium delay.*

process with *medium delay* is given by

$$\begin{aligned} y(k) &= 0.82678y(k-1) + 0.20329y(k-4) + 0.097783y(k-5) \\ &\quad - 0.13092y(k-10) - 0.040162u(k-4) - 0.026799u(k-5) \\ &\quad - 0.019746u(k-6) - 0.015958u(k-7) + 0.015502u(k-10) \\ &\quad + 0.01788u(k-11) + 0.017418u(k-12) + 0.021423u(k-13) \\ &\quad + 0.014279u(k-14) + 0.0087951u(k-15) + 0.010232u(k-16) \\ &\quad + noise\ model \end{aligned} \tag{3.42}$$

The models for *minimum delay* and *maximum delays* are fitted following a similar procedure and these are given in (3.43) and (3.44) respectively.

$$\begin{aligned} y(k) &= 1.2082y(k-1) - 0.45747y(k-6) + 0.24757y(k-7) \\ &\quad - 0.041483u(k-2) - 0.021088u(k-3) + 0.014101u(k-5) \\ &\quad + 0.020941u(k-6) + 0.023723u(k-7) + 0.0055753u(k-8) \\ &\quad - 0.00014956u(k-12) \end{aligned} \tag{3.43}$$

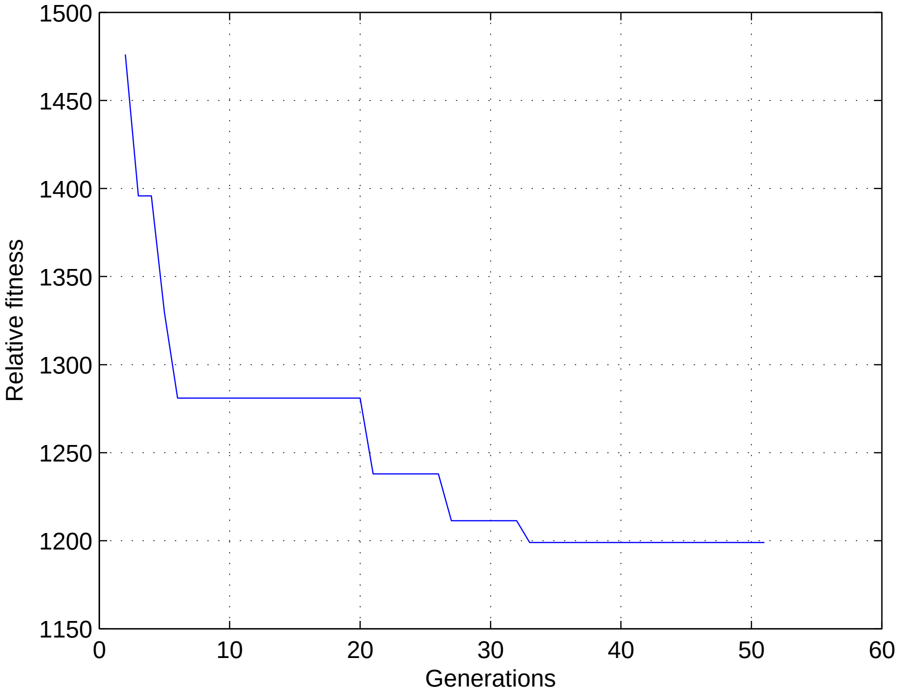


Figure 3.25: *Fitness optimisation via genetic algorithm for a thermal plant with medium delay*

$$\begin{aligned} y(k) &= 1.0415y(k-1) - 0.043677y(k-12) - 0.02634u(k-5) \\ &\quad - 0.020495u(k-6) - 0.013528u(k-7) + 0.013912u(k-12) \\ &\quad + 0.010142(k-13) + 0.0074753u(k-14) + 0.0086319u(k-15) \\ &\quad + 0.010585u(k-17) + 0.0036483u(k-18) + 0.007605u(k-22) \end{aligned} \tag{3.44}$$

3.5.3 Step 2: Optimisation of Gains for Generalised Predictive Controller for a Thermal System

Using the same procedure presented in §3.4.3, the GPC parameters are optimised. Figure 3.25 shows for the relative fitness optimisation trajectory for the *medium delay*. A similar optimisation procedure is performed for all delays using all the identified systems. The results are presented in Table 3.2. These results will be used to generate the required gains for the final implementation of the GPC.

3.5.4 Step 3: Experimental Results

Using the identified plant in (3.42), and the optimised parameters obtained using genetic algorithm algorithms, the GPC control law is implemented. The result of this simulation is shown in

	Optimised Values		
GPC Parameters	$t_{min} = 0.05s$	$t_{med} = 0.125s$	$t_{max} = 0.18s$
N_2	6	5	7
N_u	1	1	2
λ	21.4	37.4	78.18

Table 3.2: *Optimised GPC parameters obtained via genetic algorithms for thermal plant with three different transport lags.*

Figure 3.26, and are shown to be satisfactory. Similar simulations are performed for the minimum and maximum delays with similar results but are not shown here. Figure 3.26 is a very important result because it serves as a benchmark for the real behavior of the thermal plant.

The final implementation of the thermal system (set to medium delay) is shown in Figure 3.27. The performance of the bit-stream based GPC controller is comparable to multi-bit and shown to be satisfactory. The performance of the GPC using multi-bit and bit-stream on a thermal system set to minimum and maximum delay is shown in Figure 3.28 and Figure 3.29 respectively. The GPC bit-stream response is also shown to be satisfactory.

3.6 Conclusions

A bit-stream based generalised predictive controller (GPC) has been successfully designed and implemented for a linear system and a linear system with delay, namely a D.C. motor and a thermal system respectively. The success of the implementation relied heavily on the correct identification of the plants. Clear evidence of this has been presented and is used to successfully implement bit-stream based controller simulations. These simulations are validated through the implementation of multi-bit and bit-stream based GPC on actual plant prototypes. The results of the experiments demonstrated that a bit-stream based approach for predictive control of linear system is a viable alternative. Therefore, the procedures and results presented here, serve as the basis for the implementation of more complex controllers in later chapters of this thesis.

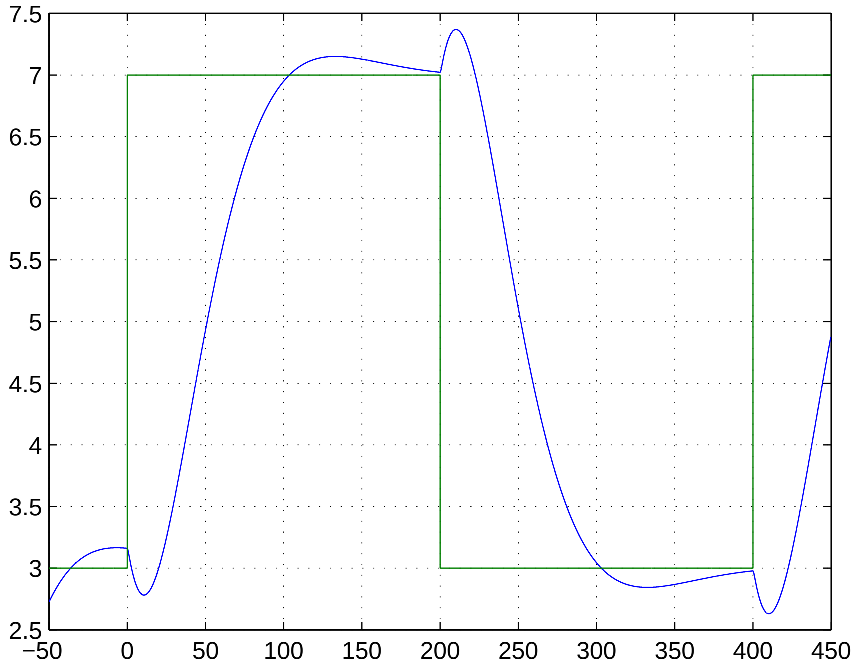


Figure 3.26: *Simulation of temperature tracking performance of a thermal system set to medium transport lag using multi-bit GPC.*

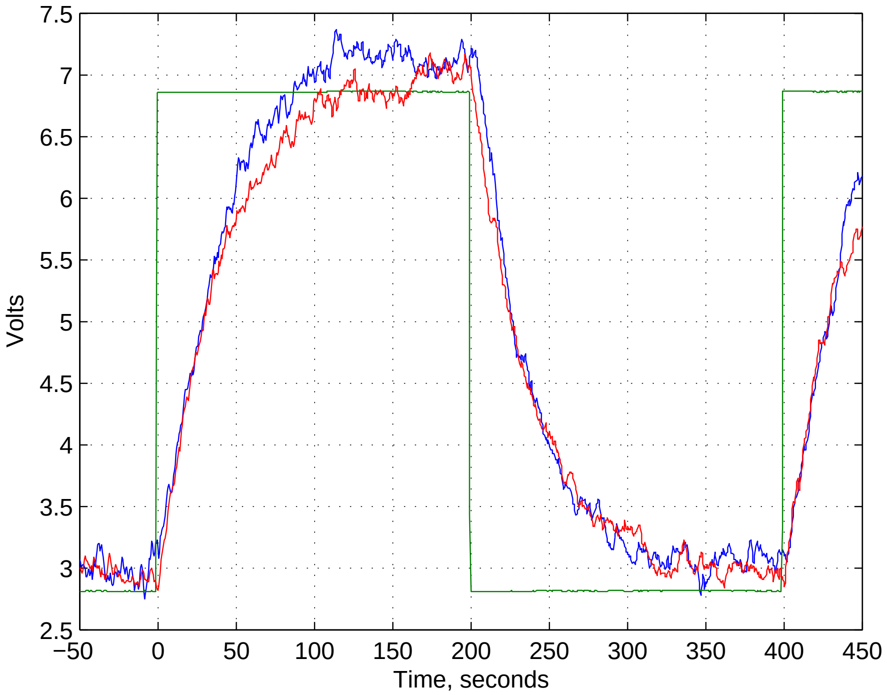


Figure 3.27: *Temperature tracking performance of a thermal system set to medium delay: Bit-stream is shown in blue, multi-bit is shown in red and setpoint is shown in green.*

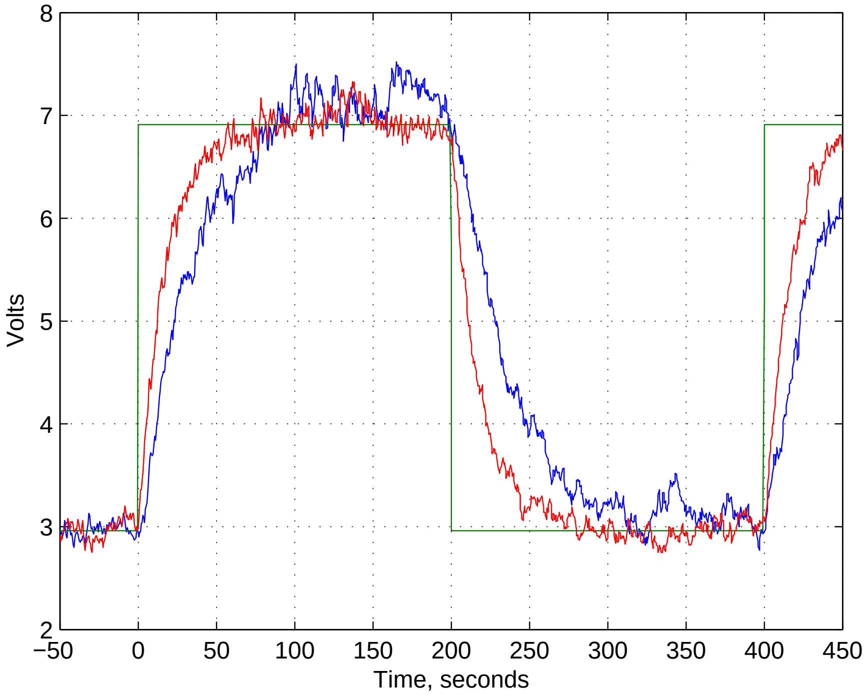


Figure 3.28: *Temperature tracking performance of a thermal system set to minimum delay: Bit-stream is shown in blue, multi-bit is shown in red and setpoint is shown in green.*

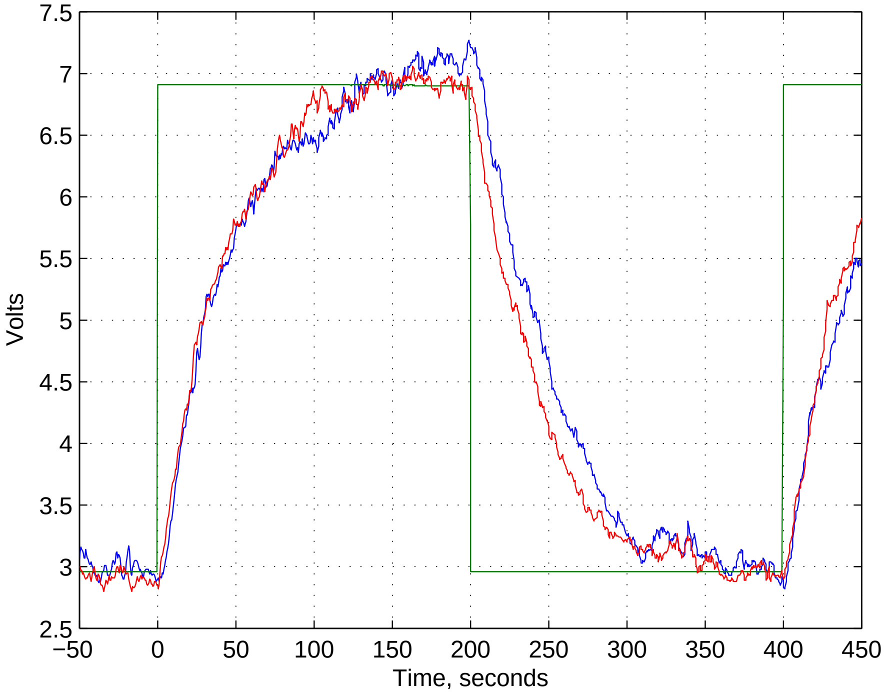


Figure 3.29: *Temperature tracking performance of a thermal system with maximum delay: Bit-stream is shown in blue, multi-bit is shown in red and setpoint is shown in green.*

CHAPTER 4

Bit-stream based Controller for an Unstable Magnetic Levitation System

4.1 Introduction

Magnetic levitation systems can suspend objects without friction. In recent years, numerous applications of such systems have been reported e.g. frictionless bearings, high speed Maglev trains, levitation of wind tunnel models, vibration isolation system, rocket guiding projects, superconductor rotor suspension of gyroscopes, high speed transportation systems, self-bearing blood pumps used in artificial hearts, levitation of molten metal in induction furnaces, levitation of metal slabs during manufacturer *et cetera* [81, 82, 83, 84, 85, 86]. The electro-mechanical dynamics of Magnetic Levitation Systems (MLS) are inherently unstable and highly nonlinear. The problem of successfully controlling such systems therefore poses considerable challenge to control engineers.

Different types of controllers, based on both linear and nonlinear techniques, have been proposed for controlling such systems [87, 88]. The standard linear technique is based on an an approximate linear model which is obtained by perturbing the system dynamics around a desired operating point. The performance of the linear controller is therefore optimal when the system operates around the linearisation point. To overcome the problems associated with linear controllers, several controllers based on nonlinear methods have been proposed e.g. nonlinear state space control, the sliding mode control, adaptive control, robust control, predictive control, *et cetera* [89, 90, 91, 92, 93, 94, 95, 96, 97].

Most of these controllers are implemented using standard digital implementation and using microcontrollers and digital signal processors which are usually optimised for fixed width binary words e.g. 8-bit, 16-bit or 32-bit. Although these implementations have been very successful, they have one thing in common and that is they are all serial processors and are capable of executing a single instruction at a time. Moreover, the choice of the word size is governed by the required dynamic range, the signal-to-noise ratio, the nature of the control algorithm or even the cost of the silicon hardware and the associated development tools [59].

In recent years, an alternate technique based on bit-stream has been proposed for selected control applications due to the several advantages they offer [59, 67, 50, 51]. The most obvious advantage is that each signal is represented by a single bit which eases place and routing operations for FPGA systems, reduces I/O usage to only one pin per signal, allows easy isolation of systems using digital optocouplers or similar equipment and is inherently parallel.

The objective of this chapter is to investigate whether a bit-streams can be used for the control of an unstable magnetic levitation plant. To achieve this, several types of controllers are implemented which include, feedback linearising cfontroller, state space based model predictive controller, generalised predictive controller and a lead-lag compensator.

4.2 Mathematical Model of Magnetic Levitation System

The schematic of a typical magnetic levitation system is shown in Figure 4.1. Two slightly different systems are shown, albeit their dynamics are identical. The system consists of a metallic ball bearing, whose weight is counteracted by the electromagnetic force applied by an iron core with DC resistance R, inductance L and magnetic force constant C.

In magnetic levitation systems, the height of the ball can be measured in various ways. Presented in Figure 4.1 are two ways to achieve this. In Figure 4.1-(A), optical sensors and LED sources are used to measure the position of a steel ball by correlating the height to the received infrared light of the optosensors. In Figure 4.1-(B), a magnetised steel ball is used, whose height is determined by the strength of the magnetic field detected by the hall effect sensor placed just below the coil. Both of these systems provide the height of the ball as a proportion of voltage. The dynamics of either system remains the same provided the height of the ball is interpreted correctly, and these are therefore given by

$$\frac{dx}{dt} = v \tag{4.1}$$

$$Ri + \frac{d}{dt}\left[L(x)i\right] = e \tag{4.2}$$

$$m\frac{dv}{dt} = mg - C\left[\frac{i}{x}\right]^2 \tag{4.3}$$

where, x is the position of the ball in meters, v is the velocity of the ball in meters per second, m is the mass of the ball in kilograms, g is gravity, i is the current in amperes, C is the force constant in Nm^{2}A^{-2}, R is the DC core resistance in Ω, e is the voltage applied to the core in volts, and $L(x)$ is the inductance of the core in Henry which is a nonlinear function of the position of the ball

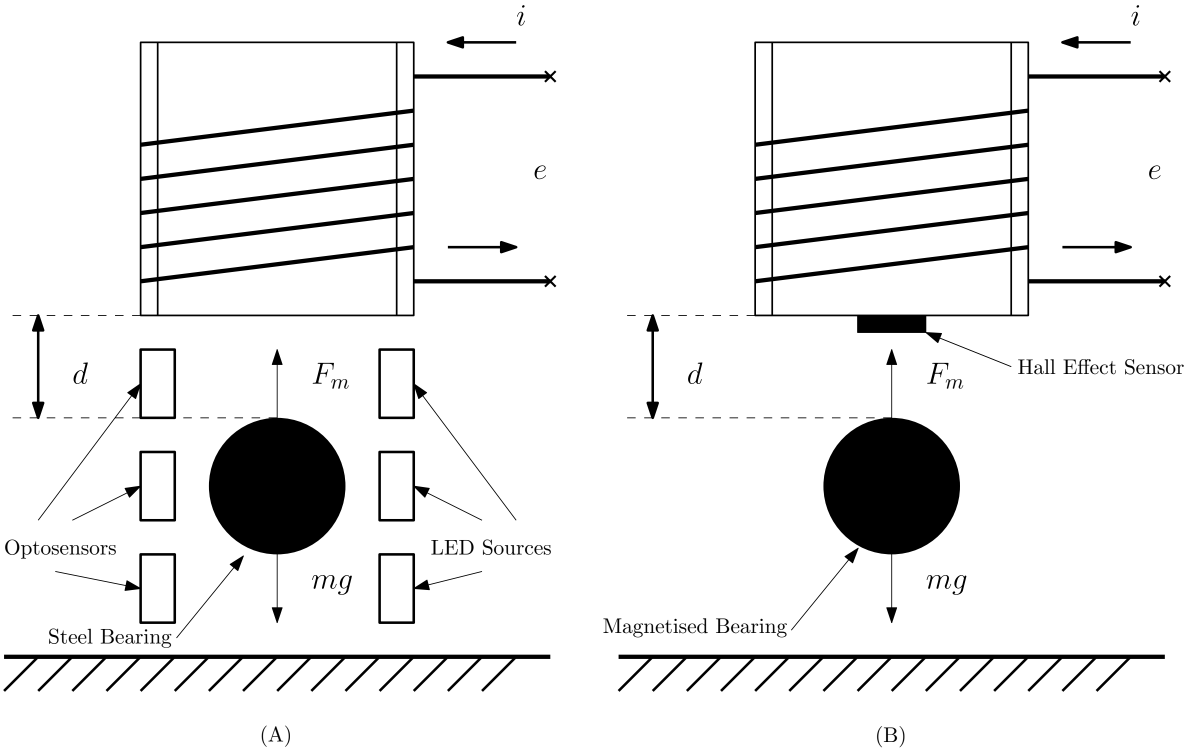


Figure 4.1: *Magnetic levitation system using (A) optical sensors and (B) a hall effect sensor.*

and is the primary source of the system's nonlinearity.

Equation (4.1) gives the speed/distance relationship, (4.2) states the relationship between current, resistance, inductance and voltage within the magnet and (4.3) equates the gravitational force with the magnetic force.

The nonlinear inductance $L(x)$ in (4.2) is a nonlinear function of the position of the ball[87], namely x_1. Several types of approximations can be found across the literature, three of which are presented here.

$$L(x) = L_1 + \frac{L_0 x_0}{x} \tag{4.4}$$

$$L(x) = L_1 + \frac{L_0}{1 + \frac{x}{a}} \tag{4.5}$$

$$L(x) = L_1 + e^{\frac{-x}{a}} \tag{4.6}$$

Equation (4.4) assumes that the inductance varies inversely with respect to the position of the ball, where x_0 is an arbitrary reference position for the inductance [98]. Equation (4.5) was proposed in [99], where parameters L_0 and L_1 are selected empirically to be 0.0244 and 0.38 respectively. The length constant a is equal to $d_{ball}/9$ where d_{ball} is the ball diameter set at 6cm. Equation (4.6) was proposed in [100] and provides a simple way of expressing nonlinear inductance by reducing its properties to single value a. The three equation are compared using typical values and this is shown in Figure 4.2. Although, (4.5) and (4.6) provide a good approximation,

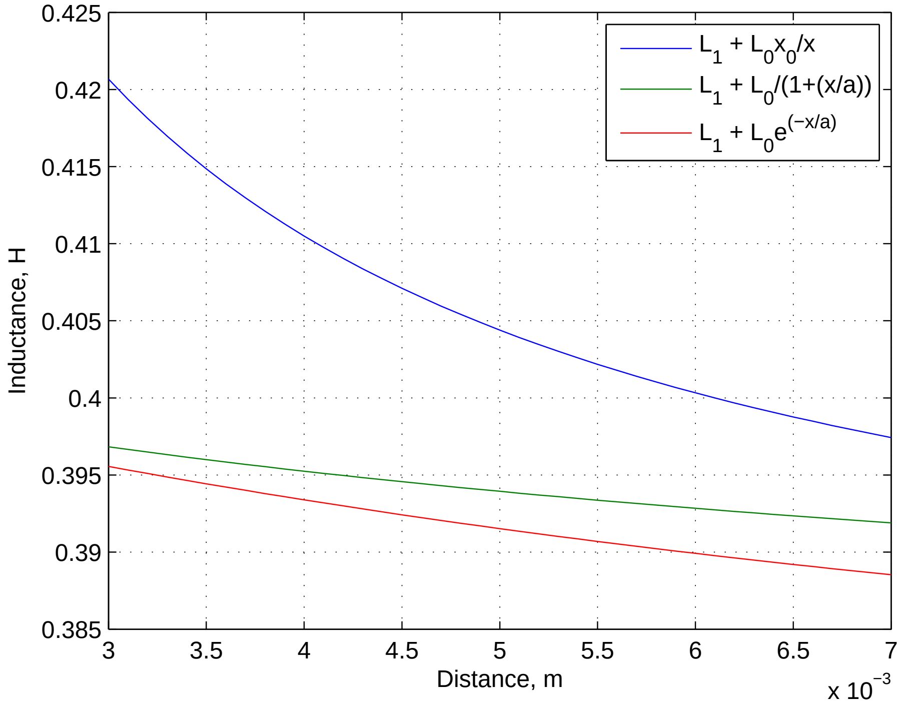


Figure 4.2: *Comparison between different estimations of nonlinear inductance* $L(x)$

(4.4) is selected in this derivation for its simplicity.

By combining (4.4) with (4.2) yields

$$\frac{dx_3}{dt} = \frac{e}{L} - \frac{R}{L}x_3 + \frac{L_0 x_0 x_3}{L x_1^2}\frac{dx_1}{dt} \tag{4.7}$$

A conservation of energy argument shown in [101] validates the expression $C = L_0 x_0 / 2$, where C is known as the magnetic force constant. Equation (4.7) can be re-written as

$$\dot{x_3} = -\frac{R}{L}x_3 + \frac{2C}{L}\frac{x_2 x_3}{x_1^2} + \frac{e}{L} \tag{4.8}$$

The dynamics of the magnetic levitation system described in (4.1), (4.2) and (4.3) can be expressed in the form of

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{4.9}$$

where $f(x)$ and $g(x)$ and $h(x)$ are given by

$$f(x) = \begin{bmatrix} x_2 \\ g - \frac{C}{m}\frac{x_3^2}{x_1^2} \\ -\frac{R}{L}x_3 + \frac{2C}{L}\frac{x_2 x_3}{x_1^2} \end{bmatrix} \qquad g(x) = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} \qquad h(x) = x_1 \tag{4.10}$$

This model will be used for designing various controllers for a magnetic levitation system.

4.3 Design of Feedback Linearising Controller

Initially, a feedback linearising controller for magnetic levitation system is designed. This controller has attracted great deal of interest from researchers during past four decades. The philosophy of this approach is to algebraically transform nonlinear dynamics into a fully or partially linear one, such that available linear control methods can be used to design controllers for nonlinear systems.

A single-input nonlinear system in the form (4.9) with $f(x)$ and $g(x)$ being smooth vector fields on $\mathbf{R}^n$, is said to be *input-state linearisable* if there exist a region $\mathbf{\Omega}$ in $\mathbf{R}^n$, a diffeomorphism $\phi : \Omega \to \mathbf{R}^n$, and a nonlinear feedback control law

$$u = \alpha(x) + \beta(x)w \tag{4.11}$$

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

$$\dot{z} = Az + bw \tag{4.12}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 & . & . & 0 \\ 0 & 0 & 1 & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & 1 \\ 0 & 0 & 0 & . & . & 0 \end{bmatrix} B = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ 0 \\ 1 \end{bmatrix} \tag{4.13}$$

The functions $\alpha(x)$ and $\beta(x)$ are given by

$$\begin{aligned} \alpha(x) =& -\frac{L_f^n h(x)}{L_g L_f^{n-1} h(x)} \\ \beta(x) =& \frac{1}{L_g L_f^{n-1} h(x)} \end{aligned} \tag{4.14}$$

The various steps for designing a feedback linearising controller can be found [102]. The feedback linearising controller as proposed in [103, 104] is obtained by performing a nonlinear change of coordinates (or transformation) given by

$$z = T(x) = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} y \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ L_f^2 h(x) \end{bmatrix}$$

The operation involves finding the Lie derivative $h(x)$, $L_f h(x)$ and $L_f^2 h(x)$. Note that $h(x) = x_1$. Hence

$$L_f h(x) = \frac{d\,[x_1]}{dx} f(x) = \begin{bmatrix} \frac{dx_1}{dx_1} & \frac{dx_1}{dx_2} & \frac{dx_1}{dx_3} \end{bmatrix} \begin{bmatrix} x_2 \\ g - \frac{C}{m}\frac{x_3^2}{x_1^2} \\ -\frac{R}{L}x_3 + \frac{2C}{L}\frac{x_2 x_3}{x_1^2} \end{bmatrix} = x_2 \tag{4.15}$$

$$L_f^2 h(x) = \frac{d\,[L_f h(x)]}{dx} f(x) = \begin{bmatrix} \frac{dx_2}{dx_1} & \frac{dx_2}{dx_2} & \frac{dx_2}{dx_3} \end{bmatrix} \begin{bmatrix} x_2 \\ g - \frac{C}{m}\frac{x_3^2}{x_1^2} \\ -\frac{R}{L}x_3 + \frac{2C}{L}\frac{x_2 x_3}{x_1^2} \end{bmatrix} = g - \frac{C}{m} \cdot \frac{x_3^2}{x_1^2} \tag{4.16}$$

Thus,

$$z_1 = x_1 \tag{4.17}$$

$$z_2 = x_2 \tag{4.18}$$

$$z_3 = g - \frac{C}{m}\left[\frac{x_3}{x_1}\right]^2 \tag{4.19}$$

The system output state is x_1, which represents the position of the ball. The system state is restricted to the region of the state-space where $x_1 > 0$ (the height of the ball) and $x_3 > 0$ (the current in the coil) to ensure that this transformation is invertible. Note that this transformation results in the new state variable z_1, z_2 and z_3 being, the position of the ball, its velocity and acceleration respectively. Using these new state variables, the dynamics is expressed as

$$\dot{z}_1 = z_2 \tag{4.20}$$

$$\dot{z}_2 = z_3 \tag{4.21}$$

$$\dot{z}_3 = L_f^3 h(x) + L_g L_f^2 h(x) u \tag{4.22}$$

By selecting $u = \alpha(x) + \beta(x)w$ where,

$$\begin{aligned}\alpha(x) =& -\frac{L_f^3 h(x)}{L_g L_f^2 h(x)} \\ =& \frac{Lx_1^2}{x_3}\left[\frac{R}{L}\frac{x_3^2}{x_1^2} + \frac{x_3^2 x_2}{x_1^3}\left(1 - \frac{2C}{Lx_1}\right)\right] \qquad (4.23) \\ \beta(x) =& \frac{1}{L_g L_f^2 h(x)} \\ =& -\frac{mLx_1^2}{2Cx_3} \qquad (4.24)\end{aligned}$$

the dynamics of the system in the transformed co-ordinates becomes

$$\begin{bmatrix} \dot{z_1} \\ \dot{z_2} \\ \dot{z_3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w \end{bmatrix} \qquad (4.25)$$

Let the arbitrary reference trajectory $z_{1ref}(t)$, $z_{2ref}(t)$ and $z_{3ref}(t)$ and reference input $j_{ref}(t)$ be given such that

$$\begin{aligned} z_{1ref}(t) &= x_{ref}(t) \\ z_{2ref}(t) &= \frac{d}{dt} z_{1ref}(t) \\ z_{3ref}(t) &= \frac{d}{dt} z_{2ref}(t) \\ j_{ref}(t) &= \frac{d}{dt} z_{3ref}(t) \end{aligned}$$

A linear feedback for w can then be chosen as

$$w = K_0 \int_0^t (z_{1ref} - z_1)dt + K_1(z_{1ref} - z_1) + K_2(z_{2ref} - z2) + K_3(z_{3ref} - z_3) + j_{ref} \qquad (4.26)$$

Since the system in (4.25) is in controllable canonical form, it is straightforward to choose the feedback gains K_0, K_1, K_2 and K_3 to place the closed loop poles of the system in the left-half plane to ensure tracking of the reference trajectory.

This control law can be used to stabilise a nonlinear magnetic levitation system. This will be shown in the next section.

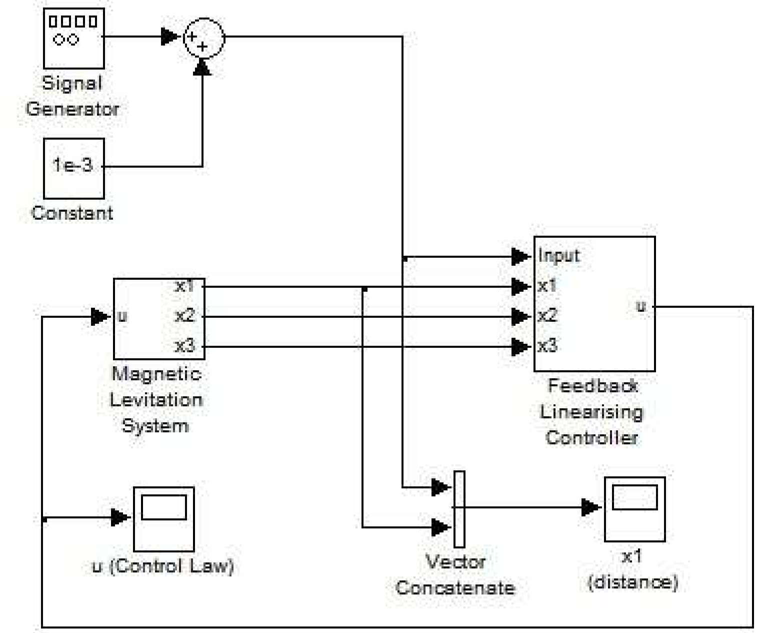


Figure 4.3: *Schematic diagram for magnetic levitation system with feedback linearising controller implemented in Simulink*^{TM}*.*

4.3.1 Simulation Results of Feedback Linearising Controller: Multi-bit Implementation

The feedback linearising controller is simulated in a Simulink^{TM}environment considering the following parameters: $m = 0.017$kg, $C = 61 \times 10^{-6}$ Nm^{2}A^{-2}, $L = 0.38$H, $R = 90\Omega$ and $x_0 = 0.005$m. It is worth noting that these values can be changed to suit the parameters of any magnetic levitation system. The complete system schematic is shown in Figure 4.3. Here, the set-point is labelled *Input*, and the states of the magnetic levitation x_1, x_2 and x_3 represent the position, velocity and current respectively. The control law is labelled as u.

The feedback linearising controller is shown in Figure 4.4. The *Transform* block implements equations (4.17), (4.18) and (4.19). The *Alpha and Beta* block implements (4.23) and (4.24). The *Feedback Loop* implements (4.11), and the *Control Law* block implements equation (4.26). A velocity *observer* block is also shown, however, this is not essential for this simulation.

The system gains are set to $K_0 = 2 \times 10^6$, $K_1 = 950000$ and $K_2 = 80000$ and $K_3 = 900$. The simulation results are shown in Figure 4.5. Successful tracking is demonstrated using a square wave signal switching from 0.8 to 1.6 millimeters.

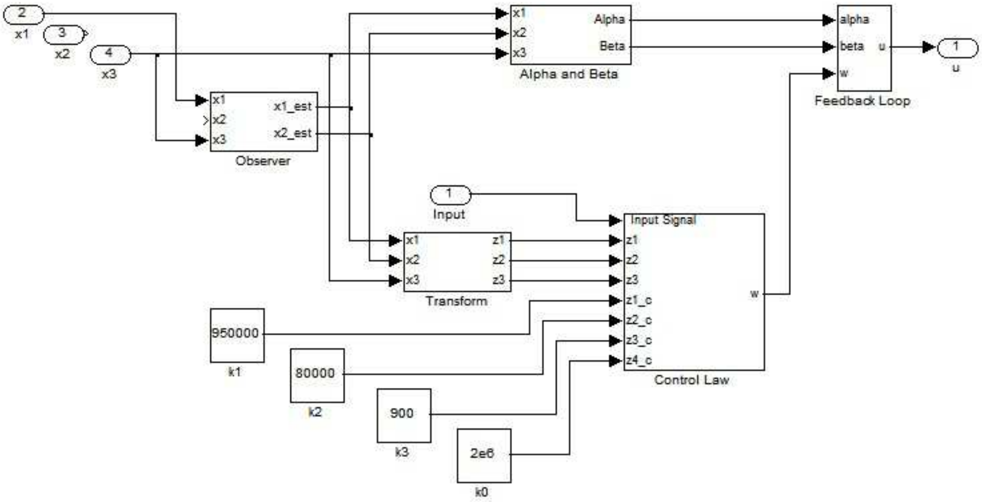


Figure 4.4: *Schematic diagram for feedback linearising controller.*

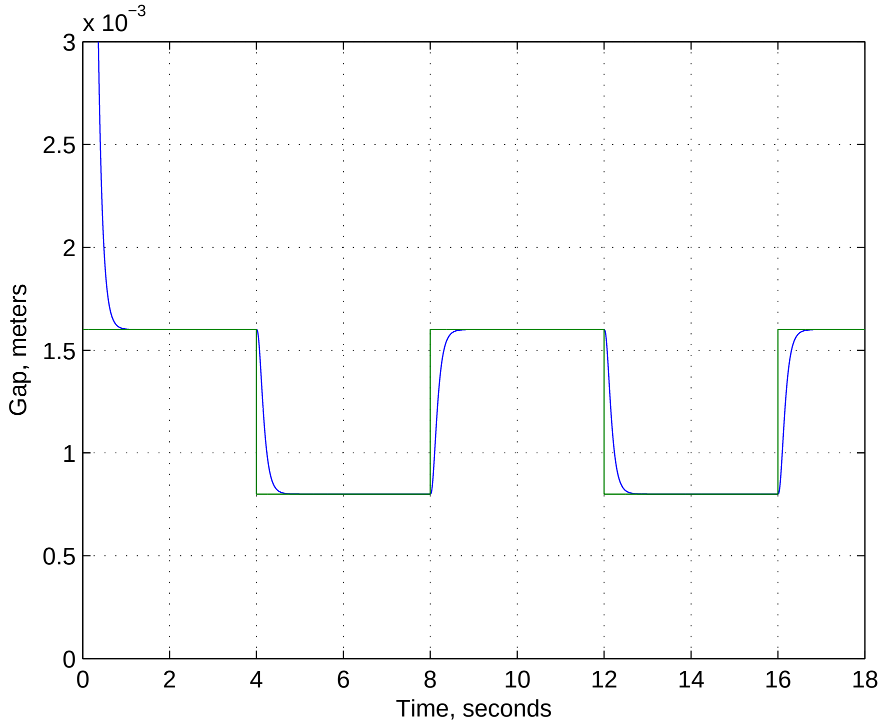


Figure 4.5: *Tracking performance simulation of a feedback Linearisation controller using multi-bit control*

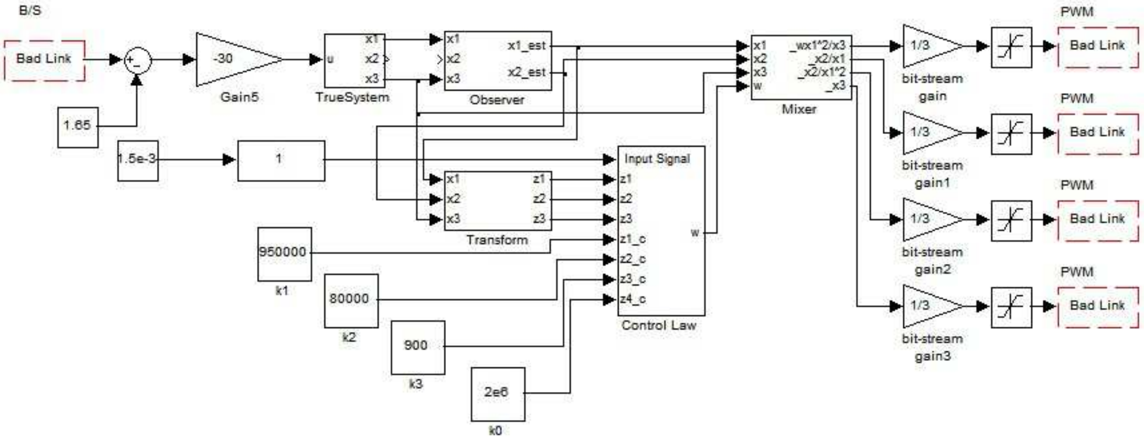


Figure 4.6: *Magnetic levitation system with feedback linearisation controller implemented for bit-streams.*

4.3.2 Simulation Results of Feedback Linearising Controller: Bit-stream Implementation

The purpose of the current research is to validate the use of bit-stream for control purposes. To further emphasise the versatility of bit-streams, the controller presented in this section will be partially implemented using bit-stream. Please note that the work presented serves as proof-of-concept only.

Figure 4.6 shows the complete setup for the feedback linearising controller using bit-streams. Due to the complexity of the nonlinear controller, certain components have to be merged. The unavailability of a *variable multiplication* function block prevents the full implementation of the nonlinear controller. However, it is possible to simulate the magnetic levitation plant in Simulink$^{\text{TM}}$using the equations (4.1), (4.2) and (4.3), and simplify the nonlinear controller by collecting variables together as shown below

$$\left[w\frac{x_1^2}{x_3}\right], \quad \left[\frac{x_2}{x_1}\right], \quad \left[\frac{x_2}{x_1^2}\right], \quad [x_3]$$

The multiplexer block labelled *Mixer* in Figure 4.6 combines all the states, and outputs the four expressions that are fed externally to the FPGA. The implementation of the aforemention blocks can be seen in Figure 4.7. Results are shown to be satisfactory.

Although results of implementing the feedback linearising controller gives satisfactory results in simulations, the control law is more complex in nature. In our quest for getting a simpler controller than the feedback linearising one, the next phase of study involves designing state space based model predictive controller considering the linesarised state space model of the system. The design procedure and implementation of state space based model predictive control is described in the following section.

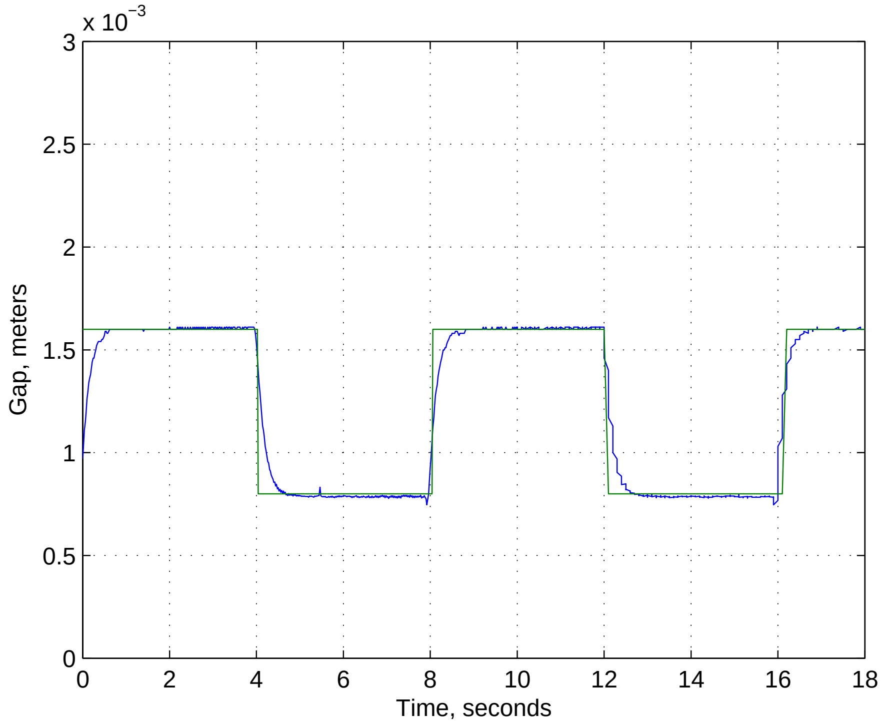


Figure 4.7: *Tracking performance simulation of a feedback linearisation controller using bit-streams control*

4.4 Linearised Model of Magnetic Levitation System

Since the state space based model predictive control design requires a linear model of the plant, initially the system is linearised around a specific operating point.

The linearised state space model of the system around the operating point $x = x_0$ is obtained from (4.1), (4.2) and (4.3) using Taylor series approximation and is given by

$$\begin{bmatrix} \dot{x} \\ \dot{v} \\ \dot{i} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{2CRI_0^2}{Lmx_0^3} & \frac{2CI_0^2}{mx_0^3} & \frac{-R}{L} \end{bmatrix} \begin{bmatrix} x \\ v \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{2CI_0}{Lmx_0^2} \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \\ i \end{bmatrix} \tag{4.27}$$

The transfer function between the output displacement x and input voltage is given by

$$H(s) = \frac{X(s)}{E(s)} = \frac{1}{s^3 \frac{L_0 m x_0^2}{2Ci_0} + s^2 \frac{Rmx_0^2}{2Ci_0} - s\frac{L_0 i_0}{x_0} - R\frac{i_0}{x_0}} \tag{4.28}$$

4.4.1 State Space Based Model Predictive Control

The design procedure of the Model Predictive Control (MPC) is briefly described here following [24]. The MPC essentially computes a trajectory of a future manipulated variable u to optimise the future behaviour of the plant output y. At every time instant, MPC requires the on-line solution of an optimisation problem to compute optimal control inputs over a fixed number of future time instants, known as the *finite horizon*. MPC systems are designed based on a mathematical model of the plant. In the present study the MPC design is based on a state space model of the system.

Consider a single-input single-output system described by

$$\begin{aligned} x_m(k+1) &= A_m x_m(k) + B_m u(k) \\ y(k) &= C_m x_m(k) \end{aligned} \tag{4.29}$$

where u is the manipulated variable or input variable; y is the process output; and x_m is the state variable vector with dimension n_1. Applying a difference operation on both sides of (4.29) and defining a new state variable vector

$$x(k) = [\Delta x_m(k)^T \quad y(k)]^T \tag{4.30}$$

where

$$\Delta x_m(k) = x_m(k) - x_m(k-1)$$

the system of (4.29) can be expressed as

$$\begin{aligned} \overbrace{\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix}}^{x(k+1)} &= \overbrace{\begin{bmatrix} A_m & o_m^T \\ C_m A_m & 1 \end{bmatrix}}^{A} \overbrace{\begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}}^{x(k)} + \overbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}^{B} \Delta u(k) \\ y(k) &= \overbrace{\begin{bmatrix} O_m & 1 \end{bmatrix}}^{C} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \end{aligned} \tag{4.31}$$

where $\Delta u(k) = u(k) - u(k-1)$, $O_m = \overbrace{[0 \quad 0 \quad \cdots \quad 0]}^{n_1}$. The triplet (A, B, C) is called the augmented model, which will be used in the design of predictive control.

The MPC determines a sequence of future control increments

$$\Delta u(k+i-1|k), \quad i = 1, \ldots\ldots, N_c$$

by optimising the following cost function

$$J = \sum_{i=1}^{N_p} [r(k+i) - y(k+i|k)]^2 + \sum_{i=1}^{N_c} \lambda \, [\Delta u(k+i-1|k)]^2 \tag{4.32}$$

where N_p is the prediction horizon and N_c is the control horizon. The term $r(k+i)$ for $i = 1, \ldots N_p$ are the future setpoints, usually presumed to be known, and $\lambda > 0$ is the control weighting sequence which can be used to tune the performance of the controller.

Denoting the predicted values of output, control and reference as:

$$\begin{aligned} R &= \left[\begin{array}{ccc} r(k+1) & \ldots & r(k+N_p) \end{array} \right]^T \\ Y &= \left[\begin{array}{ccc} y(k+1|k) & \ldots & y(k+N_p|k) \end{array} \right]^T \\ \Delta U &= \left[\begin{array}{ccc} \Delta u(k+1|k) & \ldots & \Delta u(k+N_c|k) \end{array} \right]^T \end{aligned} \tag{4.33}$$

the cost function (4.32) can be expressed as:

$$J = (R - Y)^T (R - Y) + \Delta U^T Q_w \Delta U \tag{4.34}$$

where $Q_w = \lambda I_{N_c \times N_c}$ is a diagonal matrix. The predicted value of the output over the prediction horizon N_p can be obtained from the incremental state space model (4.31) and is expressed as:

$$Y = Fx(k_i) + \phi \Delta U \tag{4.35}$$

where,

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix} \tag{4.36}$$

$$\phi = \begin{bmatrix} CB & 0 & \ldots & 0 \\ CAB & CB & \ldots & 0 \\ CA^2B & CAB & \ldots & 0 \\ \vdots & & & \\ CA^{N_P-1}B & CA^{N_p-2}B & \ldots & CA^{N_p-N_c}B \end{bmatrix} \tag{4.37}$$

For a given set-point signal $r(k_i)$ at sample time k_i, within a prediction horizon of N_p, the objective of the predictive control system is to bring the predicted output as close as possible to the setpoint signal, where it is assumed that the setpoint signal remains constant in the optimisation window. The optimal control parameter vector ΔU can be obtained by minimising the cost function (4.34) and is given by

$$\Delta U = (\phi^T \phi + Q_w)^{-1} \phi^T [R - Fx(k_i)] \tag{4.38}$$

Note that R is a data vector that contains the set-point information expressed as

$$R = \overbrace{\begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}}^{N_p}{}^T r(k_i) = \bar{R} r(k_i), \tag{4.39}$$

The optimal solution of the control signal is linked to the setpoint signal $r(k_i)$ and the state variable $x(k_i)$ via the following equation

$$\Delta U = (\phi^T \phi + Q_w)^{-1} \phi^T [\bar{R} r(k_i) - Fx(k_i)] \tag{4.40}$$

Since MPC follows receding horizon principle, only the first element of ΔU at ${k_i}^{\mathrm{th}}$ sample is taken as the incremental control and is expressed as

$$\begin{aligned} \Delta u(k_i) &= \overbrace{\begin{bmatrix} 1 & \dots & 1 \end{bmatrix}}^{N_c}{}^T (\phi^T \phi + Q_w)^{-1} \phi^T [\bar{R} r(k_i) - Fx(k_i)] \\ &= K_y r(k_i) - K_{mpc} x(k_i) \end{aligned} \tag{4.41}$$

where K_y is the first element of

$$(\phi^T \phi + Q_w)^{-1} \phi^T \bar{R}$$

and K_{mpc} is the first row of

$$(\phi^T \phi + Q_w)^{-1} \phi^T F$$

The closed loop system with the MPC based control law is given by

$$\begin{aligned} x(k+1) &= Ax(k) - BK_{mpc} x(k) + BK_y r(k) \\ &= (A - BK_{mpc}) x(k) + BK_y r(k) \end{aligned} \tag{4.42}$$

The stability and performance of the closed loop system is therefore determined from the eigenvalues of $(A - BK_{mpc})$ and applied to (4.27). The simulation results are presented below.

4.4.2 Simulation Results of State Space Model Predictive Control

The state space based Model Predictive Controller (MPC) is based on the discrete equivalent of the continuous time model of (4.27). The parameters of the model used in the present simulation are $m = 0.017$kg, $C = 61 \times 10^{-6}$ Nm^{2}A^{-2}, $L = 0.38$H, $R = 90\Omega$, $x_0 = 0.005$m and $I_0 = 0.2614$A.

The discrete time state space model of the system at a sampling interval of 10kHz is given by

$$\begin{bmatrix} x(k+1) \\ v(k+1) \\ i(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.0001 & \text{4.961e-9} \\ 0.00461 & 1 & \text{9.883e-5} \\ 91.85 & 0.39 & 0.9766 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \\ i(k) \end{bmatrix} + \begin{bmatrix} \text{3.272e-11} \\ \text{9.797e-7} \\ 0.01952 \end{bmatrix} u \tag{4.43}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ v(k) \\ i(k) \end{bmatrix} \tag{4.44}$$

To illustrate the feasibility and effectiveness of the proposed bit-stream controller, the performance of MPC is first investigated in an idealised form using Matlab$^{\text{TM}}$and then as a complete bit-stream controller using Mentor Graphics' ModelSim$^{\text{TM}}$. The performance of the bit-stream controller is assessed by comparing its response to ideal controller.

The success of MPC is dependent on several factors such as choice of prediction windows and control horizon as well as the tuning parameter λ of (4.32). Several methods exist to optimise the parameters of these controllers. However, since the objective of the present study is to investigate whether a bit-stream approach can be a viable alternative for implementing controllers for the magnetic levitation system, these are selected by trial and error which gives acceptable result. For the prediction horizon of $N_p = 300$, control horizon $N_c = 200$ and $\lambda = 0.0001$, the gains of the predictive controller becomes

$$K_y = 96.325 \tag{4.45}$$

$$\begin{aligned} K_{mpc} &= \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix} \\ &= \begin{bmatrix} 1.3529 & 1.5471 & 0.4473 & 1.2843 \end{bmatrix} \end{aligned} \tag{4.46}$$

Figure 4.8 shows the tracking performance of the MPC from Matlab$^{\text{TM}}$/Simulink$^{\text{TM}}$as the reference position is varied around the nominal value of 5mm.

The Matlab$^{\text{TM}}$/Simulink$^{\text{TM}}$based controller is ideal in the sense that it has infinite resolution. The results of the simulation show that the MPC is capable of tracking the position satisfactorily. To successfully implement a bit-stream MPC controller several issues need to be addressed. The MPC gains which are obtained using Matlab$^{\text{TM}}$/Simulink$^{\text{TM}}$are hard coded into the bit-stream implementation. The schematic layout of the bit-stream based MPC is shown in Figure 4.9 and it shows the interconnections between several blocks such as, the magnetic levitation system (MagLev Plant), a 12-bit precision float-to-bit-streams conversion block and a bit-stream to float conversion block. Since the magnitude of signals used in the control law calculations varies over a wide range, there is a need for normalisation before converting them into bit-streams. Otherwise, these may lead to the saturation of bit-stream lines. The constants c_1, c_2, c_3 and c_4 shown in Figure 4.9 essentially normalise the magnitudes of the signals. In the present example, the values of these constants are $c_1 = 20000$, $c_2 = 200$, $c_3 = 2$ and $c_4 = 75$. Note that these

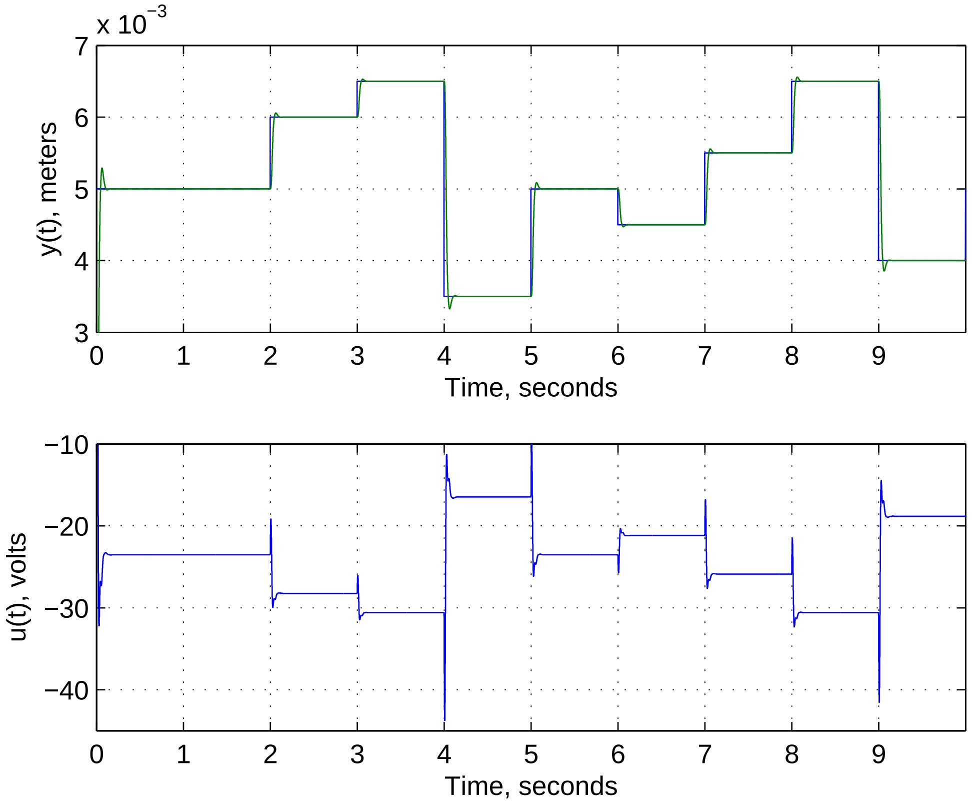


Figure 4.8: *Tracking performance of MPC implemented using multi-bit: Upper plot shows the displacement and the lower plot shows the control signal*

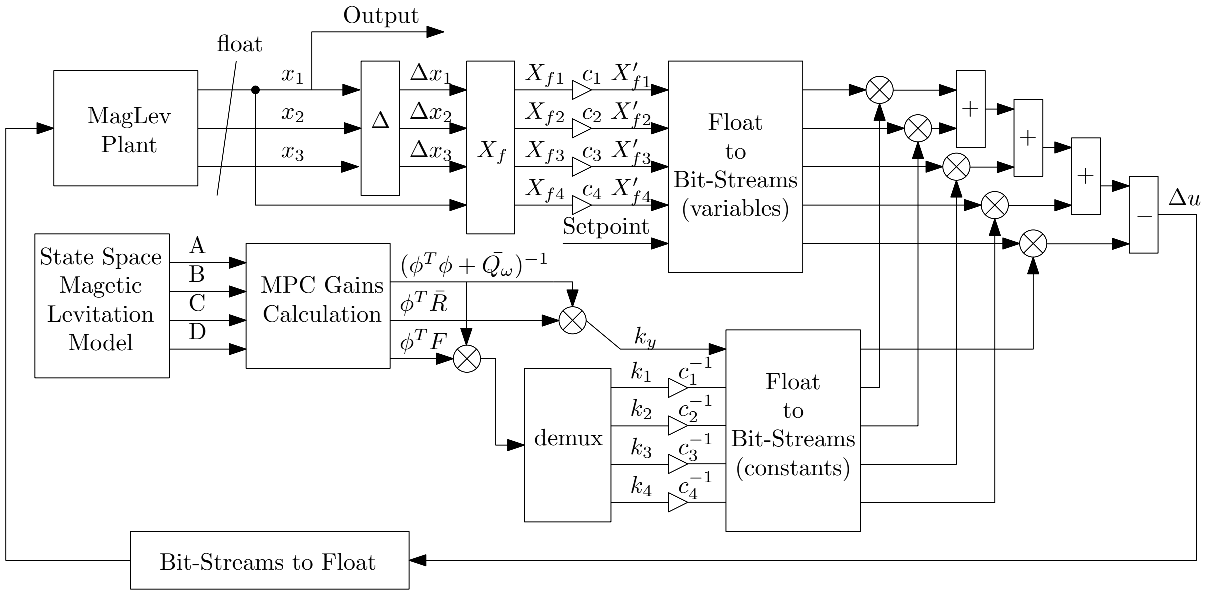


Figure 4.9: *Layout of bit-stream based MPC implementation using bit-streams.*

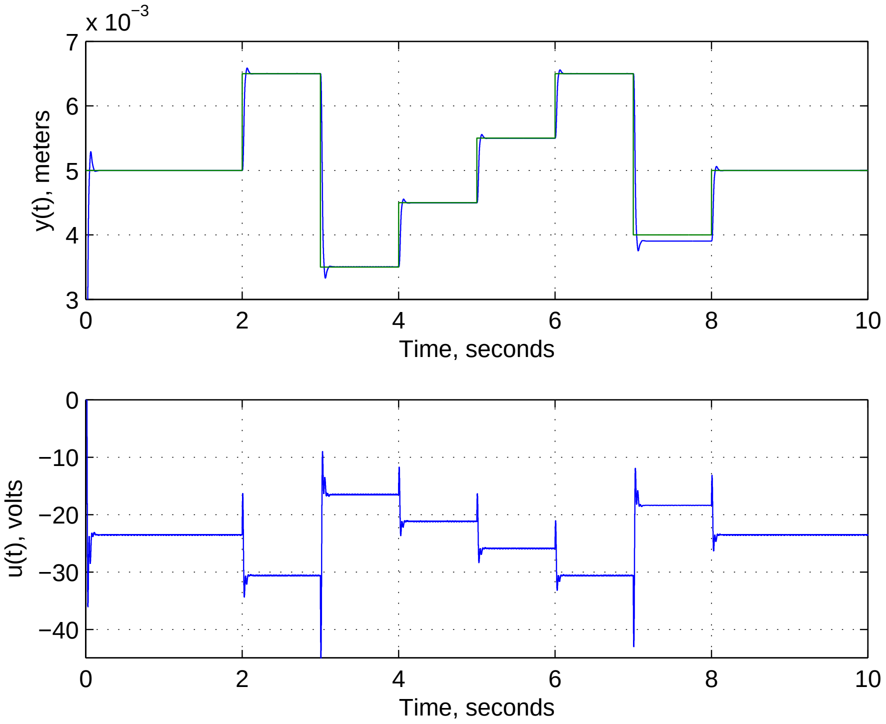


Figure 4.10: *Tracking performance of MPC implemented using bit-stream: Upper plot shows displacement and lower plot shows the control signal*

values can easily be selected with preliminary knowledge about the signal levels.

The bit-stream controller is implemented in ModelSim$^{\text{TM}}$and results of the simulation are shown in Figure 4.10 using a 12-bit window and $f_b = 5$MHz. Results of the bit-stream controller exhibit very similar response which demonstrate that the proposed bit-stream controller perform as expected. Furthermore, the performance of the bit-stream MPC is studied by tracking a sinusoidal reference and the result is shown in Figure 4.11 which shows that the proposed MPC implemented using bit-stream can effectively track the sinusoidal reference. Note that, although the control law of state space based MPC is simpler compared to the feedback linearising controller, the practical implementation of this controller requires all the state variables to be available. It is a demerit of such controller. Although, an observer can be designed to estimate the states, this option is not explored in this study due to further complexity which may arise. During the last phase of the control design of unstable magnetic levitation system, the simple generalised predictive controller is designed and implemented in bit-stream and its performance compared with another controller; the lead compensator.

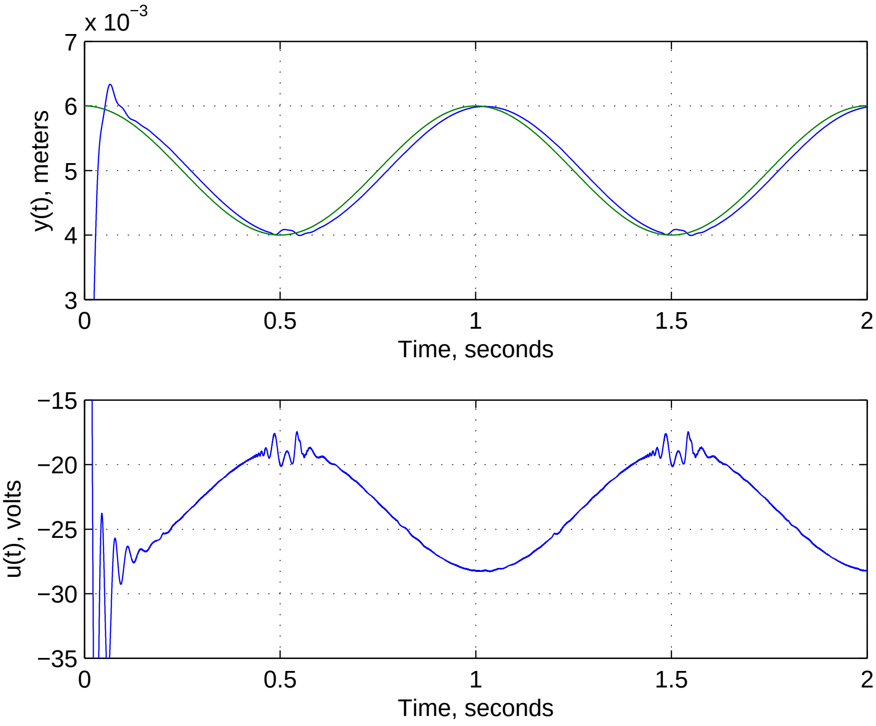


Figure 4.11: *Sinusoidal tracking performance of MPC implemented using bit-stream: Upper plot shows displacement and lower plot shows the control signal*

4.5 Generalised Predictive Controller for Magnetic Levitation System

The discrete time model of the magnetic levitation system is obtained by discretising the linearised state space model of (4.27) at a sampling frequency of 20Hz. The model is given by

$$\left(1 - 1.4252q^{-1} + 0.4966q^{-1}\right) y(t) = \left(0.0015 + 0.0012q^{-1}\right) q^{-d} u(t-1) + \frac{e(t)}{\Delta} \qquad (4.47)$$

The GPC for this system is designed following similar procedure of GPC design discussed in Chapter 3. The tuning parameters of the GPC such as prediction and control horizon as well as the control weighting λ is optimised using genetic algorithms. The optimised tuning parameters are given in Table 4.1. The controller is implemented using the experimental prototype shown in Figure 4.12. The experimental results obtained from both multi-bit and bit-stream GPC are shown in Figure 4.13 and Figure 4.14. From the results it is observed that bit-stream can be considered as a viable alternative to implement advanced controllers such as GPC.

For comparative purposes, a lead compensator tracking performance has also been implemented. Only the final implementation of the lead compensator using bit-streams will be presented. The tracking performance is shown in Figure 4.15.

GPC Parameter	Optimised Value
N_2	2
N_u	1
λ	0.50

Table 4.1: *Optimised GPC parameters obtained via genetic algorithms for magnetic levitation system.*

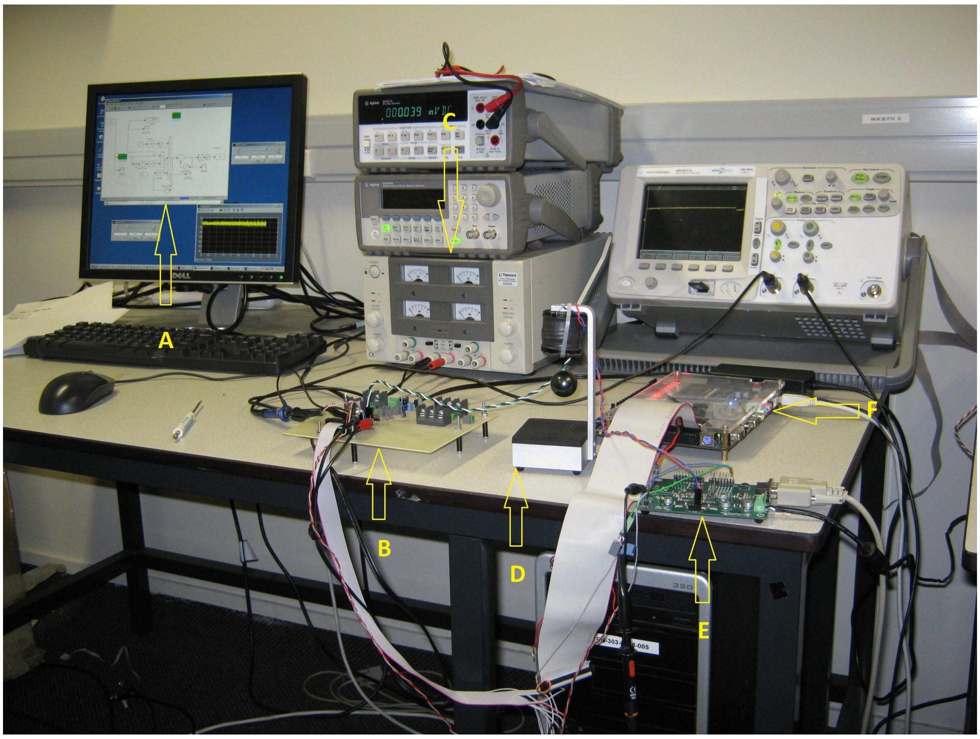


Figure 4.12: *Photograph of magnetic levitation setup: (A) Simulink^{TM}interface, (B) analogue/bit-Stream converter, (C) Power supply, (D) magnetic levitation system, (E) Hilink data acquisition board and (F) DE2 Cyclone II FPGA.*

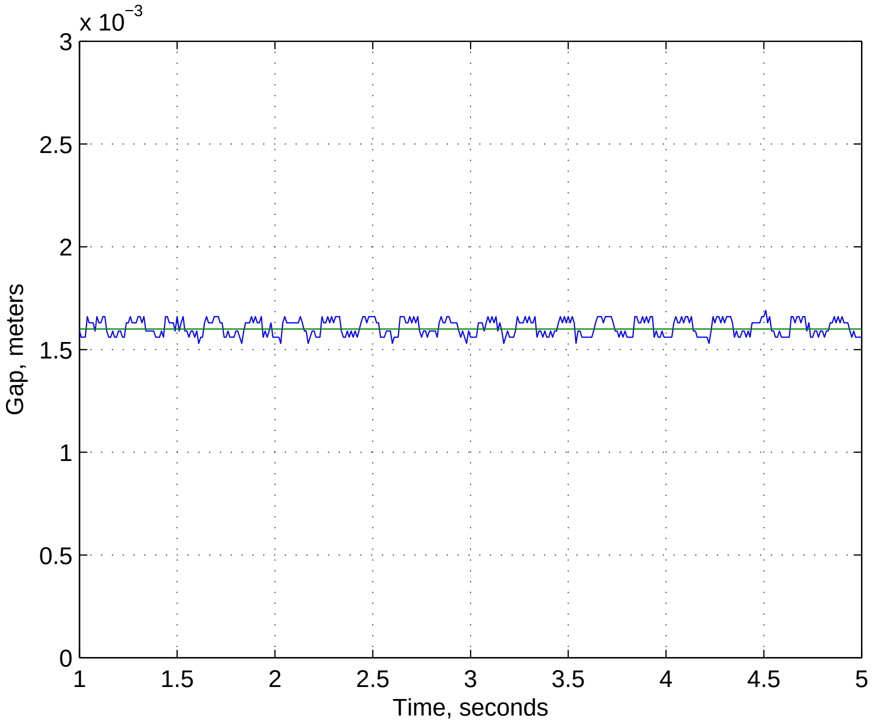


Figure 4.13: *Tracking performance of a generalised predictive controller using multi-bit control*

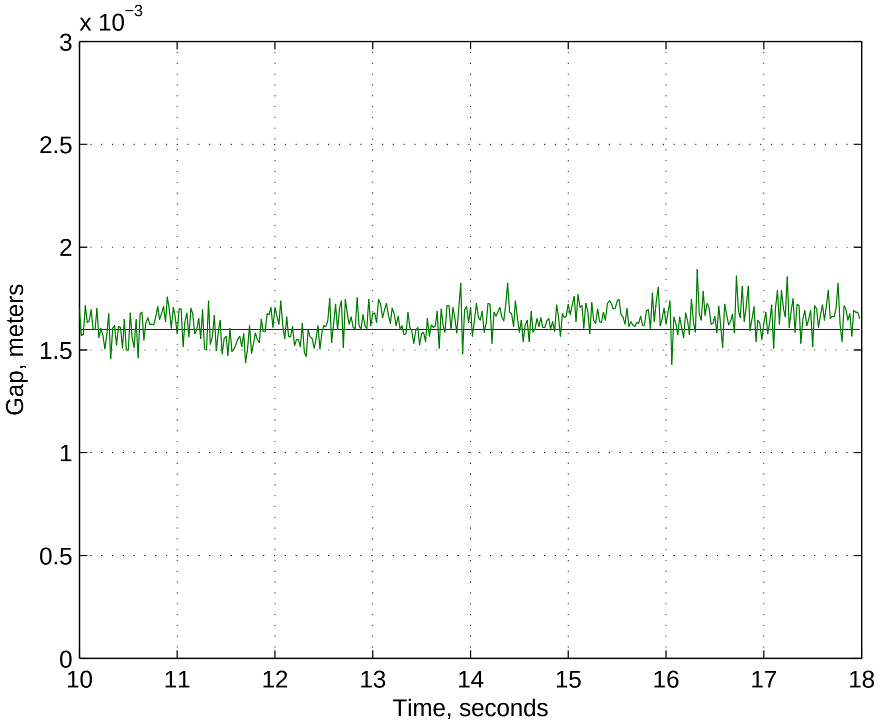


Figure 4.14: *Tracking performance of a generalised predictive controller using multi-bit and bit-stream control*

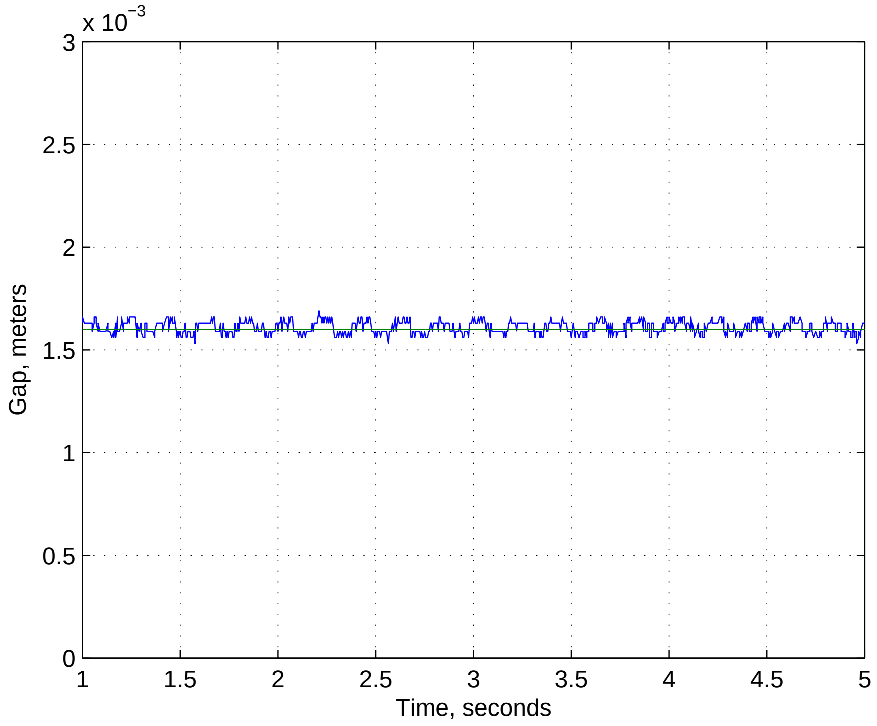


Figure 4.15: *Tracking performance of a lead compensator using bit-stream control*

4.6 Conclusions

A bit-stream based model predictive controller has been designed for an unstable magnetic levitation system considering the linearised model of the system around a nominal operating point. Unlike most other implementation approach of digital controller which use multi-bit signals on a processor, this method uses single bit-streams. The controller is hard coded and implemented using Mentor Graphics' ModelSim and its performance is compared to the ideal controller implemented using MATLAB/Simulink$^{\text{TM}}$. A nonlinear feedback linearisation controller is successfully implemented in simulation using multi-bit and bit-stream encoding. Results of the simulation convincingly demonstrate that bit-stream approach of implementing controllers is a better alternative due to several advantages this offers. A real magnetic levitation system is stabilised using multi-bit and bit-stream for the encoding of the control signals. The results show successful tracking behaviour. This approach of controller design can easily be extended to control more complex systems and is the scope of further research.

CHAPTER 5

Efficient Structure Selection of Polynomial Nonlinear Systems using Evolutionary Programming

5.1 Introduction

The problem of nonlinear system identification has attracted the attention of researchers during the past several decades and has been studied in many different frameworks and investigated from different view points using various optimisation techniques. Although several possible representation of nonlinear systems have been proposed including traditional functional series of Volterra and Wiener [33], Legendre polynomials [34], neural networks [35, 36] and weighted maps [37], the polynomial Nonlinear Auto Regressive Moving Average with eXogenous inputs (NARMAX) model proposed in [38, 39] has attracted considerable interest during last three decades.

The polynomial NARMAX model provides a concise representation of a wide class of nonlinear systems where the output is expressed as a nonlinear functional expansion of lagged inputs, outputs and noise terms. Moreover, due to the polynomial NARMAX model being linear-in-the-parameters, parameters associated with different terms can be estimated using simple least squares based algorithms. However, one of the disadvantages of the polynomial NARMAX models is that the total number of terms increases rapidly with the increase in maximum lag of inputs, outputs and noise terms and degree of nonlinearity. Thus detecting the model structure or which terms to include into the model is crucial for nonlinear system identification. Amongst several approaches which have been proposed to address this structure selection problem, the Orthogonal Least Squares (OLS) algorithm with Error Reduction ratio (ERR) introduced by [41] and later modified by Billings and co-workers [42, 105, 43, 44] provides an efficient and elegant solution. The ERR criterion of orthogonal algorithm essentially computes the significance of model terms based on their ability to explain the output variance. However, certain linear and nonlinear systems may produce erroneous results [45] as OLS-ERR may select spurious terms.

An alternate approach of structure selection are methods based on evolutionary computation. Genetic Algorithms (GA), Evolutionary Strategies (ES) and Evolutionary Programming (EP) are all part of the larger class that is evolutionary computation [76, 77]. Such programs emulate natural genetic operators to execute a parallel global search [106]. Using Darwin's theory of evolution, a selection of the fittest solutions are obtained by applying operators such as crossover, mutation and extinction.

Evolutionary Computation using GAs have been used in system identification since 1990 in controller design [76]. Since then several search methods have been developed and applied for system identification and parameter estimation [106, 107, 108, 109, 110, 111] and the literature related to this are well documented in [47]. Each of these techniques describes a new way of encoding, ranking and selecting their solutions based on some fitness guidelines. The use of these derivative-free optimisation methods for structure selection can sometimes be better than their counterparts as they do not adhere themselves to mathematical limits such as differentiability or continuity. Moreover, evolutionary computation algorithms converge to a single result via the optimisation of a cost function.

In the present study, a different strategy based on EP is proposed for selecting significant terms from a polynomial NARMAX model. The algorithm uses an adaptive mutation based on the relative fitness and introduces several parameters including an internal penalty function and a time-to-live parameter to reject spurious terms which are likely to be included into the model under noisy environment. The organisation of this chapter proceeds as follows: §5.2 briefly introduces the polynomial NARMAX model. In §5.5 the method of structure selection using EP is explained. Results of simulation are shown in §5.6.

5.2 Representation of Nonlinear Systems Using NARMAX model

Models based on difference equations constitute one of the most important classes of models for both linear and nonlinear system [112]. These models often arise naturally from physical laws and contain relatively few parameter which can be estimated using parameter estimation algorithms which are independent upon specialised input signals. A model known as the Auto Regressive Moving Average (ARMA) model can be used to represent linear systems using a difference equation model as shown below.

$$y(k) = \sum_{i=1}^{n_y} \theta_i y(k-i) + \sum_{i=1}^{n_u} \theta_{n_y+1} u(k-i) \tag{5.1}$$

where,

$y(k)$ is the output,

$u(k)$ is the input, with order n_u

n_y, n_u denote maximum lags of output and input respectively.

Due to the outstanding performance of the ARMA model in representing linear systems, the search for a nonlinear equivalent to the difference equation seems logical. Billings and Leontaritis [39, 38] proposed a model with similar representation for nonlinear systems, known as the NARMAX (Nonlinear Auto regressive Moving Average with eXogenous inputs). A large class of nonlinear systems can be expressed using the NARMAX model, which is given as:

$$y(k) = F^l\left[y(k-1)\dots y(k-N_y)u(t-d)\dots u(k-d-N_u)\epsilon(k-1)\dots\epsilon(k-N_\epsilon)\right]+\epsilon(k) \quad (5.2)$$

where $u(t)$ and $y(t)$ represent the measured input and output respectively and $\epsilon(t)$ is the prediction error. N_y, N_u, and N_ϵ represent the the maximum order of lags in the input, output and prediction error respectively, and $F^l[\cdot]$ is some nonlinear function with degree of nonlinearity l. A subset of the NARMAX description model called the NARX (Nonlinear Auto Regressive model with eXogenous input) model is represented as

$$y(k) = F^l\left[y(k-1)\dots y(k-N_y)u(k-d)\dots u(k-d-N_u)\right] \quad (5.3)$$

where the nomenclature remains the same. It has been proven in [39, 38] that a nonlinear discrete time invariant system can always be represented by (5.3) in the region around the equilibrium point provided it satisfies the following criteria

1. The original response function of the system is finitely realisable. This means that a finite dimensional systems must exist such that it realises the original function[113]. This condition excludes distributed parameter systems.

2. If the system is operated around the chosen equilibrium point, there exist a linearised model. This implies that if the system were to be perturbed by some small external input, the linearity of the system is still guaranteed.

Since the function F^l in (5.2) can take a variety of forms, the identification process becomes a much more complicated task. Three possible forms of the NARMAX model such as the polynomial model, the output-affine difference equation model and rational difference equation models are presented below.

5.2.1 Polynomial Models

In a real system, the exact form of the NARMAX model function F^l as presented in (5.2) may be too complex to be obtained. However, it is possible to approximate F^l by a polynomial expansion of degree l. The output can therefore be expressed as;

$$\begin{aligned} y(k) =& \sum_{|\theta|}^{l}[\alpha_\theta y^{\theta_1}(k-1)y^{\theta_2}(k-2)\dots y^{\theta_{n_y}}(k-n_y)e^{\theta_{n_y+n_u}}(k-n_y)e^{\theta_{n_y+n_u+1}}(k-1) \\ & \dots e^{\theta_{n_y+n_u+n_e}}(k-n_e)] + e(k) \end{aligned} \tag{5.4}$$

where $\theta = (\theta_1, \theta_2, ...\theta_{n_y+n_u+n_e})$ is a non-negative multi index with $\mid \theta \mid = \theta_1 + \theta_2 + \ldots + \theta_{n_y+n_u+n_e}$ and α_θ are real constants. One of the most important advantages of the polynomial NARMAX model is that it is linear-in-the-parameters and thus can be estimated using linear least squares methods.

5.2.2 Output-affine Difference Equation Models

The development of the output-affine difference equation model started with the works in polynomial response maps [114]. This type of equation relates the sampled output signals to sampled inputs of the form:

$$\begin{aligned} y(k) =& \frac{\sum_{i=1}^{n} f_i[u(k-1)\dots u(k-n), e(k-1)\dots e(k-n)]}{f_0[u(k-1)\dots u(k-n), e(k-1)\dots e(k-n)]} \\ & + \frac{f_{n+1}[u(k-1)\dots u(k-n), e(k-1)\dots e(k-n)]}{f_0[u(k-1)\dots u(k-n), e(k-1)\dots e(k-n)]}e(k-i) \\ & + e(k) \end{aligned} \tag{5.5}$$

where n is the order of the system and f_i, for $i = 0, \dots, n+1$, are polynomials of finite degree of nonlinearity. Note that in practical application, it is required that the system description contains an input, i.e. $[u(k-1), \dots u(k-n)] \neq 0$. The advantage of the output-affine model is that it is globally valid and admits a state affine representation as shown below:

$$\begin{aligned} x(k+1) &= A[u(k))]x(k) + B[u(k)] \\ y(k) &= C[u(k)]x(k) \end{aligned} \tag{5.6}$$

where $A(\cdot), B(\cdot)$ and $C(\cdot)$ are matrix and vector valued polynomials of finite degree of nonlinearity.

5.2.3 Rational Difference Equation Models

Polynomial NARMAX models have shown to be an excellent tool when modelling several nonlinear system as well as practical systems [112]. However, in certain type of heavily nonlinear systems, polynomial NARMAX has proven to be inadequate [115]. To overcome these problems a stochastic rational model is introduced. This model can be represented as

$$y(k) = \frac{f_{\alpha}^{l_{\alpha}}[y(k-1),\dots y(k-n), u(k-1),\dots u(k-n), e(k-1)\dots e(k-n)]}{f_{\beta}^{l_{\beta}}[y(k-1),\dots y(k-n), u(k-1),\dots u(k-n), e(k-1)\dots e(k-n)]} + e(k) \quad (5.7)$$

where $f_{\alpha}^{l_{\alpha}}$ and $f_{\beta}^{l_{\beta}}$ are polynomial functions of degree l_{α} and l_{β} respectively. These models are valid under very mild conditions.

Results from nonlinear approximations theory demonstrate that rational function $f(x) = a(x)/b(x)$ may provide a better approximation to an underlying function than polynomial functions. Rational functions need a smaller number of parameters to achieve similar or better accuracy. They can be used to represent certain types of singular or near singular behaviours which is not possible using polynomial approximations. Despite these advantages, rational models do have some disadvantage. There is a possibility of degeneracy [116]. A rational function may also be less convenient for certain analytical manipulation such as integration and differentiation. The derivatives for rational functions can be unbounded in contrast with the bounded derivatives for polynomial functions. The identification of rational models is more difficult because the models are nonlinear-in-the-parameters. A prediction error estimation algorithm derived by [117] can be used to estimate the parameters of rational models. However, this is computationally demanding and therefore difficult to apply to real nonlinear systems. However, in [118] it was demonstrated that with simple algebraic manipulation the identification of stochastic rational models can be performed using well known algorithm for linear-in-the-parameter models.

It is known that the NARMAX model can represent a wide class of nonlinear systems. Since the polynomial NARMAX model will be used in the present study, identification of this type will be reviewed in this section.

5.3 System Identification Using Polynomial NARMAX Models

Identification of discrete systems is largely based on the input/output relationships provided by linear difference equations. In contrast to Functional Series, these description may use a considerably less number of terms to describe nonlinear systems. In particular, the study of the Nonlinear Auto Regressive Moving Average with eXogenous Input (NARMAX) model as proposed in [39, 38] has gained considerable interest in the development of parameter estimation techniques, as it provides a much simpler and concise way of expressing nonlinear systems.

The NARMAX model enables the representation of a wide class of nonlinear systems where the output is expressed as a nonlinear functional expansion of delayed inputs, outputs and noise terms. Moreover, due to the polynomial NARMAX model being linear-in-the-parameters, parameters associated with different terms can be estimated using simple least squares based algorithms. However, one of the disadvantages of the polynomial NARMAX models is that the total number of terms increases rapidly with the increase in maximum lag of inputs, outputs, noise terms and degree of nonlinearity. Thus, a method is required to aid the NARMAX representation to only select the important terms.

Amongst several approaches which have been proposed to address this structure selection problem, the Orthogonal Least Squares (OLS) algorithm with Error Reduction Ratio (ERR) test introduced in [41] and later modified by Billings and co-workers [43, 42, 44, 105] provides an efficient and elegant solution. The ERR criterion of orthogonal algorithm essentially computes the significance of model terms based on their ability to explain the output variance. However, certain linear and nonlinear systems may produce erroneous results as OLS/ERR may select spurious terms [45].

It can be seen now that the NARMAX model may be used for structure and parameter identification of unknown systems [119]. To maximise the probability of a successful match, a methodology to verify, support and update the results is shown below.

- Testing for nonlinearity in data.
- Structure detection
- Parameter estimation
- Model validation and testing

The strength of the NARMAX methodology presents itself clearly as every stage becomes an updated augmented version of its previous and it is this desired sequence of events that makes the NARMAX description so powerful. It is because of these reasons that NARMAX representation will be the focus of this thesis.

5.3.1 Tests for Nonlinearity

It is pointless applying powerful nonlinear system identification algorithms if the system under test is linear [120]. There are many algorithms available which are able to distinguish between linear and nonlinear data, by means of correlation tests, filter detection method and Hilbert transform test [120]. These are easily implementable techniques and provide valuable model information information. The literature related to this can be found in [120].

5.3.2 Structure Detection and Parameter Estimation

Expanding equation (5.2) and defining $F^l\left[\cdot\right]$ to be an l degree polynomial, it is possible to express the model by

$$y(t) = \sum_{m=0}^{M} \theta_m p_m(t) + \epsilon(t) \tag{5.8}$$

Note that although the model of (5.8) is linear-in-the-parameters, the number of possible terms increases rapidly with increase in the degree of nonlinearity and maximum lag of inputs, outputs and noise terms. The maximum number of terms, M, in the NARMAX model presented in (5.2) is given by

$$M = 1 + \frac{\sum_{i=1}^{l}\left[n_{i-1}\left(N_y + N_u + N_\epsilon + i - 1\right)\right]}{i} \tag{5.9}$$

where $n_0 = 1$. For example, in the case of a second order dynamic process model ($N_y = N_u = 2$) with a first order noise model ($N = 1$) expanded as a quadratic polynomial ($l = 2$) would contain 21 unique terms. Models with excessive number of terms may be extremely effective in fitting the estimation data, but may not catch the true dynamics of the underlying system and may exhibit unwanted nonlinear behaviour. The structure selection problem is therefore crucial. The OLS algorithm with error reduction ratio (ERR) test proposed by Billings and coworker[reference] provide a better method of structure selection. In this study an alternate method based on evolutionary programming (EP) will be used for this purpose. This will be discussed in [section EP and term selection].

5.4 Model Validity Test

After fitting a NARMAX model to the input-output data, it is important to test for possible inadequacies of the fitted model, i.e. to check if the model has successfully captured the system dynamics and is not just a curve fit to one data set. Over the last few years, much work has been reported relating to the design and development of model validation tools[121, 122]. In the present study, the models are validated based on the predictive performance (cross-validation) and correlation test.

5.4.1 Cross Validation

One technique which can be used to judge the qualitative performance of a model is to assess the predictive performance based on both one-step ahead prediction of the system output defined as

$$\hat{y}_{osa}(k) = F^{N_l}\left[y(k-1),\ldots,y(k-n_y),u(k-1),\ldots,u(k-n_u),\epsilon(k-1),\ldots,\epsilon(k-n_\epsilon)\right] \tag{5.10}$$

and the model predicted output given by

$$\hat{y}_{mpo}(k) = F^{N_l}\left[\hat{y}_{mpo}(k-1),\ldots,\hat{y}_{mpo}(k-n_y),u(k-1),\ldots,u(k-n_u)\right] \tag{5.11}$$

A metric which measures the closeness of fit between the predicted output and measured output is Normalised Root Mean Square Error (NMSE) which is defined as[123]

$$NMSE = \sqrt{\frac{\sum\left[(\hat{y}) - y(k)\right]^2}{\sum\left[y(k) - \bar{y}\right]^2}} \tag{5.12}$$

where $\hat{y}$ is the predicted (either one step ahead or model predicted) output of the system. Another approach commonly known as cross validation, is based on dividing the available data set into two disjoint sets, the estimated and the test set. The former is used for estimation and the latter set is used for model validation.

5.4.2 Correlation Tests

The classical approach to validating identified linear models consist of computing the autocorrelation function of the residuals and the cross-correlation function between the residuals and the input[124]. This result has been extended to the case of validating identified nonlinear models [125, 122, 124]. The identified model produces acceptable predictions over different data sets only if its unbiased. If the model structure and the estimated parameters are correct, then the prediction error sequence $\varepsilon(k)$ should be unpredictable from all linear and nonlinear combinations of past inputs and outputs. This condition will hold if and only if the following correlation tests are satisfied [125, 120, 119].

$$\phi_{\varepsilon\varepsilon}(\tau) = E[\varepsilon(k-\tau)\varepsilon(t)] = \delta(\tau), \tag{5.13}$$

$$\phi_{u\varepsilon}(\tau) = E[u(k-\tau)\varepsilon(t)] = 0, \quad \forall\tau, \tag{5.14}$$

$$\phi_{[uu]''\varepsilon}(\tau) = E[(u^2(k-\tau) - \overline{u^2(t)})\varepsilon(t)] = 0, \quad \forall\tau, \tag{5.15}$$

$$\phi_{[uu]''\varepsilon^2}(\tau) = E[(u^2(k-\tau) - \overline{u^2(t)})\varepsilon^2(t)] = 0, \quad \forall\tau, \tag{5.16}$$

$$\phi_{(\varepsilon)[\varepsilon u]}(\tau) = E[\varepsilon(t)\varepsilon(k-1-\tau)u(k-1-\tau)] = 0, \quad \forall\tau \geq 0, \tag{5.17}$$

where $\delta(\tau)$ is the Kronecker delta, the bar signifies mean value, and $E[\cdot]$ denotes expectation. The underlying rationale of the correlation test is that for a model to be statistically valid, there should be no predictable information in the residuals. However, in practice only a finite data length is available which is contaminated with noise (which is rarely additive, Gaussian or white) measured with finite precision and subject to innumerable external influences in the environment and the measurement apparatus. Hence, confidence bands should be used to indicate if the correlation between variables is significant or not. For large N, the 95% confidence bands are approximately $\pm 1.95/sqrt(N)$ and any significant correlation will be indicated by points lying outside the respective confidence band [125, 119].

5.4.3 Generalised Frequency Response Function

The discrete polynomial NARMAX representation of a continuous time system is not necessarily unique, which means that it is possible to obtain several discrete descriptions for the same continuous time system. Thus, one cannot be certain that the difference in model structure is due to differences in the underlying physics, or if it is simply a reflection of non- uniqueness. However, no matter what the form of the model, if the model correctly captures the dynamics associated with the data sufficiently, this must reflect the correct linear and nonlinear frequency content of the system. In other words, although there may be a number of discrete time models that represent a continuous time system, the higher-order frequency response functions corresponding to each of the discrete models should correspond with those for the continuous time system description. This forms the motivation for analysing the above systems in the frequency domain.

Since continuous time differential equation models are to be estimated for different sets of data by curve fitting to the Generalised Frequency Response Function (GFRF) (computed by mapping the NARMAX models into the frequency domain), a brief review of this is given below.

The *n*th order GFRF of a system is defined as

$$H_n(j\omega_1, \dots j\omega_n) = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} h_n(\tau_1, \dots, \tau_n) e^{-j(\omega_1\tau_1 + \dots + \omega_n\tau_n)} d\tau_1 \dots d\tau_n \tag{5.18}$$

where $h_n(\tau_1, \dots \tau_n)$ is known as the nth-order Volterra Kernel or generalised impulse response function of order n. The frequency domain representation of a system having nonlinearity of degree N_l is given as

$$Y(j\omega) = \sum_{n=1}^{N_l} Y_n(j\omega) \tag{5.19}$$

$$= \sum_{n=1}^{N_l} \frac{1}{2\pi^{n-1}} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} H_n(j\omega_1, \dots j\omega_n) U(j\omega_1) \dots U(j\omega_n) d\omega_1 \dots d\omega_{n-1} \tag{5.20}$$

where $\omega = \omega_1 + \omega_2 + \dots + \omega_n$, $Y(j\omega)$ and $U(j\omega)$ represent the spectrum of the input and output respectively. The GFRF will be used in the proposed study, however due to its derivation and implementation being widely available, it will not be discussed further [72, 119, 120].

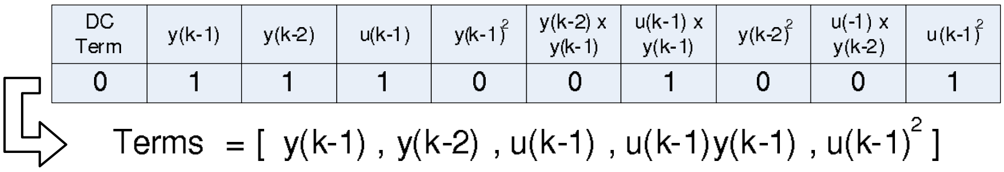
5.5 Structure Selection of Polynomial NARMAX Model using Evolutionary Programming

System identification based on evolutionary computation (EC) has attracted the attention of researchers in recent times [126]. Amongst different EC based methods, the GAs have been most popular [47] which utilises both crossover and mutation operators. The crossover operator allows for creation of new chromosomes by interchanging bit string sections within each of the two fittest chromosomes and the mutation operator prevents the results being trapped in local minima by varying a random gene within a chromosome at a low mutation rate.

Another EC based method which has been used for nonlinear system identification is genetic programming (GP) [47]. The method proposed in [111] combines the ERR criteria in [41] with GP to select the correct structure. The traditional GA based methods where all variables of interest are encoded using binary digits have been used for parameter estimation which is essentially achieved via indirect optimisation. Real coded GAs have recently been applied in parameter estimation of nonlinear systems when the structure is known a priori[127]. In the present study the problem of structure selection is addressed using Evolutionary Programming (EP) due to several advantages they offer such as the absence of a crossover operator. Although the proposed method follows closely the work found in [76, 77, 128], several strategies such as internal term penalty function, extinction operator, time-to-live parameter have been introduced to obtain a parsimonious model structure of a given nonlinear system. The algorithm applies the four main components of all evolutionary computation programs: Initialisation, Variation, Evaluation and Selection [76]. Since the original population often does not contain the true model structure, phylogenical learning is used [76].

5.5.1 Initialisation: Representation of Model Terms

It is assumed that all terms are independent and have no similarity within them. Each gene within a chromosome represents a possible term of the NARMAX model. A complete set of chromosomes is known as a population, which should be modified under the fitness search topography after each generation. The value held in each gene within a chromosome represents a binary state of the corresponding term. The state value 1 and 0 signify term availability and unavailability respectively. The entire population represent a series of possible model sets. Consider the chromosome shown in Figure 5.1. where each gene corresponds to different terms. If the genes of the chromosome is 0111001001 this corresponds to terms [$y(k-1)$, $y(k-2)$, $u(k-1)$, $u(k-1)y(k-1)$, $u^2(k-1)$].



DC Term	y(k-1)	y(k-2)	u(k-1)	$y(k-1)^2$	y(k-2) x y(k-1)	u(k-1) x y(k-1)	$y(k-2)^2$	u(-1) x y(k-2)	$u(k-1)^2$
0	1	1	1	0	0	1	0	0	1

Terms = [y(k-1) , y(k-2) , u(k-1) , u(k-1)y(k-1) , $u(k-1)^2$]

Figure 5.1: *Chromosome encoding for NARMAX model.*

5.5.2 Variation

Variation of the population matrix from generation to generation is key to successfully perform structure selection. Note that EP has only the mutation operator[76]. The mutation operator of EP therefore has to be carefully selected such that the chromosomes vary sufficiently and the best chromosomes pass on to the next generation. In the present study, an adaptive mutation rate which is proportional to the level of fitness is therefore proposed. The mutation rate is restricted to lie within 0 and 50% to achieve better convergence. It is worth noting that increasing the maximum mutation rate above 50% will often causes delay in convergence. The mutation rate used in the present study is computed from

$$\mu = 0.5 \times \frac{fit(chrom) - min[fit(pop)]}{max[fit(pop)] - min[fit(pop)]}\% \tag{5.21}$$

where the function $fit(\cdot)$ is either the scalar fitness of the chromosome or the fitness vector of the entire population. Moreover, a pseudo-elitists strategy is adopted where the single best chromosome in a generation is retained and pass on to the next generation.

5.5.3 Evaluation

The fitness of each chromosome is initially evaluated using the mean square error (MSE) function shown in (5.22).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y'_i - y_i)^2 \tag{5.22}$$

where y is the true output, y' is the model predicted output and N is the number of samples. It has been observed that this fitness function works well and the algorithm detects the correct structure only under ideal conditions when the data is noise free. When the data is corrupted by noise, using the fitness function of (5.21) results in some spurious terms being included into the model. In order to correct this problem, an internal term penalty (ITP) function, which will penalise the chromosomes with high number of terms is proposed. The ITP function is expressed as [76].

$$MSE_{ITP} = MSE + MSE \sum_{I=1}^{l_c} gene_i \tag{5.23}$$

where l_c is the length of the chromosome, $gene_i$ corresponds to the bit value of i-th bit of the chromosome. The summation over all $gene_i$ gives the total number of terms being selected. If the total number of terms are too large, then the insignificant terms need to be deleted using (5.23). The MSE is used to scale the sum thus preventing the ITP from becoming dominant. Note that (5.23) also works well under no noise conditions.

5.5.4 Selection

Equation (5.21) returns a mutation probability which lies between 0 and 50% for all the chromosomes. To spawn a new generation, every chromosome except the best one is mutated in order to produce a new population. This is called pseudo-elitism. Sometimes the coefficients of some terms in a chromosome become very small and their contribution in reducing the mean square error is insignificant. Instead of removing such terms from the gene pool as soon as they get selected, an extinction principle is followed where a time-to-live parameter is assigned to the terms whose coefficients lie below a threshold. If terms with very low coefficient appear longer than the assigned time-to-live parameter, they are removed from the gene pool. This accelerates and improves the search.

5.6 Simulation Results

The performance of the proposed method is illustrated considering several examples of nonlinear systems. The results of three examples, namely a Van der Pol oscillator, the modelling of a small scale wave force[72] and Dufffing's oscillator are presented here.

5.6.1 Example 1: Van der Pol Oscillator

To demonstrate the effectiveness of the proposed algorithm, consider the dynamic system below:

$$\frac{d^2y}{dt^2} + 2\zeta\omega_n[1 - y^2(t)]\frac{dy}{dt} + \omega_n^2 y(t) = u(t) \tag{5.24}$$

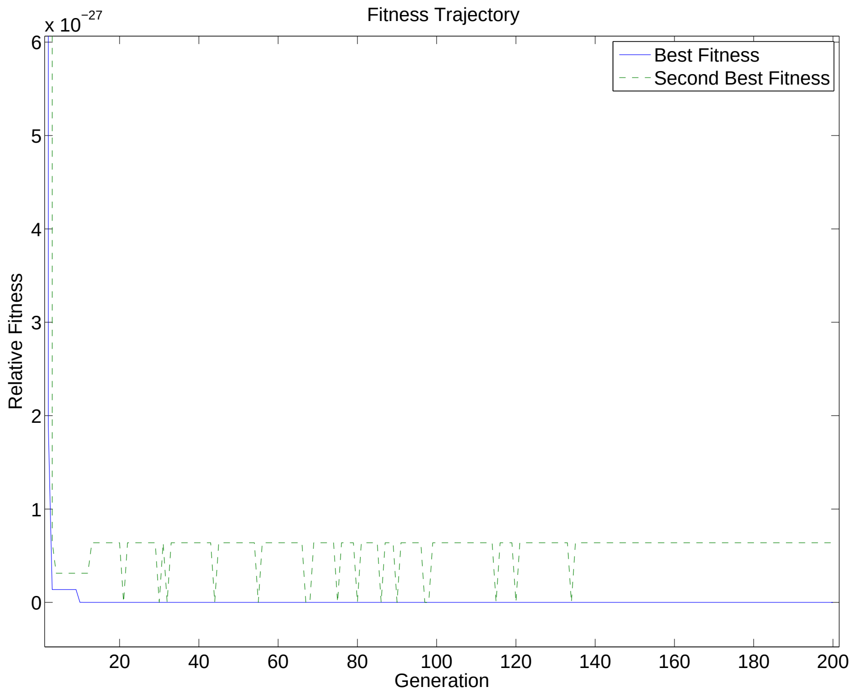


Figure 5.2: *Convergence curves for Van Der Pol Oscillator*

Equation (5.25) is the discrete equivalent of the Van Der Pol oscillator of (5.24) with a sampling rate of 200Hz.

$$\begin{aligned} y(k) &= 1.4858y(k-1) - 0.9859y(k-2) + 0.000025u(k-1) + 0.0141y(k-1)^3 \\ &\quad - 0.0141y(k-1)^2y(k-2) \end{aligned} \tag{5.25}$$

where $\zeta = 0.001$ and $\omega_n = 45$. 1000 pairs of input-output data are obtained using Matlab$^{\text{TM}}$by exciting the system with a band limited Gaussian white noise signal with zero mean. The output is corrupted by noise. In order to fit the discrete time model, 400 samples of input-output data are used for estimation purposes and the model validation is carried out with a sequence of 400 data points taken arbitrarily from the rest of the available data points. The specified model set consists of 84 possible terms which are generated from the NARMAX model of (5.2) with $l = 3$, $Ny = 3$, and $Nu = 3$. The model set is therefore encoded using an 84-bit chromosome where each gene represents one of the possible terms. The proposed algorithm is implemented through 200 generations with a population size of 5 chromosomes. The time-to-live parameter is set at 5 generations. The chromosomes are mutated using the proposed adaptive mutation rate, evaluated using MSE-ITP and finally pruned to yield an optimal structure. The convergence of algorithm for this system is shown in Figure 5.2. Figure 5.3 shows the occurrence of each term through the complete generation cycle. The result of the estimation is presented in (5.26).

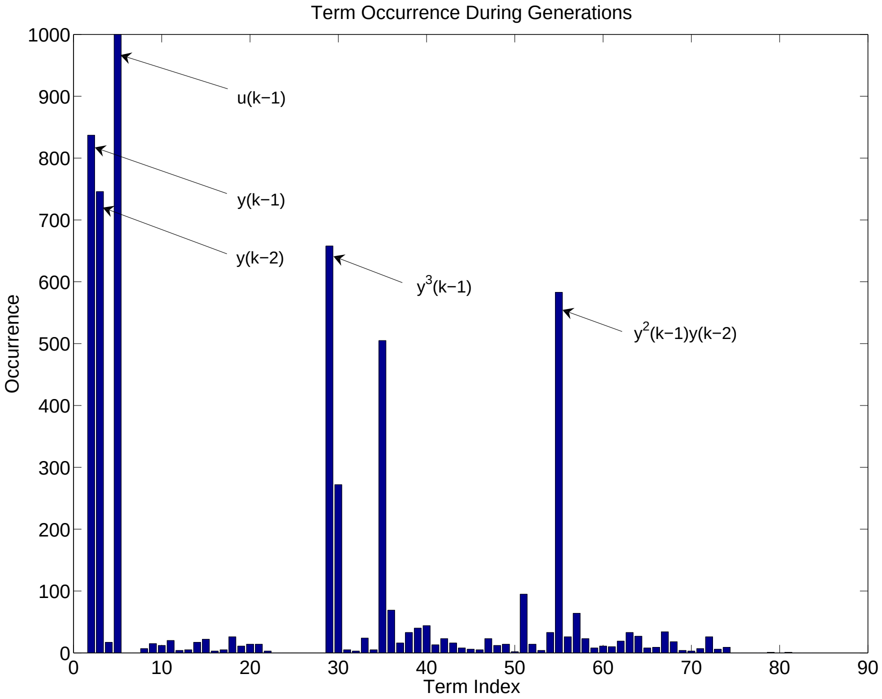


Figure 5.3: *Term occurrence for Van Der Pol Oscillator*

$$\begin{aligned} y(k) &= 1.4858y(k-1) - 0.9859y(k-2) + 0.000025u(k-1) + 0.0141y(k-1)^3 \\ &\quad - 0.0141y(k-1)^2y(k-2) \end{aligned} \tag{5.26}$$

From the model of (5.26) it is found that the structure of the models resembles the structure of (5.25) and the parameters are identical.

To further validate the model, the model predicted output (MPO) over the test set is computed and compared with the original data and is shown in Figure 5.4. The generalised frequency response functions of the true system are compared with those of the estimated model and is shown in Figure 5.5. It is observed that the proposed EP based method could successfully select the correct structure which preserves the original system dynamics.

5.6.2 Example 2: Modelling of Small Scale Wave force Data

The velocity and force time histories for 22 different rectangular wave spectra with a fixed cylinder were obtained from University of Salford [129]. The cylinder was subjected to random waves with rectangular spectral density functions having different bandwidths. The force was measured on a small cylindrical element and the input velocity is the ambient horizontal water particle velocity at the mid point of the element. Discrete NARMAX models were fitted to these different data sets [119]. Since the present objective is to illustrate the structure selection ability of the proposed

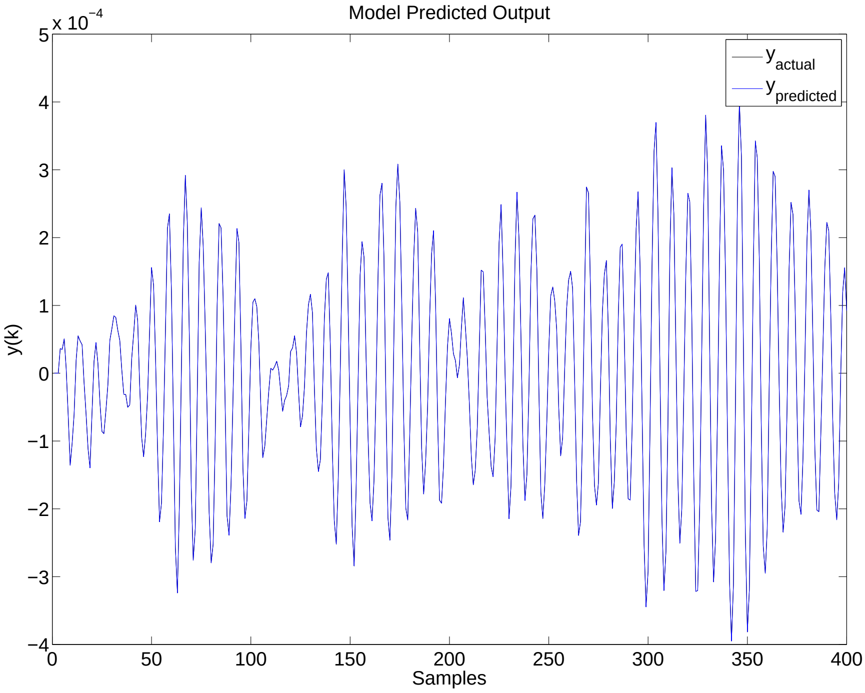


Figure 5.4: *Model Predicted Output for Van Der Pol Oscillator*

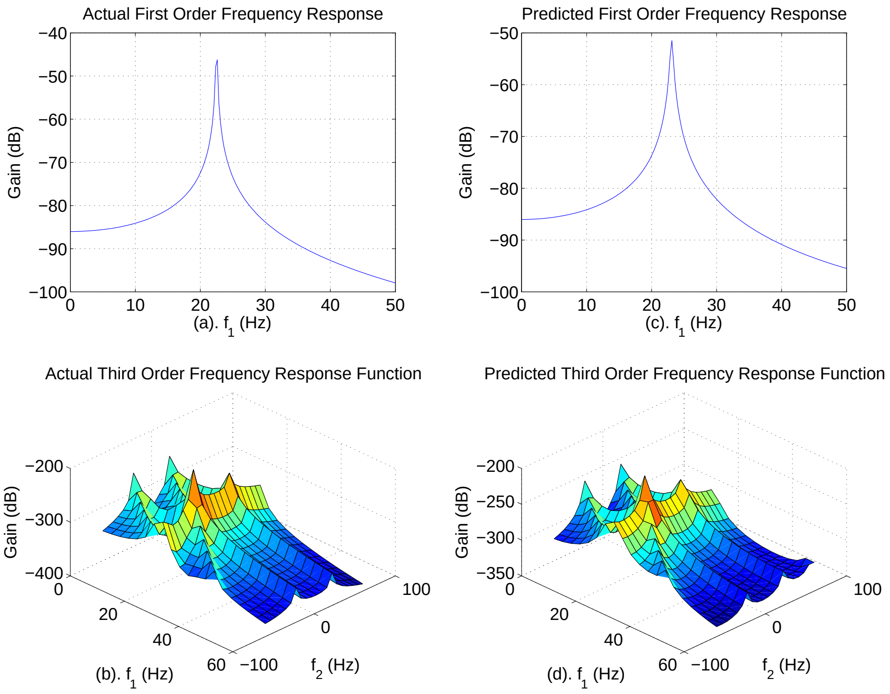


Figure 5.5: *Linear and third order frequency response functions (FRF) for a Van Der Pol Oscillator*

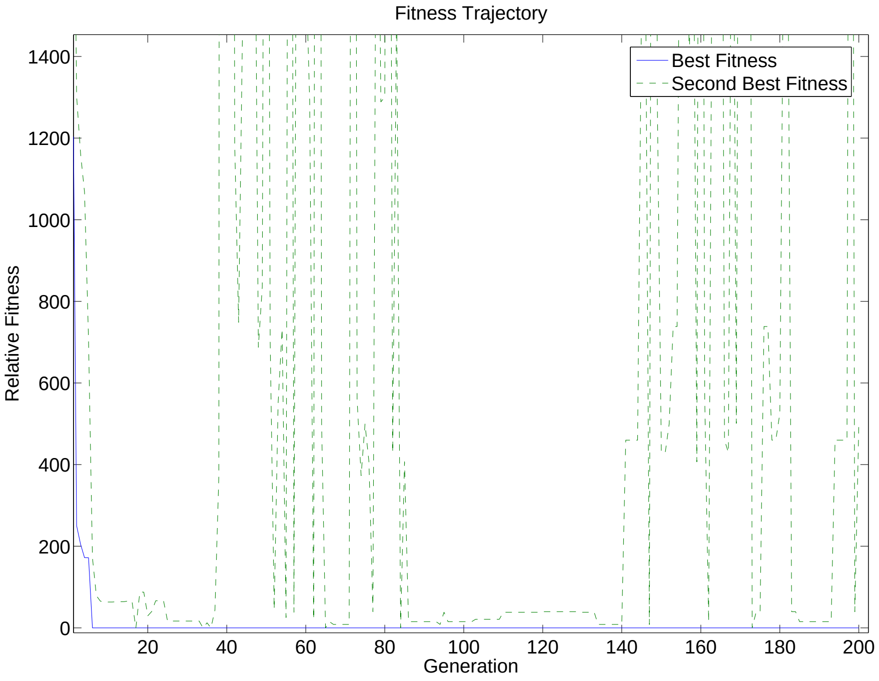


Figure 5.6: *Convergence curves for a small scale wave force data*

method, one of the discrete model fitted to such data [119] is considered here which is given by

$$\begin{aligned} y(k) = {} & 1.5593y(k-1) - 0.44738y(k-2) - 0.15585y(k-3) + 1.2829u(k-1) \\ & - 1.195u(k-2) + 4.8262u^3(k-3) \end{aligned} \tag{5.27}$$

The convergence of the algorithm is shown in Figure 5.6 The entire convergence plot has not been shown as the initial generations have a large level of relative fitness compared to the later generations. Note that the y-axis is only a relative fitness value and is not representative of the MSE. Figure 5.7 shows the occurrence of each term through the complete generation cycle. Each bar in Figure 5.7 represents one of the 84 terms produced by the NARMAX model. The selected terms are those with the highest number of occurrences. The result of the estimation is presented in (5.28).

$$\begin{aligned} y(k) = {} & 1.5593y(k-1) - 0.44738y(k-2) - 0.15585y(k-3) + 1.2829u(k-1) \\ & - 1.195u(k-2) + 4.8262u^3(k-3) \end{aligned} \tag{5.28}$$

From the model of (5.27) it is found that the structure of the model resembles the structure of (5.28) and the parameter estimates are identical.

To further validate the model, the model predicted output (MPO) over the test set is computed and compared with the original data and is shown in Figure 5.8. From the MPO plot, it is obvious that the proposed algorithm selected the correct structure and preserved the original dynamics.

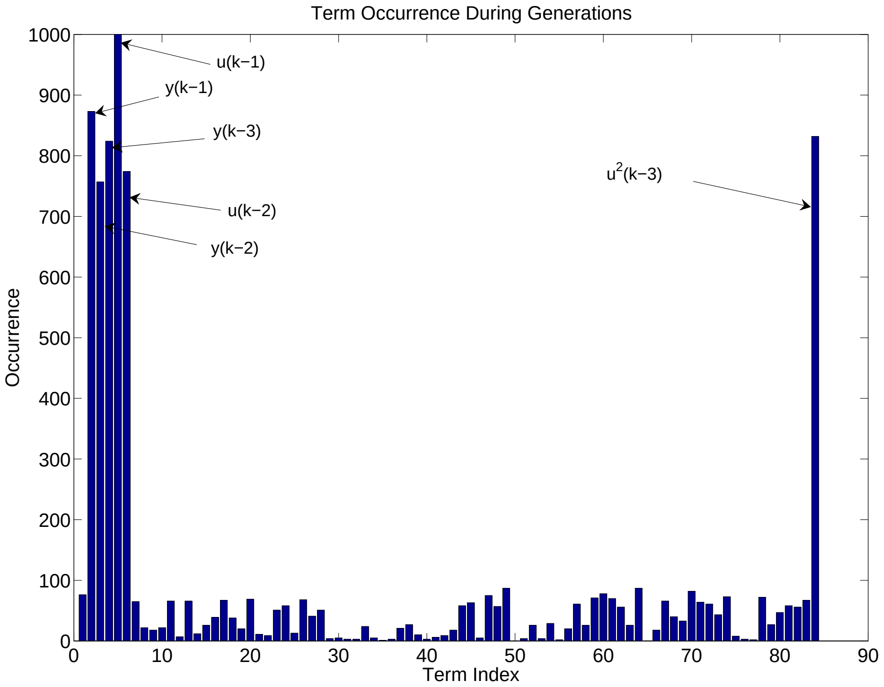


Figure 5.7: *Term occurrence for a small scale wave force data*

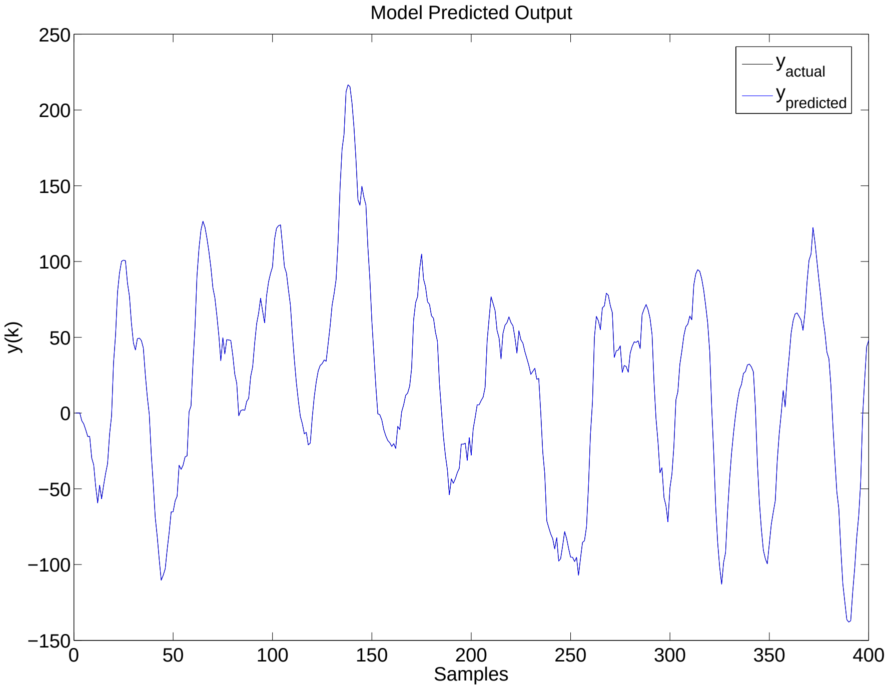


Figure 5.8: *Model predicted output for a small scale wave force data*

5.6.3 Example 3: Duffing's Oscillator

Consider the nonlinear dynamic system.

$$y(k) = 1.9144y(k-1) - 0.99435y(k-2) + 4 \times 10^{-6}u(k-1) + 0.079944y(k-1)^3 \tag{5.29}$$

Equation (5.29) is the discrete equivalent of Duffing's Equation (5.30) with $\xi = 0.01, w_n = 45\pi, \epsilon = 3$ and is obtained using forward difference method at a sampling frequency of 500Hz. Note that the dynamics shown in (5.30) matches with that of (5.30).

$$\frac{d^2y}{dt^2} + 2\xi w_n \frac{dy}{dt} + w_n^2 y(t) + w_n^2 \epsilon y(t)^3 = u(t) \tag{5.30}$$

1400 pairs of input-output data are obtained using Matlab$^{\text{TM}}$by exciting the continuous time system in (5.30) with a band limited Gaussian white noise signal with zero mean. The sampling frequency is 500Hz (i.e. the sampling time is 0.002s). The output is corrupted by 40dB of Normalised Gaussian White noise. In order to fit the discrete time model, 700 points of input/output data are used for estimation purposes and the model validation is done with a sequence of 700 data points taken arbitrarily from the rest of the available data points.

The specified model set consists of 120 possible terms which are generated from the NARMAX model of (5.9) with $l = 3$, $N_y = 5$, and $N_u = 2$. The DC term and all second order terms are removed to improve convergence. Since there is noise present, a sequence of $N_\epsilon = 10$ terms are added. The model set is therefore encoded using 101-bit chromosome where each gene represents one of the possible delay terms. The proposed algorithm is implemented through 500 generations with a population size of 5 chromosomes. The time-to-live parameter is set at 5 generations. The chromosomes are mutated using the proposed adaptive mutation rate, evaluated using MSE/ITP and finally pruned to yield an optimal structure. The convergence of the algorithm is shown in Figure 5.9.

The entire convergence plot has not been shown as the initial generations have a large level of relative fitness compared to the later generations. Note that the y-axis is only a relative fitness value and is not representative of the MSE. Figure 5.10 shows the occurrence of each term through the complete generation cycle.

Each bar in Figure 5.10 represents one of the 101 terms produced by the NARMAX model. The selected terms are those with the highest number of occurrences. In this case a selection of top 14 terms are selected and are shown on the Table 5.1.

From the model that can be represented using these values, it is found that the structure of the model resembles the structure of (5.29) and the parameter estimates match closely. To further validate the model, the Model Predicted Output (MPO) over the test set is computed and

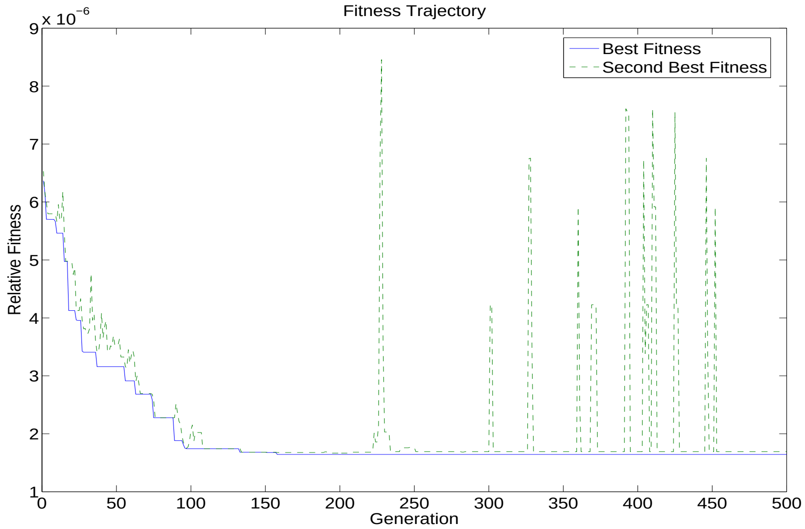


Figure 5.9: *Convergence Curves for Duffing's Equation*

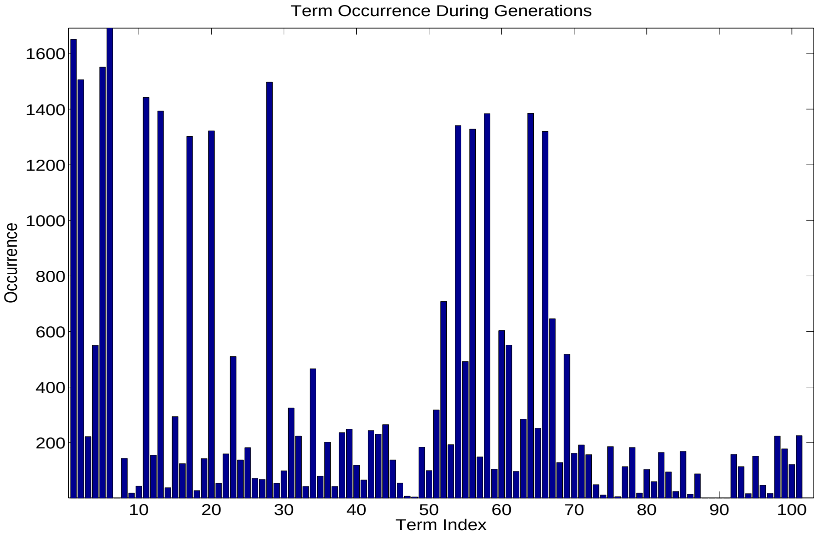


Figure 5.10: *Term Occurrence for Duffing's Equation*

Terms	**Value**
$u(k-1)$	$3.99E-06$
$y(k-1)$	0.68184
$y(k-5)$	-0.36941
$y(k-2)$	0.41093
$y(k-4)y(k-3)y(k-2)$	0.57602
$y(k-4)y(k-1)^2$	-2.9928
$y(k-2)^2y(k-1)$	0.81976
$u(k-2)y(k-4)y(k-3)$	-0.35157
$u(k-2)y(k-4)y(k-2)$	-0.66708
$u(k-2)y(k-2)^2$	1.4996
$u(k-2)y(k-3)y(k-2)$	-0.44369
$y(k-4)^2y(k-1)$	-0.7774
$u(k-2)y(k-5)y(k-3)$	0.70445
$y(k-3)^2y(k-1)$	0.12268

Table 5.1: *Algorithm Result for Duffing's Equation*

compared with the original data and is shown in Figure 5.11. The model is further validated by using correlation based model validity tests. These plots are shown in Figure 5.12.

Note that if the model has truly captured the system dynamics, then the Frequency Response Functions of the model must match those of the original system [130]. The model validation is therefore further carried out in the frequency domain. The first and third order Frequency Response Functions (FRF) are computed from the model parameters obtained in Table 5.1 using the harmonic probing algorithm [120] and are compared with the original FRFs and is shown in Figure 5.13.

From the plots of MPO, correlation and FRFs it is obvious that the proposed algorithm picked the correct structure and preserve the original dynamics.

5.7 Conclusions

An alternative method based on evolutionary programming is proposed to select the correct structure of a polynomial NARMAX model. The mutation probability is made adaptive based on relative fitness and are restricted to lie below 50% to achieve better convergence. By augmenting an internal penalty function to the MSE, it is possible to reject the spurious terms which are likely to be selected in a noisy environment. By assigning a time-to-live parameter, certain insignificant terms are discarded. The performance of the algorithm is demonstrated considering three examples of nonlinear systems where one of the example corresponds to nonlinear wave force model.

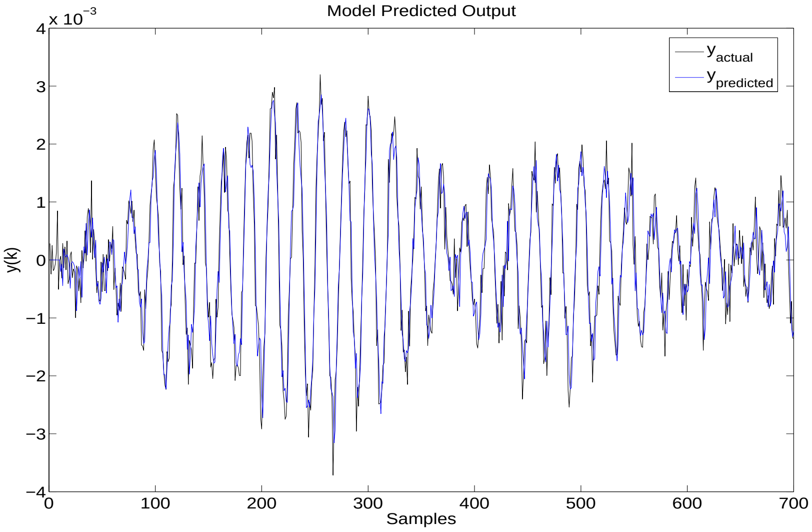


Figure 5.11: *Model Predicted Output for Duffing's Equation*

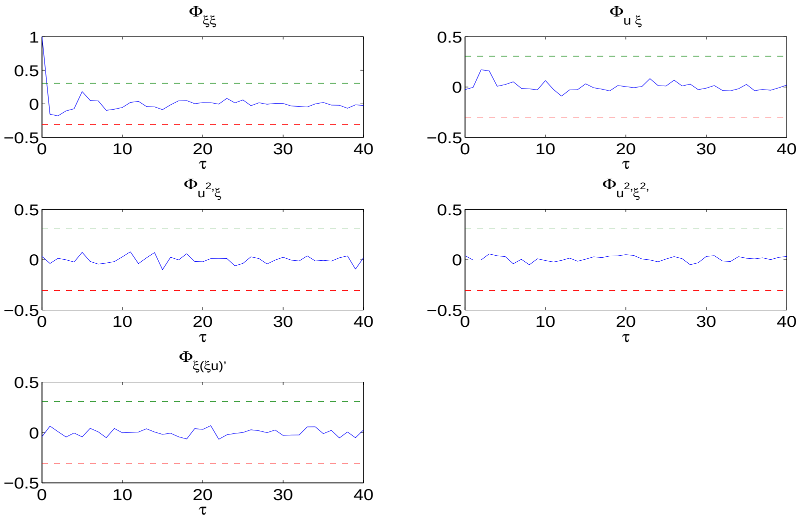


Figure 5.12: *Model Validation for Duffing's Equation*

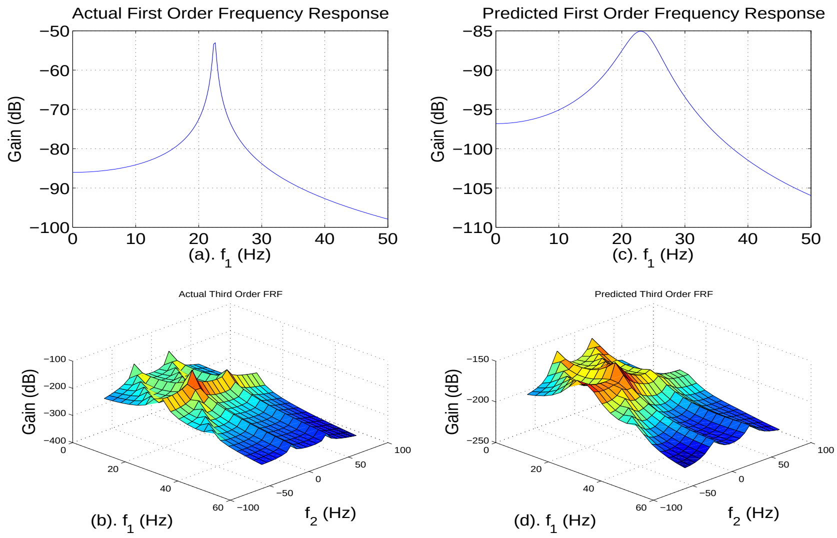


Figure 5.13: *Linear and Third Order Frequency Response Functions for Duffing's Equation*

The models are validated using Model Predicted Output (MPO) criteria in time domain and generalised Frequency Response Functions (FRF) in frequency domain. The observations of the present study demonstrate that Evolutionary Programming (EP) can provide a viable alternative to solve the model structure selection problem of nonlinear systems.

CHAPTER 6

Generalised Predictive Control for Nonlinear Systems using Evolutionary Computation and Bit-Streams

6.1 Introduction

Predictive control has attracted considerable attention from researchers across various engineering fields during the past few decades and is a widely used method across industry and academia [131] [27]. The term predictive control is used to define an ample range of control techniques which include Model Predictive Heuristic Control(MPHC), Model Algorithmic Control (MAC), Dynamic Matrix Control (DMC), Internal Model Control (IMC), Generalised Predictive Control, *et cetera*. [132, 133, 134, 25]. The key features of these apparently different types of predictive controllers are almost similar and they all use a mathematical model to predict the output over a prediction horizon and compute an optimal sequence of controls from all feasible sequences by minimising an objective function which is subjected to constraints. The feedback control law is then obtained by using some sort of receding strategy where only the first element of the computed sequence of optimal controls is applied to the system.

Amongst different types of predictive controllers, the Generalised Predictive Control (GPC) proposed by Clarke and co-workers [25] has been very successful in practical applications across a wide range of engineering fields [28, 135] and is the focus of the present study. The initial predictive controllers including GPC utilised only linear models and has been very successful when the plant is operating in the neighborhood of operating point. The performance of GPC for linear systems has been studied in Chapter 2 and Chapter 3 for a stable linear systems with and without delay, and an unstable system. However most industrial processes have inherent complex nonlinearities, and these can render the classical GPC algorithm impractical. Due to this, many researcher have extended the benefits of GPC into the nonlinear realm which can be found in [29, 27, 28, 30, 31, 136, 32] and the references therein.

Another field of interest that the author wishes to explore is bit-stream encoding. A complete overview of the bit-stream paradigm can be found in Chapter 2. In the present study, the results for a newly developed nonlinear controller will be merged with current bit-stream encoder theory for the implementation of the control law. This will prove that even in extremely sensitive conditions, such as the one presented for the nonlinear control law, a bit-stream paradigm can provide satisfactory results.

The first step of designing GPC for nonlinear systems is to develop a suitable mathematical model for the system which will be used for predicting the output over a prediction horizon. When the system is linear such predictions are obtained by solving a set of Diophantine equations. This is not possible for a nonlinear system. The present study therefore uses an alternate approach based on the concept of extended model proposed in [137] where the predictions are obtained from a selected linear model, called the reference model. The reference model, being linear, can not capture the nonlinear dynamics and will therefore give erroneous predictions. The errors in the predictions are corrected by estimating the unmodelled dynamics. In the present study the unmodelled dynamics are estimated by using a polynomial nonlinear autoregressive moving average with exogenous input (NARMAX) [39, 38] model. The NARMAX model is essentially a nonlinear expansion of linear ARMAX model and can represent a wide class of nonlinear systems. The structure and parameters of NARMAX model is determined using evolutionary computation [138]. The parameters of GPC controller such as prediction and control horizon is optimised by evolutionary programming. Once the entire control law has been obtained, it is expanded and simplified to obtain a digitally implementable expression. This is then expressed in terms of bit-streams operands and attached to the nonlinear plant.

This chapter is structured as follows. §6.2 describes briefly the algorithm of classical Generalised Predictive Control. §6.3 describes the detail procedure of designing GPC for nonlinear systems including the methods of tuning some of the controller parameters using evolutionary programming. Simulations results for two nonlinear systems are given in §6.4 and §6.4.2 with conclusions in §6.5.

6.2 The GPC Algorithm - Classical Approach

There is a need to design GPC which can effectively control nonlinear systems. But the first step of designing GPC for nonlinear systems is to find a model which can accurately represent the nonlinear dynamics.

Several possibilities exist to represent nonlinear systems. In the present study the design of GPC will be based on a polynomial nonlinear autoregressive moving average with exogenous input

(NARMAX) representation of the system. The output of such model is expressed as

$$y(t) = F^{N_l}[y(t-1)\dots y(t-n_y) \quad u(t)\dots u(t-n_u)\epsilon(t-1)\dots\epsilon(t-n_\epsilon)] + \epsilon(t) \tag{6.1}$$

where $u(t)$ and $y(t)$ represent the measured input and output respectively, and $\epsilon(t)$ is the prediction error. The terms n_y, n_u and n_ϵ represent the maximum lags in the input, output and prediction error respectively, and $F^{N_l}[\cdot]$ is some nonlinear function with degree of nonlinearity N_l.

Note that the model of (6.1) is linear-in-the-parameters. However, the number of possible terms increases rapidly with increase in the degree of nonlinearity and maximum lag of inputs, outputs and noise terms. For example, in the case of a third order output ($n_y = 3$), second order input ($n_u = 2$) with a fourth order noise model ($n_\epsilon = 4$) expanded as a third degree polynomial ($l = 3$) would contain 220 unique terms. Models with excessive number of terms may be extremely effective in fitting the estimation data, but may not catch the true dynamics of the underlying system and may exhibit unwanted nonlinear behavior. The structure selection problem or which terms to include into the model is therefore crucial and has been solved using evolutionary computation as described in Chapter 5.

6.3 Nonlinear Predictive Controller Using NARMAX Model

In this section, a new type of nonlinear GPC controller is proposed. The section begins by establishing the reasoning behind the use of the polynomial NARMAX model. The procedure for the development of the nonlinear controller are then described.

6.3.1 Controller Formulation

In a conventional generalised predictive controller, the main idea is to evaluate future system response whilst adjusting the input and minimising the cost function. The future output predictions are generally obtained by iteratively evaluating a set of Diophantine equations. However, when the system is nonlinear i.e. when it is represented by a polynomial NARMAX model, it is not possible to solve the Diophantine equations and get the control law. In the present study, an alternate approach has been adopted to control a nonlinear system using classical GPC by comparing the output of a *linear* and *controllable* reference system against the output of the nonlinear system. By studying the relationship between both of these system outputs, the dynamics of the linear plant can be mapped against those of the nonlinear plant. This model will enable the predictions of the classical GPC approach to be resolved via the Diophantine equations albeit with erroneous predictions which will be accounted for using system identification. A similar concept of linearisation can be found in [40] though its control sequence derivation is different from the present one.

Step-1: Selection of a linear Reference Model

Given a system represented by a nonlinear model consisting of both linear and nonlinear terms, obtain a linear model (to be called hereafter as the *reference* model) from the nonlinear model. The reference model can be selected using different procedures. For example, it can be obtained by retaining only the linear terms and discarding the nonlinear terms of the system; or it may be the linearised model of the nonlinear system around the desired operating point etc. The reference model should preferably be controllable.

Let the output of the reference system be denoted as y_l. The reference system can be represented by the general model of (3.1) and therefore the future predictions of the output based on this model can be represented as

$$\hat{y}_l = G\Delta u + f \tag{6.2}$$

Step-2: Prediction using an Extended Linear Model

Note that the linear model can not capture the complete dynamics of the nonlinear system. Following similar procedure of [137], define an extended linear model from (6.2) whose predicted output is given by

$$\begin{aligned} \hat{y}_{el} &= G\Delta u + f + d_{nl} \\ &= \hat{y}_l + d_{nl} \end{aligned} \tag{6.3}$$

where the additional term d_{nl} in (6.3) accounts for the effects of nonlinearities of the system as well as any possible external disturbances acting on the system. The term d_{nl} is time varying and changes at each sampling instant. This is to be estimated such that the output from the extended linear model $\hat{y}_{el}$ equals to the output of the original nonlinear system.

Step-3: Control Law from Extended Linear Model

Recall the GPC performance index presented in §3.18:

$$J = (\mathbf{G\Delta u} + \mathbf{f} - \mathbf{w})^T(\mathbf{G\Delta u} + \mathbf{f} - \mathbf{w}) + \lambda\mathbf{\Delta u}^T\mathbf{\Delta u} \tag{6.4}$$

The performance index using the extended linear model becomes

$$J = (\mathbf{G\Delta u} + \mathbf{f} + d_{nl} - \mathbf{w})^T(\mathbf{G\Delta u} + \mathbf{f} + d_{nl} - \mathbf{w}) + \lambda \mathbf{\Delta u}^T \mathbf{\Delta u} \tag{6.5}$$

The optimal control input which minimises the performance index of (6.5) is given by

$$u(t) = u(t-1) + \left(\mathbf{G}^T\mathbf{G} + \lambda\mathbf{I}\right)^{-1} (\mathbf{w} - \mathbf{f} - \mathbf{d_{nl}}) \tag{6.6}$$

Note that the optimum control law of (6.6) is obtained from the extended linear model and not for the original nonlinear system. However, the objective of the control law is to compute a time varying disturbance vector d_{nl} which takes into account the effects of nonlinearity which has not been accounted for by the linear reference model and gives an optimal input for nonlinear system. This is possible if the output of the extended reference model match the output of the nonlinear system y_{nl} at all future time instants. Thus

$$y_{el}(t+j) = y_{nl}(t+j), \qquad j = N_1, \ldots\ldots, N_2 \tag{6.7}$$

Thus it is important to accurately represent the time varying disturbance term which contains the effects of nonlinearities of the system.

Step-4: Determination of d_{nl} using System Identification Technique

One of the technique which has been used to compute d_{nl} to use a fixed point algorithm such as

$$\hat{d}_{nl}^{n+1} = \hat{d}_{nl}^{n} + \beta \left[y_{nl}^{n} - y_{el}^{n}\right] \tag{6.8}$$

where n is the iteration number and β is a factor which lies between 0 and 1 to control the region of convergence [137].

Since the unmodelled dynamics often contain nonlinearities, the model between d_{nl} and y_l will essentially be a nonlinear model.In the present study a polynomial NARMAX model is fitted between d_{nl} and y_l which takes the following form :

$$\begin{aligned} d_{nl}(k) = F^{N_l}[&d_{nl}(k-1)\ldots d_{nl}(k-N_d), y_l(k-1)\ldots y_l(k-N_{y_l}) \\ &\epsilon(k-1)\ldots\epsilon(k-N_\epsilon)] + \epsilon(k) \end{aligned} \tag{6.9}$$

The structure selection and parameter estimation of this model is carried out by using evolutionary computation approach proposed in Chapter 5 that can successfully obtain a parsimonious system description by removing spurious terms. The algorithm uses an adaptive mutation rate based on the relative fitness and introduces several parameters including an internal penalty function and a time-to-live parameter to reject unwanted terms.

6.3.2 Tracking of Asymptotically Constant References

In the present study, a nonlinear function d_{nl} is used to compensate the difference in dynamics between a linear system and a nonlinear system. However, when there is a need to track a reference signal, the controller needs further modification. Assume that the reference signal w is asymptotically constant i.e. it is constant beyond a prescribed prediction horizon N_r :

$$w(k) = w, \forall\, k \geq t + N_r \tag{6.10}$$

Perfect tracking can be achieved by including integral action in front of the system with gain K_i as has been suggested in [139]. The value of the integral gain K_i has been obtained using evolutionary programming in the present study.

6.3.3 Intelligent Tuning by Evolutionary Programming

Evolutionary Programming (EP) [128] is a useful method of optimisation when other techniques such as gradient or direct analytical methods are not possible. The EP algorithm has proven to be especially useful for difficult combinatorial functions which contain many locally optimal solutions [140]. The proposed study uses EP to optimise the parameters N_2, N_u, λ and integral gain K_i.

The learning starts with an initial population of binary encoded set of solutions (N_2, N_u, λ, K_i) called parents. Each parent is evaluated using a fitness cost function $F(x)$ which describes its suitability as a possible solution. Next, each parent creates one offspring by performing a series of mutations to the parents. The mutation probability is calculated as a function of $F(x)$ and is given by:

$$mutation = 0.5 \times \frac{F(x) - min[F]}{max[F] - min[F]}\% \tag{6.11}$$

where $F(x)$ is the fitness of parent x, $min[F]$ and $max[F]$ are the minimum and maximum cost of the entire parent population. Equation (6.11) provides a pseudo-elitist strategy where the single best parent is retained and passed on to the next generation. After the successful mutation of each parent, all the new offsprings are evaluated using $F(x)$. The offsprings now become the parents of the next generation. This process is iterated for a set number of generations, or until a terminal condition is satisfied [128].

6.4 Simulation Results

The performance of the proposed GPC is illustrated by considering two examples of nonlinear systems. The first example is a Duffing's oscillator and the second example is a nonlinear resonator.

6.4.1 Example 1: Nonlinear GPC for Duffing's Oscillator

The duffing oscillator is a well-studied nonlinear oscillator which contains a cubic stiffness term to describe the hardening spring effect observed by many mechanical problems [141, 142]. This system exhibits a wide range of dynamical behaviors, including chaotic motion [143] and is considered as common benchmark example to study nonlinear systems.

The dynamics of Duffing's oscillator is described by

$$\frac{d^2y}{dt^2} + 2\xi w_n \frac{dy}{dt} + w_n^2 y(t) + w_n^2 \epsilon y(t)^3 = u(t) \tag{6.12}$$

The discrete equivalent of this system at a sampling rate of 500Hz with $\xi = 0.01, w_n = 45\pi, \epsilon = 3$ is given by

$$y(k) = 1.9144y(k-1) - 0.99435y(k-2) + 4 \times 10^{-6} y(k-1)^3 + 0.079944u(k-1) \tag{6.13}$$

Note that the dynamics of (6.13) matches with that of (6.12). This has been verified by comparing the frequency response functions of both these systems as shown in Chapter 5.

The first step in the design of the proposed GPC is to find a suitable linear model, called the reference model. Since the system of (6.13) contains both linear and nonlinear terms, the reference model is selected by retaining the linear terms of (6.13) to give:

$$y_l(t) = 1.9144y_l(t-1) - 0.99435y_l(t-2) + 0.05u(t-1) \tag{6.14}$$

Both the original nonlinear system and the reference linear system are excited by a zero mean white Gaussian noise with standard deviation of 3 and 1000 samples of input/output are collected. The difference between the output of the linear and the nonlinear model d_{nl} are obtained. Since this contains unmodelled nonlinear dynamics, which needs to be compensated, a polynomial NARMAX model is fitted between d_{nl} and y_l using evolutionary computation following the procedure in [138]. The model is fitted from 500 samples of data and is validated using the rest of the data sets. From a specified model set which consists of 84 terms, a parsimonious model is obtained using EC which is given by:

$$\begin{aligned} d_{nl}(k) = {} & 0.9281d_{nl}(k-1) - 0.55118y_l(k-1) - 0.38738y_l(k-3) \\ & + 0.016764y_l(k-5)y_l(k-1) - 0.0063029y_l(k-5)y_l(k-3) \end{aligned} \tag{6.15}$$

The model predicted output of the model fitted to d_{nl} is shown in the Figure 6.1. From the figure it is observed that the fitted NARMAX model could accurately represent the unmodelled dynamics.

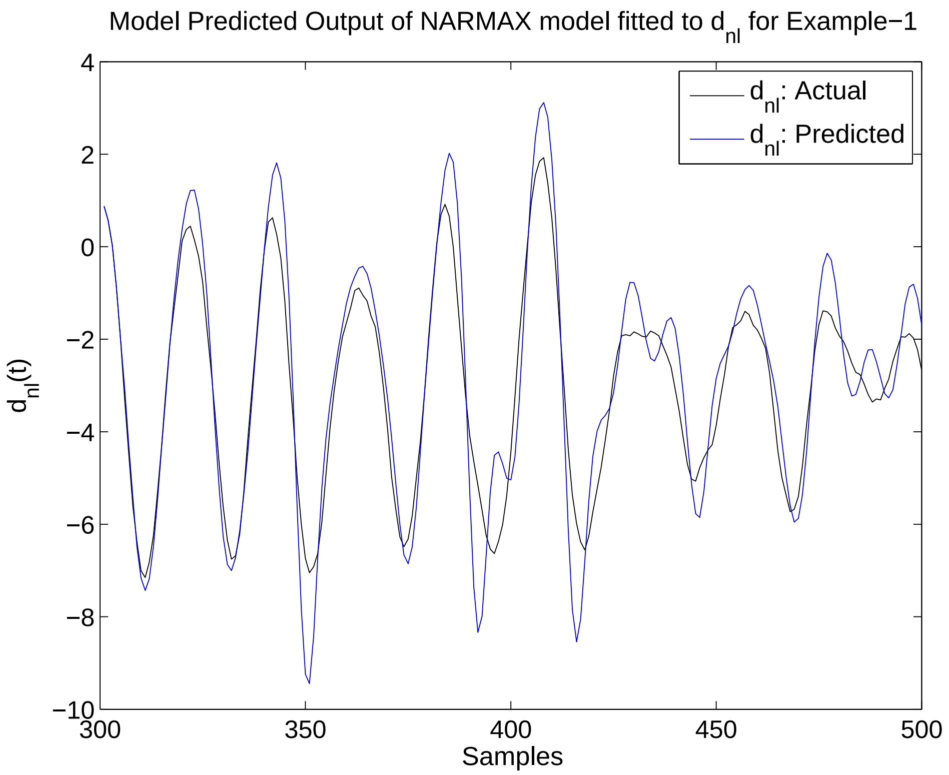


Figure 6.1: *Model predicted output of the NARMAX model fitted to the unmodelled dynamics d_{nl} of Example-1*

Nonlinear GPC Parameters	Optimised Value
N_2	4
N_u	4
λ	0.048828
K_i	3.0762

Table 6.1: *Optimised nonlinear GPC parameters obtained via evolutionary programming for Example-1*

Note that, this is required for the computation of the control law of nonlinear GPC using (6.6).

Initially, the nonlinear GPC is implemented in Matlab$^{\text{TM}}$. Some of the design parameters of the GPC such as prediction and control horizon as well as the integral gain are obtained using evolutionary programming and are summarised in Table 6.1. The tracking performance of the controller is shown in Figure 6.2 From the result it is observed that output tracks satisfactorily the square wave signal. Note that the control of this particular system has known to be very complex due to tits chaotic behaviour. There is a distinct noisy sequence present during the first few sample in Figure 6.2. This may have been caused by initial values of $y(t-1)$, $y(t-2)$, $u(t-1)$, $u(t-2)$,...*et cetera* which are set to zero during the initialisation phase. As time progresses, the values change from their initial assumed values and the output track the reference. The control signal also become much smoother.

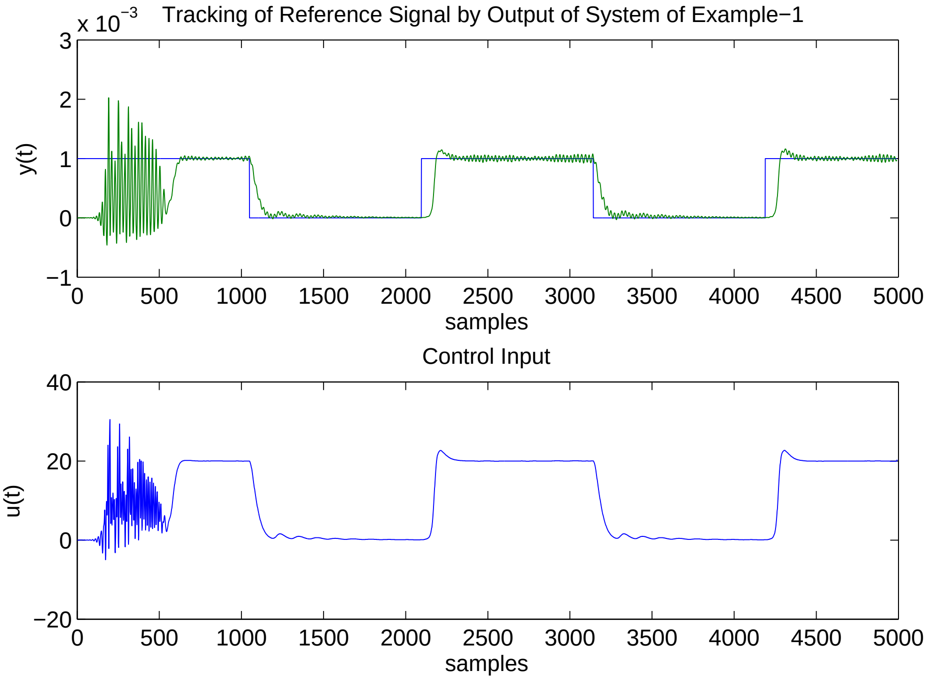


Figure 6.2: *Tracking performance for Example-1 using multi-bit: Upper graph represents the output y(t) and lower graph represents the control input.*

After implementing the nonlinear GPC in Matlab^{TM}, the next phase is to investigate if this complex nonlinear control law could be implemented in bit-stream using Mentor Graphics' Modelsim^{TM}. The control law of nonlinear GPC for this system which is implemented in bit-stream is given by

$$\begin{aligned} u(t+1) = u(t) + \{&0.035026\,[w(t) - f(1) - d_{nl}(1)] \\ &+ 0.413680\,[w(t) - f(2) - d_{nl}(2)] \\ &+ 0.220770\,[w(t) - f(3) - d_{nl}(3)] \\ &+ 0.144470\,[w(t) - f(4) - d_{nl}(4)]\} \end{aligned} \tag{6.16}$$

The result of this implementation is shown in Figure 6.3. It is observed that the tracking performance is poor when there is a step change in the reference signal. However, the controller tracks the reference satisfactorily as time progresses. This is due to the inherent slew rate of the bit-streams. Further, there exists quantisation noise during the conversion of analogue to bit-stream and bit-stream to analogue. In the present simulation, a 12-bit resolution is being used which means that the maximum resolution available is 0.000488281. Note that, the maximum resolution plays a critical part on the success of bit-stream implementation.

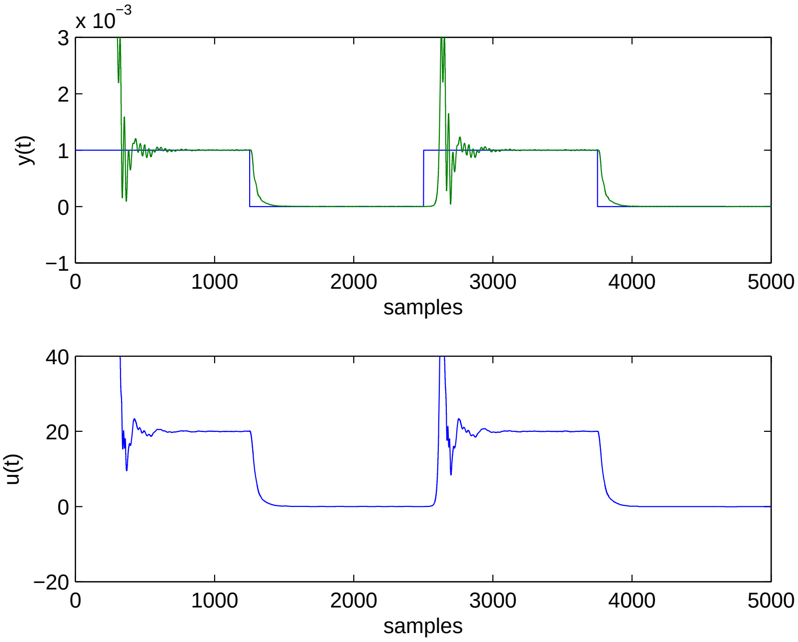


Figure 6.3: *Tracking performance for Example-1 using bit-stream: Upper graph represents the output y(t) using bit-stream control and the lower graph represents the real value interpretation of the bit-stream control input.*

6.4.2 Example 2: Nonlinear GPC for Nonlinear Resonator

To further establish and prove the efficacy of the nonlinear controller described in §6.3; another nonlinear system is considered. A similar procedure to that presented in Example 1 is followed for designing and implementing the GPC for this system.

The dynamics of the nonlinear resonator in discrete time is given by

$$y(t) = 1 + y(t-1)u(t-1) - u(t-1)^2 \tag{6.17}$$

The system of (6.17) contains only nonlinear terms and therefore the linear reference model should be selected using some rough guidelines. For example, the reference linear model should be controllable and its output should lie within the range of the output of the original nonlinear system. The linear reference model considered in the present example is given by

$$y_l(t) = 0.8y_l(t-1) + u(t-1) \tag{6.18}$$

Note that possibilities of selecting other linear models also exist. Simulations with various examples (not described) show that the choice of the reference model does not significantly affect the overall performance of the controller provided its output lies within the range of nonlinear system output.

Nonlinear GPC Parameter	Optimised Values
N_2	4
N_u	4
λ	1.7285
K_i	0.69336

Table 6.2: *Optimised nonlinear GPC parameters obtained via evolutionary programming for Example-2*

Following similar procedure as of Example 1, polynomial NARMAX model is fitted between the unmodelled dynamics d_{nl} and y_l using evolutionary computation. The EC based identification algorithm searched for a second order nonlinear system with a maximum output and input lag terms of 2 and 5 respectively, thus providing 36 possible terms from which 5 terms are selected in 50 generations. The model is given by

$$\begin{aligned} d_{nl}(t) &= 0.10896 d_{nl}(t-1) - 0.096416 y_l(t-1) - 0.19874 y_l(t-3) y_l(t-1) \\ &\quad - 0.015354 y_l(t-5)^2 + 0.11004 y_l(t-4) y_l(t-3) \end{aligned} \tag{6.19}$$

Figure 6.4 shows the Model Predicted Output (MPO) of the model of (6.19). The parameters of the GPC are obtained with an objective to achieve a desired settling time using EP and are summarised in Table 6.2. The controller is initially implemented in Matlab^{TM}which uses infinite resolution. The tracking performance of the controller is shown in Figure 6.5. From the result it is

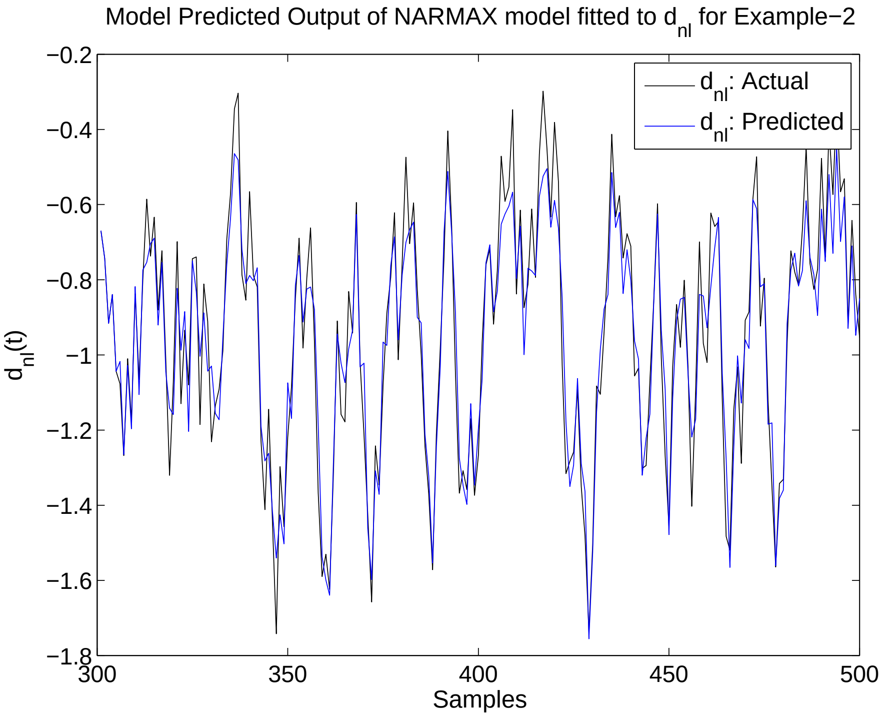


Figure 6.4: *Model predicted output of the NARMAX model fitted to the unmodelled dynamics d_{nl} of Example-2*

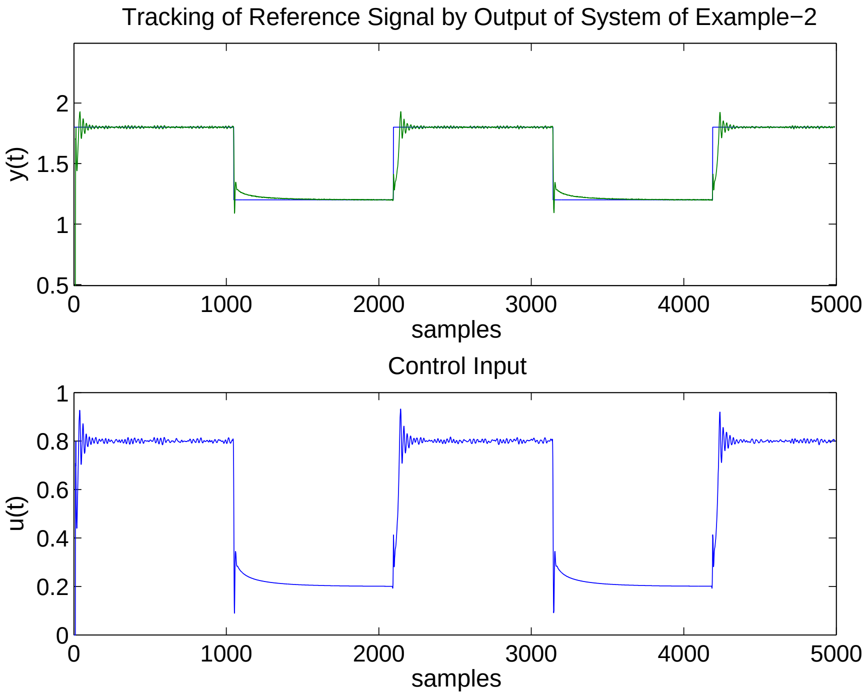


Figure 6.5: *Tracking performance for Example-2 using multi-bit: Upper graph represents the output y(t) and lower graph represents the control input.*

observed that the proposed controller could satisfactorily track the reference signal.

During the last phase, the nonlinear GPC is implemented in bit-stream using ModelSim. The control law for the nonlinear resonator which is implemented in bit-stream is given as:

$$\begin{aligned} u(t+1) &= u(t) + 0.0840Q(1) + 0.0901Q(2) + 0.0664Q(3) + 0.0387Q(4) + 0.0167Q(5) \\ &\quad + 0.0017Q(6) - 0.0087Q(7) \qquad (6.20) \\ u(t+2) &= u(t+1) - 0.0611Q(1) + 0.0186Q(2) + 0.0420Q(3) + 0.0383Q(4) + 0.0256Q(5) \\ &\quad + 0.0256Q(5) + 0.0125Q(6) + 0.0017Q(7) \qquad (6.21) \end{aligned}$$

where $Q(x)$ for $x = 0 \ldots 7$ is temporary vector containing the relationship between w, f and d_{nl}

and it is given by

$$
\begin{aligned}
Q(1) &= w(1) - f(1) - d_{nl}(1)\\
Q(2) &= w(2) - f(2) - d_{nl}(2)\\
Q(3) &= w(3) - f(3) - d_{nl}(3)\\
Q(4) &= w(4) - f(4) - d_{nl}(4)\\
Q(5) &= w(5) - f(5) - d_{nl}(5)\\
Q(6) &= w(6) - f(6) - d_{nl}(6)\\
Q(7) &= w(7) - f(7) - d_{nl}(7)
\end{aligned}
\tag{6.22}
$$

This vector is passed through a bit-stream to analogue converter and sent to a bit-stream operator that evaluates control signals (6.20) and (6.21). The values $u(t+1)$ and $u(t+2)$ is passed through a bit-stream to analogue converter and is the input to plan (6.17). The conversion is done via a 12-bit analogue to bit-stream converter. The bit-streams $ggclock$ signal runs at 1MHz. This ensures that bit-stream calculations are performed before the edge of every plant clock sample. That is, the time it takes for the bit-streams to reach the final value (i.e. slew rate) is significantly less than the time before the next value is required for the calculation of the response of the system. If this condition is not met, then the value fed to the system may not be the correct result of the control law calculation, causing a series of error that will cause the system to be unstable. The results of this simulation are shown in Figure 6.6.

The results obtained through Matlab$^{\text{TM}}$simulations are very similar to those obtained using ModelSim $^{\text{TM}}$. The first thing to observe is the range of the scales in Figure 6.5 and Figure 6.6. Ringing can be seen to be present in both simulations. However this appears to be more pronounced when using bit-streams. This can be attributed to the accuracy of conversion between bit-streams to real-valued signals and vice-versa. The conversion from analogue to bit-streams is carried at 12-bit resolution which gives a maximum resolution of 0.000488281. It is also important to acknowledge that under a Matlab$^{\text{TM}}$simulation, it is possible to run the simulation using a much higher resolution, i.e. using a very small step size. This is problematic when performing a bit-stream simulation since bit-stream require a lot of computational power to simulate. Overall, the implementation of the proposed nonlinear controller has shown successful tracking of a nonlinear plant. Moreover, the augmentation of this controller with bit-stream encoding for the control law has clearly shown that bit-streams can be used to implement the nonlinear control law.

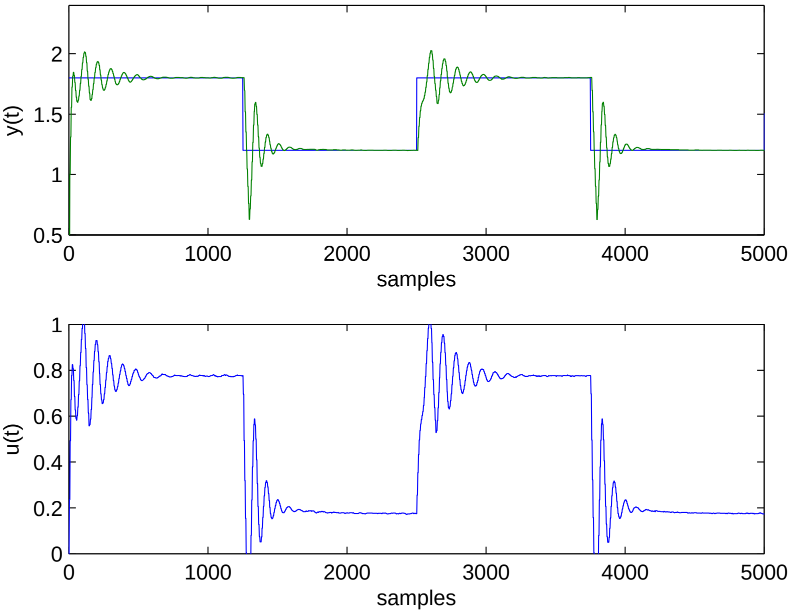


Figure 6.6: *Tracking performance for Example-2 using bit-streams: Upper graph represents the output y(t) using bit-stream control and the lower graph represents the real value interpretation of the bit-stream control input*

6.5 Conclusions

An alternative method of designing generalised predictive controller (GPC) for nonlinear system has been proposed. The design is based on extending the concept of the linear GPC to nonlinear system. Initially a linear reference model is selected to predict the output of the nonlinear system which results in erroneous predictions. The errors in the prediction are estimated using a polynomial NARMAX model via evolutionary computation. The linear GPC law is then modified by including the estimated prediction error. The parameters of GPC such as prediction and control horizon and the integral gain are optimised using evolutionary programming. The capabilities of the controller are tested using two systems, namely Duffing's oscillator and a nonlinear resonator. The performance of the nonlinear GPC controller has been verified by successful tracking of a reference signal.

Upon the successful implementation of the nonlinear GPC, a direct equivalent for this nonlinear controller is made using ModelSim^{TM}. In this case, bit-stream signals are digitally implemented instead of a multi-bit implementation using Matlab^{TM}. The bit-stream controller also shows satisfactory tracking performance for both systems. The importance of this task lies in complexity and sensitivity of the examined systems. Many practical adjustments had to be made to use bit-streams for the nonlinear control law and these are all carefully outlined. Since bit-streams are inherently digital in nature, they are best suited for hardware implementation using FPGA. With suitable hardware the performance of bit-streams should be on par with that of multi-bit controllers, as well as proving many of the advantages presented in Chapter 2. Overall, the success of the controller presented in this chapter has been successfully validated through the control of two very complex systems. Furthermore, the use of bit-streams for nonlinear control has also proved to be successful and it therefore expands the potential of bit-stream to be used in controller implementation.

Chapter 7

Conclusion

With an objective to find the feasibility of an alternate method of digital controller implementation than the traditional multi-bit implementation, the research carried out in this thesis has gone a long way to achieve this. Most of the research carried out in this study has combined two distinct fields of research namely digital systems and control systems. This dissertation studies and investigates the possibility of implementing controllers using 1-bit signal processing; popularly known as bit-stream signal processing. Design of controllers using bit-stream paradigm, where the analogue or multi-bit signal are converted to a simple 1-bit stream, offers a relatively new and highly promising approach for the controller implementation for many types of real world systems. However, the application of bit-streams in implementing complex controllers is relatively new and this research therefore focused on design and implementation of Generalised Predictive Controllers (GPC) for both linear (stable and unstable) and nonlinear systems in bit-stream.

The first step in the pursuit of bit-stream implementation is to study about various basic functional blocks which are needed to implement a complete bit-stream based system. Following an introductory review in Chapter 1, Chapter 2 presents the bit-stream methodology by establishing the background, advantages and the tools used when developing a bit-stream system. Specifically, the relevance, concept and representation of an analogue signal using quanta is presented here. The functions of various tools such as bit-stream adder, subtractor, multiplier and delay element are clearly outlined. The validity of these components are demonstrated by simulating both linear and nonlinear systems which are represented respectively by CARIMA and NARMAX models. The results of simulations in a bit-stream environment are compared to those obtained from multi-bit environment and are found to be satisfactory.

Encouraged by the success of bit-stream in simulating both linear and nonlinear systems, the next phase of the research involves designing generalised predictive controller for both linear systems and linear systems with delay. Essentially, the GPC is a predictive controller which generates a sequence of future control signals in each sampling interval to optimise the control effort of the controlled signals and offers numerous advantages, and, has been very popular in industrial process control systems in recent years. Two case studies i.e. control of a D.C. servo mechanism and control of a thermal system has been taken up for investigation. Since the design of GPC in this study is based on CARIMA (Controlled Auto Regressive Moving Average) model of the system, discrete parametric models are fitted to both these systems using standard procedure of system identification. In this study, orthogonal least squares with error reduction ratio test (OLS with ERR) algorithm has been used to identify the models of the systems. These models are validated using several model validation criteria which include one step ahead prediction, model predicted output and correlation tests. After representing these systems by CARIMA model, GPC is designed for these systems. Since the performance of GPC is dependent on several tuning parameters such as control and prediction horizon as well as the control weighting, these are optimised using Genetic Algorithm (GA). The GPC is implemented in bit-stream and its performance is compared with the multi-bit implementation. The success of bit-stream implementation depends critically on the choice of several parameters such as bit-stream frequency f_b, input/output signal voltage levels, bit-stream slew rate *et cetera*. The impact of these parameters on the performance of bit-stream GPC is analysed.

Having successfully implementing the GPC in bit-stream for stable linear systems, the next phase of the research focus on investigating if it can be applied for unstable systems. A typical magnetic levitation system, which is both nonlinear and unstable, are considered as a benchmark to study the effectiveness of bit-stream based controller in stabilising the unstable system. Due to enormous complexity of this system, several options of designing controllers are adopted in a progressive manner. Initially, an exact feedback linearising controller is designed for the magnetic levitation system and it is implemented in bit-stream simulation environment using Mentor Graphics' Modelsim$^{\text{TM}}$. Although the performance of this controller in bit-stream is satisfactory in simulations, the feedback linearising control law is more complex and is difficult to be implemented in practice. Therefore, in the next stage, a state space based model predictive controller (MPC) is designed using a linear model of the system. The linear model of the system is obtained by linearising the system around an operating point using standard methods of Taylor series approximation. The MPC is implemented using Modelsim$^{\text{TM}}$and its performance is found to be comparable with that of multi-bit implementation. However, the MPC control law requires all the state variables to be measured directly or to be estimated using an observer. Since, the introduction of an observer will make the overall controller implementation complex, a generalised predictive controller, similar to that used in Chapter 3, are designed using the linearised model of the system. Further, a lead compensator is designed for this system. The GPC and the lead controller are implemented in bit-stream in real time using an experimental prototype and the results of the experiment are compared to those obtained from simulations. It is observed that the performance of bit-stream based controller is comparable with multi-bit implementation and hence this can be considered as an alternative method of controller implementation due to various advantages it offers.

After successfully implementing controllers in bit-stream for both stable and unstable linear systems, the last phase of the research investigates if bit-stream paradigm can be adopted for implementing controllers for nonlinear systems. Since the focus of the present research has been to design and implement GPC, a novel method of designing GPC for nonlinear systems, which are represented by polynomial NARMAX model is developed. The first step in designing the nonlinear GPC is to represent the nonlinear system by a polymomial NARMAX model. The primary advantage of a polynomial representation is that it is linear-in-the-parameters and thus can be estimated using linear least squares methods. However, the number of possible terms increases exponentially with the increase in the degree of nonlinearity and maximum lags of inputs and outputs. An efficient structure selection algorithm, based on evolutionary programming, is proposed to obtain a parsimonious model of the system. The proposed algorithm eliminates spurious terms of the model by using an internal penalty function and implements an adaptive pruning strategy by assigning a time-to-live parameter to remove insignificant terms from the model. By introducing an adaptive mutation rate, the convergence of the algorithm becomes faster. The proposed method is found to be effective in identifying parsimonious models for several nonlinear systems.

After identifying the nonlinear system by polynomial NARMAX model, generalised predictive controller is designed for nonlinear systems. This design is essentially based on representing the system by a linear model, which is controllable, and a nonlinear disturbance model which accounts for the nonlinear effects of the system. This model is identified using the approach developed in Chapter 5. The optimum values of the GPC are computed using evolutionary programming. The nonlinear GPC is implemented in bit-streams in a simulated environment using ModelSim$^{\text{TM}}$and their performance is compared with multi-bit implementation and found to be satisfactory.

The main contributions of the thesis can be summarised as:

- A generalised predictive controller has been designed and implemented in bit-stream for both linear systems (with and without) delay. The effectiveness and feasibility of bit-stream paradigm has been demonstrated in controller implementation.

- Different types of controllers, such as feedback linearising controller, state space based model predictive controller and generalised predictive controller have been designed for unstable magnetic levitation system and implemented in bit-stream.

- New method of structure selection algorithm for nonlinear system identification has been developed based on evolutionary programming and a novel method of designing GPC for polynomial NARMAX systems have been proposed and implemented in bit-stream.

7.1 Future Work

Although, there has been some success in implementing controllers in bit-stream environment, much remains to be investigated. The results of this research can be considered as preliminary but nevertheless a relevant starting point which will prompt further interest and investigation of controller implementation from the perspective of bit-stream.

Some of the immediate future work are:

- Develop a general mathematical frame work of bit-stream control.

- Since bit-stream conversion introduces errors into the system, robust controllers can be developed to reduce the effects of such errors.

- Design and implement adaptive and self-tuning controllers in bit-stream.

References

[1] T. Lee, “Tales of the continuum: a subsampled history of analog circuits,” *Solid-State Circuits Newsletter, IEEE*, vol. 12, no. 4, pp. 38–51, fall 2007.

[2] V. Bush and H. L. Hazen, “The differential analyzer: a new machine for colving differential equations,” *Journal of the Franklin Institute*, vol. 212, no. 4, pp. 447–48, Oct. 1931.

[3] E. Dudek, M. Mosiadz, and M. Orzepowski, “Uncertainties of resistors temperature coefficients,” *Measurement Science Review*, vol. 7, no. 3, pp. 23–26, 2007.

[4] S. Renukunta and D. Wells, “Optical memory and blue lasers,” *Potentials, IEEE*, vol. 13, no. 4, pp. 14 –18, oct.-nov. 1994.

[5] M. Fagen, *A History of Engineering and Science in the Bell System*, B. T. Laboratories, Ed. Murray Hill, N.J., 1978.

[6] D. Linkens, *CAD for Control Systems*, M. Dekker, Ed. Taylor & Francis, Inc., 1993.

[7] A. Cavallo, R. Setola, and F. Vasca, *Using MATLAB, SIMULINK and Control System Toolbox*. Prentice Hall, 1996.

[8] R. Dorf, *Electrical Engineering Hanbook*. CRC Press, 1998.

[9] D. Sbarbaro-Hofer, “Control of a steel rolling mill,” *IEEE Control Systems*, pp. 69–75, June 1993.

[10] R. C. Dorf and R. H. Bishop, *Modern Control Systems*, M. J. Horton, Ed. Prentice Hall, 2008.

[11] X. Wu, V. A. Chouliaras, J. L. Nunez-Yanes, and R. M. Goodall, “A novel delta-sigma control system processor and its VLSI implementation,” *IEEE Transactions on very large scale integration systems*, vol. 16, no. 3, pp. 217–288, 2008.

[12] P. M. Paziz, H. V. Sorensen, and J. Van der Spiegel, “An overview of sigma-delta converters,” *IEEE Signal Processing Magazine*, vol. 13, pp. 61–84, January 1996.

[13] W. Forsythe and M. Goodall, *Digital Control: Fundamentals, Theory and Practice*. McGraw-Hill, 1991.

[14] P. Paraskevopoulos, *Digital Control Systems*. Prentice Hall, 1996.

[15] R. Middleton and G. Goodwin, *Digital Control and Estimation - A Unified Approach*. Prentice-Hall, 1990.

[16] R. Goodall, “Perspectives on processing for real-time control,” *Annual Reviews in Control*, vol. 25, pp. 123 – 131, 2001.

[17] N. Patel, “Bit-streams: Applications in Control,” Ph.D. dissertation, University of Auckland - School of Engineering, May 2006.

[18] J. Candy and G. Temes, *Oversampling methods for A/D and D/A conversion*. Wiley-IEEE Press, 1992.

[19] C. B. Wang, “A 20-bit 25-khz delta-sigma A/D converter utilizing a frequency-shaped chopper stabilization scheme,” *IEEE Journal of Solid-State Circuits*, vol. 36, no. 3, pp. 566–569, March 2001.

[20] M. Kurosawa, M. Kawakami, K. Tojo, T. Katagiri, and T. Higuchi, “Single-bit digital signal processing for current control of brushless dc motor,” in *Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE International Symposium on*, vol. 2, 2002, pp. 589 – 594 vol.2.

[21] N. Patel and U. Madawala, “Brushless DC motor control using bit-streams,” in *10th International Conference on Control,Automation, Robotics and Vision (ICARCV 2008)*, Dec. 2008, pp. 85–90.

[22] X. Shao and D. Sun, “Development of an FPGA-based motion control ASIC for robotic manipulators,” *Proceedings of the 6th World Congress on Intelligent Control and Automation*, vol. 2, pp. 8221–8225, 2006.

[23] S. Chidrawar and B. Patre, “Generalized predictive control and neural generalized predictive control,” *Leonardo Journal of Science*, no. 13, pp. 133–152, December 2008.

[24] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*, 1st ed. Springer Publishing Company, Incorporated, 2009.

[25] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, “Generalized predictive control - part I: The basic algorithm,” *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[26] C. Camasca, A. Swain, and N. Patel, “Generalized predictive control of nonlinear systems using evolutionary computation,” in *TENCON 2009 - 2009 IEEE Region 10 Conference*,

jan. 2009, pp. 1 –6.

[27] P. S. Agachi, Z. K. Nagy, M. V. Cristea, and A. Imre-Lucaci, *Model based control : Case Studies in Process Engineering*, 1st ed. Wieley-VCH, 2006.

[28] E. F. Camacho and C. Bordons, *Model Predictive Control in the Process Industry*, 1st ed. Springer, 2004.

[29] G. P. Liu, V. Kadirkamanathan, and S. A. Billings, "Predictive control for nonlinear systems using neural networks," *International Journal of Control*, vol. 71, no. 6, pp. 1119–1132, 1998.

[30] S. Said and F. M'Sahli, "Output feedback nonlinear GPC using high gain observer," *Computational Intelligence and Intelligent Informatics*, pp. 125–129, 2007.

[31] G. D. Nicolao, L. Magni, and R. Scattolini, "Stabilizing predictive control of nonlinear ARX models," *Automatica*, vol. 33, no. 9, pp. 1691–1697, 1997.

[32] S. J. Qin and T. A. Badgwell, "An overview of nonlinear model predictive control applications," in *Nonlinear predictive control*, F. Allgower and A. Zheng, Eds. Birkhauser, Basel, 2000.

[33] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, 1st ed. Wieley, 1980.

[34] J. Cremers and A. Huber, "Construction of differential equation from experimental data," *Z. Naturforsch*, vol. A42, pp. 797–802, 1987.

[35] J. B. Elsner, "Predicting time series using a neural network as a method of distinguishing chaos from noise," *J. Phys.*, vol. A25, pp. 843–850, 1992.

[36] J. C. Principe, A. Rathie, and J. M. Juo, "Prediction of chaotic time series with neural networks and the issue of dynamic modelling," *Int. J. Bifurcation and Chaos*, vol. 2(4), pp. 989–996, 1992.

[37] L. Stokbro and D. K. Umberger, *Forecasting with weighted maps*, M. Casdagli and S. Eubank, Eds. New York: Addison Wesley, 1992.

[38] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for non-linear systems part II: stochastic non-linear systems," *International Journal of Control*, vol. 41, pp. 329–344, 1985.

[39] ——, “Input-output parametric models for non-linear systems part I: detereministic non-linear systems,” *International Journal of Control*, vol. 41, pp. 303–328, 1985.

[40] L. Bai and D. Coca, “Nonlinear predictive control based on NARMAX models,” *11th International Conderence on Optimization of Electrical and Electronic Equipment*, pp. 3–10, 2008.

[41] M. Korenberg, S. A. Billings, Y. P. Liu, and P. J. McIlroy, “Orthogonal parameter estimation algorithm for non-linear stochastic systems,” *International Journal of Control*, vol. 48, no. 1, pp. 193–210, 1988.

[42] K. Z. Mao and S. A. Billings, “Algorithms for minimal model structure detection in nonlinear system identification,” *International Journal of Control*, vol. 68 No 2, pp. 311–330, 1997.

[43] L. A. Aguirre and S. A. Billings, “Improved structure selection for nonlinear models based on term clustering,” *International Journal of Control*, vol. 62, no. 3, pp. 569–587, 1995.

[44] A. K. Swain and S. A. Billings, “Weighted complex orthogonal estimator for identifying linear and nonlinear continuous time models from generalized frequency response functions,” *Mechanical Systems and Signal Processing*, vol. 12, no. 2, pp. 269–292, 1998.

[45] E. M. A. M. Mendes, “Identification of Nonlinear Discrete Systems with Intelligent Structure Detection,” Ph.D. dissertation, University of Sheffield, 1995.

[46] J.-S. R. Jang, *Neuro-fuzzy and soft computing : a computational approach to learning and machine intelligence*. Upper Saddle River, NJ : Prentice Hall, 1997.

[47] P. J. Fleming and R. C. Purshouse, “Evolutionary algorithms in control systems engineering: a survey,” *Control Engineering Practice*, vol. 10, pp. 1223–1241, 2002.

[48] H. Fujisaka, R. Kurata, M. Sakamoto, and M. Morisue, “Bit-stream signal processing and its application to communication systems,” *Circuits, Devices and Systems, IEE Proceedings*, vol. 149, no. 3, pp. 159 – 166, 2002.

[49] H. Fujisaka, M. Sakamoto, and M. M., “Bit-stream signal processing circuits and their applcation,” *IEICE Trans Fundam*, vol. 85-A, pp. 853–860, 2002.

[50] N. Patel and U. Madawala, “A bit-stream based scalar control of an induction motor,” in *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, nov. 2008, pp. 1071 –1076.

[51] J. Bradshaw, U. Madawala, and N. Patel, “Techniques for conditioning high-speed bit-

stream signals for power electronic applications," in *35th Annual Conference of Industrial Electronics, (IECON '09.)*, Nov. 2009, pp. 142 –147.

[52] K. Tojyo, M. Kurosawa, and T. Higutchi, "Resolution of digital servo control system using single-bit digital signal processing," *Trans IEE Japan*, vol. 118-D, pp. 623–629, 1998, (in Japanese).

[53] X. Wu and R. Goodall, "One-bit processing for digital control," *IEEE Proc.-Control Theory Appl.*, vol. 152, no. 4, pp. 403–410, July 2005.

[54] N. Patel, S. Nguang, and G. Coghill, "Bit-streams: Applications in neural network implementation," in *American Control Conference, 2007. ACC '07*, July 2007, pp. 4780 –4785.

[55] D. Braendler, T. Hendtlass, and O. O'Donoghue, "Deterministic bit-stream digital neurons," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1514–1525, November 2002.

[56] Y. Murahashi, S. Doki, and S. Okuma, "Hardware realization of novel pulsed neural networks based on delta-sigma modulation with gha learning rule," *Asia-Pacific Conference on Circuits and Systems*, vol. 2, pp. 157–162, 2002.

[57] I. Bostan, I. Ionescu, and M. C., "Fpga implementation of a deterministic bit-stream neuron," *International Semiconductor Conference*, vol. 2, pp. xvii+410, Sept-Oct 2003.

[58] Y. Murahashi, H. Hotta, S. Doki, and S. Okuma, "Pulsed neural networks based on delta-sigma modulation suitable for hardware implementation," *IEEE International Joint Conference on Neural Networks*, vol. 4, pp. 2607–2612, July 2004.

[59] N. Patel, S. K. Nguang, and G. Coghill, "Neural network implementation using bit streams," *IEEE Transactions on Neural Networks*, vol. 18, no. 5, pp. 1488 –1504, Sept. 2007.

[60] N. Patel and T. K. Madawala, "A bit-stream based PWM technique for sine-wave generation," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 7, pp. 2530–2539, July 2009.

[61] J. Bradshaw, U. Madawala, and N. Patel, "Bit-stream control of three phase reversible rectifiers," *IEEE Energy Conversion Congress and Exposition*, pp. 2777–2783, September 2009.

[62] Y. Suzuki, H. Fujisaka, T. Kamio, and K. Haeiwa, "Digital filters built of locally connected nanoelectronic devices," *7th IEEE Conference on Nanotechnology*, pp. 873–878, August 2007.

[63] T. Katao, Y. Suzuki, H. Fujisaka, T. Kamio, C.-J. Ahn, and K. Haeiwa, "Single-electron arith-

metic circuits for sigma-delta domain signal processing," *8th IEEE Conference on Nanotechnology*, pp. 729–732, August 2008.

[64] D. Jarman, "A brief introduction to sigma delta conversion," Intersil Corporation, Application Note, May 1995.

[65] N. Nise, *Control System Engineering*, 5th ed. John Wiley & Sons, 2008.

[66] Y. Ito, S. Doki, and S. Okuma, "Digital filter design with state space methods for 1-bit signal processing based on delta-sigma modulation," *Electrical Engineering in Japan*, vol. 128, no. 12, pp. 1781–1788, December 2011.

[67] N. Patel, S. Nguang, G. Coghill, and A. Swain, "Online implementation of servo-controllers using bit-streams," in *IEEE Region 10 Conference (TENCON 2005)*, November 2005, pp. 1 –6.

[68] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized predictive control - part II: Extensions and interpretations," *Automatica*, vol. 23, no. 2, pp. 149–160, 1987.

[69] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice- a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[70] R. Gorez, V. Wertz, and Z. Kuan-Yi, "On a generalized predictive control algorithm," *Systems &amp; Control Letters*, vol. 9, no. 5, pp. 369 – 377, 1987. [Online]. Available: http://www.sciencedirect.com/science/article/pii/016769118790065X

[71] M. Minkova, D. Minkov, J. Rodgerson, and R. Harley, "Adaptive neural speed controller of a dc motor," *Electric Power Systems Research*, vol. 47, no. 2, pp. 123 – 132, 1998. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378779698000571

[72] J.-J. Liu, "A new gui interpretation tool for the nonlinear frequency response function," in *Journal of Franklin Institute*, vol. 339, 2002, pp. 431–454.

[73] A. M. Sharaf and A. A. El-Gammal, "Multi-objective PSO/GA optimal control strategies for energy efficient PMDC motor drives." *European Transactions on Electrical Power*, vol. 21, pp. 2080–2097, 2011.

[74] N. Thomas and P. Poongodi, "Position control of dc motor using genetic algorithm based pid controller," *Proceedings of the World Congress on Engineering*, vol. 2, July 2009.

[75] J. H. Holland, *Adaptation in natural and artificial systems*, A. Arbor, Ed. MI: Univ. Mich. Press, 1975.

[76] T. Back, D. B. Fogel, and Z. Michalewicz, *Evolutionary Computation 1: Basic Algorithms and Operators*. Bristol : Institute of Physics Publishing, 2000.

[77] ——, *Evolutionary Computation 2: Advanced Algorithms and Operators*. Bristol : Institute of Physics Publishing, 2000.

[78] K. Y. Rani and H. Unbehauen, “Study of predictive controller tuning methods,” in *Automatica*, vol. 33, 1997, pp. 2243–2248.

[79] F. M. Rahmat, K. Y. Hoe, S. Usman, and N. A. Wahab, “Modelling of pt326 hot air blower trainer kit using PRBS signal and cross-correlation technique,” in *Jurnal Teknologi*, vol. 42, no. D. Universiti Teknologi Malaysia, Jun 2005, pp. 9–22.

[80] S. Billings and I. Leontaritis, “Model selection and validation methods for nonlinear systems,” *International Journal of Control*, vol. 44, pp. 311–341, 1986.

[81] M. Ono, K. Shunsaku, and O. Hisao, “Japan superconductive train,” *IEEE Transactions on Instrumentation and Measurement Magazine*, vol. 5, no. 1, pp. 9–15, 2002.

[82] K. Nagaya and M. Ishikawa, “A noncontact permanent magnet levitation table with electromagnetic control and its vibration isolation method using direct disturbance cancellation combining optimal regulators,” *IEEE Transactions on Magnetics*, vol. 31, no. 1, pp. 885–896, 1995.

[83] e. a. S. C. Mukhopadhyaya, “Design, analysis and control of a new repulsive-type magnetic bearing system,” *IEE proceedings Electric power applications*, vol. 146, no. 1, pp. 33–40, 1999.

[84] T. Masuzawa, S. Ezoe, T. Kato, and Y. Okada, “Magnetically suspended centrifugal blood pump with an axially levitated motor,” *Arificial Organs*, vol. 27, no. 7, pp. 631–638, 2003.

[85] M. Varvella, E. Calloni, L. D. Fiore, L. Milano, and N. Arnaud, “Feasibility of a magnetic suspension for second generation gravitational wave interferometers,” *Astroparticle Physics*, vol. 21, no. 3, pp. 325–335, 2004.

[86] H. Lee, K. Kim, and J. Lee, “Review of maglev train technologies,” *IEEE Transaction on Magnetics*, vol. 42, no. 7, pp. 1917–1925, 2006.

[87] W. Barie and J. Chiasson, “Linear and nonlinear state-space controllers for magnetic levitation,” *International Journal of Systems Science*, vol. 27, pp. 1153–1163, 1996.

[88] A. J. Joo and J. H. Seo, “Design and analysis of nonlinear feedback linearizing control for an

electromagnetic suspension system," *IEEE Transactions on Control System Technology*, vol. 5, pp. 135–144, 1997.

[89] M. D. Queiroz and D. M. Dawson, "Nonlinear control of active magnetic bearings: a backstepping approach," *IEEE Transactions on Control System Technology*, vol. 4, pp. 545–552, 1996.

[90] A. E. Hajjaji and M. Ouladsine, "Modelling and nonlinear control of magnetic levitation system," *IEEE Transactions on Industrial Electronics*, vol. 48, pp. 831–838, 2001.

[91] Z.-J. Yang, K. Miyazaki, S. Kanae, and K. Wada, "Robust position control of a magnetic levitation system via dynamic surface control technique," *IEEE Transactions on Industrial Electronics*, vol. 51, no. 1, pp. 26–34, 2004.

[92] F. J. Lin, T. L. Teng, and P. H. Shieh, "Intelligent sliding-mode control using RBFN for magnetic levitation system," *IEEE Transaction on Industrial Electronics*, vol. 54, no. 3, pp. 1752–1762, 2007.

[93] R. J. M. Afonso and R. K. H. Galvao, "Predictive control of a magnetic levitation system with infeasibility handling by relaxation of output constraints," *ABCM Symposium Series in Mechatronics*, vol. 3, pp. 11–18, 2008.

[94] Z. J. Yang, Y. Fukushima, S. Kanae, and K. Wada, "Robust nonlinear output feedback control of a magnetic levitation system by K-filter approach," *IET Control theory and applications*, vol. 3, no. 7, pp. 852–864, 2009.

[95] J. Baranowski and P. Piatek, "Observer-based feedback for the magnetic levitation system," *Transactions of Insitute of Measurement and Control*, vol. DOI: 10.1177/0142331210389650, pp. 1–14, 2011.

[96] H. Karampoorian and R. Mohseni, "Control of a nonlinear magnetic levitation system by using constraint generalized model predictive control," in *International Conference on Control Automation and Systems (ICCAS),2010*, Oct. 2010, pp. 48 –51.

[97] K. Kemih, O. Tekkouk, and S. Filali, "Constrained generalised predictive control with estimation by genetic algorithm for a magnetic levitation system," *International Journal of Innovative Computing, information and control*, vol. 2, no. 3, pp. 543–552, 2006.

[98] T. H. Wong, "Design of a magnetic levitation control system - an undergraduate project," *Education, IEEE Transactions on*, vol. E-29, no. 4, pp. 196 –200, nov. 1986.

[99] H. Woodson and J. Melcher, *Electromechanical Dynamics - Part 1: Discrete Systems*.

New York : Wiley, 1968.

[100] W. Hurley and W. Wolfle, “Electromagnetic design of a magnetic suspension system,” *Education, IEEE Transactions on*, vol. 40, no. 2, pp. 124 –130, may 1997.

[101] W. Barie, “Design and implementation of a nonlinear state-space controller for a magnetic levitation system,” Master’s thesis, University of Pittsburgh, Pennsylvania U.S.A, 1994.

[102] J.-J. E. Slotine and W. Li, *Applied Nonliear Control*, J. Wenzel, Ed. Prentice Hall, 1991.

[103] A. Isidori, *Nonlinear Control Systems*. New York : Springer-Verlag, 1989.

[104] H. Nijmeijer and A. Der Schaft, *Nonlinear Dynamical Control Systems*. New York : Springer-Verlag, 1990.

[105] E. M. A. M. Mendes and S. A. Billings, “An alternative solution to the model structure selection problem,” *IEEE Transaction on System Man and Cybernetics part A: System and Humans*, vol. 31, no. 6, pp. 597–608, 2001.

[106] K. Kristinsson and G. A. Dumont, “System identification and control using genetic algorithm,” in *IEEE Transactions on Systems, Manm and Cybernetics*, vol. 22, no. 5, September/October 1992.

[107] T. Hachino and H. Takata, “Structure selection and identification of Hammerstein type nonlinear systems using automatic choosing function model and genetic algorithm,” in *IEICE Trans. Fundamentals*, vol. E88-A, no. 10, October 2005.

[108] H. M. Abbas and M. M. Bayoumi, “Volterra system identification using adaptive genetic algorithms,” *Applied Soft Computing*, vol. 5, no. 1, pp. 75–86, December 2004.

[109] X. Hong and S. A. Billings, “Parameter estimation based on stacked regression and evolutionary algorithms,” *IEE Proc.- Control Theory Appl.*, vol. 146, no. 5, pp. 406–414, September 1999.

[110] T. Kumon, M. Iwasaki, T. Suzuki, T. Hashiyama, N. Matsui, and S. Okuma, “Nonlinear system identification using genetic algorithm,” *Industrial Electronics Society, 2000 IECON 200. 26th Annual Conference of the IEEE*, vol. 4, pp. 2485–2491, 2000.

[111] J. Madar, J. Abonyi, and S. Ferenc, “Genetic programming for the identification of nonlinear input-output models,” *Industrial and Engineering Chemistry Research*, vol. 44, no. 9, pp. 3178–3186, 2005.

[112] A. K. Swain, “Dynamic modelling and control of robotic manipulator with an investigation of evolutionary computation methods,” Ph.D. dissertation, University of Sheffield, 2001.

[113] E. D. Sontag, “Realization theory of discrete-time nonlinear systems: Part i - the bounded case,” *IEEE Transactions on Circuits and Systems*, vol. cas-26, no. 4, pp. 342–356, 1979.

[114] E. Sontag, *Polynomial response maps*, ser. Lecture notes in control and information sciences. Springer-Verlag, 1979. [Online]. Available: http://books.google.co.nz/books?id=TBuoAAAAIAAJ

[115] S. A. Billings and S. Chen, “Extended model set, global data and threshold model indentification of severely non-linear systems.” *Internation Journal of Control*, vol. 50, pp. 1897–1923, 1989.

[116] D. Braess, *Nonlinear approximation theory*. Berlin: Springer-Verlag, 1986.

[117] S. A. Billings and S. Chen, “Identification of nonlinear rational systems using a prediction error estimation algorithm,” *Int. J. Systems Science*, vol. 20, no. 3, pp. 467–494, 1989.

[118] Q. M. Zhu and S. A. Billings, “Parameter estimation for stochastic nonlinear rational models,” *Int. J. Control*, vol. 57, no. 2, pp. 309–333, 1993.

[119] A. Swain, S. A. Billings, P. Stansby, and M. Baker, “Accurate prediction of non-linear wave forces: Part i (fixed cylinder),” *Mechanical Systems and Signal Processing*, vol. 12, pp. 449–485, 1998.

[120] S. A. Billings and K. M. Tsang, “Spectral analysis for nonlinear systems, part 1: Parametric nonlinear spectral analysis,” *Mechanical Systems and Signal Processsing*, vol. 4, pp. 319–339, 1989.

[121] S. Billings and W. Voon, “Correlation based model valdiaty tests for non-linear models,” vol. 44, no. 1, pp. 235–244, 1986.

[122] S. Billings and Q. Zhu, “Structure detection for nonlinear dynamic rational models.” *International Journal of Control*, vol. 59, pp. 1439–1463, 1994.

[123] S. Weigend and N. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past. Proceedings of the NATO Advanced Research Workshop on Comparative Time Series Analysis held in Santa Fe, SFI studies in Sciences of Complexity XV*. Prentice Hall, 1993.

[124] T. Soderstrom and P. Stoica, *System identification*, 1st ed. Prentice Hall: Englewood

Cliffs, NJ, 1989.

[125] S. A. Billings and W. S. F. Voon, “Correlation based model validity test for non-linear models,” *International Journal of Control*, vol. 44, no. 1, pp. 235–244, 1986.

[126] M. J. Perry, C. G. Koh, and Y. S. Choo, “Modified genetic algorithm strategy for structural identification,” *Computers and Structures*, vol. 84, pp. 529–540, 2006.

[127] W. Chang, “An improved real-coded genetic algorithm for parameters estimation of nonlinear systems,” *Mechanical systems and signal processing*, vol. 20, p. 236246, 2006.

[128] G. B. Fogel, G. W. Greenwood, and K. Chellapilla, “Evolutionary computation with extinction: Experiments and analysis,” *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 2, pp. 16–19, 2000.

[129] M. Baker, “Wave loading on a small diameter flexibly mounted cylinder in random waves,” Ph.D. dissertation, Univerisyt of Salford, U.K., 1994.

[130] W. J. Rugh, *Nonlinear system theory-The Volterra/Wiener Approach,*. The Johns Hopkins University Press, 1981.

[131] M. Morari and J. H. Lee, “Model predictive control: past, present and future,” *Computers and Chemical Engineering*, vol. 23, pp. 667–682, 1999.

[132] J. A. Richalet, A. Rault, J. Testud, and J. Papon, “Model predictive control heuristic control: applications to an industrial process,” *Automatica*, vol. 14, pp. 413–428, 1978.

[133] C. R. Cutler and B. L. Ramaker, “Dynamic matrix control- a computer control algorithm,” in *AIChE National Meeting, Houston, Texas*, 1979.

[134] C. E. Garcia and M. Morari, “Internal model control-1: A unifying review and some new results,” *Ind. Eng. Chem. Process Des and Dev*, vol. 21, pp. 308–323, 1982.

[135] M. A. Henson, “Nonlinear model predictive control: current status and future directions,” *Computers and Chemical Engineering*, vol. 23, pp. 187–201, 1998.

[136] G. R. Sriniwas and Y. Arkun, “A global solution to the nonlinear model predictive control algorithms using polynomial ARX models,” *Computers and Chemecal Engineering*, vol. 21, no. 4, pp. 431–439, 1997.

[137] T. Peterson, E. Hernandez, Y. Akrun, and F. J. Schork, “A nonlinear DMC algorithm and its application to a semibatch polymerization reactor,” *Chemical Engineering Science*, vol. 47,

no. 4, pp. 737–753, 1992.

[138] C. Camasca, A. Swain, and N. Patel, “Intelligent structure selection of polynomial nonlinear systems using evolutionary programming,” in *9th International Conference on Control, Automation, Robotics and Computer Vision*, Dec 5-8 2006.

[139] L. Magni, G. D. Nicolao, and R. Scattolini, “Model predictive control: output feedback and tracking of nonlinear systems,” in *Nonlinear predictive control*, B. Kouvaritakis and M. Cannon, Eds. The Institution of Electrical Engineers, 2001.

[140] M. Wong, W. Lam, and K. Leung, “Using evolutionary programming and minimum description length principle for data mining of bayesian networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 2, pp. 174–175, Feb 1999.

[141] T. L. Sze, “System Identification Using Radial Basis Functions Network,” Ph.D. dissertation, University of Sheffield, August 1995.

[142] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vectors Fields*. Springer-Verlag, 1983.

[143] P. Holmes, “A nonlinear oscillator with a strange attractor,” *Philos. Trans. Royal Soc. London A*, vol. 292, pp. 419–448, 1979.

LIST OF AUTHOR'S PUBLICATIONS

Camasca, C.; Swain, A.K.; Patel, N.D., ”Bit-Stream Based Model Predictive Controller For A Magnetic Levitation System”, *TENCON 2011-IEEE Region 10 Conference* pp. 1055-1059, 21-24 Nov **2011**, Bali, Indonesia.

Camasca, C.; Patel, N.D.; Swain, A.K., ”Implementation of Generalized Predictive Controller for Linear Systems Using Bit-Streams”, *TENCON 2011-IEEE Region 10 Conference*, pp. 1075-1079, 21-24 Nov **2011**, Bali, Indonesia.

Camasca, C.; Swain, A.K.; Patel, N.D., ”Generalized Predictive Control of Nonlinear Systems Using Evolutionary Computation”, *TENCON 2009-IEEE Region 10 Conference*, pp 1-6, November 23rd-26th **2009**, Singapore.

Camasca, C.; Swain, A.K.; Patel, N.D., ”Efficient Structure Selection of Polynomial Nonlinear Systems Using Evolutionary Programming”, *International Journal of Systems and Control*, **2007**.

Camasca, C.; Swain, A.K.; Patel, N.D., ”Intelligent Structure Selection of Polynomial Nonlinear Systems Using Evolutionary Programming”, *Proceedings International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, 5th-8th December **2006**.