

Loughborough University Institutional Repository

Strategies for non-uniform rate sampling in digital control theory

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: <https://dspace.lboro.ac.uk/2134/6321>

Publisher: © Mohammad Samir Khan

Please cite the published version.

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

 **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

STRATEGIES FOR NON-UNIFORM RATE
SAMPLING
IN DIGITAL CONTROL THEORY

By
Mohammad Samir Khan

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
LOUGHBOROUGH UNIVERSITY
LOUGHBOROUGH, LE11 3TU
JUNE 2010

© Copyright by Mohammad Samir Khan, 2010

Abstract

This thesis is about digital control theory and presents an account of methods for enabling and analysing intentional non-uniform sampling in discrete compensators.

Most conventional control algorithms cause numerical problems where data is collected at sampling rates that are substantially higher than the dynamics of the equivalent continuous-time operation that is being implemented. This is of relevant interest in applications of digital control, in which high sample rates are routinely dictated by the system stability requirements rather than the signal processing needs. Considerable recent progress in reducing the sample frequency requirements has been made through the use of non-uniform sampling schemes, so called '*alias-free*' signal processing. The approach prompts the simplification of complex systems and consequently enhances the numerical conditioning of the implementation algorithms that otherwise, would require very high uniform sample rates. Such means of signal representation and analysis presents a variety of options and thus is being researched and practiced in a number of areas in communications. However, the control communities have not yet investigated the use of intentional non-uniform sampling, and hence the ethos of this research project is to investigate the effectiveness of such sampling regimes, in the context of exploiting the benefits.

Digital control systems exhibit bandwidth limitations enforced by their closed-loop frequency requirements, the calculation delays in the control algorithm and the interfacing conversion times. These limitations pave the way

for additional phase lags within the control loop that demand very high sample rates. Since non-uniform sampling is propitious in reducing the sample frequency requirements of digital processing, it proffers the prospects of being utilised in achieving a higher control bandwidth without opting for very high uniform sample rates.

The concept, to the author's knowledge, has not formally been studied and very few definite answers exist in control literature regarding the associated analysis techniques. The key contributions adduced in this thesis include the development and analysis of the control algorithm designed to accommodate intentional non-uniform sample frequencies. In addition, the implementation aspects are presented on an 8-bit microcontroller and an FPGA board. This work begins by establishing a brief historical perspective on the use of non-uniform sampling and its role for digital processing. The study is then applied to the problem of digital control design, and applications are further discoursed. This is followed by consideration of its implementation aspects on standard hardware.

Keywords: Digital control, non-uniform sampling, delta transform, real-time signal processing, Fourier analysis, FPGA.

Acknowledgements

First and above all, I praise Almighty ALLAH for providing me this opportunity and granting me the capability to proceed successfully during this course. He has always stood by me through the most difficult times, and has filled my life with His limitless bounties and mercy. May your name be Exalted, Honored, and Glorified.

I would like to express my deepest thanks to my esteemed supervisors, Prof. Roger Goodall and Dr. Roger Dixon, who have not only paid attention to me and my work than I could possibly hope for but, have also been good friends. I am grateful for their encouragement to join the Ph.D. program at Loughborough, and for taking me on as a student. You have done a great deal for me over the years and I have particularly enjoyed our fruitful research discussions, and the atmosphere of freedom which characterises them.

Equally, my most sincere thanks to my parents who, despite my absence from their lives, have stood by me without question. I am grateful to my brother Munir, for his encouragements, and my sister Mariam, whose never gotten tired of my complaining. There is nothing I can say here that will do justice to how I feel about my family's support, but without their patience and *love*, this work would never have come into existence (literally).

My thanks must also extend to everybody in the Control Systems Research Group, without whom I could not possibly have completed this project. I am

particularly indebted to Karmjit, Jessica, Chris and Paul for several intellectual and largely *non-intellectual* contributions in my discussions, and for truly being good friends and a source of much needed humour. For the others in the group: Kostas, Bill, Thomas, Duncan, Hairy, Pete, Roy, Guy, Hazlina, Cheryl, Rosemarie, Yimin, Argyrios, Andrew and Ale, thank you for contributing to my time in many ways. I have had the good fortune to work in this very inspiring environment and it has been a privilege to work together with such intelligent and friendly colleagues at the department.

There is not enough space in this entire thesis to thank all the people, who have one way or the other been connected to me during this venture, that deserve my most heartfelt gratitude and any effort made in this vain will only ever be incomplete - to those who I have left out I apologise but rest assured that I will not forget what you have done for me during this period.

Loughborough, Leicestershire
February 7, 2010

Mohammad Samir Khan

Table of Contents

Abstract	i
Acknowledgements	iii
Table of Contents	v
Notational conventions	xvi
Acronyms	xvii
1 Introduction	1
1.1 Problem definition	2
1.1.1 Non-uniform control systems	5
1.2 Research contributions	8
1.3 Thesis outline	10
2 Background and Literature Review	11
2.1 The sampling process	12
2.1.1 Sampling of control systems	17
2.2 Non-uniform sampling	23
2.2.1 Developments and applications	26
2.3 Control systems design	30
2.3.1 Digital control	31
2.3.2 Implementation operators	33
2.4 Non-uniform sampling in control theory	38
2.4.1 The impact of sample time variations	41
2.5 General remarks	43
2.6 Summary	44

3	Continuous-time Transfer Function <i>Emulation</i> with Non-uniform sampling	46
3.1	Design of discrete equivalents	48
3.1.1	Numerical integration methods	49
3.1.2	Mapping techniques	53
3.1.3	Transient management for control systems	53
3.2	Non-uniform rate discrete equivalents	58
3.2.1	General assumptions	59
3.2.2	Mathematical formulation	60
3.2.3	Limitations on design	68
3.3	Suppressing the transient effects	69
3.3.1	The importance of implementation structure	69
3.3.2	Discussion	80
3.4	Frequency analysis method for non-uniform rate controllers	83
3.5	Performance evaluation criteria	92
3.6	Open-loop analysis of the phase lead-lag compensator	93
3.7	General conclusions	110
3.8	Practical example: closed-loop with DC motor	112
3.8.1	Discussion	127
3.9	Summary	128
4	Control algorithms for implementation	131
4.1	Software considerations	132
4.1.1	Algorithm design	133
4.1.2	Embedded software development	137
4.2	Microcontroller implementation	141
4.2.1	Plant description	141
4.2.2	Hardware realisation	142
4.2.3	Hardware simulation: open-loop	143
4.2.4	Experimental Results: closed-loop with SMAC actuator	146
4.3	Implementation with programmable Logic	150
4.3.1	RTL modelling of the non-uniform rate controller	151
4.3.2	Hardware-in-the-loop simulation	157
4.3.3	Hardware results: closed-loop with DC motor	158
4.4	Summary	160
5	Conclusions	161
5.1	Summary of research	161
5.2	Future work	167

5.2.1 Possible application to system identification	168
A Uniform sampling	171
B Non-uniform sampling: Frequency estimation	175
C Code implementation details	179
D Derivation of the non-uniform rate compensator coefficient values	184
D.1 The z -filter	184
D.2 The δ -filter	186
E Publications	189
F Sampling models	190
Bibliography	194

List of Figures

1.1	Discrete-time control system	3
1.2	Non-uniform rate discrete equivalent control system	6
2.1	Sampling process pulse train ($0 \leq \gamma \leq T$)	13
2.2	The signal sampling process	15
2.3	The frequency response of the Zero-Order-Hold.	19
2.4	The event-based approach	23
2.5	(a) The frequency response to a 80Hz sinusoidal signal with a uniform sampling frequency at 100Hz. The aliases are clearly visible and have corrupted the signal information. (b) The frequency response of a 80Hz sinusoidal signal with a non-uniform sampling pattern with an average sampling frequency at 100Hz. The aliases are suppressed and the true signal content is apparent.	28
2.6	Structures of a 2^{nd} order digital z -filter.	35
2.7	Structures of a 2^{nd} order digital δ -filter	37
3.1	Continuous and discrete-time representation for a 2^{nd} order transfer function	50
3.2	Demonstrating the transient phenomenon (filter structure: z -canonic)	55

3.3	Non-uniform sampling algorithm validation (uniform sample rate T_n : 0.01s). Note that, this simulation utilises the algorithm without sample time variations in the sampling period; filter structure: canonic- z	64
3.4	Accommodating for sample period change (uniform distribution with an average sample rate: 0.015s)	66
3.5	The transient phenomenon with non-uniform sampling (uniform distribution ($T_n \sim U(0.01,0.02)$, $0.01 \leq T_n \leq 0.02$) with an average sample rate: 0.015s); filter structure: canonic- z	67
3.6	The non-uniform sampling being used belong to a uniform distribution $T_n \sim U(0.01,0.02)$, $0.01 \leq T_n \leq 0.02$, average sampling frequency at 0.015s	72
3.7	The filter output for non-uniform sampling (uniform distribution $T_n \sim U(0.01,0.02)$, $0.01 \leq T_n \leq 0.02$, average sampling frequency at 0.015s); filter structure: canonic- δ	73
3.8	The non-uniform sampling being used belong to a uniform distribution $T_n \sim U(0.01,0.02)$, $0.01 \leq T_n \leq 0.02$, average sampling frequency at 0.015s	76
3.9	The filter output for non-uniform sampling (uniform distribution $T_n \sim U(0.01,0.02)$, $0.01 \leq T_n \leq 0.02$, average sampling frequency at 0.015s); filter structure: modified canonic- δ	77
3.10	The filter setup for non-uniform sampling (uniform distribution with an average sampling frequency at 0.015s)	78
3.11	The direct z -filter output with non-uniform sampling with a uniform distribution $T_n \sim U(0.01,0.02)$ ($0.01 \leq T_n \leq 0.02$, and an average sampling frequency at 0.015s); filter structure: direct- z	79

3.12	The filter frequency response at 5Hz. After an initial transient, the values settle down towards a steady state to provide the magnitude and phase values at the frequency of interest. The non-uniform sampling used here belongs to a uniform distribution with an average sampling frequency at 0.015s, where $T_n \sim U(0.01,0.02)$	87
3.13	The magnitude (dB) values for the digital filters under observation, at various frequencies.	89
3.14	The phase (degrees) values for the digital filters under observation, at various frequencies.	90
3.15	Histogram of uniformly distributed sample rates $T_n \sim U(0.01,0.02)$	94
3.16	The time response of the non-uniform sampling filter implementing with uniformly distributed sample rates i.e. $T_n \sim U(0.01,0.02)$	95
3.17	The frequency response of the non-uniform sampling filter implementing with uniformly distributed sample rates i.e. $T_n \sim U(0.01,0.02)$	96
3.18	The performance loss in the frequency response of the non-uniform sampling filter implementing with uniformly distributed sample rates i.e. $T_n \sim U(0.01,0.02)$	97
3.19	Histogram of <i>normally</i> distributed sample rates $T_n \sim N(0.015,0.1^2)$	98
3.20	The time response of the non-uniform sampling filter implementing with truncated gaussian distributed sample rates i.e. $T_n \sim N(0.015,0.1^2)$	99
3.21	The frequency response of the non-uniform sampling filter implementing with truncated gaussian distributed sample rates i.e. $T_n \sim N(0.015,0.1^2)$	100

3.22	The performance loss in the frequency response of the non-uniform sampling filter implementing with truncated gaussian distributed sample rates i.e. $T_n \sim N(0.015, 0.1^2)$	101
3.23	Histogram of dual sample rates $T_n = 0.01s$ or $T_n = 0.02s$. . .	102
3.24	The time response of the filter implementing dual sample rate (digital pattern) i.e. $T_n = 0.01s$ or $T_n = 0.02s$	103
3.25	The frequency response of the filter implementing dual sample rate (digital pattern) i.e. $T_n = 0.01s$ or $T_n = 0.02s$	104
3.26	The performance loss in the frequency response of the filter implementing dual sample rate (digital pattern) i.e. $T_n = 0.01s$ or $T_n = 0.02s$	105
3.27	Histogram of sin wave patter sample rates $0.01s \leq T_n \leq 0.02s$	106
3.28	The time response of the filter implementing sample rates according to a sin wave pattern i.e. $0.01s \leq T_n \leq 0.02s$	107
3.29	The frequency response of the filter implementing sample rates according to a sin wave pattern i.e. $0.01s \leq T_n \leq 0.02s$	108
3.30	The performance loss in the frequency response of the filter implementing sample rates according to a sin wave pattern i.e. $0.01s \leq T_n \leq 0.02s$	109
3.31	The DC motor block diagram	112
3.32	The step response with the continuous controller in a closed-loop	113
3.33	The overall control scheme	113
3.34	The digital controller output to a step input	115
3.35	The plant output response to a unit step input	116
3.36	The frequency response of the closed-loop system implementing the digital controller with uniformly distributed sample rates i.e. $T_n \sim U(0.01, 0.02)$	117
3.37	The digital controller output to a step input	118
3.38	The plant output response to a unit step input	119

3.39	The frequency response of the closed-loop system implementing the digital controller with truncated gaussian distributed sample rates i.e. $T_n \sim N(0.015, 0.1^2)$	120
3.40	The digital controller output to a step input	121
3.41	The plant output response to a unit step input	122
3.42	The frequency response of the closed-loop system implementing the digital controller with dual sample rate (digital pattern) i.e. $T_n = 0.01s$ or $T_n = 0.02s$	123
3.43	The digital controller output to a step input	124
3.44	The plant output response to a unit step input	125
3.45	The frequency response of the closed-loop system implementing the digital controller with a sinusoid distribution i.e. $0.01s \leq T_n \leq 0.02s$	126
4.1	Control program dataflow diagram	135
4.2	Sample rate regulation dataflow diagram	136
4.3	The SMAC actuator	142
4.4	Block diagram of the actuator control system	143
4.5	The response of the control signal to a step input, with constant sampling $T_n=0.015s$	144
4.6	The response of the control signal to a step input, with non-uniform sampling period: dual rate sampling $T_n = 0.01s, 0.02s$, average sample rate= $0.015s$	144
4.7	The response of the control signal to a step input, with non-uniform sampling period: uniformly distributed sample rates (Sawtooth wave pattern) $T_n \sim U(0.01, 0.02)$, average sample rate= $0.015s$	145
4.8	The experimental setup: includes the digital controller and the actuator.	145
4.9	With constant sampling period: $T_n = 0.015s$	147

4.10	With non-uniform sampling period: Dual rate sampling, $T_n = 0.01\text{s}, 0.02\text{s}$, average sample rate = 0.015s	147
4.11	With non-uniform sampling period: uniformly distributed samples (sawtooth wave pattern), $T_n \sim U(0.01,0.02)$, average sample rate = 0.015s	148
4.12	Simulink view of a 1 st order non-uniform sample rate filter; implementation structure: modified canonic- δ	150
4.13	RTL view of a 4 th order non-uniform rate controller	153
4.14	Open-loop output to a unit step input (in Matlab) from the non-uniform rate controller, designed using Xilinx TM System Generator. The non-uniform sample rate: Dual rate with $T_n=0.01\text{s}, 0.02\text{s}$	156
4.15	The HIL schematic block diagram of motor model	157
4.16	The Spartan3E Development kit	158
4.17	Closed-loop output produced using the non-uniform rate control algorithm, with the DC motor model in a closed-loop. The non-uniform sample rate: Dual rate with $T_n=0.01\text{s}, 0.02\text{s}$	159
A.1	The uniform sampling process	171
A.2	The reconstruction process	174
B.1	Uniformly sampled data. Sampling below the Nyquist frequency causes aliases which can clearly be seen to have corrupted the signal	176
B.2	Non-uniformly sampled data. Adding variations to the sampling scheme can mitigate the effects of aliasing	177
F.1	Probability density functions for jitter sampling. (a), (b), (c), (d) are the functions of the time intervals $t_1-t_0, t_2-t_0, t_3-t_0$ and t_7-t_0 respectively. (e) concluding sampling point density function.	191

F.2	Probability density functions for additive random sampling. (a), (b), (c), (d) are the functions of the time intervals t_1-t_0 , t_2-t_0 , t_3-t_0 and t_7-t_0 respectively. (e) concluding sampling point density function.	193
-----	--	-----

List of Tables

3.1	Accommodating for sample period change	56
3.2	Comparing the Mean Squared Error of the various filter structures for various sampling conditions.	82
3.3	The filter magnitude values (dB)	86
3.4	The filter phase values (degrees)	86
3.5	The filter magnitude values (dB) for the compensators under observation.	91
3.6	The filter phase values (degrees) for the compensators under observation.	91
4.1	IAE with various sampling schemes	148
4.2	PI coefficient parameters	155
4.3	Phase advance coefficient parameters	155
4.4	Notch filter coefficient parameters	155

Notational conventions

a_n, b_n	Fourier coefficients
δ	Delta operator
$\delta()$	Delta function
ΔT	Fraction of a period
Δt	Time interval
f	Frequency (Hz)
f_s	Sampling frequency
μ	Mean value of time intervals T_n
N	Number of samples
$P(t)$	Pulse function
s	Laplace operator
\S	Section
σ	Standard deviation of time intervals T_k
t	Time variable
t_k	Time instant
T	Sample time
w	Angular frequency
w_s	Angular sampling frequency
$x(t), y(t)$	Signals
$X(w), Y(w)$	Fourier result of $x(t)$ and $y(t)$, respectively
z	discrete operator

Acronyms

ADC	Analog-to-digital converter
DAC	Digital-to-analog converter
DASP	Digital alias-free signal processing
DFT	Discrete Fourier transform
DSP	Digital signal processing
FFT	Fast Fourier transform
FPGA	Field programmable gated array
HDL	Hardware descriptive language
HIL	Hardware-in-the-loop
IIR	Infinite impulse response
IAE	Integral of absolute error
LTI	Linear time invariant
MAC	Multiply-accumulate
MSE	Mean squared error
RTL	Register transfer level
SNR	Signal-to-noise ratio

Chapter 1

Introduction

A perspective on Digital Control

Digital control is a branch of control theory that makes use of digital computers or microcontrollers to modify the behaviour of a target system. Application examples range from electromechanical servo mechanisms to micro surgeries, where real-time control plays a crucial role in the coordination of the dynamics of these systems. The advantages of using digital approaches increases the flexibility of control algorithms and the decision making capability of digital controllers, which are placed in closed loops to meet specific system requirements. Additionally, controllers can be used with several different software variations to provide a profound range of solutions, thus simplifying and reducing the design time.

Chapter overview

This chapter sets the scene by discussing the purpose of this work, some basic classifications, and abstractions of non-uniform sampling; its recently identified benefits and how it can be correlated to control theory. The important issues are called attention to, and a compendium of where these issues are discussed in this thesis is given.

1.1 Problem definition

It is the purpose of this thesis to provide an account of some principal theories and analysis methods for enabling intentional non-uniform sampling in digital control.

Non-uniform sampling has shifted into an era of research where its theoretical analysis can be realized as a practical solution. Whilst traditional engineering fields have always been aimed towards uniform sampling, irregular sampling is slowly becoming the focal point for research as an ultimate cheap alternate for countering issues that otherwise cannot be solved when using uniform sample rates. The fact is that there is no known study that explores the relevance or benefits of deliberate non-uniform sampling for digital control applications and therefore the opportunity for research in this area is unique.

Classical digital control is often implied to be regularly, synchronously, and equally time spaced (Isermann 1989), and even though non-uniform sampling has been an area of popular research, it has hardly been noticed by the control communities. As a consequence, this thesis focuses on two comprehensive motifs in control theory, which are: the representation of non-uniform sampling algorithms for sampled-data systems and their pertinence to digital controllers; with particular interest to the application of studying the impact on the operating control bandwidth of a closed-loop system with various non-uniform sampling schemes.

In DSP applications, the methodology of non-uniform sampling has enabled the processing of digital signals at much slower rates without restrictions from the well-known Nyquist limit. Carefully designed sampling schemes can therefore be used to effectively mitigate the effects of aliasing and permit significant

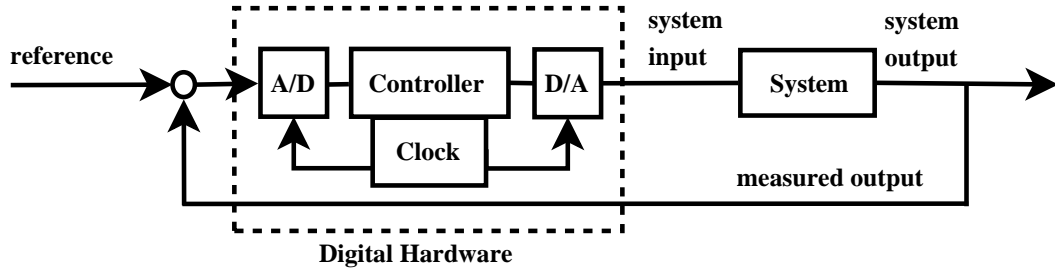


Figure 1.1: Discrete-time control system

reductions in the average sampling frequency, leading to more efficient processor utilisation. Summarising the control application potential for non-uniform sampling, it could be used for the:

- reduction in the bit flow process (Artyukh, Medniks & Vedin 1997)
- use of simpler electronics due to a reduction in overall processing (Bilinskis 2007)
- Overall simplification of complex system designs (Sonnaillon & Bonettot 2007)

A typical digital control system arrangement with a uniform rate controller is depicted in Fig. 1.1. The complete design and analysis of such real-time control systems can be lacking in many ways, due to the involvement of various issues of controller design and processor implementation. Modern control engineers pay scant attention to the implementation constraints, conspicuously because it is assumed that digital controllers are being executed on dedicated processors that are fast and deterministic enough to ignore the implications of any timing constraints on control activities that may have an effect on the implementation. Furthermore, the diffusion of FPGA technology in the DSP market has allowed significant progress in the execution of very fast signal processing applications (Goslin 1996). Despite all this, processing resources

are always limited in practice and variations in timing instances in control algorithms do occur.

Classical discrete-time implementations only assume uniform sampling theory for execution, which obviously imposes a restriction if a non-uniform sampling rate is to be adopted. Practically, several factors during controller implementation (such as the sampling process, the control algorithm and the actuation) may not be sequential due to the variability in job executions produced by sampling and latency jitter (Arzen, Cervin, Eker & Sha 2000). This inadvertent *jitter* between sampling intervals tends to deteriorate the control system performance and, in the worst-case situation, bring the system to instability¹ (Marti, Fuertes & Fohler 2001). There can be two perspectives to this: sample time variations can degrade the control performance and may even lead to the instability of the feedback control system. On the other hand, any efforts to reduce the possibility of sample time variations during the algorithm execution may over-constrain the control processor and affect the execution of other important tasks.

Thesis aims

This work investigates the creative use of non-uniform sampling for the field of digital control; with the foremost questions focusing upon:

Q1 Can intentional non-uniform sampling administer any benefits in digital control applications?

Q2 With time varying instances, how should the discrete-time transfer function relationship be examined in real-time? Can the z -transform be used for this purpose?

¹Such limitations are particularly pertinent when there are high complexity and/or high bandwidth requirements.

Q3 If the answer to Q2 is true then, how can the frequency response of such a function be evaluated?

Q4 Can the non-uniform rate controller be realized with a real physical system?

Chapter 2 fills the need for a concise discussion concerned with Q2 by providing a brief overview of how non-uniform sample instances are typically dealt with in control related applications. Chapter 3 is the core chapter of the thesis since it addresses Q1, Q2 and Q3. It helps in identifying the appropriate analysis methods and the implementation aspects of non-uniform rate controllers. Chapter 4 helps to answer Q4 and shows that control signals may correctly be processed with intentional non-uniform sample rates using standard technologies.

1.1.1 Non-uniform control systems

Control literature, for the most part, disregards sample non-uniformity in the controller design of continuous-time linear systems (Marti, Fuertes & Fohler 2001, Marti, Fuertes, Fohler & Ramamritham 2001, Marti 2002). This is probably ignored due to the lack of convenient analysis techniques of the close loop systems. In theory, whenever a signal is digitized, there is loss of information and an overall performance degeneration by the quantisation and sampling errors incurred when reading continuous data. However, Edwards & Durkin (1968) presented an analysis and attempted to reduce this degenerative effect by exercising a non-uniform quantiser at the input. It was demonstrated that, under small signal conditions, the spectrum of the non-uniform quantisation errors is essentially white and hence the technique readily improved the steady-state accuracy of the existing system.

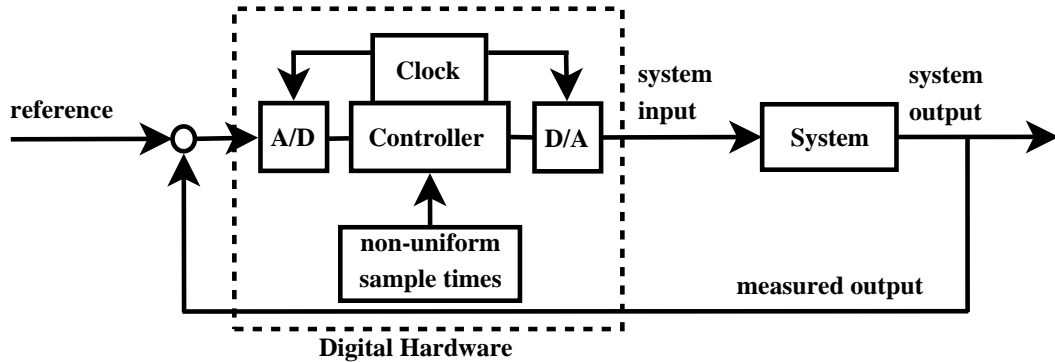


Figure 1.2: Non-uniform rate discrete equivalent control system

The work presented in this thesis proposes a similar approach to the traditional concepts of digital control, and extends the idea by endeavouring to use intentional non-uniform sampling² in the sampling instances as a tool. This is largely due to the gaining popularity of non-uniform sampling theory as a cheap alternative to uniform sampling. Forasmuch as there are many ways in which non-uniformity can occur, this work hence adopts a much more flexible design approach and assumes that the correct processing of sampled signals with non-uniform time intervals is valid only if the time intervals are known in advance (or before the next sample period). The control system of a non-uniform rate equivalent controller can then be setup as depicted in Fig. 1.2. As compared to Fig. 1.1, the only modification is the addition of a separate non-uniform sampling instances block which stores the various sample periods that are to be used. It also enables the interfacing mechanisms to operate in-sync with the controller operations.

According to Bilinskis (2007), whenever a continuous-time signal is to be digitized and the best sampling technique has to be determined, then the aspects to be considered are the:

²—in the context of a uniform, truncated gaussian and other probability distributions.

- spectrum of the sampled signal
- acceptable sampling rate
- subsequent processing of the digitized signal

However, overlapping of additional frequency components do occur in the case with uniform sampling. This phenomenon is known as aliasing, which enables many sinusoids to be drawn through a given sample value set. With non-uniform sampling, the sampling operation is carried out in such a way that the sequence of signal samples obtained is as closely related to the original signal as possible. Over the past couple of decades, studies in non-uniform sampling (Bilinskis & Mikelsons 1990, Wojtiuk 2000) have propounded that signals formed by irregular spacings have features strongly differing from typical ones obtained in the case when signals are sampled periodically. Since it avoids the overlapping of high frequency sinusoidal signals, it opens up the possibility of distinguishing all spectral components of the signal, even if their frequencies substantially exceed the mean sampling rate.

Due to the ongoing importance of producing adequate digital controllers that can process signals with wider bandwidths (Goodall 2001) for control systems, this concept lays down the foundation of this research. From a control engineering point of view, the postulated technique of ‘*alias-free*’ sampling may allow significant reductions in the requirements of speed when sampling, which will help to improve the sensitivity issues of real-time controllers.

Assumptions

Prior works in non-uniform sampling have made some general assumptions regarding the characterisation of the system structures. For instance, the sampling periods of the digital sampling device is operating with some criterion

defined by the user and hence could be deterministic or vice-versa. In the non-deterministic case, the sample periods $\{T_i, i=1,2,\dots\}$ are assumed to be a sequence of random variables with some probabilistic property³ in which the sample periods are bounded by a maximum/minimum value (Marti 2002, Eng 2007).

Other important assumptions are that the system is linear and time invariant (LTI), and it is stable in the sense that all the poles of the closed-loop Laplace transfer function are in the left half s -plane. This simple criterion suffices to the analysis and design techniques for classical continuous-time sampled data control systems (Middleton & Goodwin 1990). Unfortunately, once an element of randomness is added in the controller sampling frequency, the stability problem becomes more complex. However, if it is assumed that the output remains bounded for all combinations of possible sample time variations, the system is still stable in the sense above (Dannenberg 1972).

1.2 Research contributions

The scientific contributions of this thesis have been categorised into two groups. The main contributions are summarised as follows:

1. Brief study on the role of non-uniform sampling in control, with particular emphasis on the use of deliberate variations in the sampling regimes: Variations experienced during control implementation often result in poor system performances, or even instability. The reasons that bring about variations are highlighted and the techniques to compensate

³The probability distribution describes the range of possible values that a random variable can attain and the probability that the value of the random variable is within any (measurable) subset of that range, such as a uniform distribution or a truncated gaussian distribution

their effects are discussed.

2. Non-uniform rate control design: A flexible controller design approach is adopted that goes beyond the classic discrete-time control theory timing assumptions of uniform sampling that are given by constant sampling period values. Instead of specifying a single value for the sampling period, the controller is designed for a set of pre-defined sampling times. This strategy relies on the idea of adjusting the controller parameters at run time according to the specific implementation timing behaviour. The calculations can either be
 - performed online; if the processing overheads allow, or
 - determined off-line to form look-up tables
3. The study develops the analysis techniques for evaluating non-uniform rate controllers. It makes use of the Fourier analysis and assesses the control systems in the time domain. The technique is used to evaluate the frequency characteristics under uniform and non-uniform sampling conditions.

The additional contributions are:

1. Implementation structure importance: The modified canonic δ and the direct z structures are identified to provide a much more robust implementation and are better suited for non-uniform sampling due to their transient suppression capabilities. The transient phenomenon is extensively discussed in Chapter 3 §3.3.
2. The control algorithms are implemented on standard hardware to demonstrate their functionality on existing technologies. These include an 8051 microcontroller and an FPGA starter board.

1.3 Thesis outline

The thesis is structured as followed:

Chapter 2 discusses a brief overview of the role of signal sampling in general. Moreover, it also presents some basic but important concepts in both non-uniform sampling and control systems theory that are related to the context of this work. It highlights the potential benefits of using non-uniform sampling regimes and discusses the direction of this thesis.

Chapter 3 further explains the objective of this research. It then identifies the relevant methods for enabling intentional non-uniform sampling in discrete controllers. §3.2.3 places an emphasis on the limitations of the developed control algorithm and the possible solutions are discussed. In addition, §3.2.2 highlights a technique for estimating the frequency response of the non-uniform rate controller over time and provides examples for validating the procedure.

Chapter 4 is aimed towards the hardware implementation aspects of the non-uniform rate control algorithm and the essential components for its practical realisation as real-time controllers. Some of the issues that were mentioned in the previous chapters are now discussed in more detail. Specific attention is given to the software structure and C/C++ is used to develop the program routines for the algorithm. In addition, the application is applied to the 8051 microcontroller family and the Xilinx FPGA development board *Spartan-3E*.

Chapter 5 addresses the future of non-uniform sampling in control theory in the context of exploiting the benefits from such variations and finally, draws the conclusions on this thesis. It highlights the main contributions in the control literature and discusses extensions and open problems for potential future work.

Chapter 2

Background and Literature Review

Chapter overview

It comes as no surprise that the theory of discrete-time signals has had significant developments in the last century, from which has emanated numerous techniques and mathematical tools for digital signal processing purposes. The idea for analysing and implementing non-uniform sampling schemes in a feedback control systems is therefore another step forward in this vibrant engineering research field, which requires a sound knowledge and understanding in between two ubiquitous disciplines: control theory and signal analysis. To date, the approach of benefiting from non-uniform sampling patterns has chiefly been a part of communication applications only.

This chapter presents a brief overview of some basic but fundamental concepts in sampling theory and for non-uniform sampling in particular. Avoiding too many intricate technicalities of the technique, the chapter places an emphasis on the recent developments and discusses the possible role of non-uniform sampling in control theory. In approaching the problem in this fashion, the motivation is to introduce the rudimentary notions for understanding sampled-data systems and non-uniform sampling, at the same time as bridging the gap

between the two. The objective of this chapter can therefore be summarised as:

- To review the relevant background material on signal processing and digital control.
- To identify the potential benefits of non-uniform sampling schemes.
- To present an argument on dealing with intentional non-uniform sampling for control related applications.

2.1 The sampling process

What is sampling?

Sampling is the process of converting an analogue signal (for example, a function of continuous time or space) into a numeric sequence (a function of discrete time or space). Sampling theory has been elaborately expounded in literature (Nise 2007, Marvasti 2001, Feuer & Goodwin 1996) and is well understood. Since sampling is a linear operation, linear system theories can be applied to its analysis. A continuous analogue signal $x(t)$ can be presented where all the variables are known at all times. To model these variables in the digital domain $x_s(t)$, the signal is multiplied by a infinite sequence of pulses $p(t)$. Such a sampling action will fix and store the characteristics of the analogue signal by the digital system for analysis. The continuous time sampled data can be represented by:

$$x_s(t) = p(t)x(t) \tag{2.1.1}$$

Fig. 2.1 illustrates the sampling process where T is the sampling period. It may be noted that as the pulse duration, γ , approaches zero, the impulse

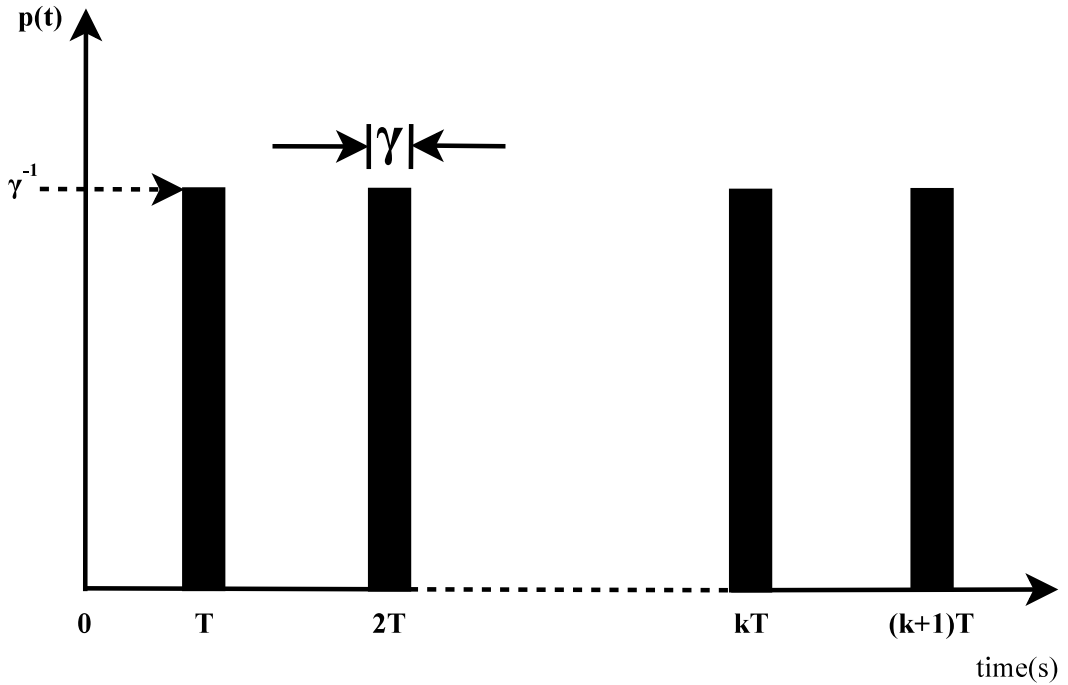


Figure 2.1: Sampling process pulse train ($0 \leq \gamma \leq T$)

function in the sampling process will approach to a unit strength or *impulse sampling*. To enable an effective reconstruction from a sampling process, it is essential that the frequency content of the pulse signal $x_s(t)$ contains all the frequency information that existed previously in $x(t)$. These samples of the pulse signal are available at a succession of uniform time intervals and are acquired using data acquisition devices.

It is well known that, with a uniform sample rate, a continuous set of data must be sampled at a minimum of twice the signal bandwidth in order to avoid adding replicas or false images of the signal in its spectral content. This implies $f_s > 2f_0$ where f_s is the sampling frequency and f_0 is the signal bandwidth of interest. The Sampling theorem is a fundamental result in the field of information theory, in particular the telecommunications fields (Nyquist 1928). The theorem stipulates: *if a function $x(t)$ contains no frequencies higher than*

B^1 cps, it is completely determined by giving its ordinates at a series of points spaced $1/(2B)$ seconds apart.

In essence the theorem shows that an analogue signal that has been sampled can be perfectly reconstructed from the samples if the sampling rate exceeds $2B$ samples per second, where B is the highest frequency in the original signal. The theorem also leads to a formula for reconstruction of the original signal (Shannon 1949). The constructive proof of the theorem leads to an understanding of the aliasing phenomenon that can occur whenever a sampling system does not satisfy the conditions of the theorem.

Generalised sampling model

The mathematical description of a uniformly sampled signal $x_s(t)$ can be generated on a pulse train $p(t)$ (Bilinskis 2007):

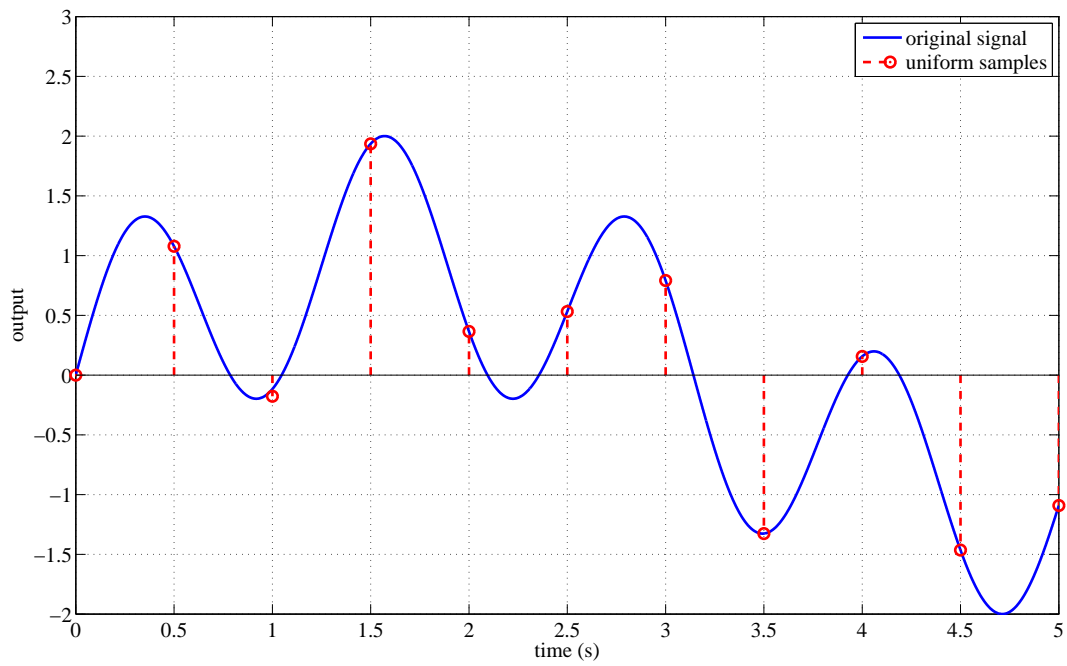
$$p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (2.1.2)$$

The sampling instants in Eq. 2.1.2 are separated on the time axis by sampling interval T and can only be applied to periodic sampling. To enable the analysis of non-uniform samples, this equation is modified as:

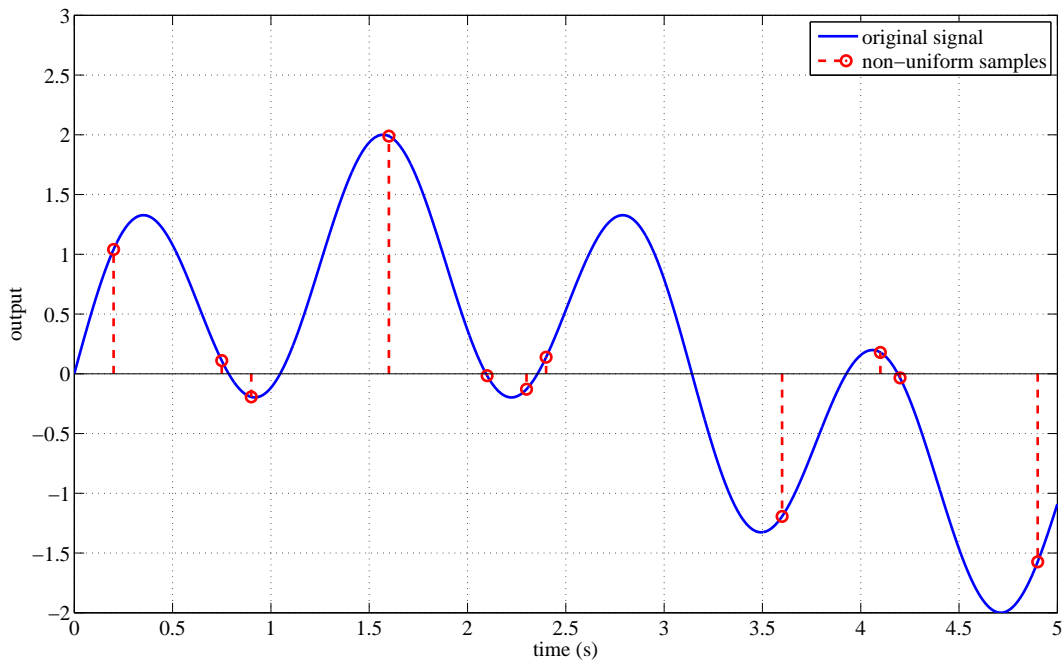
$$p_{NUS}(t) = \sum_{n=-\infty}^{\infty} \delta(t - t_n) \quad (2.1.3)$$

where $\delta(t - t_n)$ is the delta function. The sampled signal can be represented similarly as in Eq. 2.1.1. Graphically, this process can be seen in Fig. 2.2. If the time instances at which these samples are taken are equidistant, for example every T seconds, $x[n] = x(nT)$, then the signal is uniformly sampled (Fig. 2.2a). If the time instances are not equidistant, that is the samples

¹A baseband bandwidth (B) is equal to the highest frequency of a signal or system, or an upper bound on such frequencies.



(a) Uniform sampling



(b) Irregularly sampling

Figure 2.2: The signal sampling process

are taken at arbitrary points $t_k \in \mathbb{R}$, $x[n] = x(t_n)$ then this is known as non-uniform or irregular sampling (Fig. 2.2b).

Digital processing requirements

Sampling is an integral part for processing analogue signals whose exact description can be represented by a multiplication of the signal with a sequence of Dirac delta pulses to form a digital equivalent. The process is carried by a sample-and-hold device which processes according to the current value until the next sample arrives. The sample-and-hold device usually has a fixed number of bits, where each sample value is compared to a predefined binary number that corresponds to the level closest to the input value. This *quantisation* process rounds the input value to the nearest level and therefore, the digital representation is only an approximation to the original input signal. However, amplitude quantisation issues are beyond the scope of this thesis, which concentrates mainly on the randomization in the sampling time instances.

As adduced earlier the most common problem in digital processing applications is aliasing, which is a result of the violation of the sampling theorem. Therefore in signal processing applications, the selection criterion for an appropriate sample rate is limited by the Sampling theorem. Historically, the sampling rate selection is a compromise between the simplicity of high performance prediction from high rates and the practical processing capability. Therefore, during the preliminary design of signaling applications, the issues of sampling rates and other real-time requirements must be addressed (Mittra 2001).

2.1.1 Sampling of control systems

As outlined in §2.1, the foundation for selecting the sampling frequency in signal processing is the Sampling theorem. The theorem specifies that sampling rate should be at least twice the operating bandwidth of the system. Houpis & Lamont (1985) argue that the theorem might further suggest the assumption of two properties:

- the sampled signal is to be reconstructed
- infinite accuracy of computations

However, when designing digital control systems, these two conditions will eventually become invalid due to issues such as coefficient sensitivity and the limited processing capability of the controller. In a real-time control system, there will often be induced time delays due to the discretization process and computational processes, even though the signal is not yet being reconstructed. Many authors (Nise 2007, Li & Fang 2006) strongly recommend that the time delays should be assumed to be less than the sampling time so that the theoretical development can proceed but, this assumption might not always be true in a given implementation due to the hardware constraints and extensive multiprocessing in the system. Therefore, the selection of the sampling frequency will depend on the nature of the system characteristics, and the overall system requirements.

Furthermore, the finite word length of the processor will truncate the original values of the controller coefficients, ADC and DAC conversions, and computer arithmetic operations that will affect the accuracy performance (Feuer & Goodwin 1996). Other accuracy considerations include the compensator structure, disturbance signals, noise, uncontrollable system modes at high frequencies and the inherent time delays that introduce phase lags in a closed-loop

(Forsythe & Goodall 1991). In essence, *the primary influence on the sampling frequency is the bandwidth characteristics of various signals such as measurement noise, disturbance signals and time delays* (Houpis & Lamont 1985). It is worthy to make a note that the single most important impact of the sample rate in a control system is the delay associated with the reconstruction device.

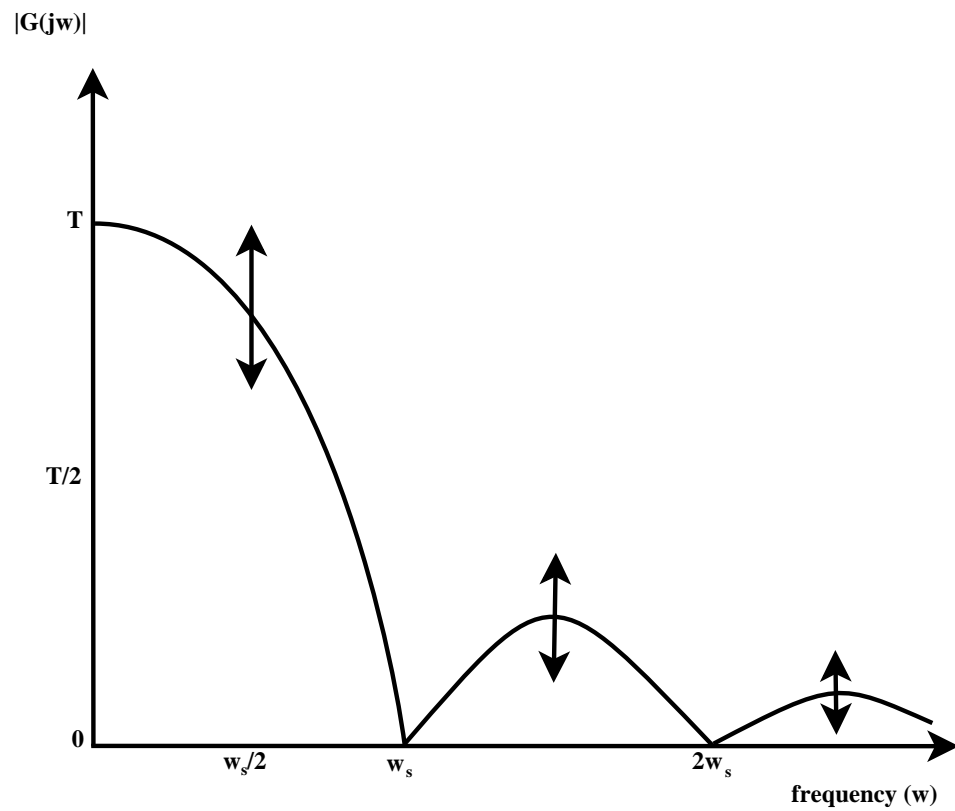
A sampled signal $y[t_n]$ can be converted back into a continuous signal $y(t)$ by the Zero-Order-Hold (ZOH). It is thus important to assess its effects for a given system. Fig. 2.3 depicts the frequency response characteristics of a ZOH. The magnitude response of the device has a characteristic similar to a low-pass filter with a drop in the gain at high frequencies, and the phase of the device is also of concern since it contributes to a lag of $\frac{\pi}{2}$, which degrades the degree of system stability at high frequencies. The phase delay introduced is approximately (Wu 2005):

$$\phi_s \simeq 360f_0 \frac{1}{2f_s} \quad (2.1.4)$$

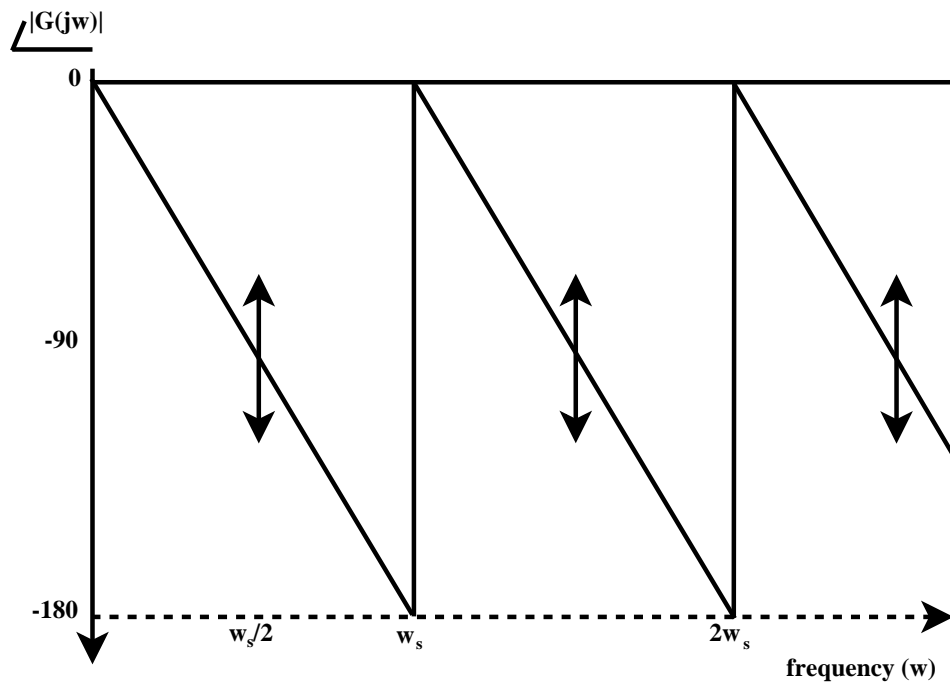
where ϕ_s is the phase delay introduced by sampling, f_s is the sampling frequency and f_0 is the signal bandwidth.

In general, selecting a sample rate solely based on the Sampling theorem is not enough as it will introduce a phase difference which is not satisfactory for real-time control². A common rule of thumb when selecting the sample rate is to start with the highest frequency of interest and use the Sampling theorem to acquire a rough idea of the minimum bound of the sampling frequency required. Then this bound will be increased by a factor of 10, 20 or even up to 100 times the control bandwidth according to the desired performance criterion.

²—i.e., the phase lag will seriously limit the ability to provide a high performance controller.



(a) The magnitude response



(b) The phase response

Figure 2.3: The frequency response of the Zero-Order-Hold.

However, the question to be asked during digital controller executions should be: *What is the optimum sample frequency for the required performance?* Control literature discusses various methods of accomplishing the sampling rate selection. For example, for a system with resonant modes which are higher than the closed-loop bandwidth, the sample rate is sometimes selected to be a multiple of the highest important resonant mode. Johnson (1974) has shown that the appropriate value of this multiple is from 4 to 5. Others sources (Scaechter 1982) used a sample rate of 12 to 15 times the highest important resonant mode frequency. Conversely, Gran & Berman (1974) have suggested that the proper sample rate should be selected based solely on the disturbance effects, independent of the resonant frequency.

Goodall (2001) points out that achieving a very high sampling frequency is not the problem using modern day chips, but rather it is necessary to establish a proper understanding of the recursive processes and design of the computational aspects. The author further discusses the impact of the increasing phase lag in a closed loop, which will degrade the stability margins. Since the computation may in fact take up a whole sample period, the implications on the selection of the sampling frequency become more stringent. For example, Wu (2005) argues that to achieve a phase delay no more than 5° , the corresponding sampling frequency should be increased by 72 times the operating bandwidth of the control system. This criterion should be enough to suffice the control stability requirements and this is why, particular emphasis has to be placed on the numerical requirements needed from the controller.

Furthermore, it should be pointed out the other techniques that are used to compensate for lags/delays, such as forward extrapolation, can aid in reducing the sampling frequency requirements. However, most practical applications may still have unmodelled dynamics at high frequencies e.g. structural

resonance, that will cause problems as the prediction process approaches the sampling frequency. Moreover, Goodall, Jones & Cumplido-Parra (1998) outlined the difficulties of the implementation requirements of digital filters used in control systems and proposed a highly efficient processor core for a wide variety of control applications. Discussing the various types of controller structures and numerical requirements, the authors emphasised the importance of the implementation operators.

The selection of the sampling frequency of a digital control system is usually a compromise among many aspects of the design. The basic motivation for lowering the sample rate is cost (Clarke & Maslen 2007). A slow sample rate directly reduces the hardware costs and makes it possible for a slower computer to achieve a given control function; or provides greater capability for a given computer. The potential disadvantages of slow sampling, relative to controller bandwidth, may lead to open loops between samples³ or a control input with large steps⁴. On the other hand, a very fast sample time can assure stability and performance of a system, based on certain selection criterion that can provide the overall frequency response after the reconstruction process. Strictly speaking, real signals will not have any bandwidth limits, i.e. there always exist small energy components outside the bandwidth (Middleton & Goodwin 1990), but when implementing a digital control system, it is always required to sample higher than the theoretical minimum (Feuer & Goodwin 1996).

³The sampled output will be a poor representation of the actual continuous-time response; inter-sample ripple (Goodwin & DeSouza 1984).

⁴These can feed significant energies to high frequency mechanical resonances.

Clock-driven and event-driven approaches

Two different approaches could be utilised when implementing real-time control systems: clock-based and event-based. The clock-driven approach is a time-based strategy where the control algorithms are executed at predefined times. A periodical sample rate is an example of a time-based approach since it is independent of the signal being sampled. On the contrary, an event-based approach is when the sampling scheme is implemented by specifying adequate conditions when an event or activity occurs. Since most control algorithms usually update their controller outputs at fixed sampling frequencies or are *time-driven*, event-based approaches offer a method for producing the output at a non-uniform rate, where the decisions are made only when significant information has been changed (McCann, Gunda & Damugatla 2004)⁵. It may provide a faster response and hence is much more suitable where the input signal changes dynamically (Astrom & Bernhardsson 2002). Therefore, event-based strategies assist in improving control delays, however, they cater time-varying systems which become difficult to analyse (Nilsson 1998).

More recently, two approaches were compared by Cuenca, Garcia, Arzen & Albertos (2009), for dealing with variable network delays and scarce data availability when the control system is integrated in a network environment. The first approach used a predictor-observer in order to recover lost information and eliminate delays. In addition, an event-based control system was used which sampled signals only when certain events occurred. After several simulation results, the main conclusion was that the event-based sampling approach achieves a shorter settling time than the other. Event-based measurement

⁵Examples of event-driven strategies include adaptive sampling, integral sampling, level-crossing/send-on-delta sampling, etc., which are all signal dependent techniques (Miskowicz 2007).

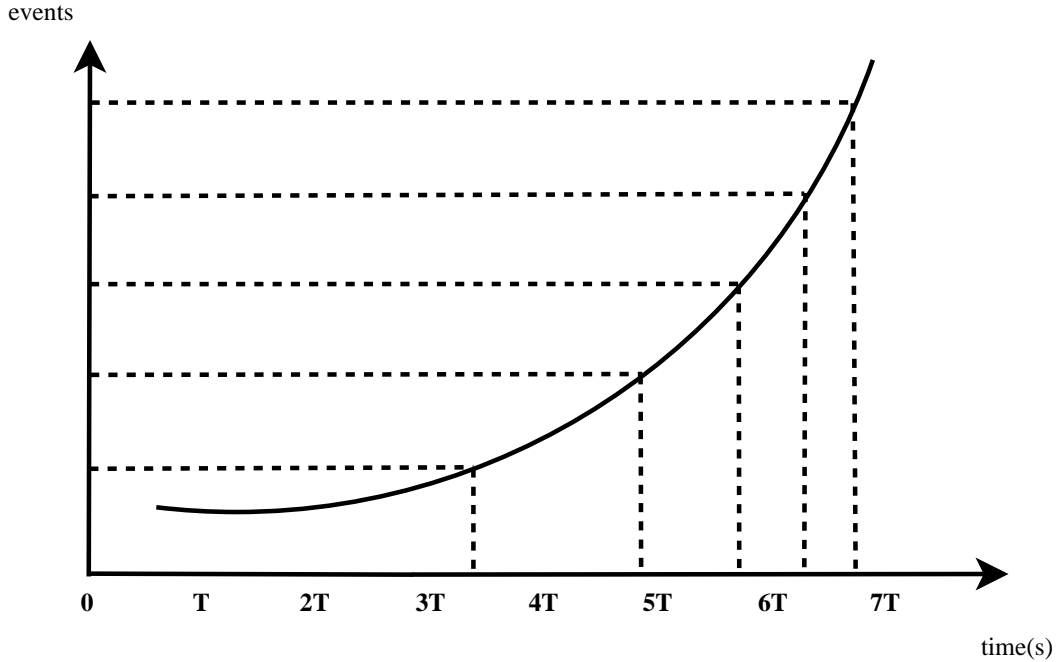


Figure 2.4: The event-based approach

updating methods have also been introduced for discrete Kalman filters to estimate the state feedback (Le & McCann 2007). The method significantly reduced the number of network communications events while contributing towards improved accuracy. Fig. 2.4 illustrates an event based approach scheme in operation. Note that events are not necessarily equally spaced.

2.2 Non-uniform sampling

The basic signal analysis techniques related to digital control have been well defined for several decades and a great deal of research has continued in the areas of implementation accuracy, quantisation effects and many other topics. These studies almost unanimously assume periodic sampling, in which the difference in sample times ($t_k - t_{k-1}$) remains constant, or the mathematical

understandings of sampling periods is uniform.

The minimum sampling frequency is imposed by the Sampling theorem, which states that it is possible to recover the original signal if, *and only if*, its contents are less than half the sampling rate — the Nyquist limit. If this condition is not met, the effect of aliasing or overlapping frequencies becomes discernible. This requirement imposes technological limits to signal processing at higher frequencies and it is often advocated to sample more than twice the maximum bounds, for instance, in applications such as radio communications and cellular phones (Wojtiuk 2000). However, instead of pushing the hardware to its limits, there are other techniques to reduce or even eliminate the adverse effects of aliasing when sampling below the Nyquist rate. Non-uniform sampling addresses the problem of acquiring a set of irregularly spaced samples from a continuous signal $x(t)$ to form an irregular sample set. It has successfully been used to identify the underlying spectrum of a signal, even at substantially lower average sample rates as compared to their uniform sampling counterpart requirements. Though non-uniformity of sampling can exist in various forms, it is often characterised according to some probability distribution:

Truncated gaussian *or normal* distribution To achieve normally (Gaussian) distributed sample times, the sample time fluctuation is added to the sampling system that can be given by:

$$t_k = kT + \tau_k, \quad T > 0, k = 0, 1, 2, \dots, \quad (2.2.1)$$

where $\{\tau_k\}$ is a set of independent and identically distributed random variables with zero mean. Bilinskis (2007), Marvasti (2001), Martin (1998) and many others claim that this type of sampling regime is present in any practical sampling system for reasons such as the phase noise of the sampling clock or

the finite aperture uncertainty within the sampling device and can be used in the analysis of various applications. In most cases the effects of the fluctuations are ignored and hence can contribute to performance degradation of the overall system. Wojtiuk (2000) presented several experiments on this distribution and concluded that increasing the randomness does not appear to have any benefits. In fact, since the Probability Density Function (PDF) is oscillatory in nature, a significant number of samples in the sampling sequence violate the condition $t_n > t_{n-1}$.

Uniform distribution Here the probability that a uniformly distributed random variable falls within any interval of fixed length is independent of the location of the interval itself. Sampling according to sawtooth or triangle waves at predetermined intervals can produce samples times with a uniform distribution.

Another uniform distribution is based on the sampling model proposed by Shapiro & Silverman (1960) called the additive pseudo-random sampling scheme, that can be constructed by adding the sampling fluctuation to the previous sampling time:

$$t_k = t_{k-1} + \tau_k, \quad k = 0, 1, 2, \dots, \quad (2.2.2)$$

where τ_k is a set of independent and identically distributed random variables having a mean μ , and variance σ . The PDF of the sampling instances can be presented as a convolution of the random variables:

$$p_k(t) = p_{k-1}(t) \star p_\tau(t) \quad (2.2.3)$$

where the \star denotes the convolution operation. The mean sampling rate can be given by $p(t) = \frac{1}{\mu}$. Bilinskis & Mikelsons (1992) presented the mathematical description that the sampling sequence eventually approaches a mean sample

rate due to the build up of the variance in the PDF, and that stationarity is achieved more rapidly with a higher value of initial variance. However, too much variance might result in statistical errors and therefore an intermediate value must be chosen that can provide reasonable results.

Other distributions Other sampling schemes include sampling regimes produced by using a sine wave $p(t) = \sin(x)$. The events can be viewed as a vector that is rotating at an angle and moving at a speed varied according to the frequency of the sine wave. Likewise, a digital wave can also be used to produce a sampling regime that comprises two consecutive sample rates.

2.2.1 Developments and applications

Non-uniform sampling comes naturally in many applications, for instance, imperfect sensors, mismatched clocks or event-driven systems. Examples of such can be found in the fields of communications, astronomy, medicine, etc. Yet, most of the literature to a large extent focuses on algorithms and analysis of uniformly sampled data. The past few decades have witnessed an important number of publications that present the diverse possibilities of non-uniform sampling patterns in engineering applications.

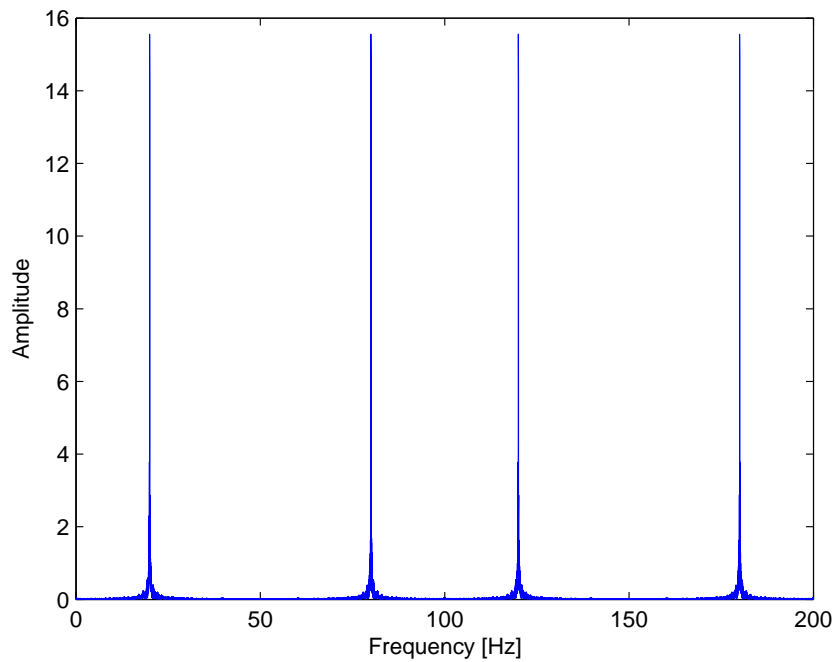
Much work has been devoted for calculating the frequency spectrum and towards studying its statistical properties (Beutler & Leneman 1966, Beutler 1970, Eng 2007, Martin 1998), whose contributions are not only of theoretical importance but very practical as well. Other pioneering works have been recorded by Marvasti (2001) who presents an exhaustive compendium of techniques that make use of non-uniform sampling, in particular the reconstruction aspects. However, it is the work of Bilinskis & Mikelsons (1992) that has triggered the idea to use intentional sample time randomization as the ultimate

economical tool for alias suppression. The authors' efforts have concentrated on enabling the possibility of increasing the frequency range of engineering applications. Fig. 2.5 effectively demonstrates the application of the concept, which is discussed in more detail in Appendix B. Since there is no Nyquist rate associated with the spectral estimation: an *alias-free* sampling scheme in principle permits correct identification of the underlying spectrum regardless of the sample rate (Martin 1998).

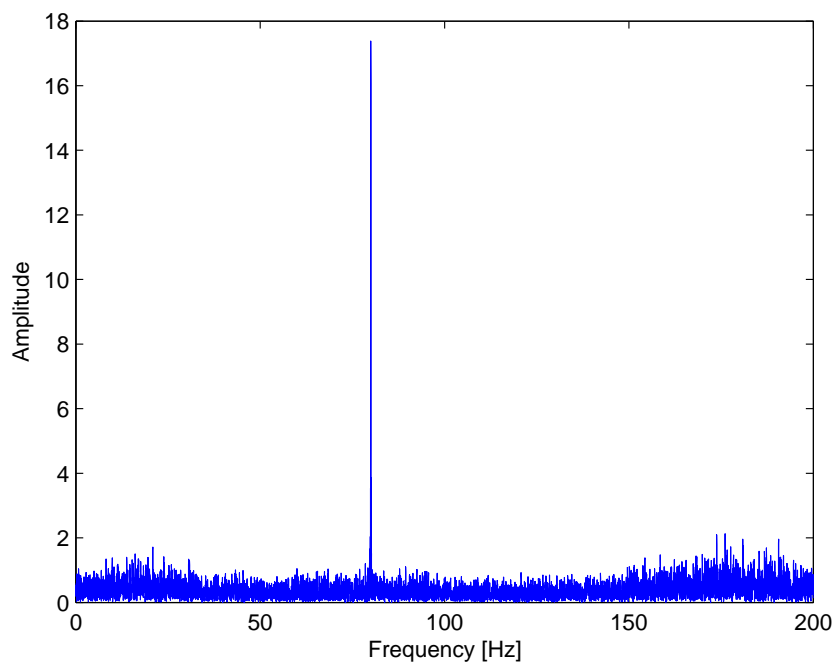
The idea of *alias-free* sampling was first proposed by Shapiro & Silverman (1960). Soon, other investigators (such as Beutler (1970), Masry (1978)) became aware of the concept and extended the idea. However, most of the research was conducted by mathematicians, and hence the study on random sampling was not directly applicable for practical engineering purposes. Moreover, modern scientists often ended up contributing their feats to the theory of sampling and ignoring the strong limitations imposed by the technology and existing analysis techniques (Martin 1998). Yet, the concept is very attractive simply because of the advantages it can offer to the signal processing needs.

A copious review of the literature on sampling theory and methods for analysing signals that have been non-uniformly sampled can be found in Martin (1998). The work has contributed towards various aspects of prediction, filtering, spectrum analysis and the use of non-linear prediction for analysing signals of nonlinear dynamical origin. The author further indicates that classical signal processing techniques such as system identification, convolution and filtering are not the preserves of uniform sampling and can readily be extended in the case with irregular sampling.

Bilinskis & Mikelsons (1992) published several key papers on non-uniform sampling, which has since been investigated for numerous applications. Their efforts has influenced other notable research areas that can demonstrate the



(a) The magnitude response



(b) The phase response

Figure 2.5: (a) The frequency response to a 80Hz sinusoidal signal with a uniform sampling frequency at 100Hz. The aliases are clearly visible and have corrupted the signal information. (b) The frequency response of a 80Hz sinusoidal signal with a non-uniform sampling pattern with an average sampling frequency at 100Hz. The aliases are suppressed and the true signal content is apparent.

applicability of alias-free sampling. This includes the work done by Sonnailon, Urteaga & Bonettot (2008) who proposed the use of random sampling to implement a high frequency lock-in amplifier. The analysis showed that the maximum operating frequency can be several times higher than the Nyquist frequency, and that the maximum frequency is only limited by the minimum time step of the acquisition device. Moreover, Wojtiuk (2000) investigated the application of random sampling schemes to design a radio transceiver. The author analysed the power spectrum with different types of sample distributions and studied the degree of alias suppression achieved. In addition, the signal reconstruction aspects was highlighted and suggestions for a new receiver front end architecture were made.

Principal applications

It should be noted that non-uniform measurement and computing methods are not yet regarded as the ultimate achievement in signal processing. However, the potential contributions have been demonstrated repeatedly by results of successful engineering attempts to use this approach for development of high performance instruments; such as in oscilloscopes (Artyukh, Medniks & Vedin 1997) and ADCs (Papenfuss, Artyukh, Boole & Timmermann 2003).

Various systems have been developed that are capable of processing signals in a frequency range many times exceeding the mean sampling rate. The most popular one is the computer based system of Virtual Instruments called the DASP-Lab Systems, which is a versatile analyser of wideband RF signals (Artyukh, Bilinskis, Greitans & Vedin 1997, Bilinskis 2007). It serves as a demonstrator that can undertake radio frequency signal analysis in the time and frequency domains, with an operating bandwidth ranging up to 1.2GHz

at a mean non-uniform sample rate of 80MS/s⁶. Besides anti-aliasing measures, there is another system aspect improvement that can be resolved using the technique. This includes massive data acquisition from distributed signal sources, making it technically and economically feasible to gather cost-effective data (Artyukh, Bilinskis, Sudars & Vedin 2008). Since the operational environment of the data acquisition system often plays an important role, the sample values are taken at random unpredictable time instants. The obtained sequence of the signal sampling values under these conditions are unavoidably non-uniform and have to be treated as it occurs *unintentionally* as a side effect. Even so, instruments can exploit this and enable the correct processing to resolve the uncertainty.

The main benefit of non-uniform sampling designs for spectral analysis is where direct digital processing of a wide band is desirable. That is why the use of such strategies is generally justified for high frequency applications; i.e to reduce the speed requirements of converters and subsequently, the speed of calculations in the processor. The main disadvantage acknowledged in literature is the presence of a ‘*noise*’ floor in the output spectrum⁷.

2.3 Control systems design

A control system is an interconnection of components or set of devices to manage, command, direct or regulate the behaviour of other devices or systems (Ragazzini & Franklin 1958). Classical control analysis makes use of transfer functions⁸, which are mathematical representation of the relation between the input and output of a *linear time-invariant* system. In the *s*-domain, transfer

⁶—MS/s — Million samples per second.

⁷This is discussed with detail in Appendix B.

⁸—also known as the network function.

functions commonly take the form:

$$H(s) = \frac{n_0 + n_1s + \dots + n_i s^i}{1 + m_1s + \dots + m_i s^i} \quad (2.3.1)$$

where n_i are the numerator coefficients, m_i are the denominator coefficients and i is the order of the function.

Control systems can be considered in two cases: open-loop control where the feedback of the process is not observed to determine if its output has achieved the desired goal. A typical cause-effect relationship in an open-loop (also known as feed-forward loop)⁹.

Conversely, closed-loop control monitors the output process and continuously adjusts the control input as necessary to keep the control error to a minimum. Feedback on how the system is actually performing can allow a controller to dynamically compensate for disturbances to the system. An ideal feedback control system would cancel out all errors, effectively mitigating the effects of any forces that may or may not arise during operation and producing a response in the system that perfectly matches the reference input.

2.3.1 Digital control

Continuous-time control is implemented using analogue electronics. However, if the analogues are replaced by a digital computer within the loop, it becomes a computer-controller system. The controllers may therefore be designed in the discrete-time domain¹⁰. Since a digital computer is a discrete system, the Laplace transform is superseded with the z -transform. Also since a digital

⁹Sometimes closed-loop and open-loop control can also be used simultaneously. In such systems, the open-loop control is termed feed-forward and serves to further improve reference tracking performance.

¹⁰A discrete control system is described schematically in Fig. 1.1 on page 3.

computer has finite precision¹¹, extra care is needed to ensure the error in coefficients, ADC conversion, DAC conversion, etc. are not producing undesired or unplanned effects. Unlike analogue electronics, digital computers cannot integrate. In order to solve the differential equation using a computer, the equation must be approximated by using mapping rules that can reduce the maths to algebraic expressions. These approximations are often referred to as numerical integration rules. In particular, a simple technique to solve the differential equation using a computer is Euler's method. It follows from the definition of a derivative that:

$$\dot{x} = \lim_{\partial t \rightarrow 0} \frac{\partial x}{\partial t} \quad (2.3.2)$$

where ∂x is the change in x over a time interval ∂t . Even if ∂t is not equal to zero, the following relationship will be approximately true, and:

$$\dot{x} \cong \frac{x(k+1) - x(k)}{T} \quad (2.3.3)$$

where

$T = t_{k+1} - t_k$ (sample interval)

$t_k = kT$ (time instant)

k is an integer

$x(k)$ is the value of x at time t_k

$x(k+1)$ is the value of x at time t_{k+1}

This approximation can be substituted in place of all derivatives that appear in the controller differential equations to produce difference equations that can be solved by the computer. Obviously, the faster the sampling, the better the approximation, although a price that is paid in terms of higher computation requirements. Apart from conforming to the Sampling theorem, the

¹¹—quantisation effects.

sampling frequency for discrete-time control systems is often chosen according to one of various rules-of-thumb mentioned in §2.1.1.

2.3.2 Implementation operators

The z -transform The transfer function of a linear, time-invariant and discrete system can be obtained by the method of z -transform analysis. The z -transform is widely used in digital control and has the same role in discrete systems that the Laplace transform has in the analysis of continuous systems. The transfer function for a discrete controller can be obtained by using established $s - z$ mapping rules¹². A popular rule is the bilinear transform¹³ which provides an accurate representation and ensures stability:

$$s = \frac{T}{2} \frac{z + 1}{z - 1} \quad (2.3.4)$$

The Bilinear transform is a first-order approximation that substitutes into the continuous-time transfer function $H(s)$, as $s \leftarrow \frac{2}{T} \frac{z-1}{z+1}$, i.e. $H(s) \approx H(\frac{2}{T} \frac{z-1}{z+1}) = H(z)$. It is a highly preferable technique in digital control since it preserves stability and maps every point of the continuous-time filter frequency response, $H(j\omega)$, to the corresponding points in the digital domain to estimate the discrete-time filter frequency response, $H(e^{j\omega T})$. However, since this is only an approximation, it is likely that the corresponding digital transfer function, $H(z)$, will give similar gain and phase margins in its frequency response only at low frequencies. The differences become quite evident at higher frequencies or close to the Nyquist limit.

The generalised discrete equivalent transfer function in the z -domain for

¹²The resulting discrete transfer function will be similar in form to that of the s -plane transfer function of Eq. 2.3.1 on page 32.

¹³See Schneider, Kaneshige & Groutage (1991) for more mapping techniques.

Eq. 2.3.1 is then defined as:

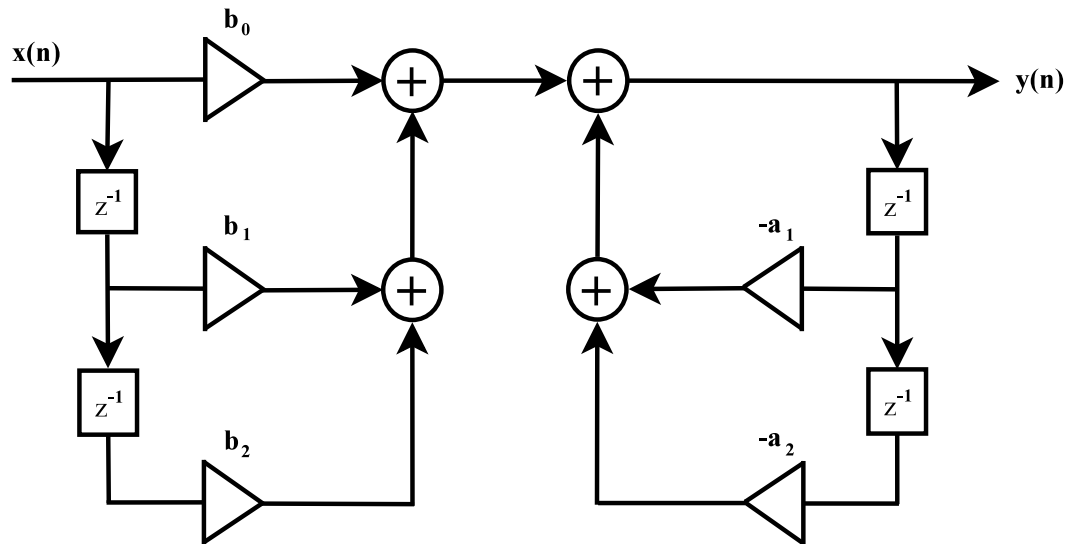
$$H(z) = \frac{a_0 + a_1 z^{-1} + \dots + a_i z^{-i}}{1 - b_1 z^{-1} - \dots - b_i z^{-i}} \quad (2.3.5)$$

where a_i are the numerator coefficients, b_i are the denominator coefficients and i is the order of the function.

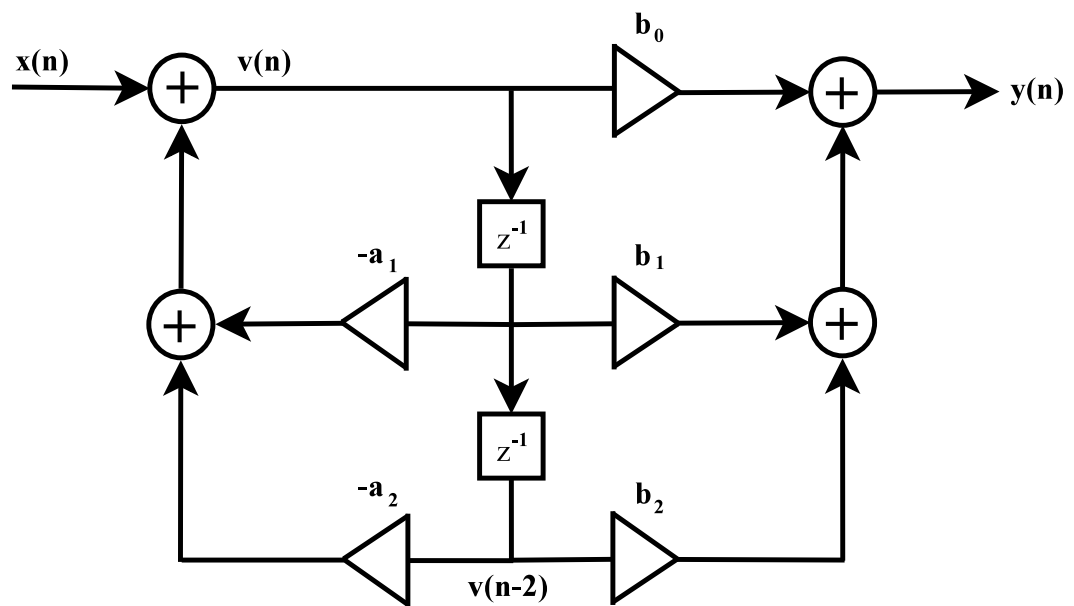
In order to implement the discrete controller using the z -operator, an implementation structure will have to be determined. These structures reflect the ways in which the discrete transfer functions can be interpreted both theoretically and diagrammatically. The most commonly used z -operator implementation methods are the *direct* and *canonical* form's, shown in Fig. 2.6 for a 2^{nd} order filter. It is widely recognised that the canonical form has certain benefits over the direct form since there are fewer stored variables and shift operations and hence is the most popular choice for implementation (Forsythe & Goodall 1991).

However, considerable progress has been made towards ameliorating problems of the z -operator through the use of a δ -transform, that enhances the numerical conditioning of control algorithms at high sample rates. Middleton & Goodwin (1990) argue that that this change in the foundations of the system representation and analysis has the potential of superseding the traditional analysis methods based on the conventional shift operator. Although the choice of the z -operator is quiet natural for control engineers, opting for the opting for the δ -operator can provide:

- better closed-loop stability margins
- low coefficient sensitivity
- significant reduction in the wordlengths for the coefficients



(a) Direct form



(b) Canonical form

Figure 2.6: Structures of a 2nd order digital z -filter.

The δ -transform The difference operator can also be used defined by:

$$\delta = \frac{dx}{dt} = \frac{x(k) - x(k-1)}{T} \quad (2.3.6)$$

where T is a small time difference. The difference application becomes more precise as T approaches 0. Middleton & Goodwin (1990) presented a historical perspective on the use of this operator along with the general system calculus to unify the continuous-time and discrete-time system theories. The δ operator is defined as:

$$\delta = \frac{z-1}{T} \quad (2.3.7)$$

As compared to the z -filter, it seems that the only alteration in the δ -filter implementation equations is that the original ‘shift’ operation have to be replaced by additions. The discrete transfer function in δ can be written in identical form to that of the z , although the coefficient values will naturally be different:

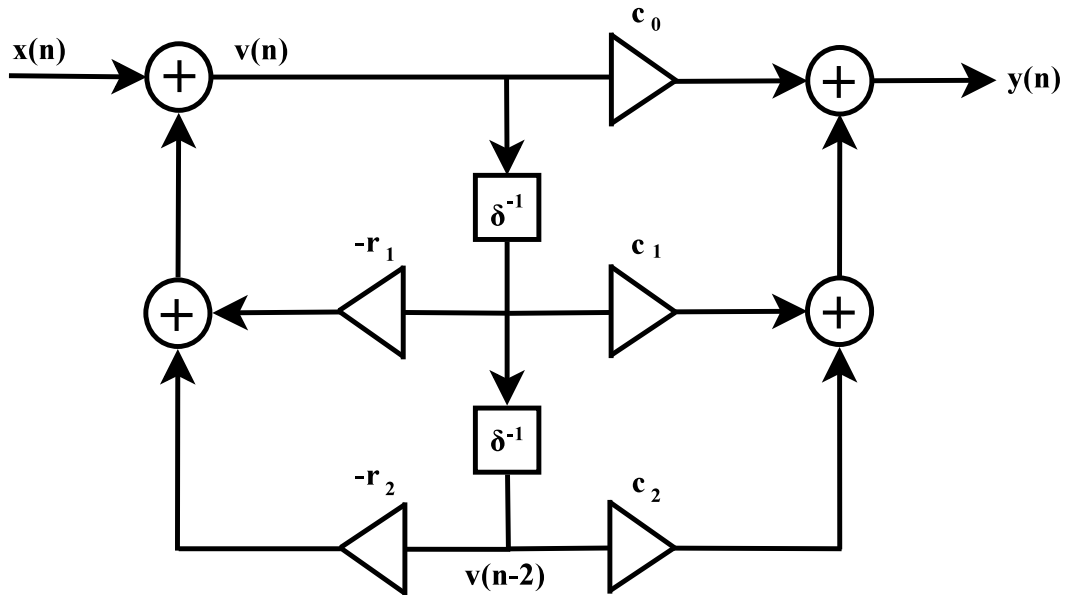
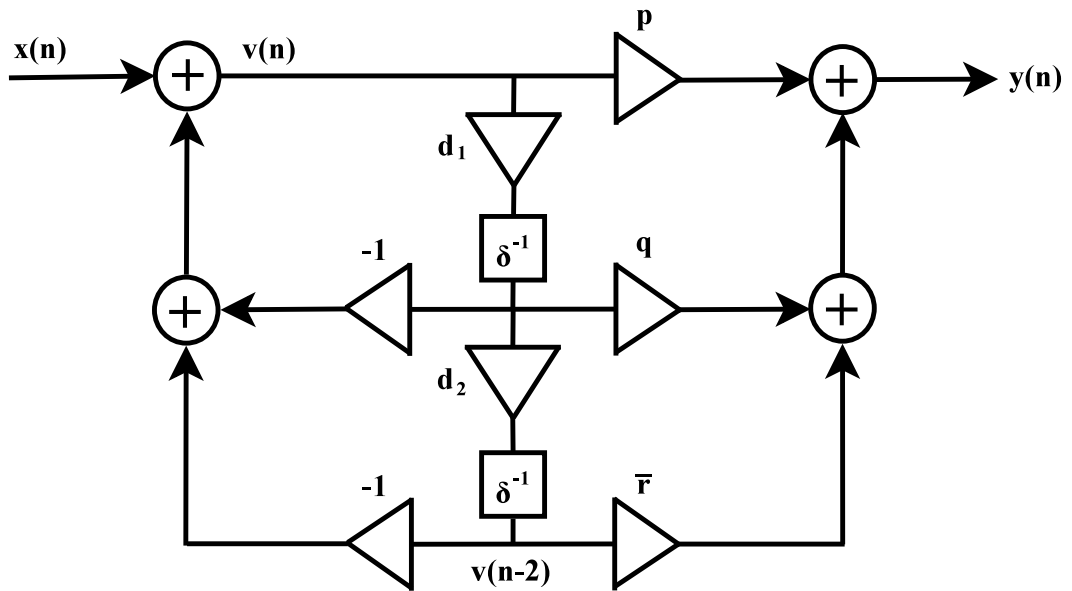
$$H(\delta) = \frac{c_0 + c_1\delta^{-1} + \dots + c_i\delta^{-i}}{1 + r_1\delta^{-1} + \dots + r_i\delta^{-i}} \quad (2.3.8)$$

Forsythe & Goodall (1991) have offered another definition for the δ -operator:

$$\delta = z - 1 \quad (2.3.9)$$

This definition has the important benefit that the feedback coefficients have now been moved to the forward path of the digital filter, with suitable adjustments to the coefficient values. In addition, this modification leads to an extra benefit that some of the numerator coefficients may now be approximated to 0 or 1 in certain situations, resulting in a reduction in the number of multiplications needed. The discrete-time transfer function for the modified canonic δ -filter presented by Forsythe & Goodall (1991) can be written as:

$$H(\delta) = \frac{p_0 + d_1p_1\delta^{-1} + \dots + d_1 \dots d_i p_n \delta^{-i}}{1 + d_1\delta^{-1} + \dots + d_1 \dots d_i \delta^{-i}}, \quad (2.3.10)$$

(a) Canonic δ -filter(b) Modified canonic δ -filterFigure 2.7: Structures of a 2nd order digital δ -filter

where p_n is the last feed-forward coefficient.

The problem with the z -operator is that, high sample rates can result in very long wordlengths when representing the filter coefficient values. The δ -operator however, uses the differences between successive intervals and therefore provides a lower coefficient sensitivity in its representations. Such means of signal analysis and function implementations have shown considerable promise and therefore is expected to perform better under non-uniform sampling regimes. Fig. 2.7 shows the diagrammatic representation of the two δ -filter structures for a 2^{nd} order digital filter. Note that the feedforward coefficients in Fig. 2.7b have been renamed p , q and \bar{r} for simplicity of the 2^{nd} order filter as compared to its generalised definition in Eq. 2.3.10.

2.4 Non-uniform sampling in control theory

A limited number of control applications have been considered with random sampling. However, most industrial control applications are rather forced to use different sampling schemes largely due to the nature of the systems being used and the way information is handled. Therefore, processing irregular sampling correctly requires a synergy between well-designed control algorithms and carefully implemented electronic systems.

For instance, the control signal may be updated at a fixed rate, while the output feedback signal might be measured by different sensors, each one possibly having a different sampling rate, noise characteristics and reliability. In such cases, the control updating rate can be faster than the measurement output rate, inducing discontinuities¹⁴ (Chaumette, Mansard & Remazeilles 2009). The situation also raises in the use of multi-rate control schemes, where

¹⁴In some cases the output might not even be available at every sampling time.

the selection of the sampling rate is based on its suitability to control each controlled variable. This of course, imposes the use of discrete-time models with different sampling periods, even for the same process (Balbastre, Ripoll & Crespo 2000).

Furthermore, situations might appear in networked control systems whose sensors, controllers and actuators are connected through a communications link. The insertion of this *communications element* introduces timing variations in the control loop, and the application of dynamic bandwidth allocation/scheduling techniques may vary the sampling rates for each control loop (Marti, Frigola & Velasco 2005). Wittenmark, Nilsson & Torngren (1995) focused upon the timing issues from a sampled-data point of view. They acknowledged that, while inaccuracies, disturbances, etc. have been extensively treated at the process side, very little work has treated deficiencies in the real-time control system. In addition, Colandairaj, Irwin & Scanlon (2007) suggested a technique in which the sample rate is adapted to network and control parameters. By increasing the sample rate during periods of high traffic, the control and network performance were maintained simultaneously.

Interests in the development of analysis and design of non-uniform sampling is also found in intelligent sensing technology. It essentially uses event-based systems to detect specified events of interest in a sensor field. Miskowicz (2007) discussed an event-driven approach where the input is sampled according to some integral criterion. The work presented an analytical method for estimating the communication bandwidth and the sampling effectiveness for event-based integral sampling.

More recently, Albertos & Crespo (1999) reviewed and proposed improvements in control design techniques under non-uniform sampling schemes mainly

concentrating on scheduling and multitasking timing constraints. They examined industrial perspectives for real-time control with non-uniform sampling and discussed the interaction between control algorithms and their implementations. Their conclusion of the analysis was that due to the complexity of modern control systems, sample update irregularities become inevitable. Therefore, controllers should provide certain timing constraint guarantees to implement task scheduling, or the issues could be solved by the use of additional specific hardware (such as multiprocessors, parallel processing, or digital signal processor units).

Other basic issues that result in non-ideal implementation of digital control algorithms are due to:

- time delays (Marti, Fuertes, Fohler & Ramamritham 2001)
- lost of data samples¹⁵ (Sanchis, Sala & Albertos 1997, Wittenmark, Nilsson & Torngren 1995)
- hardware constraints from the data acquisition mechanisms (Clarke & Maslen 2007)
- various software interactions during event-driven procedures¹⁶(Nilsson 1998)

To deal with such issues in real-time control, the solutions usually consider the following alternatives:

- The sample rate is often increased according to some user defined criterion¹⁷.

¹⁵Such as lost data packets due to errors in the communications medium.

¹⁶—also known as scheduling issues.

¹⁷This is discussed with detail in §2.1.1.

- In the case with multi-rate control systems, it is assumed that, although the sampling rate for all the variables is not the same, the whole system is considered to be periodic to simplify the analysis (Voulgaris & Bamieh 1993).
- Control algorithms can be modified when dealing with random sampling. Linear extrapolation techniques might be used to predict the output at the next sampling instant and compensate for lost information (Bibian & Jin 2000).

2.4.1 The impact of sample time variations

From the above brief, it is evident that the ideology of non-uniformity is by no means new to control theory. Even so, most of the work done is moderately targeted towards compensating the adverse effects that unintended non-uniform sampling might impose on the control system. Concepts of adding intentional variations in the sampling frequency are primarily lacking and hence various methods of employing additional controllers and look-up tables are demonstrated to eliminate the effects of sample variations.

It is possible to adapt to a changing sampling frequency by modifying the digital controller coefficients. This must be done in correspondence to the desired filter characteristics that are to be achieved and essentially requires the discrete operation to preserve the impulse response represented by the continuous filter¹⁸ (Akira 2006).

Recently, several works combining control and scheduling co-design approaches have focused on the *jitter* problem (Marti 2002, Arzen et al. 2000). Here the main goal, again, has been to demonstrate the degradation effects on

¹⁸This concept is covered with detail in Chapter 3.

controlled system performance due to scheduling jitters in control task executions and how to minimise these effects by computing and switching controllers accordingly. In addition, specific tools have been presented for simulation and performance analysis of real-time control systems (Cervin, Henriksson, Lincoln & Arzen 2002). Obviously this adds more constraints onto the digital controllers, and therefore would demand more high speed processors. Surprisingly, none of the above work considered the possibility of intentionally randomizing the sampling rate of a controller.

The added randomness might possibly be utilised for some particular purpose, principally as a result of enabling a lower sampling frequency without compromising the operating bandwidth of the digital compensator, with reductions in the overall processing. Furthermore, it would be valuable if the control design determines and specifies a range of allowable sampling intervals, where larger periods give deteriorated but still acceptable control performance. This would be very useful and open up a new degree of freedom for design of real-time systems. Some work with non-uniform sampling was conducted by Jugo & Arredondo (2007), to design an estimation algorithm for an adaptive vibration controller. The controller updated its parameters for the compensation signals on non-uniform time instances and the resulting signals effectively counteracted the effects of any perturbations.

Either way, as far as the author can tell, there has been no research reported that investigates the opportunities of using intentional non-uniform sample rates for feedback control systems. More recently, Bilinskis (2007) coined together the term '*alias-free signal processing*' which demonstrates the reality of expanding the frequency range of a signal without being corrupted by aliasing. The implications of the phrase *alias-free* are that it is possible to determine information at frequencies well in excess of the average sample

rate. The corollary for control, perhaps, is that the effect of sampling delays could be reduced such that the average sampling frequency does not need to be as high¹⁹.

2.5 General remarks

- Phase lag issues are acknowledged to be the predominant reasons for causing instability in control systems. As a result, the sample rates are increased to ensure stability at high frequencies.
- Intentional random sampling offers potential in reducing the high sample rate requirements for signal processing. It is able to recover the almost complete spectral contents of an under-sampled signal, even if the average sample rate is below the Nyquist criterion. However, it has not yet been used in control theory due to a lack of design and analysis methods.
- Some non-uniform sampling concepts in digital control are encountered in relation to event-based sampling. However, such strategies are predominately used to reduce the amount of processing required²⁰ and do not necessarily improve the quality/performance of overall control process.
- Control theory norms: Control algorithms are inherently designed for uniform sampling. Therefore, an applicable algorithm (or filter structure) must be identified that can perform the control operations correctly.

¹⁹The idea has been discussed in Goodall (2001).

²⁰—by reducing the number of samples being processed.

- Evaluation techniques: Appropriate methods must be developed that can allow the identification of the frequency response of a non-uniform rate control system.

2.6 Summary

Digital control theory deals with real time control issues. Nowadays, the use of powerful design tools often lead to solutions with numerical problems. Despite the fact that the obtained control structures, their parameters and their operational properties are derived through standard mathematical procedures, the physical meaning of the actions are often hidden which makes it difficult to interpret and change unexpected behaviours. The tendency to generalise to a generic property by employing rules of thumb could lead to wrong actions, and the selection of the sample rate is often at the forefront of such rules. It is well known that in digital control, the faster sampling rate is not always the best one, due to numerical and resolution problems (Forsythe & Goodall 1991). Thus, the selection of the sample rate is a trade-off between loss of information²¹ and the processing capability of the digital system.

This work investigates the application of applying intentional variations in the sampling frequency of digital controllers. Non-uniform sampling has largely been part of mathematical research and theories, but it has gained much attention due to the added advantages it can provide. Nonetheless, the field is still very immature and has yet to highlight the border line of its preference over uniform sampling.

Evidently, certain signal processing applications have embraced the random sampling nature of their process operations and have used it for their benefit.

²¹Which increases the phase lag in the loop.

Random sampling has been noted as a useful tool to improve the quality of numerical computation of signals for tracking problems by adapting the sample rate to the frequency content in the signal at that time. Moreover, non-uniform sampling presents new possibilities of processing signals without the stringent restrictions of the Nyquist limits.

From a controls perspective, processing controller data at irregular times poses a complex challenge but it may ensure a good compromise between the control performance and algorithmic computations. Since conventional algorithms are exclusively designed to process uniformly sampled signals, it is worth exploring its impact on the operating bandwidth of digital controllers.

Chapter 3

Continuous-time Transfer Function *Emulation* with Non-uniform sampling

Chapter overview

Various applications in digital systems require the involvement of concepts from signal processing and filtering. These specific problems often need the linear dynamic systems to have a transfer function that can specify the behavioural characteristics of the system. When operating in the digital domain, such functions can effectively be used to approximate the same characteristics over the frequency range of importance as any given continuous-time transfer function. Therefore, this chapter aims to explore the possibilities of implementing non-uniform rate transfer functions and begins the detailed description for control system development. It highlights the adopted approach in this work for non-uniform sampling and control co-design, in order to build an effective real-time controller. The objectives of this chapter are to:

- further investigate the issues identified in Chapter 2, *the impact of sample time variations*

- establish design assumptions for implementing non-uniform rate control systems
- identify and examine the controller design techniques that can be utilised and subsequently, the experimental work is carried out
- discuss the limitations of implementing the developed non-uniform rate algorithm
- realise a convenient method for estimating the frequency response of non-uniform rate control systems
- establish the performance evaluation criteria

Instinctively, the first technique of interest is the topic of numerical methods with varying step sizes for integration of differential equations. Ordinary differential equations have been the subject of much study for a very long time and numerous methods with many refinements have been derived for their numerical solution. However, an alternative approach can often be used which, though by no means new, has been subject to less sustained examination. This is to replace the n^{th} order differential equation by the n^{th} order difference equation through the use of mapping rules. This will be looked at in detail and the major limitations that reside with the implementation modes of the non-uniform rate transfer functions are identified and their solution is demonstrated through examples.

This chapter is organised as follows: §3.1 introduces relevant design techniques for incorporating non-uniformity in discrete-time transfer functions. These techniques are based on classical controller design methods used in control theory. The chapter then progresses on to identify the limitations that non-uniform sampling adds during implementation of digital controllers.

However, the effects of these limitations can be reduced and the solutions are provided in §3.3. §3.4 demonstrates a simple but important technique for calculating the frequency response of non-uniform rate control systems, the accuracy of which can be measured by some performance criterion defined by the designer. Finally, a DC motor controller example is provided that is designed, implemented, simulated and analysed using the tools and techniques outlined in the chapter.

3.1 Design of discrete equivalents

The primary design methods for discretization of linear continuous-time transfer functions include the standard z transformation or *impulse invariant* method, the *bilinear* transform and related transformation methods, and the matched z transform techniques. All these methods are essentially algebraic in nature and differ primarily only in the details of the approximation from the continuous-time domain to the discrete. They can be used to yield a digital filter approximation for a continuous-time linear system, that is of the recursive form¹.

Although highly-developed theories on control design are available for LTI systems, both in continuous-time and discrete-time cases (Nise 2007, Houpis & Lamont 1985), these would be insufficient to deal with loops that experience time variations within its controller actions. Historically, the time-domain formulation is the most commonly used method for performing a time-varying analysis due to its simplicity. Although, such analysis may not provide an

¹—i.e., the calculation of the next output value depends not only on the present and a select few of the past values of the input but also on the previous values of the output.

accurate answer since the convolution rule does not exist in the case with time-varying systems (Wang 2008). Most of the literature on sampled-data control therefore is not applicable for non-uniform rate transfer function design.

An obvious obstruction to the development of the non-uniform rate theory are the added time variations. According to Freudenberg, Middleton & Braslavsky (1995), most of the control literature implies that transfer functions cannot effectively be used to describe the input-output properties in such cases. However, if the Laplace transform of the response and the time instants are available, then it might be possible to redesign the controller in the discrete domain according to the changing time instants.

3.1.1 Numerical integration methods

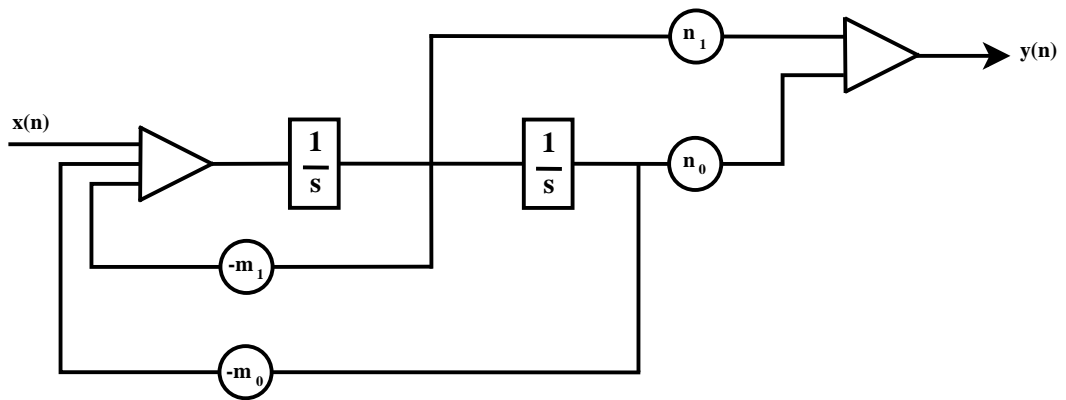
It is a fundamental concept to represent a continuous-time transfer function as a differential equation and then to derive a discrete-time *or difference* equation from it. Many numerical approximates exist in literature that are able to calculate the discrete-time solution of a differential equation, e.g. *Euler's* method, *Tustin's rule*², etc. Consider the 2^{nd} order continuous-time transfer function:

$$\frac{y(s)}{x(s)} = \frac{n_0 + n_1 s}{m_0 + m_1 s + s^2} \quad (3.1.1)$$

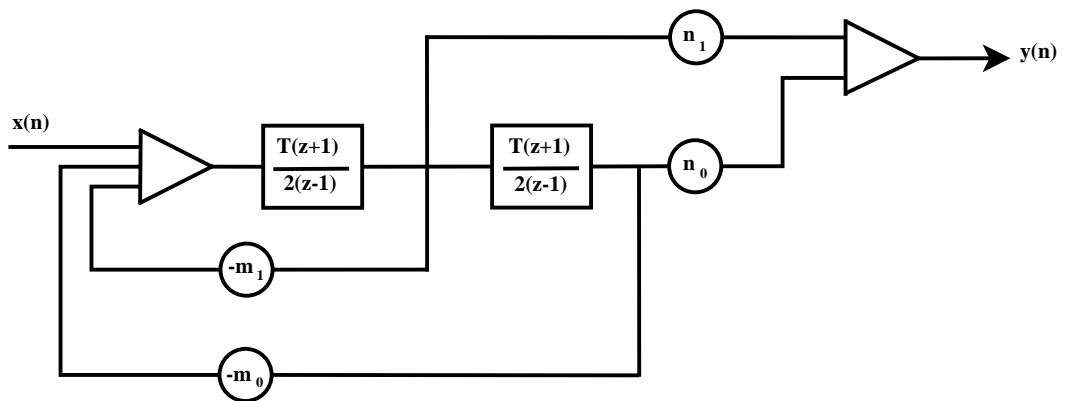
Forsythe & Goodall (1991) remark that the structure of Eq. 3.1.1 can be expressed as in Fig. 3.1a, where x_n and y_n are the input and output signals, respectively, n_0 , n_1 & n_2 are the feed-forward coefficients, m_0 & m_1 are the feedback coefficients and the $\frac{1}{s}$ is the integration operation in the filter.

Rabbath & Hori (2001) have discussed a similar setup for obtaining the discrete equivalent representation of a continuous-time controller by replacing

²—also known as *Trapezoidal* integration or in engineering terminology '*the Bilinear transform*' as discussed on page 33.



(a) A continuous-time filter



(b) A discrete-time equivalent filter using the bilinear transform

Figure 3.1: Continuous and discrete-time representation for a 2^{nd} order transfer function

the integrator term of Fig. 3.1a by a numerical integration rule (as illustrated in Fig. 3.1b which uses the bilinear transform). Any numerical integration rule can be used for this purpose. This *direct approach* is obviously valid for both uniform and non-uniform step sizes. Furthermore, the filter coefficients in both situations will remain the same since they are independent of the sampling interval.

Even though this is a very convenient way to look at the problem for general purpose applications, poor accuracies result in the discrete equivalent, unless the sampling interval is chosen sufficiently small. For instance, a stable continuous-time transfer function may go unstable if discretised with the *direct approach* using a large sampling interval. This is due to the fact that the discrete integrator gain does not depend on the controller parameters.

Another approach is based on working in the z -domain. It is demonstrated by Albertos & Salt (1990), where a classical PID controller is considered with non-uniform time steps. Acknowledging that a continuous-time control action of a PID controller can be given as:

$$u(t) = K_p \cdot e(t) + K_d \cdot \frac{d}{dt}e(t) + K_i \cdot \int_0^t e(\tau) d\tau \quad (3.1.2)$$

the discrete-time equivalent can be described as:

$$u_k = K_p \cdot e_k + \frac{K_d}{T_n} \cdot (e_k - e_{k-1}) + K_i \cdot T_n \cdot \sum_{j=0}^{k-1} e_j \quad (3.1.3)$$

the generalised difference equation in the discrete domain with a constant sample period T_n takes the form:

$$u_k = u_{k-1} + q_0 \cdot e_k + q_1 \cdot e_{k-1} + q_2 \cdot e_{k-2} \quad (3.1.4)$$

where the subindex k stands for time kT_n , i.e. $u_k = u[kT_n]$. The discrete-time coefficients q_i , if applying an *Euler's* approximation, can be expressed by

making substitutions in Eqs. 3.1.3:

$$\begin{aligned}
 q_0 &= K_p + \frac{K_d}{T_n} \\
 q_1 &= -K_p - 2 \cdot \frac{K_d}{T_n} + K_i \cdot T_n \\
 q_2 &= \frac{K_d}{T_n}
 \end{aligned} \tag{3.1.5}$$

The discrete-time coefficient equations of Eqs. 3.1.5 indicate their dependence on the sampling period due to the T_n factor. Simple formulae can hence be derived from continuous-time equations to allow the discrete-time controller coefficients to be updated directly in case the sampling period changes. Therefore, when the sampling is irregular, i.e. it takes place at the time sequence $\{\dots, t_{k-2}, t_{k-1}, t_k \dots\}$, the discrete-time coefficients can be recalculated by using the following generalised discrete-time coefficient equations:

$$\begin{aligned}
 q_0 &= K_p + \frac{K_d}{t_k - t_{k-1}} \\
 q_1 &= -K_p - \frac{K_d \cdot (t_k - t_{k-2})}{(t_k - t_{k-1}) \cdot (t_{k-1} - t_{k-2})} + K_i \cdot (t_k - t_{k-1}) \\
 q_2 &= \frac{K_d}{t_{k-1} - t_{k-2}}
 \end{aligned} \tag{3.1.6}$$

Likewise, when the sampling becomes periodic/constant, Eqs. 3.1.6 will compute the same coefficient results of Eqs. 3.1.5 i.e. of a uniform discrete-time transfer function.

3.1.2 Mapping techniques

The Laplace transform makes it much easier to solve linear differential equations. It essentially converts the equations into an algebraic expression, which can easily be manipulated. In the design of digital control systems, it is possible to go through the design process entirely in the discrete domain, often using mapping rules, to realise the discrete-time transfer function $F(z)$. The process of converting the Laplace equation $F(s)$ to its discrete equivalent $F(z)$ is sometimes referred to as *emulation*, which is usually regarded as a viable approach, whenever the sampling rates are kept high. The resulting discrete-time equation will be arranged in a similar manner to that of the original continuous-time equation, apart from the coefficients values, which will plausibly be different. The discrete-time filter takes the form as described by Eq. 2.3.5 on page 34: Consequently, the design method presented in this chapter adopts the philosophy of emulation, but performs the transition from s to z in such a manner as to make due allowance for variations in the sampling period.

3.1.3 Transient management for control systems

Adaptive filtering is a promising solution for systems where digital filters with time-invariant conditions are often inappropriate. Such techniques can improve the filtering operation by accommodating coefficient variations in order to retain the continuous-time filter characteristics. However, such reconfigurations often cause undesirable transients which may (momentarily) degrade the overall system performance.

Transients usually occur when a digital filter switches from one operational mode to another³. The phenomenon appears as a damped oscillatory

³Such as a change in the coefficient values.

motion, that persists for a short while after the switch happens. An introductory example that demonstrate this effect is shown in the Example A.

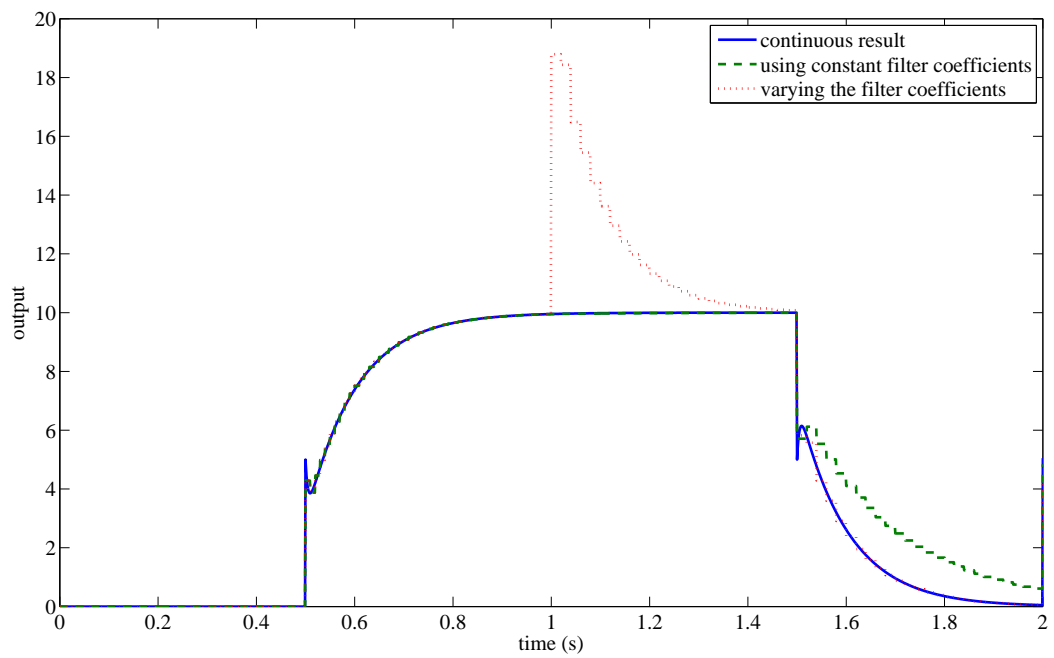
Example A: Transient analysis

The following is an emulation of a practical phase lag-lead compensator based on IIR filtering, where the filter coefficients are switched just once at $Time = 1s$. The compensator has the transfer function:

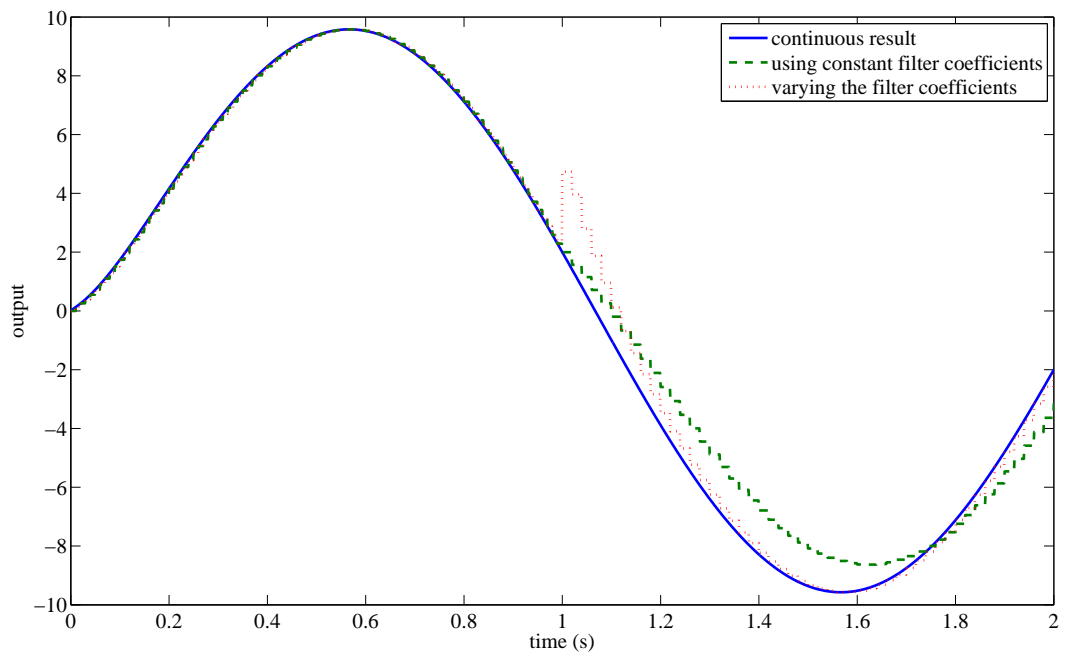
$$H(s) = \frac{10 + 0.35s + 0.0025s^2}{1 + 0.105s + 0.0005s^2} \quad (3.1.7)$$

The filter makes use of the canonic z -filter structure for implementation. The digital filter coefficients of Eq. 3.1.7 are changed according to the sample rate parameter, T_n , as listed in Table. 3.1. The filter stability is maintained for all chosen sample rates, e.g. in this case, the filter will remain stable for both sample rates being used in the experiment. For simplicity in this demonstration, the sample rate is changed just once during operation, at $Time=1s$. This means that two filter coefficient sets will be used during the simulation, set-1 from $0s \rightarrow 1s$ (where $T_n = 0.01s$), and set-2 from $1s \rightarrow 2s$ (where $T_n = 0.02s$). In addition, the simulation will also attempt to account for non-uniform sampling with constant coefficients i.e. the digital filter coefficients remain unchanged under the non-uniform sampling conditions. These coefficients are calculated using the sampling interval $T_n=0.01s$, and applied to the filter.

Since the step response of a LTI system is well understood, it is an adequate tool to demonstrate the transient effects from a digital filter. The step response can readily be calculated from its corresponding differential or difference equation. The reason for the unwanted transients is



(a) The digital filter output to a step input



(b) The digital filter output to a 0.5Hz sinusoid input (amplitude = 1)

Figure 3.2: Demonstrating the transient phenomenon (filter structure: z -canonic)

Discrete coefficients	$T_n = 0.01s$ 0s→1s	$T_n = 0.02s$ 1s→2s	$T_n=0.01s$ 0s→2s
a_0	4.285714	4.242424	4.285714
a_1	-4.285714	-1.818182	-4.285714
a_2	0.952381	0	0.952381
b_1	-0.904762	-0.484848	-0.904762
b_2	0	-0.272727	0

Table 3.1: Accommodating for sample period change

as follows: the output of a discrete-time filter can be considered as the sum of different components. One is the response to the input, while the others are the responses due to the internal variables. When there is a change in the sample rate, the internal variables produce an output that is incorrect for the new set of coefficients⁴. These conditions then generate an impulse response from the output of the storage device to the output of the filter. Fig. 3.2 is a pictorial description of the effects that a change in the sample rate will cause to the output of the filter. It can be observed in Fig. 3.2a that with a step input, the switch has significantly affected the digital filter's output response by causing an incorrect transient in the output response at $Time=1s$. Fig. 3.2b shows a similar behaviour on the output response for a 0.5Hz sinusoid input (amplitude=1).

Comparing the case of holding the digital filter coefficient values constant under non-uniform sampling conditions also presents the motivation for using the non-uniform rate control algorithm. It is clear from the simulation that by not accounting for the variations in the sampling regime not only affects the desired characteristics of compensator but can also deteriorate its performance. However, in order for the non-uniform rate

⁴—dynamically or in steady state.

control algorithm to operate correctly, the transient issue demonstrated in this example must be addressed.

It should be noted that if implemented in the correct way, such transient phenomenon will not occur in the case of non-recursive filters (Tarczynski, Valimaki & Cain 1997, Valimaki & Tarczynski 1996). Furthermore, a recursive time-varying filter can be *transient-free* only when its feedback coefficients are kept unchanged throughout the whole process. Zetterberg & Zang (1988) presented a solution to this problem based on the assumption that images of recursive filters are running for each coefficient set that is ever encountered in the system, but only one of them is connected to the output at one time. However, this approach requires a very large number of filters running in parallel which makes it increasingly complex. In practice, this is not computationally viable and Valimaki, Laakso & Mackenzie (1995) further suggested modifications to this method and presented a solution for transient suppression that could give an acceptable performance at a reasonable implementation complexity. However, using these techniques to resolve the non-uniform sampling being considered in this work will be insufficient due to the continuous variations in the time periods.

Some more transient management techniques are discussed by (Simon, Kovacsazy & Peceli 2002, Valimaki & Laakso 1998), such as smoothing techniques that can reduce the transient at the price of a much longer transition interval, or whenever possible, using look-up tables and make changes to the internal state variables of the digital filter, because the transients are due to the mismatch of these variables before and after reconfiguration. Again, such solutions are valid only for cases where the abrupt changes in the parameters

during the operation are limited. The important questions that arise here are:

- What will happen to the digital filter output when the sample frequency changes at every consecutive sample period?
- How can these undesirable transients be eliminated/suppressed during run-time?

Despite the fact that there are some problems of reconfiguration transients reported in the audio signal processing communities, very few research reports exist on strategies for eliminating or reducing these transients in time-varying systems. An approach to utilize the structure dependence of transients (see §3.3.1), is, to the author's knowledge, largely missing in control literature. Therefore, it must be established that transient management is an important aspect to be considered with context to this work and two factors that can effectively be used to influence its behaviour are:

- the proper run-time transient management techniques as mentioned in this section (Simon et al. 2002)
- the implementation structure, as discussed in §3.3

3.2 Non-uniform rate discrete equivalents

This section derives an algorithm to process a discrete-time controller with non-uniform sampling. If the sampling becomes uniform, the algorithm will give the same result as with the conventional uniform rate transfer function.

A typical control setup is composed of a number of elements. The input-s/outputs are continuous-time signals, with the outputs being generated by

passing the inputs through the analogue plant and sensor dynamics. Hold devices (such as DAC converters) are then used to maintain a continuous signal driving the plant.

The controller design is usually done in the continuous plane and later, once a sampling frequency is selected, the discrete-time equivalent is computed and used to replace the continuous-time design. The problem with a time-varying sampling frequency poses an obvious question: *Can the z -transform equivalent be used in a time varying sampling frequency?* This section therefore attempts to answer this question.

3.2.1 General assumptions

Due to the variations in the sample period, several simplifying assumptions are made in order to obtain a more tractable system model for analysis purposes. The primary assumptions made are as follows:

- Sensor dynamics are negligible
- Plant dynamics are described by linear, time invariant differential equations
- All sampling occurs simultaneously and independent of the input signal

The first two assumptions are not restrictive in any way, but are primarily for subsequent notational convenience. An LTI plant places the same restrictions on the analysis that exist for all classical control system techniques. However, the assumption of simultaneous, signal-independent sampling is somewhat of a restrictive approach, but it is crucial to the development of the theory to follow (Marti 2002).

Opting to use the z -transform technique for analysing continuous-time filters with a periodic sample frequency can provide a ready access to the system response characteristics and stability margins as the control gains and filter coefficients. But, due to the involvement of the random time variable, the transform might not necessarily exist. Although, if the sample time instants are known before hand, the transfer function characteristics can be adjusted by recalculating its coefficients in the z -transform according to the changing sampling frequency.

3.2.2 Mathematical formulation

For the sake of clarity, the arguments presented here are built up around specific examples of differential equations and subsequently generalised. The problem under discussion is the representation of a given differential equation by a difference equation. The equations for calculating the coefficients of a digital filter can be generalised in the digital domain. The approach makes use of the difference equations rather than the z -transform, though it is often convenient to express the results in term of z . Although the concept can be generalised, for simplicity consider the 2^{nd} order differential equation:

$$y + m_1\dot{y} + m_2\ddot{y} = n_0x + n_1\dot{x} + n_2\ddot{x} \quad (3.2.1)$$

where x and y denote $x(t)$ and $y(t)$ are the control and dependent variables, respectively; m_1, m_2, n_0, n_1 and n_2 are the continuous-time coefficients; $\ddot{y}, \dot{y}, \dot{x}$ and \ddot{x} are the derivatives. It must be noted that although this section considers a 2^{nd} order equation, the procedure can be generalised for an n^{th} order equation.

Choice of digital algorithm

Eq. 3.2.1 can also be written as transfer function in the Laplace domain as:

$$\frac{y(s)}{x(s)} = \frac{n_0 + n_1s + n_2s^2}{1 + m_1s + m_2s^2} \quad (3.2.2)$$

Applying a mapping rule (such as the Tustin's rule) to Eq. 3.2.2 and rearranging the equation according to Eq. 2.3.5 on page 34, the difference equation can be represented arbitrarily in discrete form as:

$$y[t_k] = x[t_k] \cdot a_0 + x[t_{k-1}] \cdot a_1 + x[t_{k-2}] \cdot a_2 - y[t_{k-1}] \cdot b_1 - y[t_{k-2}] \cdot b_2 \quad (3.2.3)$$

where x and y are functions of time

Computation of coefficients a_i , b_i and implementation

The coefficients equations must be generalised to accommodate any changes in the sample time value. This means that the filter coefficients will have to be recalculated accordingly⁵:

$$a_0 = \frac{n_0 \cdot (T_n)^2 + 2 \cdot n_1 \cdot T_n + 4 \cdot n_2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2}$$

$$a_1 = \frac{n_0 \cdot (T_n)^2 - 2 \cdot n_1 \cdot T_n - 4 \cdot n_2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} + \frac{n_0 \cdot (T_{n-1})^2 + 2 \cdot n_1 \cdot T_{n-1} - 4 \cdot n_2}{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} + 4 \cdot m_2}$$

$$a_2 = \frac{n_0 \cdot T_n \cdot T_{n-1} - n_1 \cdot (T_n + T_{n-1}) + 4 \cdot n_2}{T_n \cdot T_{n-1} + m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2}$$

$$b_1 = \frac{(T_n)^2 - 2 \cdot m_1 \cdot T_n - 4 \cdot m_2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} + \frac{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} - 4 \cdot m_2}{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} + 4 \cdot m_2}$$

⁵The derivation of the coefficient equations is shown in Appendix D.1.

$$b_2 = \frac{T_n \cdot T_{n-1} - m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2}{T_n \cdot T_{n-1} + m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2} \quad (3.2.4)$$

With a constant sample rate value, Eqs. 3.2.4 will compute the same coefficient results as the uniform sampling equations. The implementation equations for the canonic z -filter can be expressed as:

$$\begin{aligned} v_0 &= u_{t_k} - b_1 \cdot v_1 - b_2 \cdot v_2 \\ y_{t_k} &= a_0 \cdot v_0 + a_1 \cdot v_1 + a_2 \cdot v_2 \\ v_2 &= v_1 \\ v_1 &= v_0 \end{aligned} \quad (3.2.5)$$

where y_{t_k} & u_{t_k} are the output and input variables, respectively, and v_0 , v_1 & v_2 are used for internal variable shifting of past input and output values.

It is worth mentioning that, as the *order* of the discrete-time filter increases, the coefficient calculations will need to take the preceding sample rates into consideration. For example, assuming the non-uniform time sequence $\{\dots, t_{k-2}, t_{k-1}, t_k, \dots\}$, a 3^{nd} order filter will have to consider the sample rate values $\{T_n, T_{n-1}$ and $T_{n-2}\}$ to calculate the correct digital filter coefficients that will be used to produce the output result.

Validation of the algorithm

After deriving the generalised equations that calculate the coefficients, the next step is to answer: *How to verify the ‘functionality’ of the non-uniform rate sampling algorithm?* As mentioned earlier, in the case when the sampling becomes uniform, the non-uniform rate sampling algorithm should give the same result as with the conventional uniform rate transfer function.

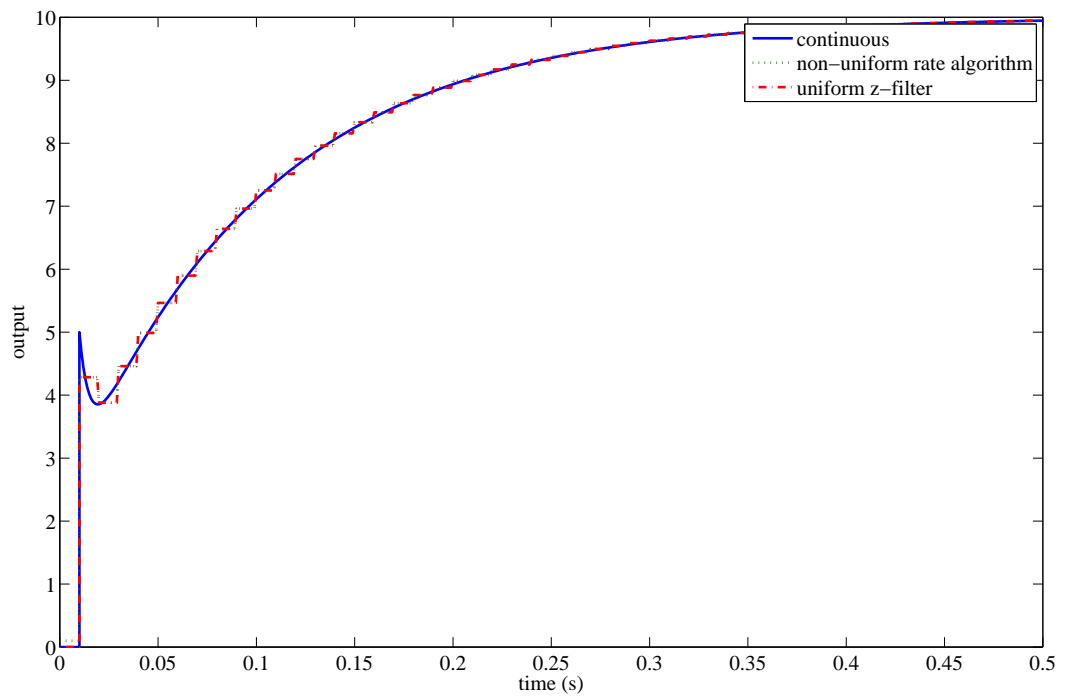
Example B: simulations with the Non-uniform rate algorithm

This example considers the same digital phase lag-lead controller used earlier in Example A on page 54. The following simulations will illustrate the output responses for uniform and non-uniform sampling.

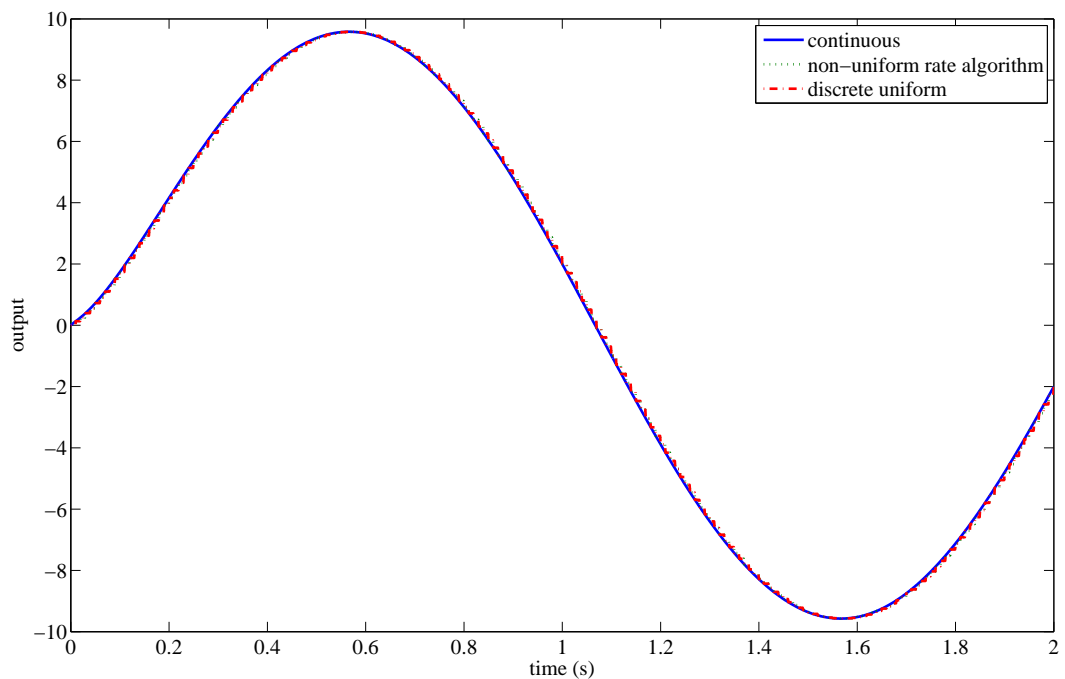
With uniform sampling

The primary aim for these simulations is to validate the non-uniform sample rate algorithm. Since the sampling rate is periodic, the algorithm should produce an identical result to that with the uniform sampling algorithm. During implementation, the digital filter coefficient values will remain constant due to the uniform sample period. Simulations are carried on with a uniform sample period of $T_n=0.01\text{s}$ (100Hz), with a step input and a 0.5Hz sinusoid input (amplitude = 1).

Observing the results of Fig. 3.3 produced with the uniform sample rate, it is established that the algorithm is operating correctly and outputs valid results.



(a) The digital filter output to a step input



(b) The digital filter output to a 0.5Hz sinusoid input (amplitude = 1)

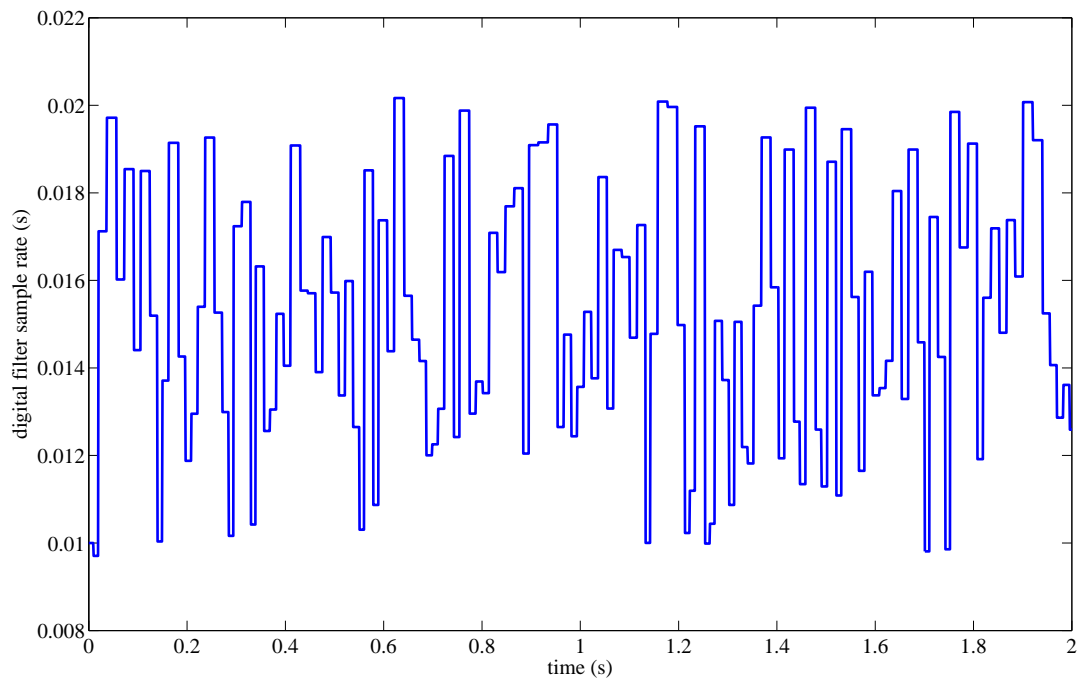
Figure 3.3: Non-uniform sampling algorithm validation (uniform sample rate T_n : 0.01s). Note that, this simulation utilises the algorithm without sample time variations in the sampling period; filter structure: canonic- z

With non-uniform sampling

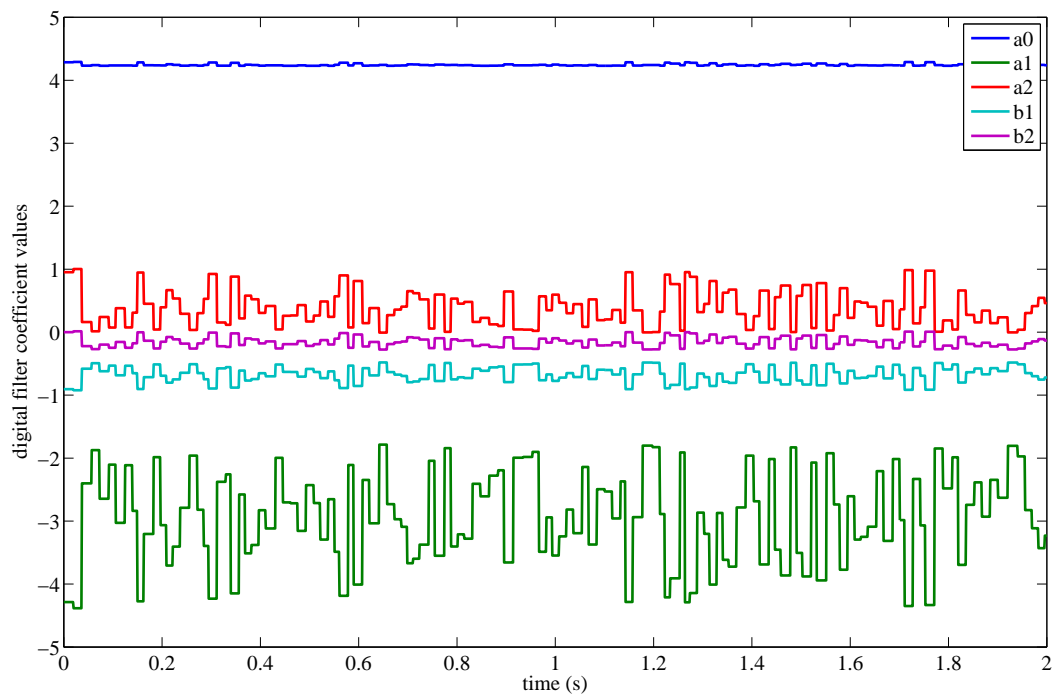
Going a step further in the analysis, consider the case when a time varying sampling scheme is used. Once again, the simulation will make use of the phase lead-lag compensator of Eq. 3.1.7. The sampling scheme under consideration belongs to a *uniformly distributed* set of samples: $T_n \sim U(0.01, 0.02)$, $0.01 \leq T_n \leq 0.02$ and uses a different sample rate value for each consecutive sample period. Fig. 3.4a depicts the filter sample rate values being used over time. The average sampling rate is 0.015s (66.67Hz).

Fig. 3.5 depicts when non-uniform sampling is applied to the digital filter. Clearly, the discrete response has been corrupted and hence it is concluded that although the non-uniform sampling algorithm operates correctly with uniform sampling, the transient phenomenon has a deleterious effect on it during implementation⁶.

⁶—this effect was expected; the digital filter response is clearly corrupted by the transient phenomenon that was discussed in §3.1.3. The reason for such an effect is due to the recursive nature of IIR filters. In the case with non-uniform sampling, the discrete-time filter coefficient values keep changing accordingly. The output result is then calculated based on incorrect internal variables that depend on different delays. Such variations hence introduce unexpected changes within the digital filter, causing disturbances to the output.

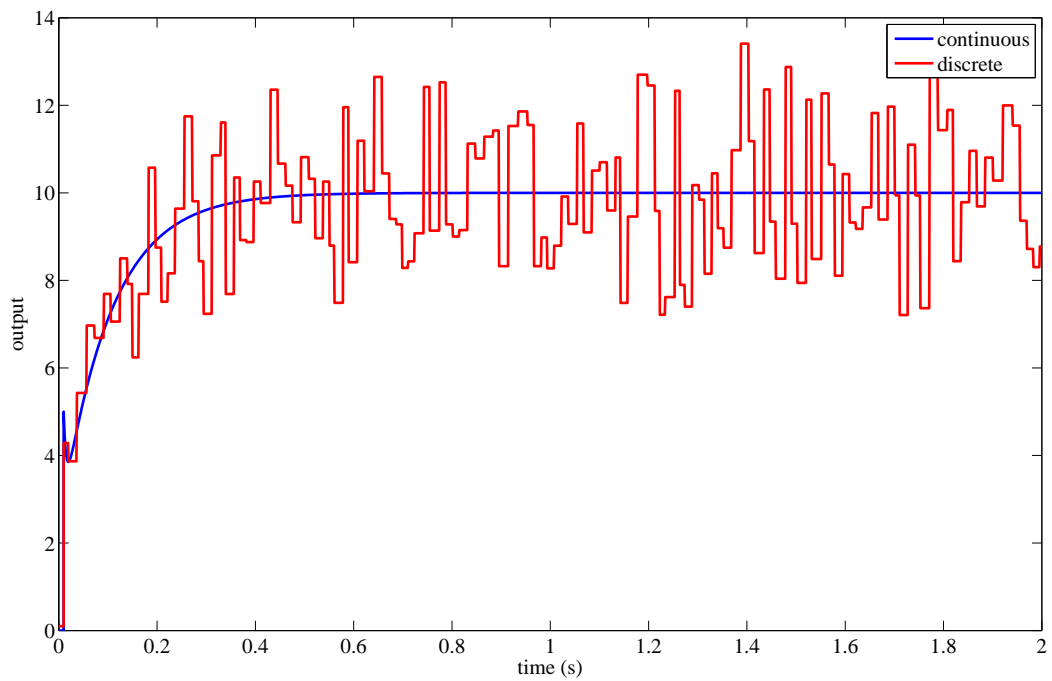


(a) The sample rate of the digital filter changes for consecutive sampling periods

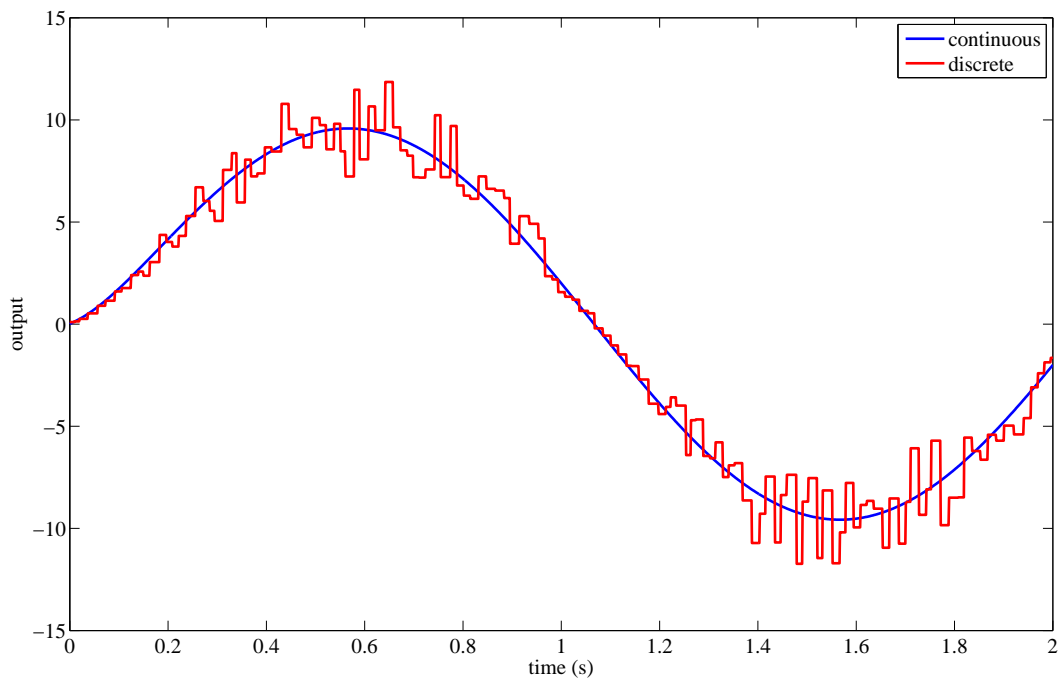


(b) The digital filter coefficient values change depending on the sample rate

Figure 3.4: Accommodating for sample period change (uniform distribution with an average sample rate: 0.015s)



(a) The digital filter output to a step input



(b) The digital filter output to a 0.5Hz sinusoid input

Figure 3.5: The transient phenomenon with non-uniform sampling (uniform distribution ($T_n \sim U(0.01, 0.02)$, $0.01 \leq T_n \leq 0.02$) with an average sample rate: 0.015s); filter structure: canonic- z

3.2.3 Limitations on design

The major limitations identified when designing digital controllers with non-uniform sample times are the reconstruction aspects and the issue with unwanted transients during IIR filter implementation.

As cited in signal processing literature, a bandlimited signal can be reconstructed from its non-uniformly spaced samples provided the average sampling rate is at least the Nyquist rate⁷. However, in most of the published methods, the algorithms derived provide a fast and numerically robust reconstruction by utilising FIR filters with a number of taps (Johansson & Lowenborg 2004). Furthermore, they require a modest amount of over-sampling to achieve high accuracy. Other reconstruction algorithms published over the last two decades cannot be used in real-time implementation and hence are not applicable for real-time control purposes. Marvasti (2001) gives a good review of reconstruction techniques, but in order for a low-pass reconstruction to be successful, some sort of minimal (average) sampling rate will be required. However, the principal aim of this thesis has been to develop methods for analysing non-uniformly sampled control systems and hence signal reconstruction is not of prime importance due to the inclusion of an *over-sampling* factor with digital controllers⁸.

The other limitation is the existence of unwanted transients when trying to implement IIR filters in real-time with non-uniform sampling. This effect becomes increasingly critical during the implementation of higher order filters, where the difference of the prior sample times result in incorrect gain values in the interval variables. The next section briefs on a convenient method

⁷This theory is primarily based upon the work done by Shannon, C. on non-uniform sampling (Marvasti 2001).

⁸See discussion in §2.1.1 on page 17.

for suppressing the transients during digital implementation when using non-uniform sampling, essentially by utilising an appropriate filter implementation structure.

3.3 Suppressing the transient effects

The coefficients of digital controllers depend on the sampling interval being used in the process. With periodic sampling, these coefficients are usually calculated just once at the start of the control program execution. However, when a non-uniform rate sampling frequency is employed, the filter coefficients are updated every time the sample interval changes. This is needed to ensure that the discrete-time filter retains the desired characteristics set out during the analogue filter design. In the case of recursive filter implementations with non-uniform sampling, the output signal suffers from a transient phenomenon that can corrupt the filter response. This is because in such a non-uniform setup, the filter is loaded with its internal variables based on previous coefficient sets. In addition, the severity of a transient signal depends on the filter input signal and the size of magnitude change in the filter coefficients.

3.3.1 The importance of implementation structure

The simulation results from Example A & B show that the usual canonic- z implementation structure, often used in digital control, is unsuitable for non-uniform sampling. Therefore, other possibilities need to be examined to ensure the correct implementation. The significance of choosing the right structure for the purpose of transient reduction was first documented by Kovacshazy, Peceli & Simon (2001). Using the proper structure for the controller realization

can aid in suppressing transients in the steady-state of the system, and some structures can assure smaller transients than others for small disturbances.

Consider the δ -operator which is discussed on page 36. It provides a much superior performance over the shift law implementation. Middleton & Goodwin (1990) recognised that the application of the δ -operator can lead to reliable and robust numerical algorithms. Since the internal variables in the delta structure are no longer successive values of the same quantity, the operation is rather an accumulation of the previous values with the new values.

Canonic δ -form Recall Eq. 2.3.8 on page 36 of the canonic δ -filter. It can be represented in 2^{nd} order as:

$$H(\delta) = \frac{c_0 + c_1\delta^{-1} + c_2\delta^{-2}}{1 + r_1\delta^{-1} + r_2\delta^{-2}}$$

The corresponding generalised coefficient equations for recalculating a 2^{nd} order canonic δ -filter with non-uniform sampling periods are⁹:

$$\begin{aligned} c_0 &= \frac{n_0 \cdot (T_n)^2 + 2 \cdot n_1 \cdot T_n + 4 \cdot n_2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} \\ c_1 &= \frac{2 \cdot n_0 \cdot (T_n)^2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} + \frac{2 \cdot n_0 \cdot (T_{n-1})^2 + 4 \cdot n_1 \cdot T_{n-1}}{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} + 4 \cdot m_2} \\ c_2 &= \frac{4 \cdot n_0 \cdot T_n \cdot T_{n-1}}{T_n \cdot T_{n-1} + m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2} \\ r_1 &= \frac{2 \cdot (T_n)^2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} + \frac{2 \cdot (T_{n-1})^2 + 4 \cdot m_1 \cdot T_{n-1}}{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} + 4 \cdot m_2} \\ r_2 &= \frac{4 \cdot T_n \cdot T_{n-1}}{T_n \cdot T_{n-1} + m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2} \end{aligned} \quad (3.3.1)$$

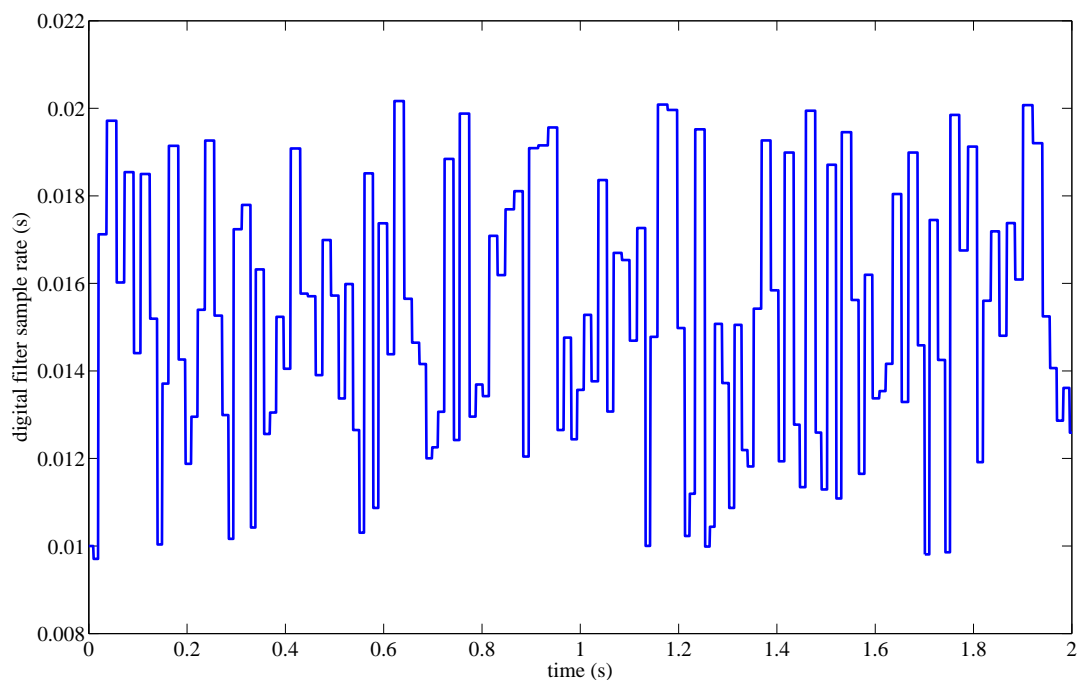
⁹The derivation of the coefficient equations is shown in Appendix D.2.

The implementation equations are written as:

$$\begin{aligned}
 v &= u_{t_k} - r_1 \cdot w - r_2 \cdot x \\
 y_{t_k} &= c_0 \cdot v + c_1 \cdot w + c_2 \cdot x \\
 x &= x + w \\
 w &= w + v
 \end{aligned} \tag{3.3.2}$$

where u_{t_k} and y_{t_k} are the input and output signals respectively, and v , x & w are the three internal variables that perform the delta operations.

The simulation setup utilised in Example A & B is repeated with the canonic δ -filter. The non-uniform sample rate has a uniform distribution $T_n \sim U(0.01,0.02)$, $0.01 \leq T_n \leq 0.02$. Fig. 3.6 illustrates the digital filter coefficient variations depending on the sample rate. The time responses to a step & 0.5Hz sinusoid input (amplitude = 1) are shown in Fig. 3.7. Once again, the *switching transients* have a significant and unwanted effect. Hence, it is concluded that this structure is not suitable for implementation with non-uniform sampling. The reason for this is due to the fact that the static/dynamic values of the internal variables are still dependent upon the sample period and coefficient sets.

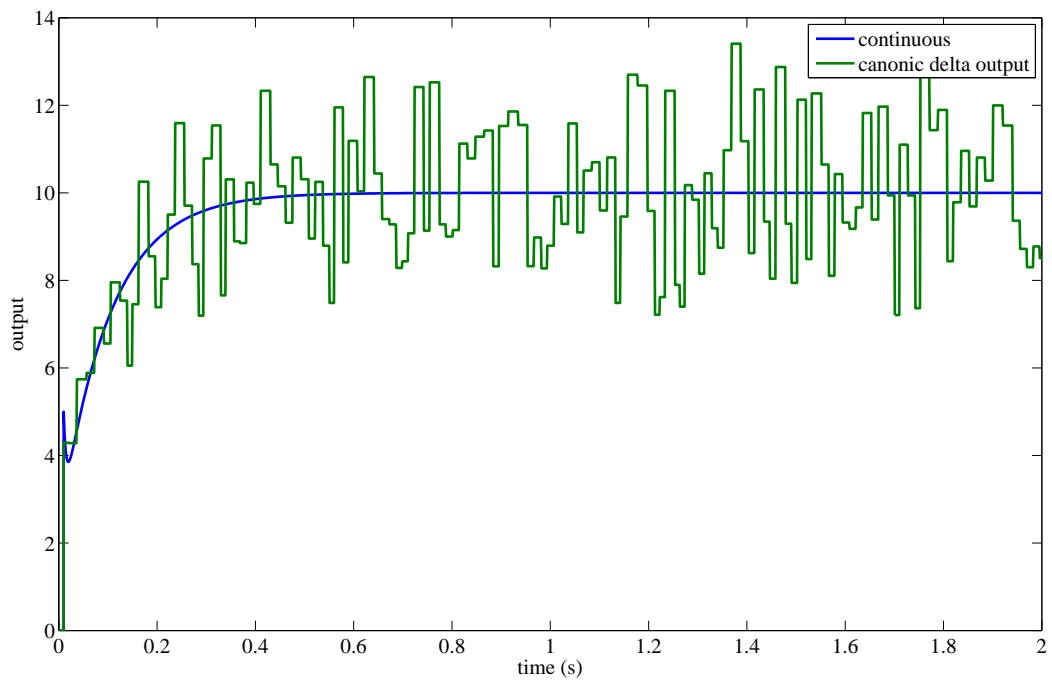


(a) The sample rate of the digital filter changes for consecutive sampling period, $T_k \sim U(0.01, 0.02)$

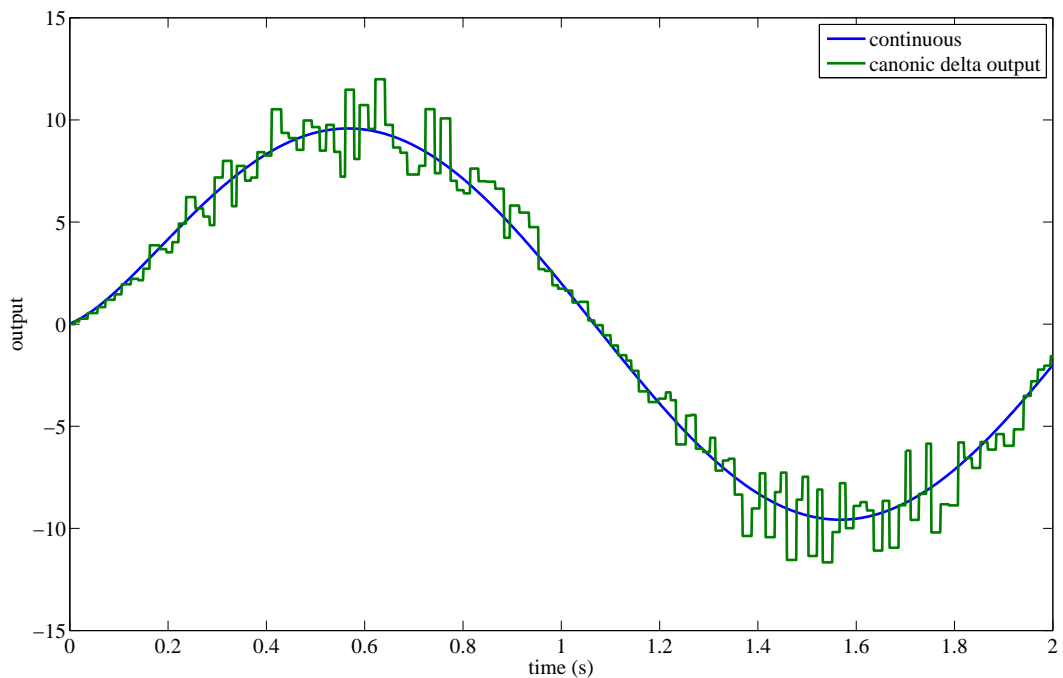


(b) The digital filter coefficient values change depending on the sample rate for the canonic δ -filter

Figure 3.6: The non-uniform sampling being used belong to a uniform distribution $T_n \sim U(0.01, 0.02)$, $0.01 \leq T_n \leq 0.02$, average sampling frequency at 0.015s



(a) The digital filter output to a step input



(b) The digital filter output to a 0.5Hz sinusoid input

Figure 3.7: The filter output for non-uniform sampling (uniform distribution $T_n \sim U(0.01,0.02)$, $0.01 \leq T_n \leq 0.02$, average sampling frequency at 0.015s); filter structure: canonic- δ

Modified canonic δ -form Recall Eq. 2.3.10 on page 36, which described how a discrete-time transfer function can be written in the modified canonic δ -filter form. It can be represented in $2^n d$ order as:

$$H(\delta) = \frac{p + d_1 q \delta^{-1} + d_1 d_2 \bar{r} \delta^{-n}}{1 + d_1 \delta^{-1} + d_1 d_2 \delta^{-n}}$$

where \bar{r} is the last feed-forward coefficient.

The corresponding generalised coefficient equations for recalculating a 2^{nd} order modified canonic δ -filter with non-uniform sampling periods are computed using the substitutions:

$$\begin{aligned} p &= c_0 \\ d_1 &= r_1 \\ q &= \frac{c_1}{d_1} \\ d_2 &= \frac{r_2}{d_1} \\ \bar{r} &= \frac{c_2}{d_1 \cdot d_2} \end{aligned} \tag{3.3.3}$$

where c_0, c_1, c_2, r_1 & r_2 are computed using Eq. 3.3.1

The implementation equations are written as:

$$\begin{aligned} v &= u_{t_k} - w - x \\ y_{t_k} &= p \cdot v + q \cdot w + r \cdot x \\ x &= x + d_2 \cdot w \\ w &= w + d_1 \cdot v \end{aligned} \tag{3.3.4}$$

where u_{t_k} and y_{t_k} are the input and output signals respectively, and v, x & w are the three internal variables that perform the delta operations.

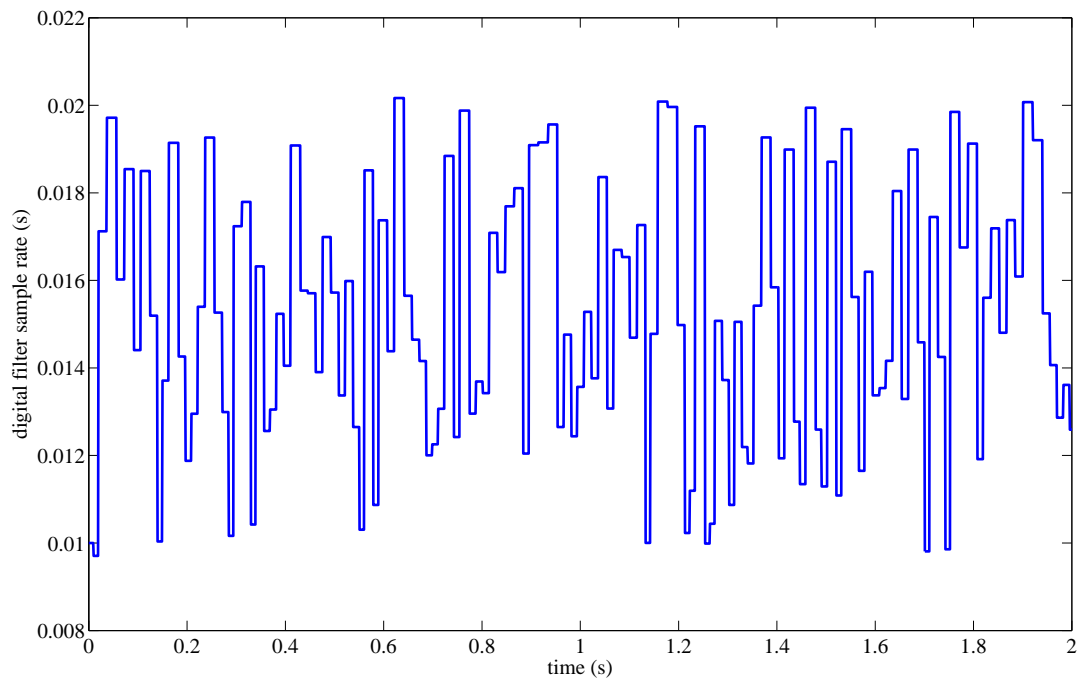
Figs. 3.8 and 3.9 demonstrate the simulations done with the modified canonic δ -filter structure. The non-uniform sample rates have a uniform distribution $T_n \sim U(0.01, 0.02)$, $0.01s \leq T_n \leq 0.02s$. It is remarked that this

structure performs better than the canonic δ -filter in the case with non-uniform sampling, as it is able to suppress the unwanted transients during operation. This is largely because in this case, the state variables of this structure are independent of the coefficient values (look at the filter structure diagram in Fig. 2.7b on page 37).

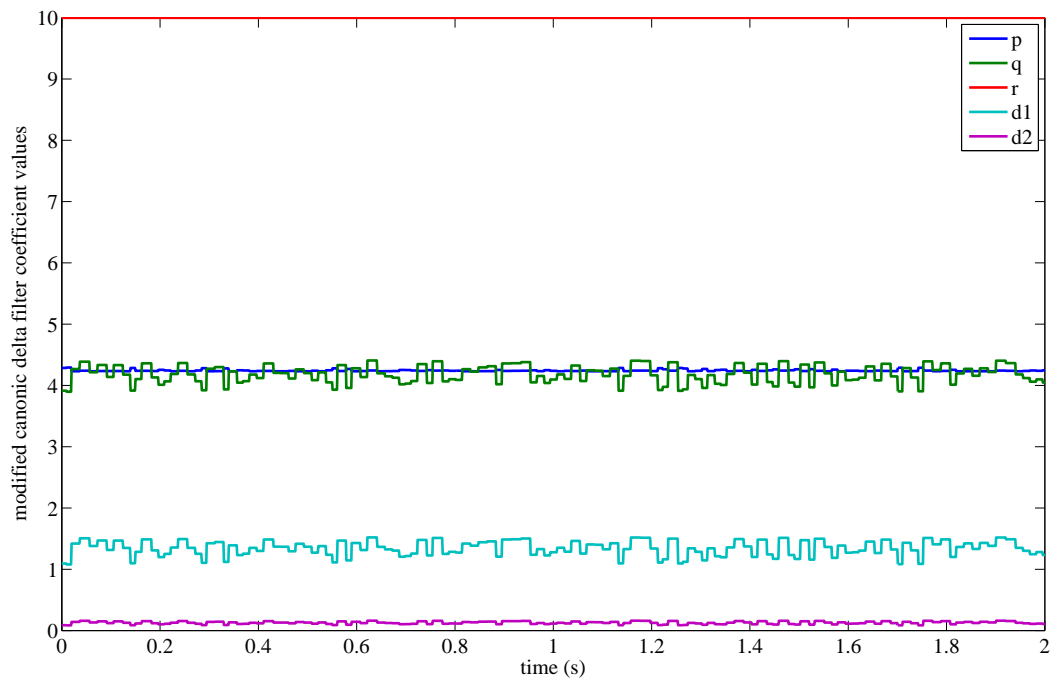
Direct- z form The direct z structure makes use of the same coefficient equations of Eq. 3.2.4 on page 62 to compute its filter coefficient values. The filter implementation, on the other hand, is performed with:

$$\begin{aligned}
 y_t &= a_0 \cdot u_{t_k} + a_1 \cdot x_{t-1} + a_2 \cdot x_{t-2} - b_1 y_{t-1} - b_2 y_{t-2} \\
 x_{t-2} &= x_{t-1} \\
 x_{t-1} &= u_{t_k} \\
 y_{t-2} &= y_{t-1} \\
 y_{t-1} &= y_t
 \end{aligned} \tag{3.3.5}$$

These equations show how to compute the next output sample, y_t , in terms of the past outputs, y_{t-1} & y_{t-2} , the present input, u_{t_k} , and the past inputs, x_{t-1} & x_{t-2} . The simulation results are illustrated in Figs. 3.10 and 3.11. Observing the output response, the direct z -structure performs better than its z -canonic counterpart and can be used for processing non-uniform sample rates. The reason is due to fact that the stored variables are only the previous input and output values, which do not depend upon the sample period and coefficient sets.

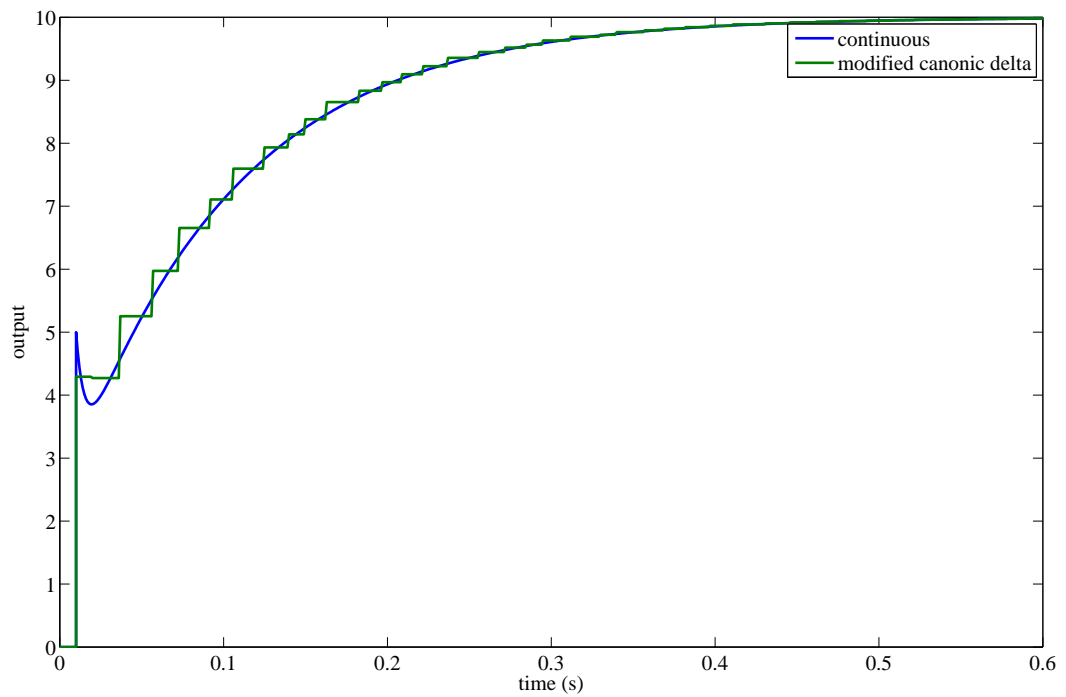


(a) The sample rate of the digital filter changes for consecutive sampling period, $T_k \sim U(0.01, 0.02)$

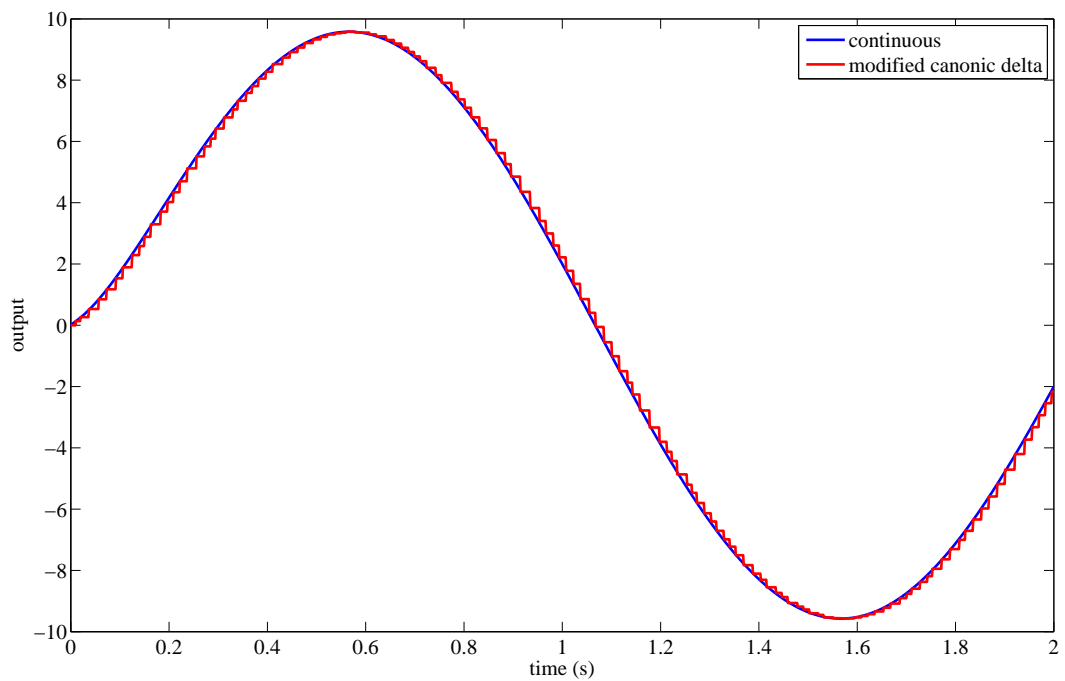


(b) The digital filter coefficient values change depending on the sample rate for the modified canonic δ -filter

Figure 3.8: The non-uniform sampling being used belong to a uniform distribution $T_n \sim U(0.01, 0.02)$, $0.01 \leq T_n \leq 0.02$, average sampling frequency at 0.015s

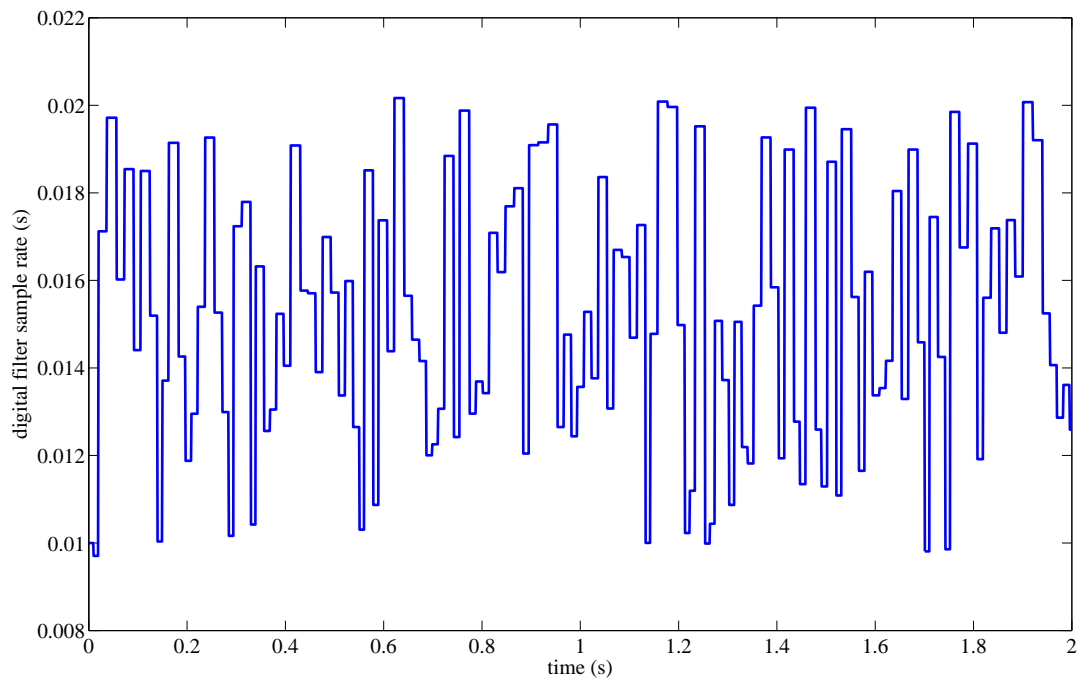


(a) The digital filter output to a step input

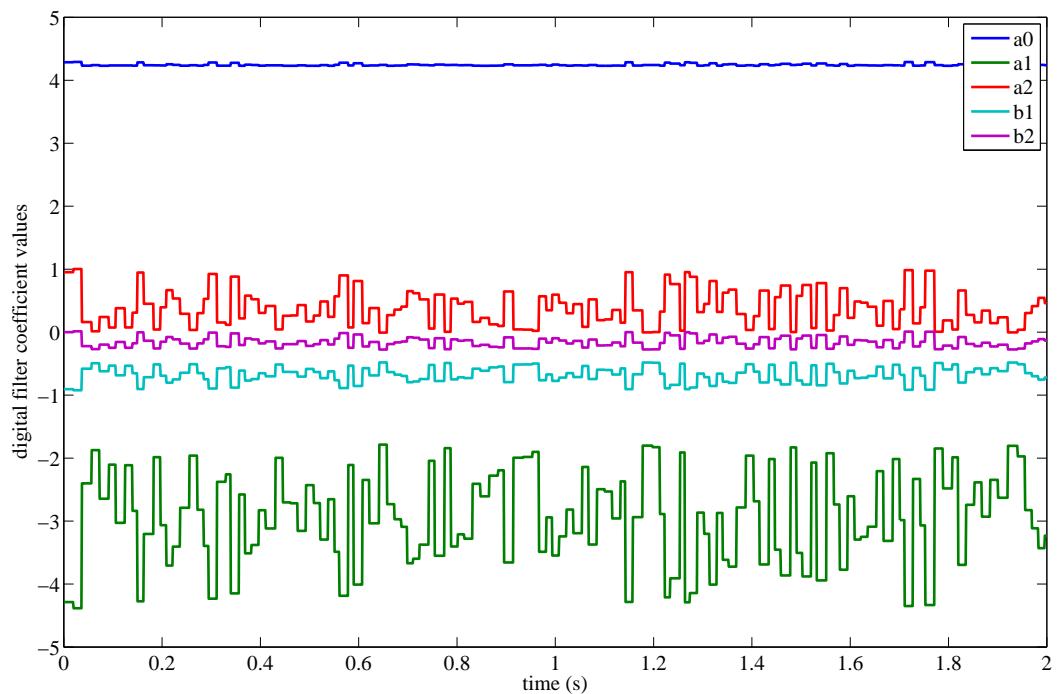


(b) The digital filter output to a 0.5Hz sinusoid input (amplitude = 1)

Figure 3.9: The filter output for non-uniform sampling (uniform distribution $T_n \sim U(0.01,0.02)$, $0.01 \leq T_n \leq 0.02$, average sampling frequency at 0.015s); filter structure: modified canonic- δ

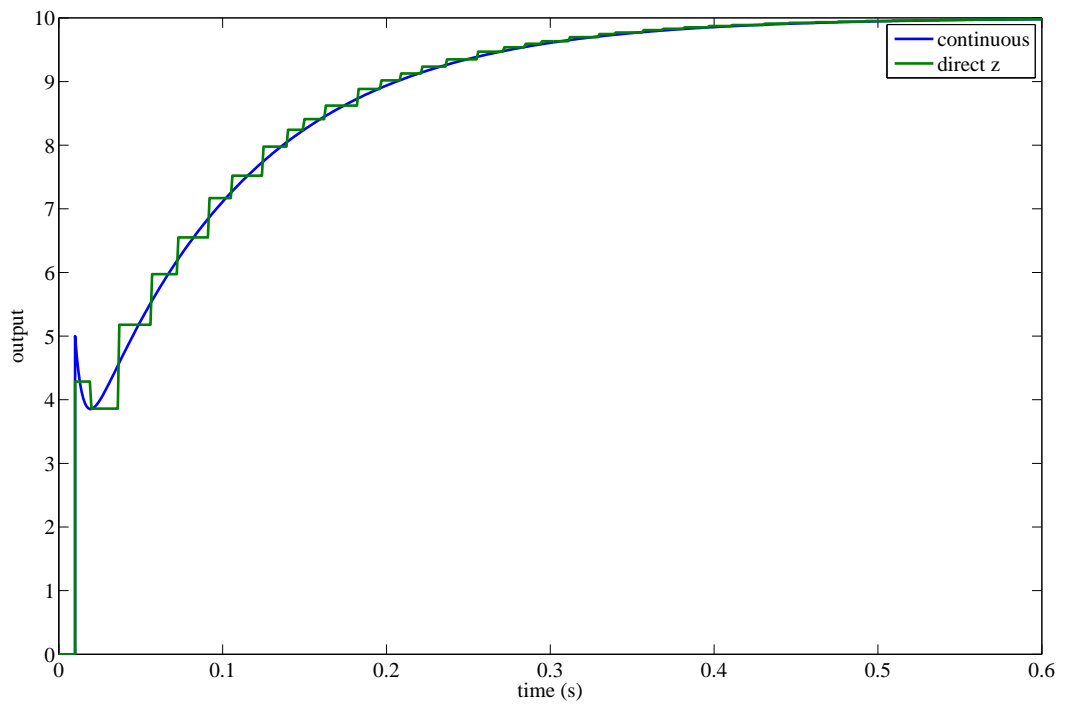


(a) The sample rate of the digital filter changes for consecutive sampling period, $T_k \sim U(0.01, 0.02)$

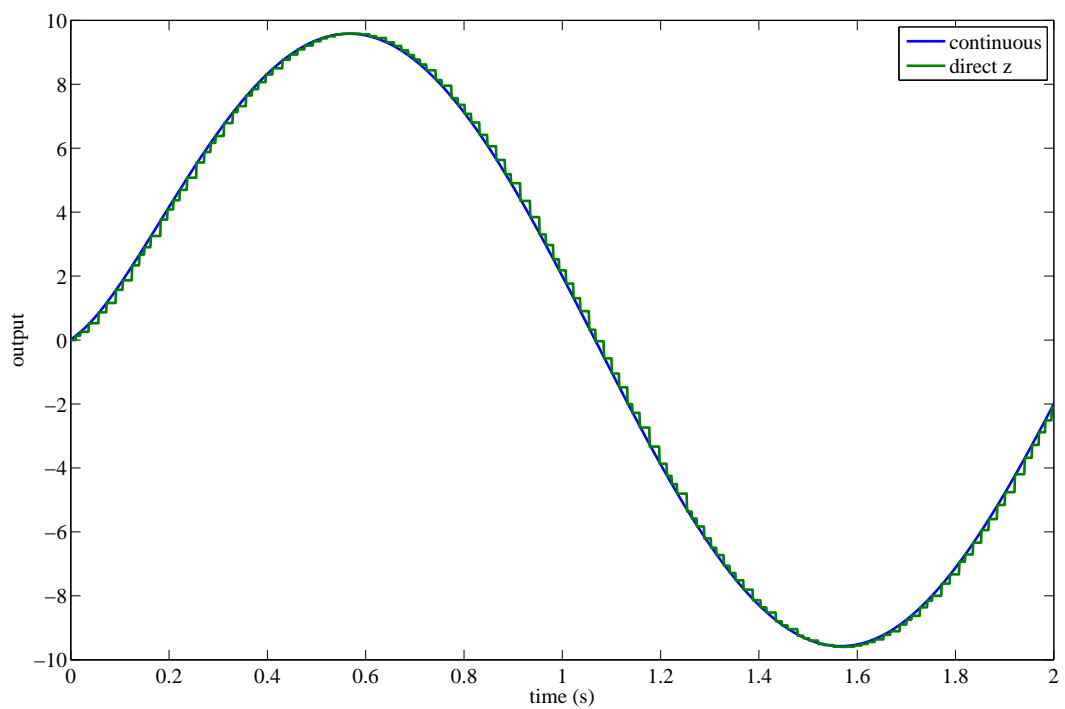


(b) The digital filter coefficient values change depending on the sample rate

Figure 3.10: The filter setup for non-uniform sampling (uniform distribution with an average sampling frequency at 0.015s)



(a) The digital filter output to a step input



(b) The digital filter output to a 0.5Hz sinusoid input

Figure 3.11: The direct z -filter output with non-uniform sampling with a uniform distribution $T_n \sim U(0.01, 0.02)$ ($0.01 \leq T_n \leq 0.02$, and an average sampling frequency at 0.015s); filter structure: direct- z

3.3.2 Discussion

The simulations carried out in this section explicitly show that there exists a broad range of behaviours depending on the implementation structure in use. The results are related to the specific set of filters being used in the analysis, where it is pointed out that correctly choosing the proper structure at the design time can significantly reduce the undesirable effects of the digital filter reconfigurations. Some important conclusions from the analysis are that the:

- transient phenomenon is caused when the internal variables of the filter are not scaled according to the variations in the sampling period
- severity of a transient signal depends on the size of magnitude change in the filter coefficients. This indicates that *very* large changes in between consecutive sample periods cause increasingly bad effects
- canonic z -filter and the canonic δ -filter structures cannot be used for implementing the non-uniform sampling algorithm
- modified canonic δ -filter and the direct z -filter structures offer a better choice for suppressing the transient phenomenon. This is largely due to the way the state variables are handled internally by the filter structure

The above findings lead to another interesting question: *Between the direct z and modified canonic δ , which implementation structure suits best for non-uniform sample rates*¹⁰? Perhaps, quantifying the difference between the discrete and the continuous transfer function outputs with a sinusoid input¹¹

¹⁰—with consecutive variations.

¹¹The reason for opting for a sinusoid instead of a step input is because with a step input, the system eventually reaches to a steady-state value. In which case some sampling schemes, such as the dual rate, may not have enough time to be regulated.

may provide an answer. The Mean Squared Error (MSE) is therefore calculated for all the non-uniform sampling schemes considered in this thesis, applied to the various filter structures. The filter used in the experiment is from Eq. 3.1.7 used earlier in this section, with a 0.5Hz sinusoid signal. The results are tabulated in Table 3.2. The simulation run-time is 4s.

As expected, the output produced by the canonic- z and the canonic- δ filters give an unacceptably high value for the MSE¹². The direct- z and the modified canonic- δ structures, which are able to suppress the transient issue, produce the lowest MSE. Both structures produce similar results and there is no conclusive evidence to differentiate the better structure amongst them. However, from this point onwards, the simulations in this thesis will make use the modified canonic δ -filter structure for implementing the non-uniform rate algorithm, unless specified otherwise. An interesting consequence when the filter coefficients are held constant throughout the simulation time under non-uniform sampling conditions is evident from the table. Although unchanging the coefficients helps avoid the transient issue, the output performance of the filter deteriorates and results with a high MSE value.

After ensuring that the derived algorithm is operating correctly with non-uniform sampling, the next step is to ensure the correct method to evaluate the frequency response of the non-uniform rate transfer function. This is important since the existing techniques¹³ do not necessarily apply in the case with non-uniform sampling. The next section is concerned with this issue and looks to answer Q3 from page 4: *How can the frequency response of a non-uniform rate controller be evaluated?*

¹²This is because of the uncontrollable transients produced during implementation.

¹³See Rake (1980) and Dorf & Bishop (2007) for some of the most common frequency analysis methods.

Sampling scheme	Constant sampling $T_n=0.015s$	Uniformly distributed $T_n \sim U(0.01, 0.02)$	Dual sample rate $T_n=0.01s$ or $0.02s$	Normally distributed $T_n \sim N(0.015, 0.1^2)$	Sine wave distribution $0.01 \leq T_n \leq 0.02$
canonic z -filter	0.0288	0.4350	1.5548	0.3062	0.6505
direct z-filter	0.0287	0.0266	0.0319	0.0219	0.0319
canonic δ -filter	0.0291	0.4349	1.555	0.3089	0.6589
modified canonic δ-filter	0.0288	0.0265	0.0316	0.0218	0.0318
direct z -filter (no coefficient change)	0.0288	0.1504	0.1433	0.1343	0.1505
modified canonic δ -filter (no coefficient change)	0.0288	0.1504	0.1432	0.1343	0.1504

Table 3.2: Comparing the Mean Squared Error of the various filter structures for various sampling conditions.

3.4 Frequency analysis method for non-uniform rate controllers

In control theory, a common technique for closed-loop analysis is to determine the steady-state frequency response of the system, often plotted using the Bode plot. The Bode plot represents the natural behaviour of a linear system over a range of selected frequencies. The easiest way to calculate it is to substitute $s \rightarrow jw$ in the continuous-time transfer function and obtain the magnitude and phase at different frequencies. However, an important question that arises with context to this research is: *How to analyse the frequency response of a system with a non-uniform rate controller?*

In the case with uniform sampling, linear systems theory can directly provide the answer i.e. simply by substituting $z = e^{jw}$ in the z transfer function. Unfortunately, the same may not be used to perform an accurate frequency analysis for the case with non-uniform sampling. Speculating the concept, the most probable method is to revisit the basic foundations of signal analysis i.e. the Fourier series.

As widely known, the Fourier analysis is a process that decomposes a given function into various sinusoids of different frequencies. These sinusoids are actually the harmonics of the fundamental frequency of the original function that is being analysed. In general, a Fourier analysis can be performed over a running window of the fundamental frequency, that can allow the calculations of the magnitude and phase of the observation signal. A much more comprehensive description of the analysis technique, its use, and its limitations is given in Katznelson (1976).

Rethinking the basic Fourier definition, a periodic signal $f(t)$ can be expressed by a Fourier series in the form:

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n(t) \cos(nw_1t) + b_n(t) \sin(nw_1t) \quad (3.4.1)$$

where $\frac{a_0}{2}$ is the DC component or *the average value of the signal*, and n represents the rank of the harmonics ($n = 1$, corresponds to the fundamental component). The remaining variables are described as:

$$a_n(t) = \frac{2}{T_F} \int_{t-T_F}^t f(t) \cos(nw_1t) dt$$

$$b_n(t) = \frac{2}{T_F} \int_{t-T_F}^t f(t) \sin(nw_1t) dt$$

$$T_1 = \frac{1}{f_1} = \frac{2\pi}{w_1}$$

$$T_F = kT_1$$

where f_1 is the fundamental frequency and T_F is the integration time being averaged via a moving window over k periods of the fundamental for the Fourier analysis.

The definition of the Fourier coefficients, a_n and b_n , presented in Eq. 3.4.1 are considered to be components of time and hence can be used to describe the behaviour of the signal frequency characteristics in the time domain. The magnitude and phase of the observation signal $f(t)$, or the *selected harmonic component*, can be calculated by the following equations:

$$\begin{aligned} \angle H_n &= \arctan \left(\frac{a_n(t)}{b_n(t)} \right) \\ |H_n| &= \sqrt{a_n^2(t) + b_n^2(t)} \end{aligned} \quad (3.4.2)$$

After performing the Fourier analysis of the filter output and input signals,

the frequency response of the transfer function is computed by:

$$\begin{aligned} Gain &= \frac{H_{output}}{H_{input}} \\ Phase(\varphi) &= \angle H_{output} - \angle H_{input} \end{aligned} \quad (3.4.3)$$

The steady state response of a system can be evaluated for a sinusoidal input at a given frequency. For a continuous-time system, the response will be the same frequency as the input, but the frequency response parameters will be modified with respect to the transfer function of the system being assessed under the input frequency¹⁴. Example C demonstrates the method with a 0.5Hz input sinusoid signal (amplitude = 1). The frequency response parameters of the transfer function under observation can thus be obtained by performing a complete analysis with different frequency values. These can then be compared with the ideal frequency response to understand the characteristics when using a non-uniform rate setup.

Example C: evolution of Fourier transform coefficients

Consider the phase lag-lead compensator of Eq. 3.1.7 from page 54. The non-uniform sample rate being used in this exercise has a uniformly distributed set of samples between 0.01s and 0.02s, as illustrated in Fig. 3.8a. Therefore, the average non-uniform sample rate for the process is 0.015s. Using the approach highlighted in this section, a Fourier analysis of the transfer function with a moving window of $k=3$. Fig. 3.12 illustrates the technique performed with the continuous filter, the uniform filter (with sample rate at 0.015s) and the non-uniform sampling filter

¹⁴On the contrary, the response of a digital system to an input signal will consist of a sum of many sinusoids spaced at integer multiples of the sampling frequency.

(with uniformly distributed samples at an average sampling frequency of 0.015s). It depicts the behaviour of the magnitude and phase at the fundamental frequency of 5Hz after reaching the steady state value¹⁵. This is the ‘Evolution of the Fourier Transform’ coefficient calculations over time. From the simulation, the frequency response parameters are tabulated below:

Frequency (Hz)	Continuous (dB)	Uniform rate (dB)	Non-uniform rate	
			max	min
5	12.11	11.88	12.26	11.48

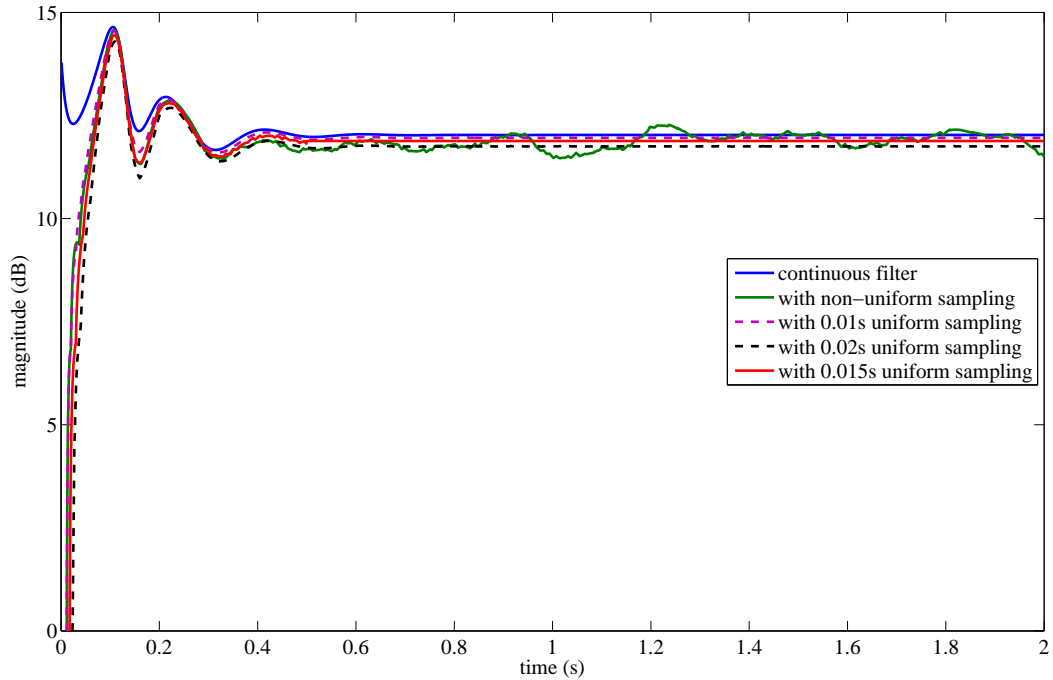
Table 3.3: The filter magnitude values (dB)

Frequency (Hz)	Continuous (degrees)	Uniform rate (degrees)	Non-uniform rate	
			max	min
5	-25.65	-38.8	-36.5	-42

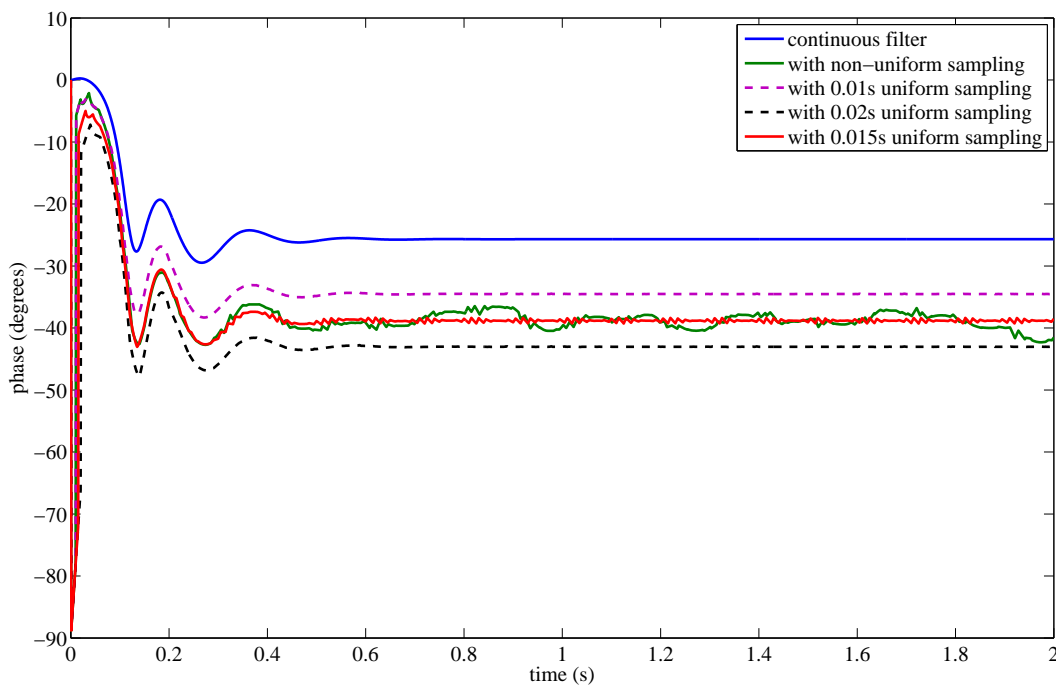
Table 3.4: The filter phase values (degrees)

As illustrated Fig. 3.12, the scope of the technique may further be extended to observe the changes that may occur in the magnitude and phase of non-uniform rate controllers over time. This will provide a much accurate answer to the frequency response characteristics for non-uniform sample rates. Furthermore, this procedure can be repeated for a range of frequencies and a complete Bode diagram can now readily be plotted for the non-uniform rate digital filter.

¹⁵The Fourier result reaches to a steady-state value after an initial transient. The later simulations carried out in this thesis using this technique will consider **only** the final steady-state values i.e. when this preliminary stage has died out.



(a) The filter magnitude response at 5Hz



(b) The filter phase response at 5Hz

Figure 3.12: The filter frequency response at 5Hz. After an initial transient, the values settle down towards a steady state to provide the magnitude and phase values at the frequency of interest. The non-uniform sampling used here belongs to a uniform distribution with an average sampling frequency at 0.015s, where $T_n \sim U(0.01,0.02)$

Example D: approximate frequency domain response using the Fourier analysis technique

The method demonstrated using Example C can effectively be used to obtain the dynamic frequency response of a digital compensator. It is a combination of the frequency domain and time domain analyses based on the Fourier series. It can be used to evaluate the magnitude and phase characteristics of uniform and non-uniform sample rate systems.

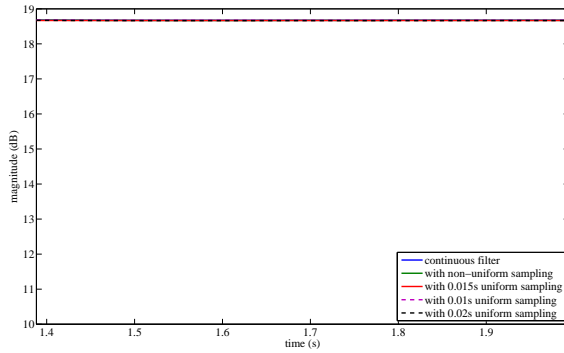
The following demonstration is an extension of Example C. The complete frequency response of the phase lag-lead compensator of Eq. 3.1.7 is estimated.

Simulation setup

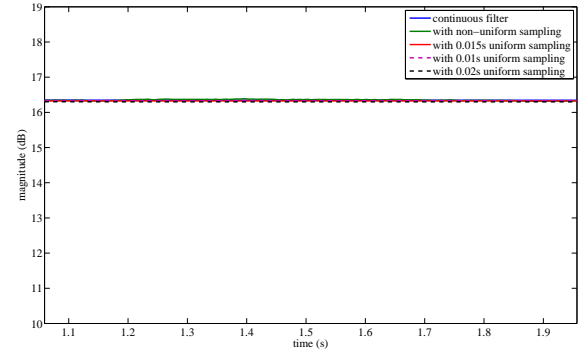
The analysis consists of three compensators:

- the continuous controller
- the discrete controller with a uniform sampling rate T_n of 0.015s
- the discrete controller with non-uniform sampling with uniformly distributed sample rates i.e. $T_n \sim U(0.01,0.02)$. The average sample rate in which case will be 0.015s

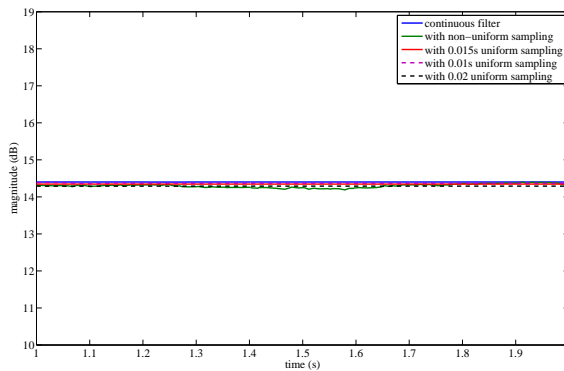
Figs. 3.13 and 3.14 show the obtained magnitude and phase plots, respectively, from the analysis of the three compensators, with frequencies of 1, 2, 3, 4, 5 and 6Hz. The frequency response is then tabulated in Tables 3.5 and 3.6 accordingly.



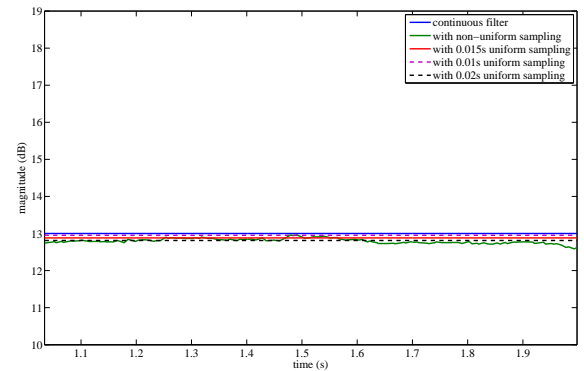
(a) At frequency = 1



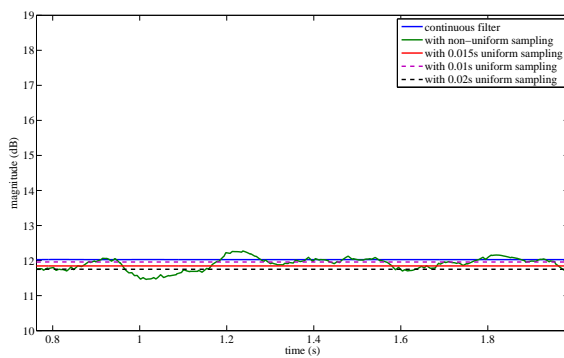
(b) At frequency = 2



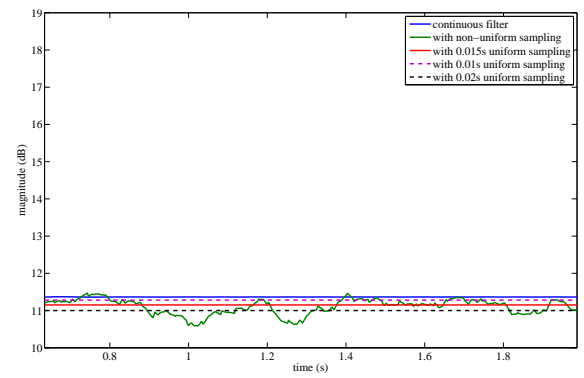
(c) At frequency = 3



(d) At frequency = 4

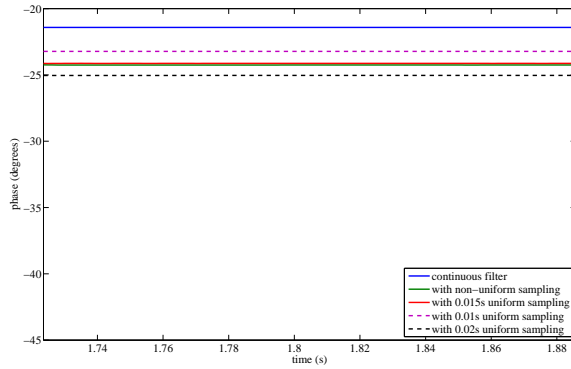


(e) At frequency = 5

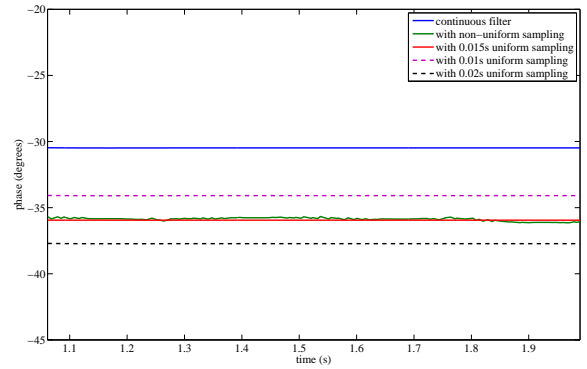


(f) At frequency = 6

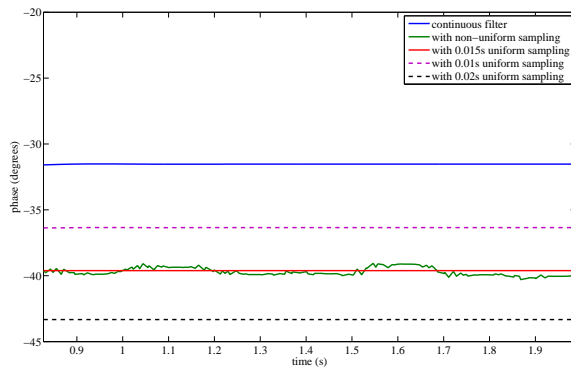
Figure 3.13: The magnitude (dB) values for the digital filters under observation, at various frequencies.



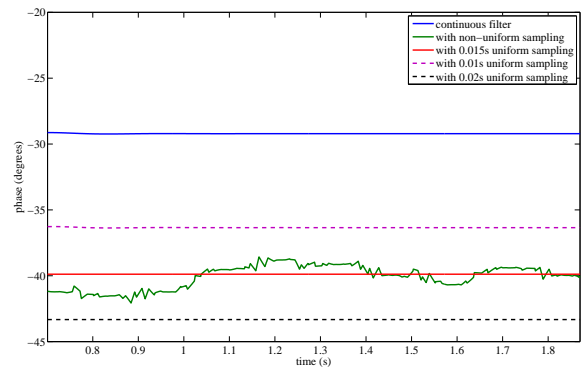
(a) At frequency = 1



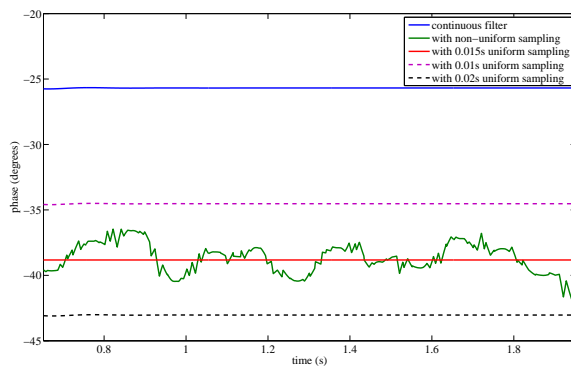
(b) At frequency = 2



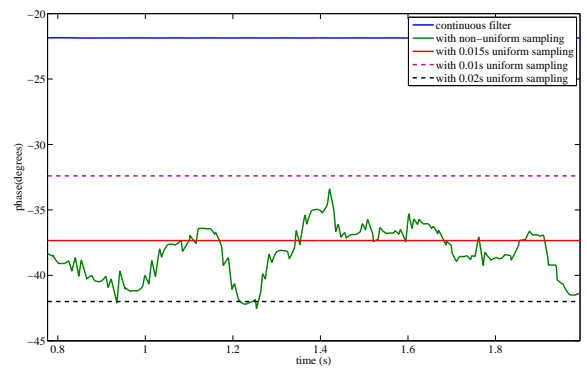
(c) At frequency = 3



(d) At frequency = 4



(e) At frequency = 5



(f) At frequency = 6

Figure 3.14: The phase (degrees) values for the digital filters under observation, at various frequencies.

Frequency (Hz)	Continuous (dB)	Uniform rate (dB)	Non-uniform rate	
			max	min
1	18.6	18.67	18.67	18.67
2	16.3	16.32	16.39	16.31
3	14.4	14.34	14.4	14.2
4	13	12.89	12.95	12.6
5	12.1	11.87	12.27	11.5
6	11.4	11.16	11.45	10.6

Table 3.5: The filter magnitude values (dB) for the compensators under observation.

Frequency (Hz)	Continuous (degrees)	Uniform rate (degrees)	Non-uniform rate	
			max	min
1	-21.4	-24.25	-24.13	-24.35
2	-30.4	-35.7	-35.7	-36.15
3	-31.4	-39.2	-39.2	-40.3
4	-29	-38.6	-38.6	-40.7
5	-25.65	-38.8	-36.48	-42.3
6	-21.7	-39.25	-36.8	-42.4

Table 3.6: The filter phase values (degrees) for the compensators under observation.

The complete Bode plot has been constructed using the values obtained in Tables 3.5 and 3.6; see Fig. 3.17 on page 96.

3.5 Performance evaluation criteria

The objective here is to evaluate, in terms of the frequency response, the compensation approach as a design methodology to be used as a non-uniform rate discrete equivalent for digital control tasks. Since it was made clear in §3.2.2 that the variations of sample periods can be taken into account by the control algorithm, this section will formally specify the performance index that can be used for its evaluation.

To evaluate the performance with the non-uniform rate discrete equivalent controller, a performance loss criterion can be defined. The applicability of this criterion will allow the comparison of various frequency responses. One common technique that is usually adopted to compare the performance is based on the discrepancy in between an exact value and the approximation. Typically, the Absolute Error, Squared Error and their derivations are used to tabulate data. Such criteria can give a measure of the error in the frequency response of a transfer function. Therefore, using the same concept of such classic performance criterions of error evaluation, the *performance loss* criterion for some given value v and its approximation v_{approx} can be defined by its error.

$$\text{Absolute Error, } \epsilon(f) = |v_{approx} - v| \quad (3.5.1)$$

where v is the true value of the frequency response, that is to be used as the reference and v_{approx} is obtained from the discrete-time controllers.

3.6 Open-loop analysis of the phase lead-lag compensator

The compensator under consideration is Eq. 3.1.7 from page 54 which is a combination of two 1st order filters: a phase lag and a phase lead. The following exercise is used to observe if a non-uniform sampling frequency can enable improvements in the phase response of a digital compensator. The digital filter is implemented using the control algorithm outlined earlier to allow variations in the sampling frequency with the modified canonic- δ structure. The filter is implemented using the following sampling schemes accordingly

- uniform distribution of sample rates i.e. $T_n \sim U(0.01,0.02)$
- a truncated gaussian distribution i.e. $T_n \sim N(0.015,0.1^2)$
- dual sample rate (digital pattern) i.e. $T_n = 0.01\text{s}$ or $T_n = 0.02\text{s}$
- a sine wave pattern at 5Hz i.e. $0.01\text{s} \leq T_n \leq 0.02\text{s}$

All the possible values for T_n for the above sampling schemes will remain confined within 50Hz and 100Hz i.e. $0.01\text{s} \leq T_n \leq 0.02\text{s}$. These sample rate values are computed and stored before hand and loaded into the control algorithm for computations at run-time.

Uniform distribution

Consider the case when the non-uniform sample rates to be used, generated such that all intervals have a constant probability, where $T_n \sim U(0.01,0.02)$. A histogram of the sample rates is illustrated in Fig. 3.15. The following simulations of the compensator include the time response (Fig. 3.16), frequency response (Fig. 3.17) and performance loss in the frequency plot (Fig. 3.18).

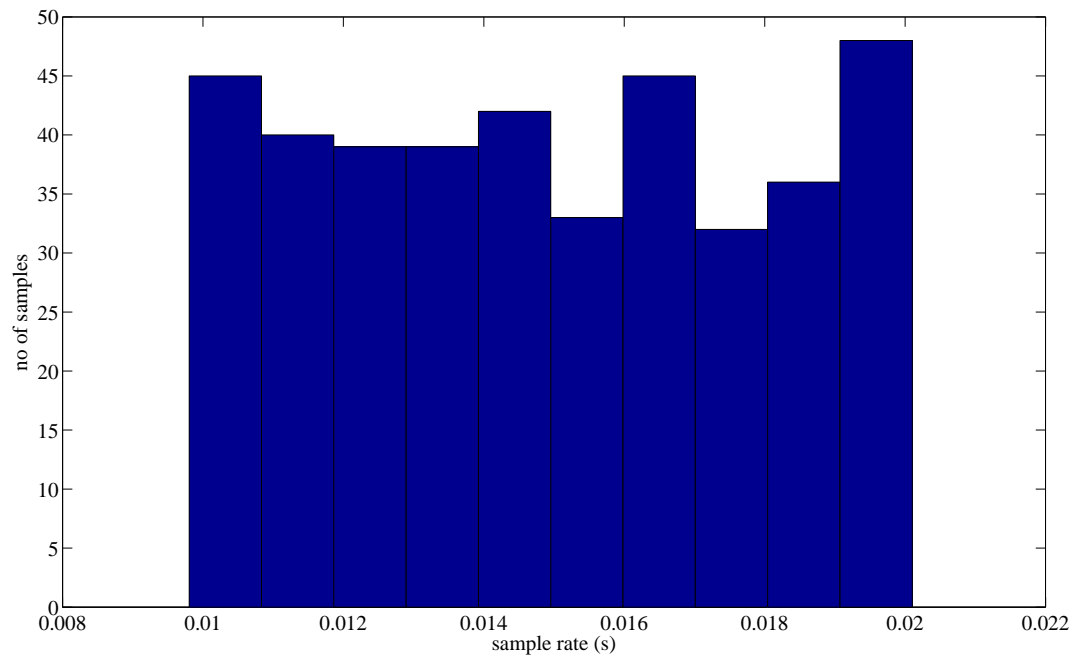
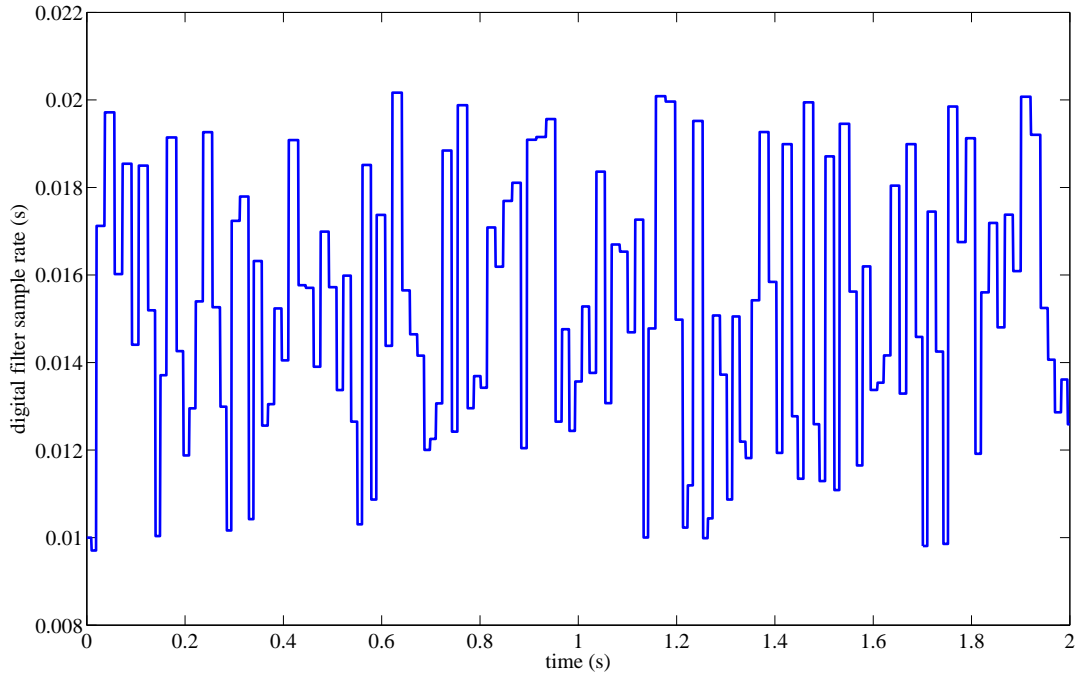
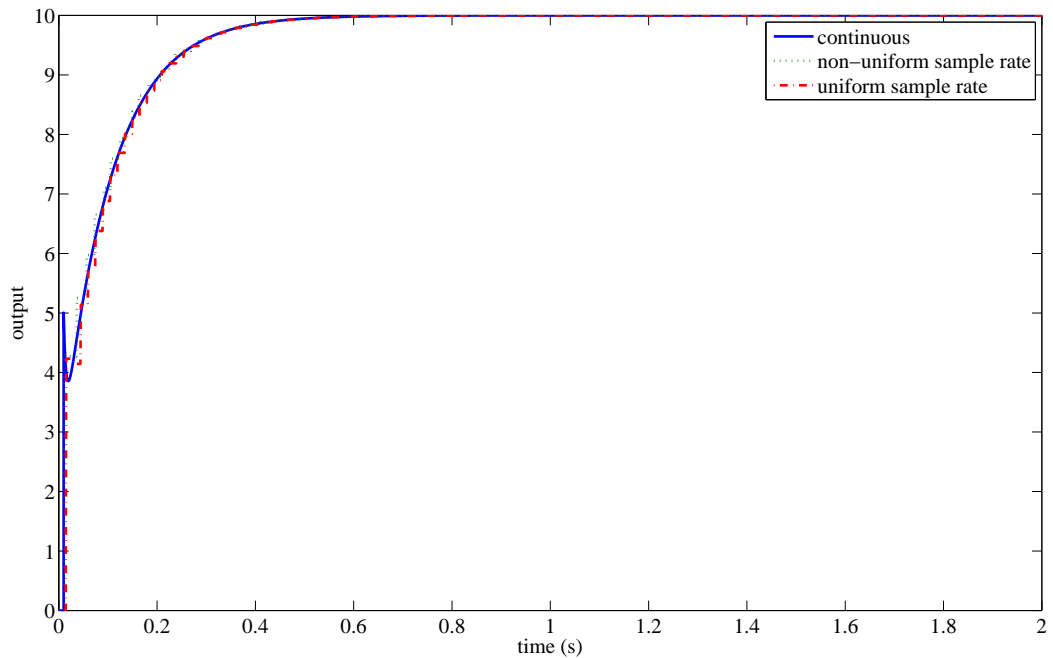


Figure 3.15: Histogram of uniformly distributed sample rates $T_n \sim U(0.01,0.02)$

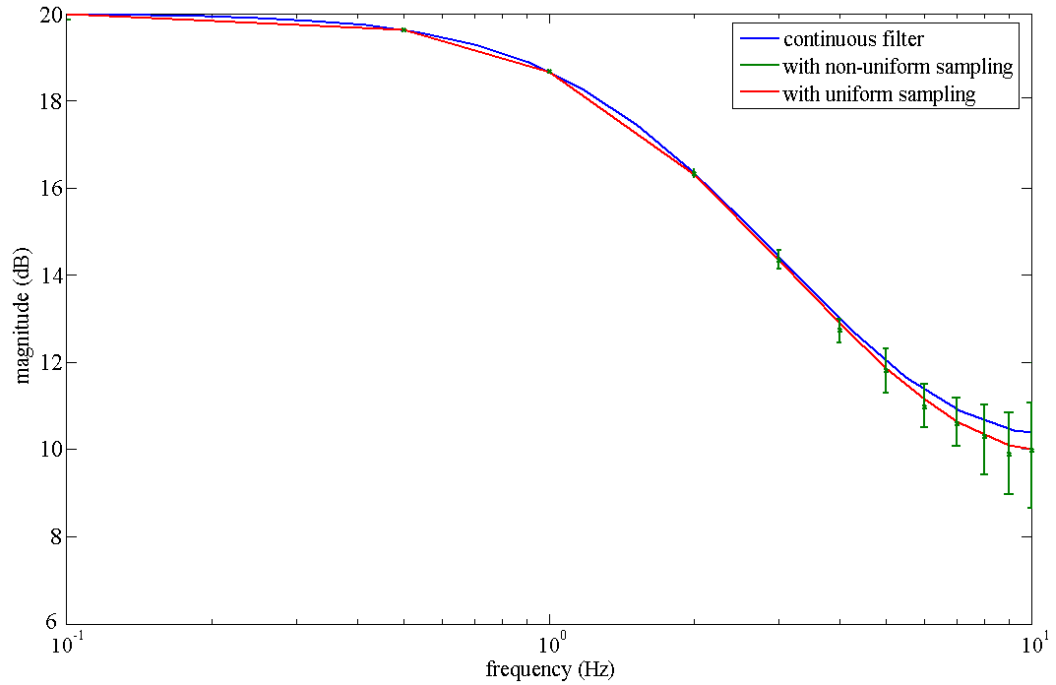


(a) The sample rate being used

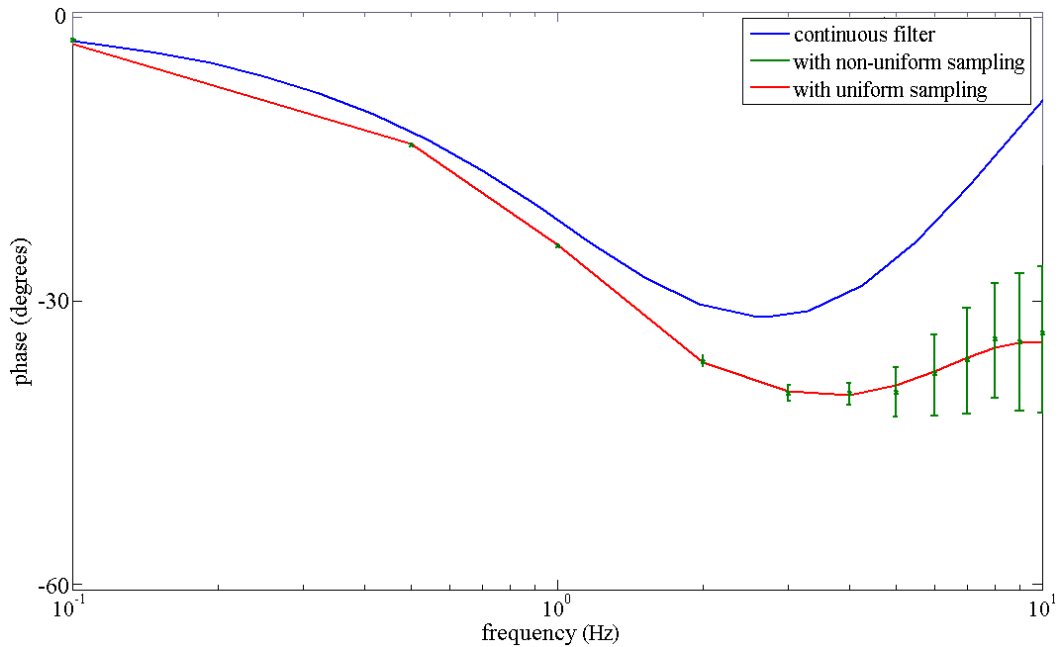


(b) The digital filter output to a step input

Figure 3.16: The time response of the non-uniform sampling filter implementing with uniformly distributed sample rates i.e. $T_n \sim U(0.01,0.02)$

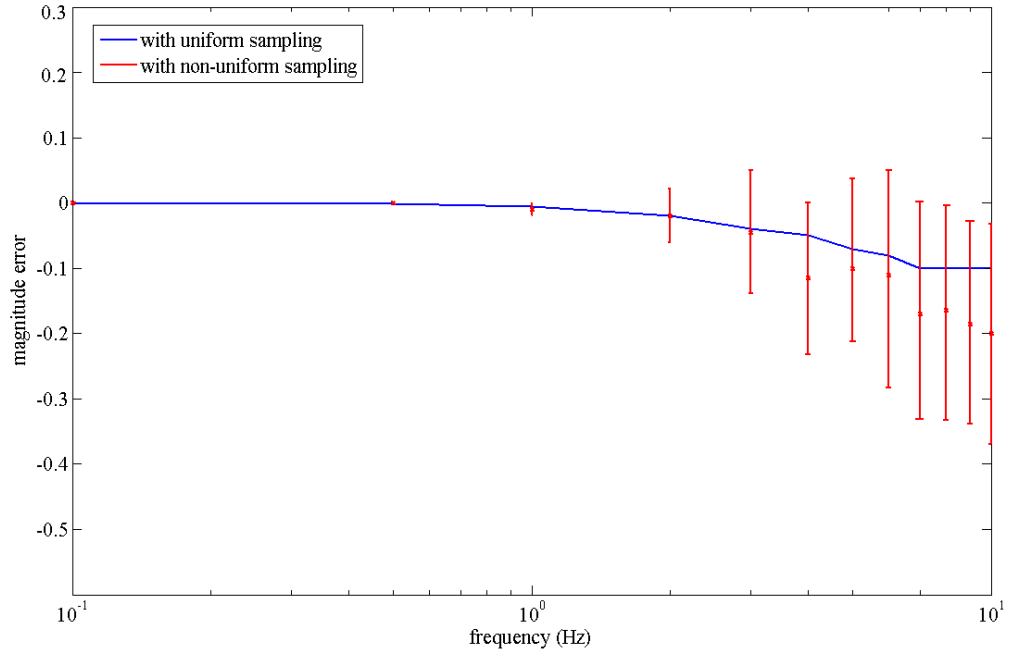


(a) Digital filter magnitude estimation

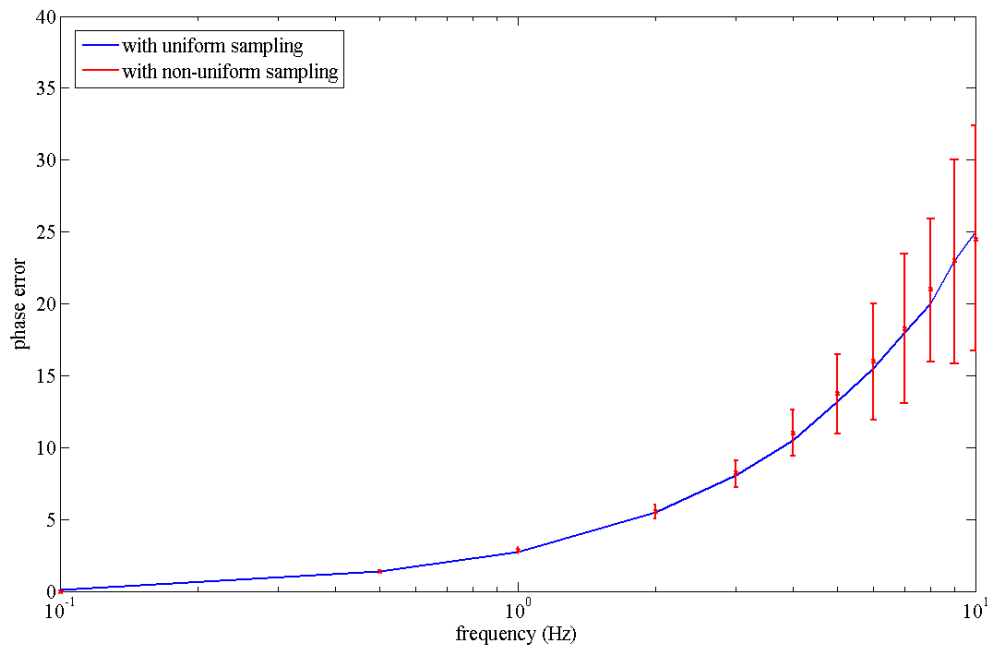


(b) Digital filter phase estimation

Figure 3.17: The frequency response of the non-uniform sampling filter implementing with uniformly distributed sample rates i.e. $T_n \sim U(0.01,0.02)$



(a) Digital filter magnitude estimation performance loss



(b) Digital filter phase estimation performance loss

Figure 3.18: The performance loss in the frequency response of the non-uniform sampling filter implementing with uniformly distributed sample rates i.e. $T_n \sim U(0.01,0.02)$

Truncated gaussian distribution

Consider the case when the non-uniform sample rates to be used, generated such that most of the intervals are clustered around the mean sample rate, where $T_n \sim N(0.015, 0.1^2)$. A histogram of the sample rates is illustrated in Fig. 3.19. The following simulations of the compensator include the time response (Fig. 3.20), frequency response (Fig. 3.21) and performance loss in the frequency plot (Fig. 3.22).

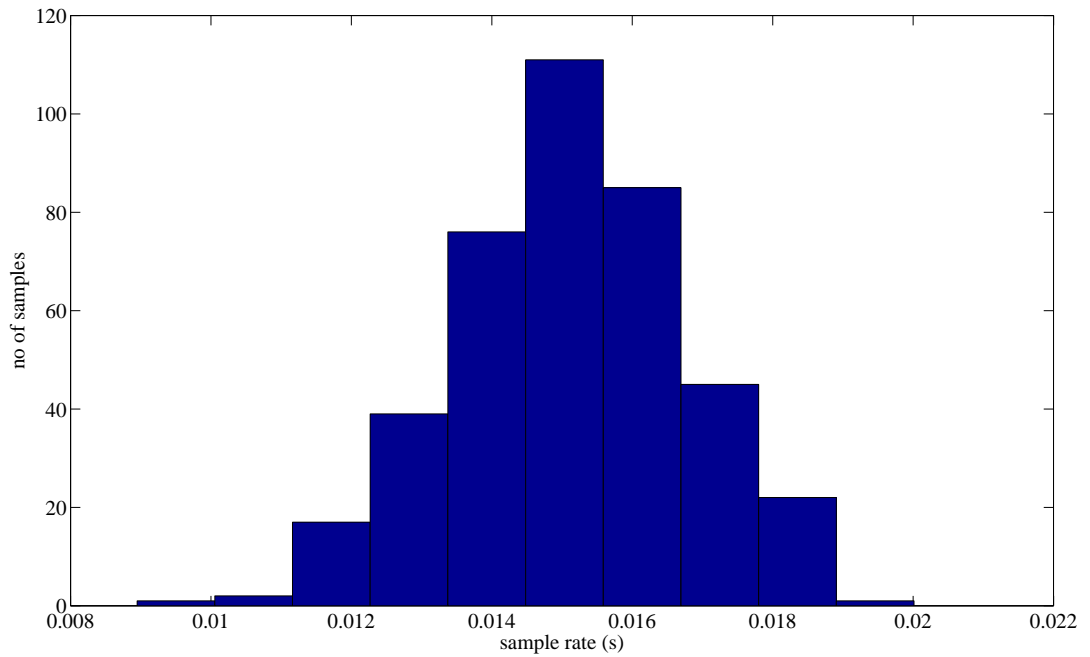
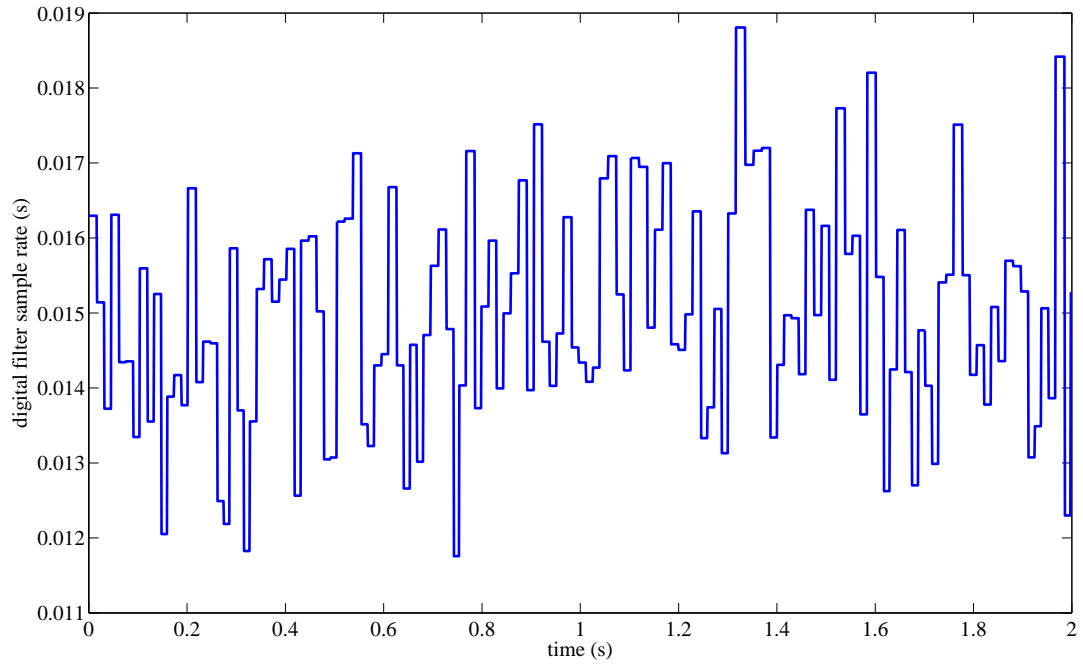
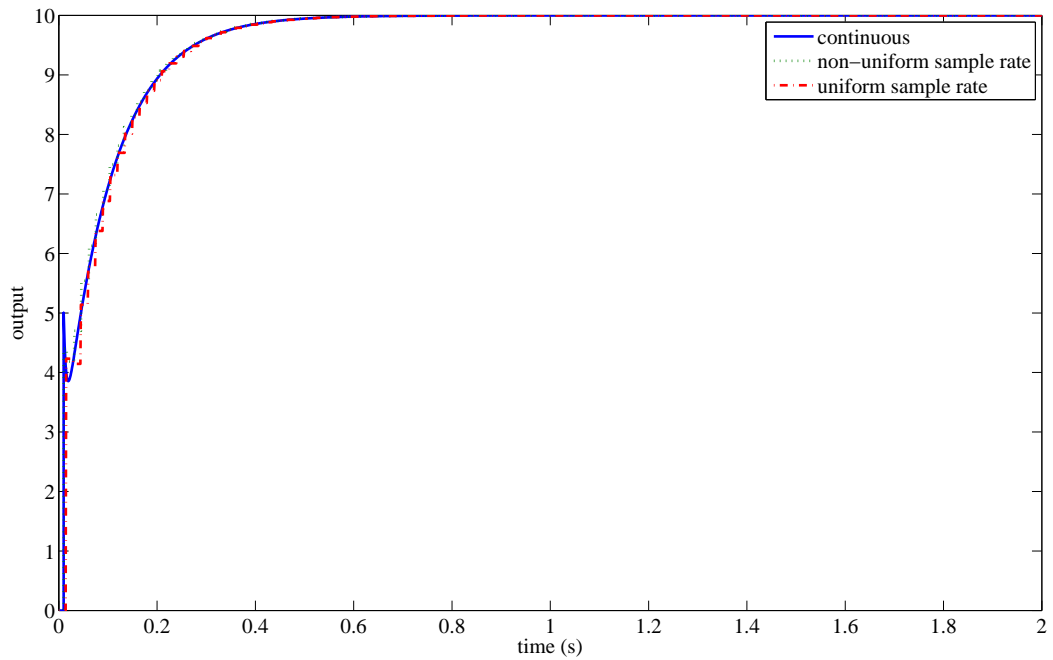


Figure 3.19: Histogram of *normally* distributed sample rates $T_n \sim N(0.015, 0.1^2)$

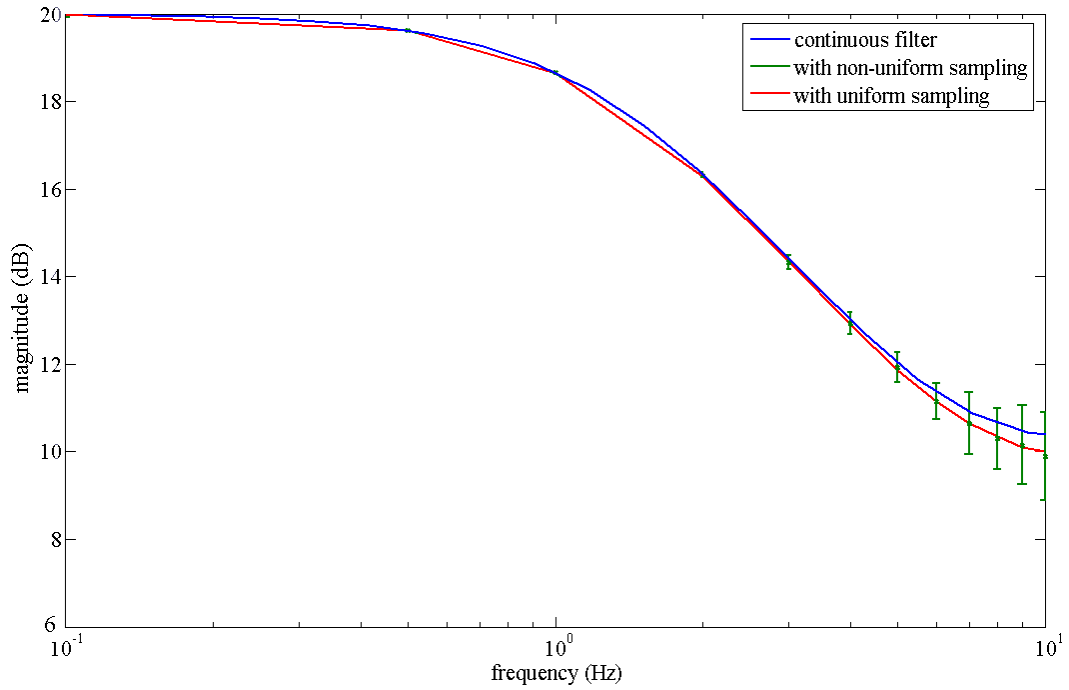


(a) The sample rate begin used

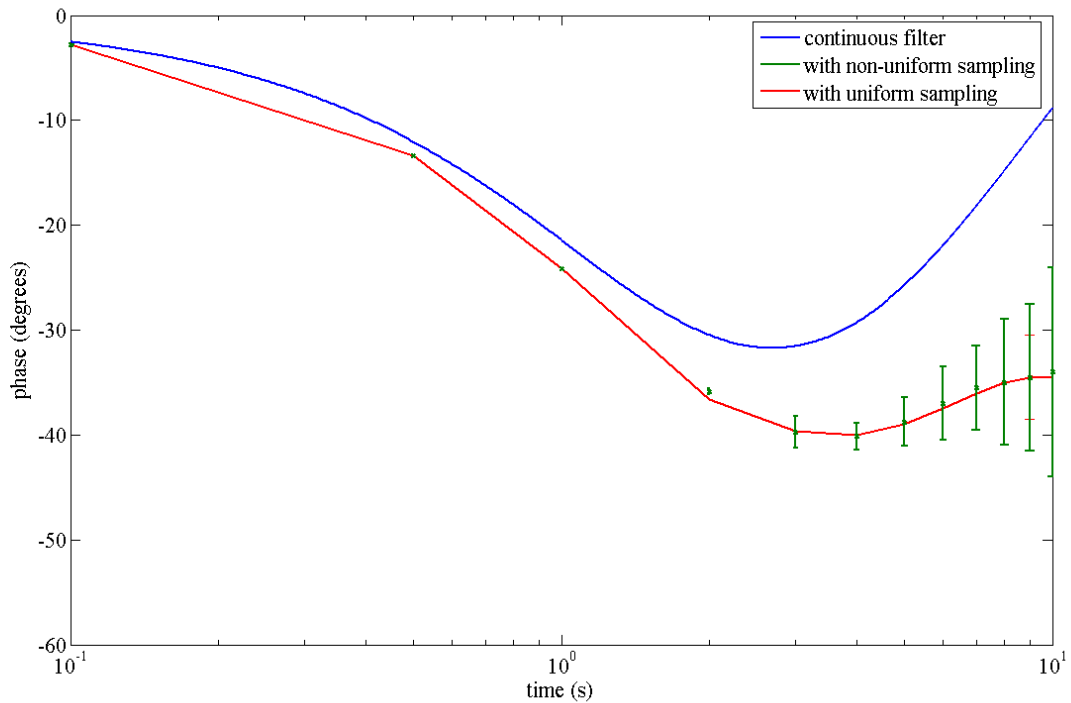


(b) The digital filter output to a step input

Figure 3.20: The time response of the non-uniform sampling filter implementing with truncated gaussian distributed sample rates i.e. $T_n \sim N(0.015, 0.1^2)$

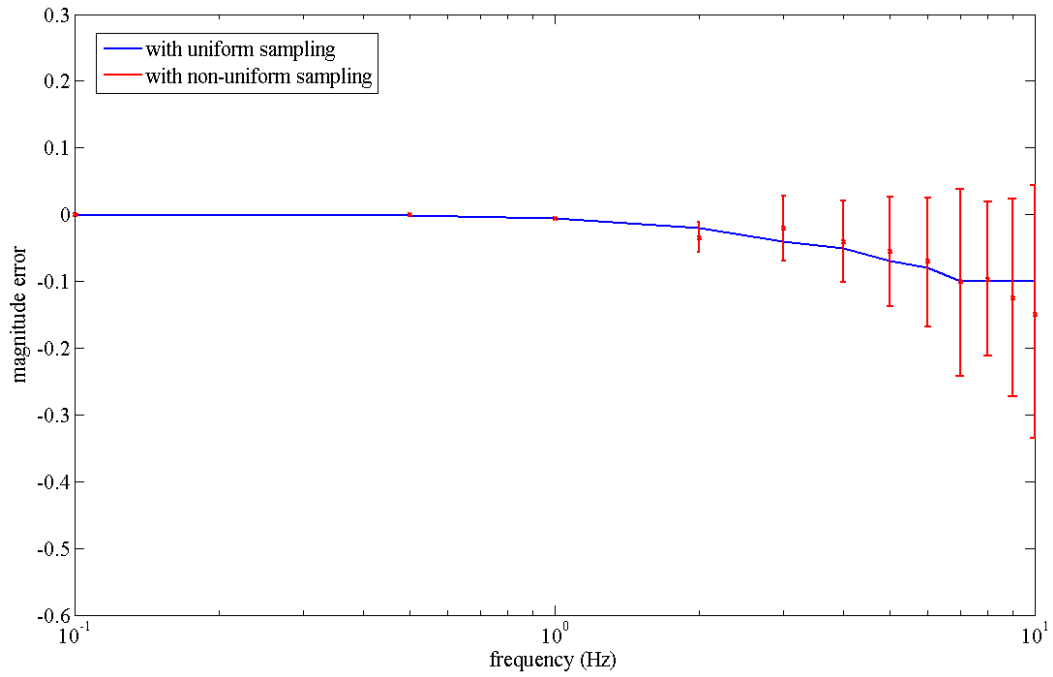


(a) Digital filter magnitude estimation

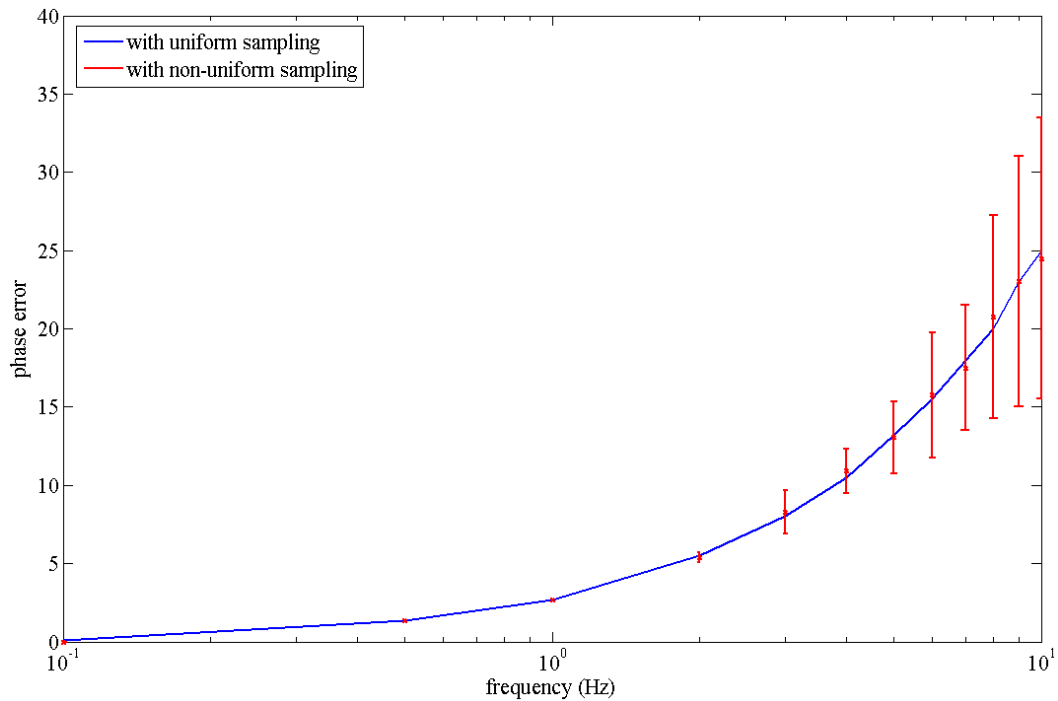


(b) Digital filter phase estimation

Figure 3.21: The frequency response of the non-uniform sampling filter implementing with truncated gaussian distributed sample rates i.e. $T_n \sim N(0.015, 0.1^2)$



(a) Digital filter magnitude estimation performance loss



(b) Digital filter phase estimation performance loss

Figure 3.22: The performance loss in the frequency response of the non-uniform sampling filter implementing with truncated gaussian distributed sample rates i.e. $T_n \sim N(0.015, 0.1^2)$

Dual sample rate

Consider the case when the non-uniform sample rates to be used, generated according to a digital signal, where $T_n = 0.01\text{s}$ or $T_n = 0.02\text{s}$. A histogram of the sample rates is illustrated in Fig. 3.23. The following simulations of the compensator include the time response (Fig. 3.24), frequency response (Fig. 3.25) and performance loss in the frequency plot (Fig. 3.26).

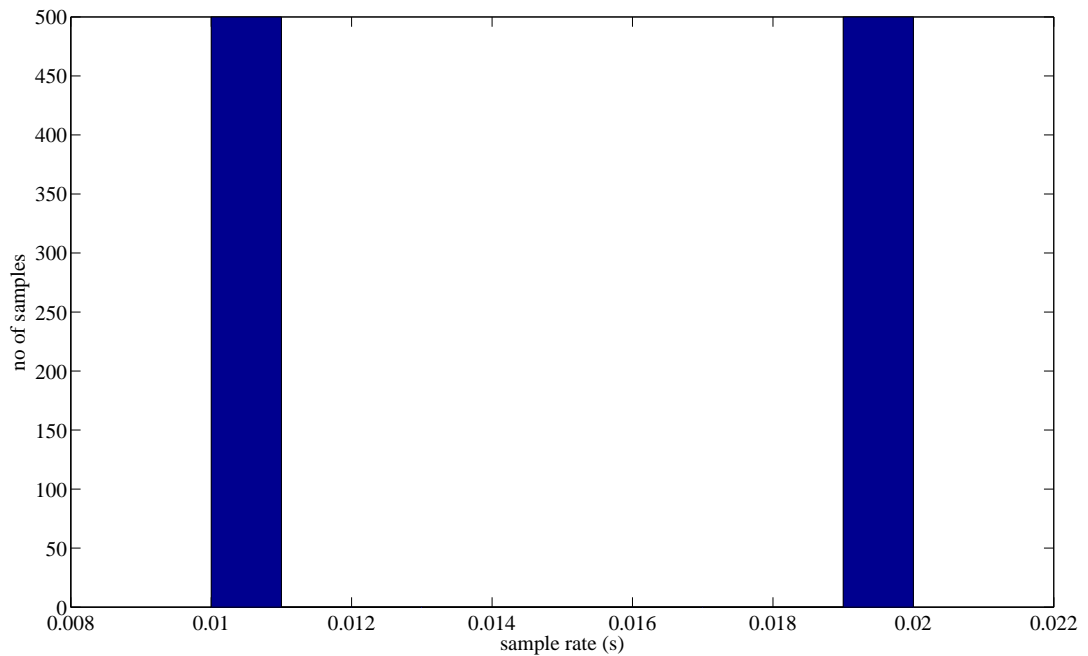
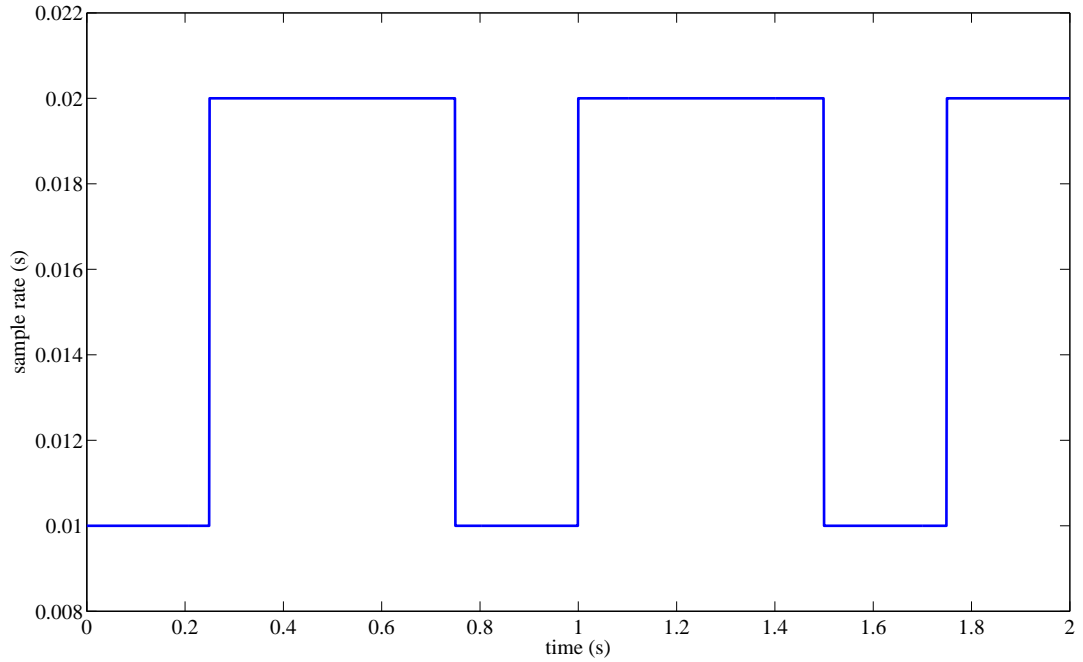
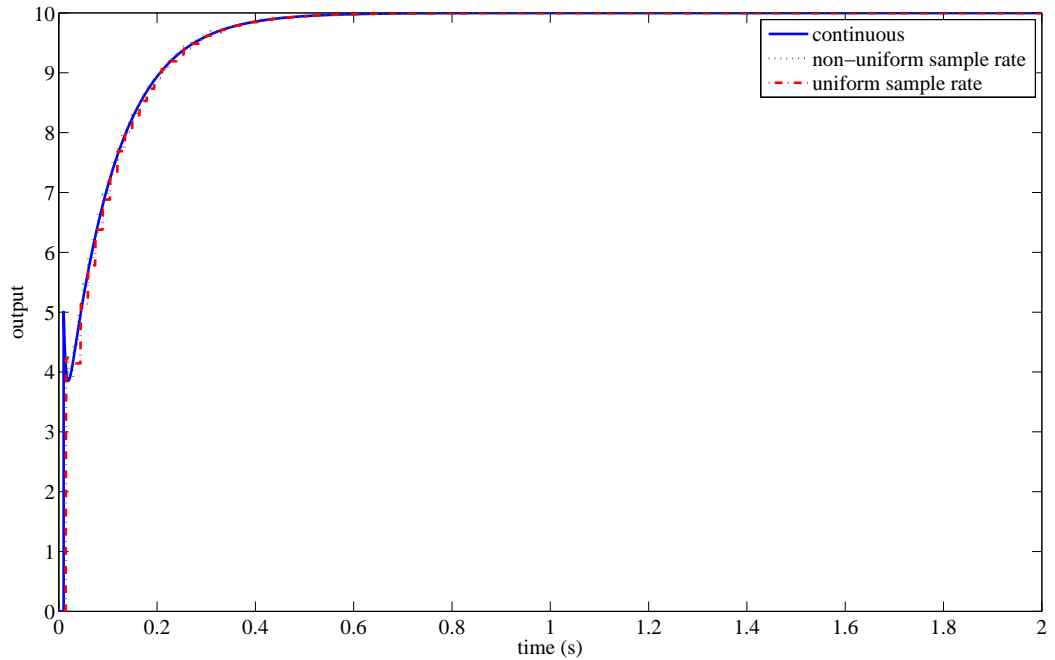


Figure 3.23: Histogram of dual sample rates $T_n = 0.01\text{s}$ or $T_n = 0.02\text{s}$

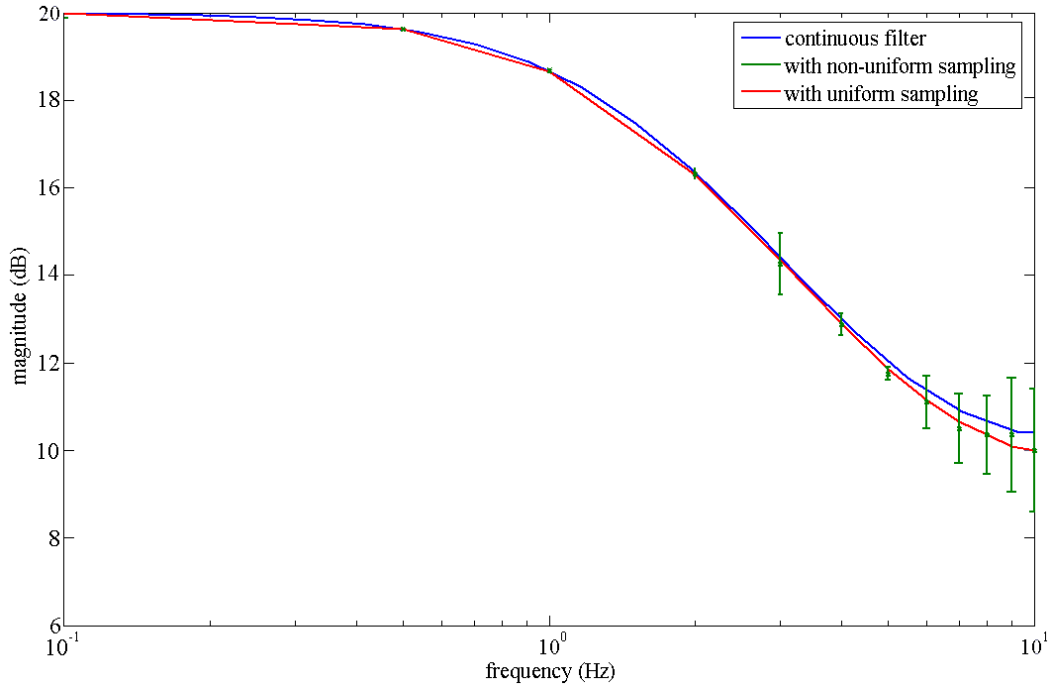


(a) The sample rate begin used

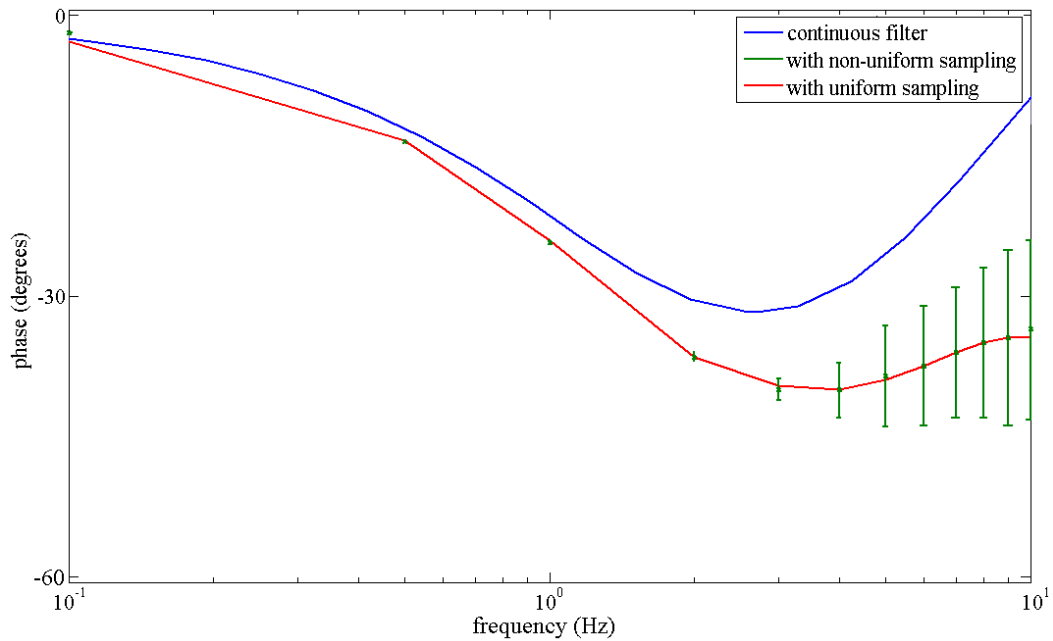


(b) The digital filter output to step input

Figure 3.24: The time response of the filter implementing dual sample rate (digital pattern) i.e. $T_n = 0.01\text{s}$ or $T_n = 0.02\text{s}$

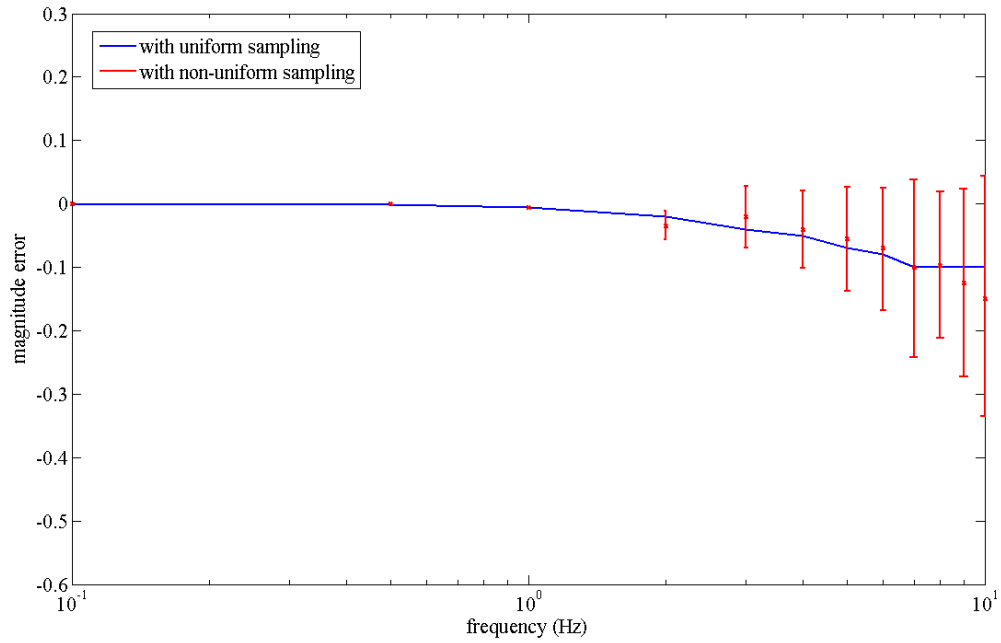


(a) Digital filter magnitude estimation

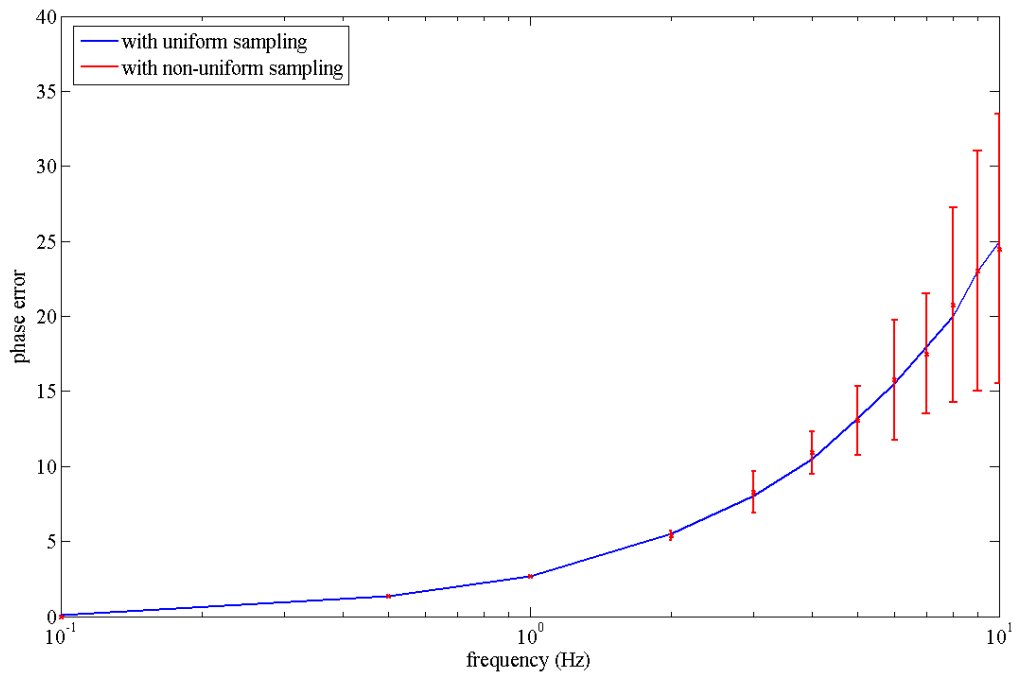


(b) Digital filter phase estimation

Figure 3.25: The frequency response of the filter implementing dual sample rate (digital pattern) i.e. $T_n = 0.01s$ or $T_n = 0.02s$



(a) Digital filter magnitude estimation performance loss



(b) Digital filter phase estimation performance loss

Figure 3.26: The performance loss in the frequency response of the filter implementing dual sample rate (digital pattern) i.e. $T_n = 0.01s$ or $T_n = 0.02s$

Sinusoid distributed samples

Consider the case when the non-uniform sample rates to be used, generated according to a sine wave pattern where the samples are distributed according to a 5Hz sinusoid, $0.01s \leq T_n \leq 0.02s$. A histogram of the sample rates is illustrated in Fig. 3.27. The following simulations of the compensator include the time response (Fig. 3.28), frequency response (Fig. 3.29) and performance loss in the frequency plot (Fig. 3.30).

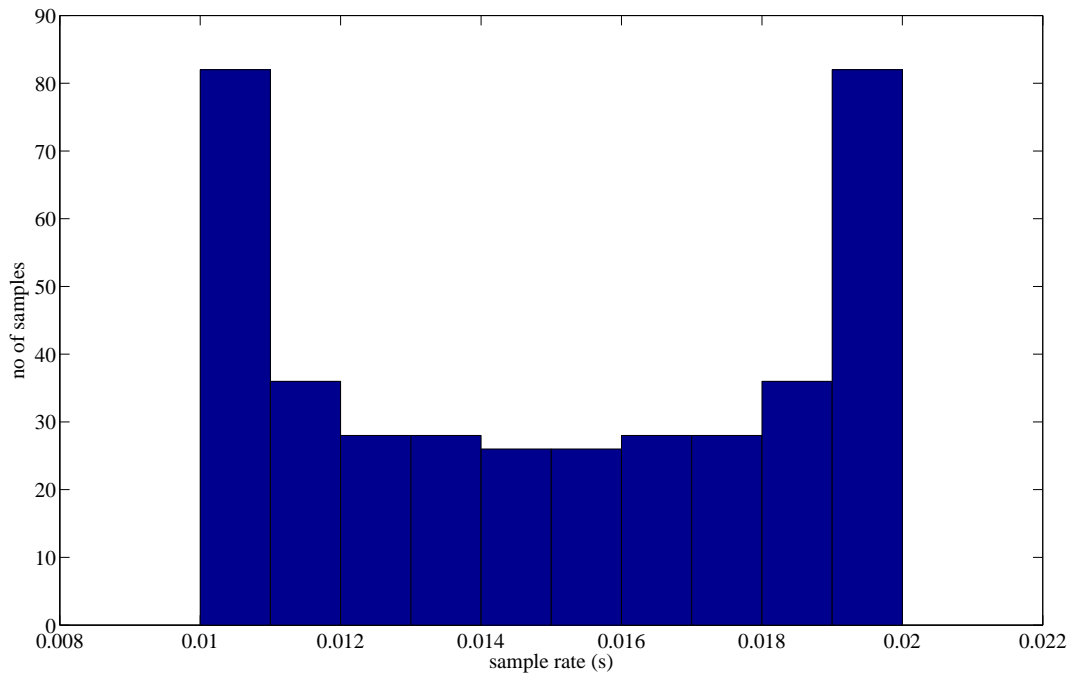
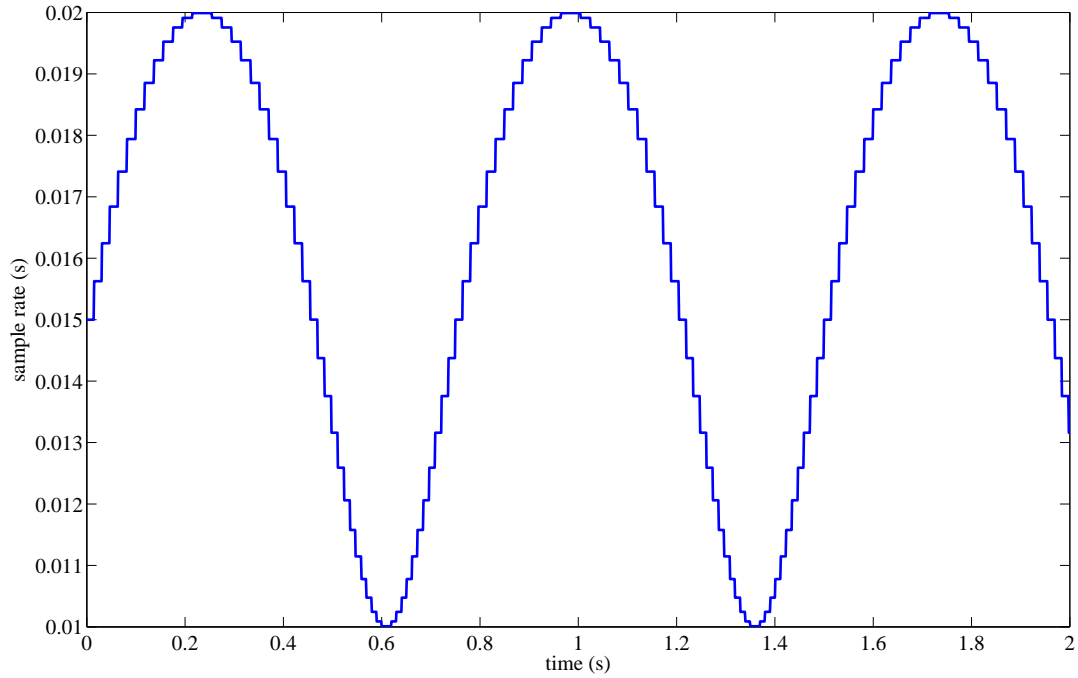
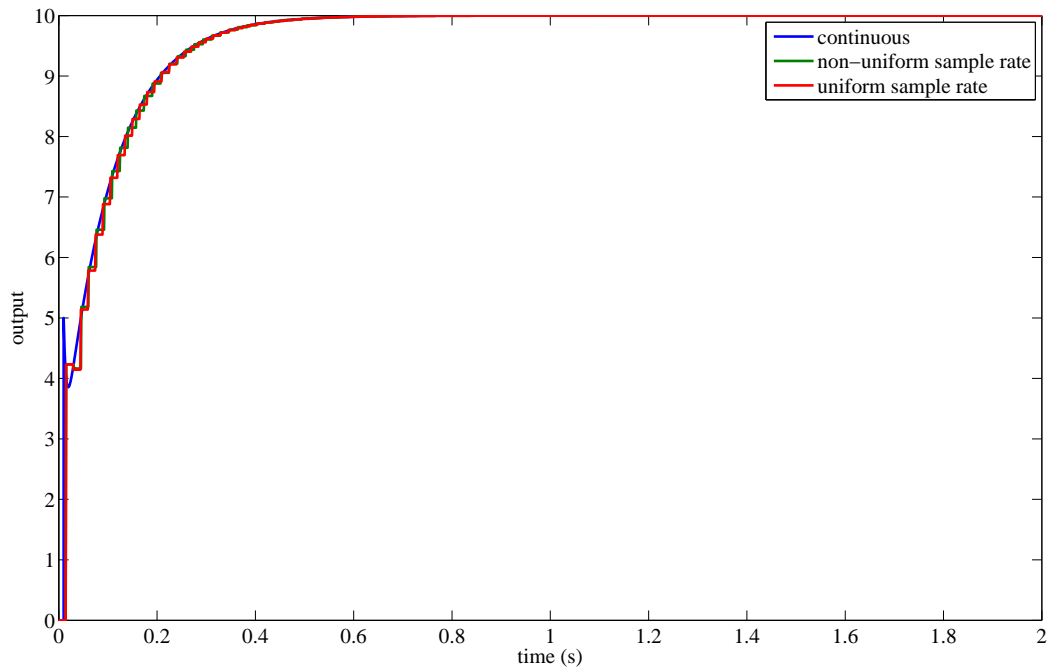


Figure 3.27: Histogram of sin wave patter sample rates $0.01s \leq T_n \leq 0.02s$

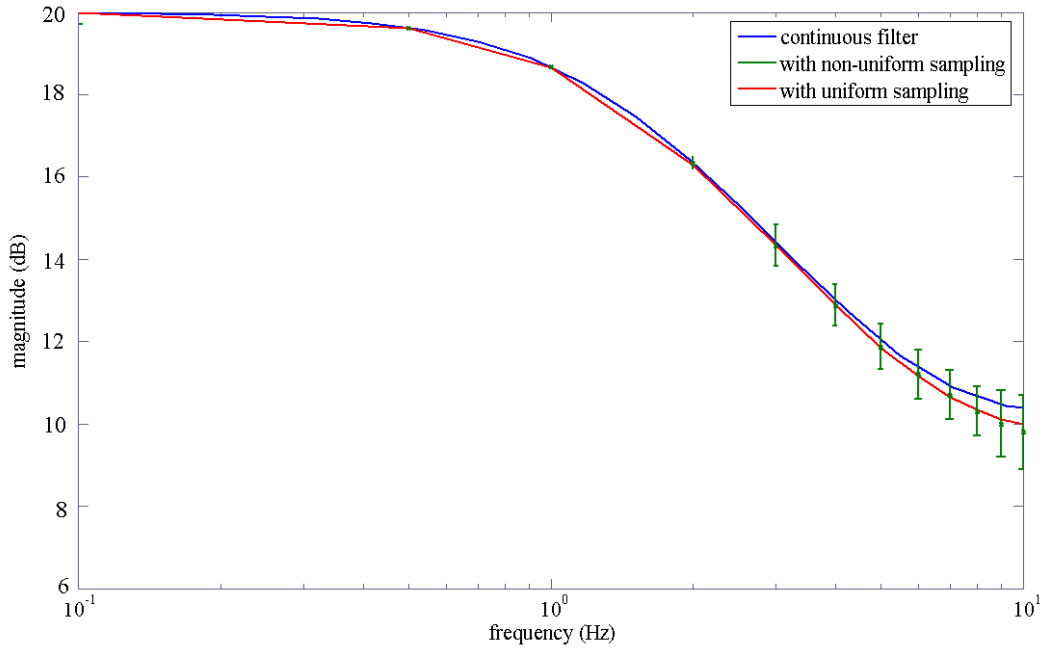


(a) The sample rate begin used

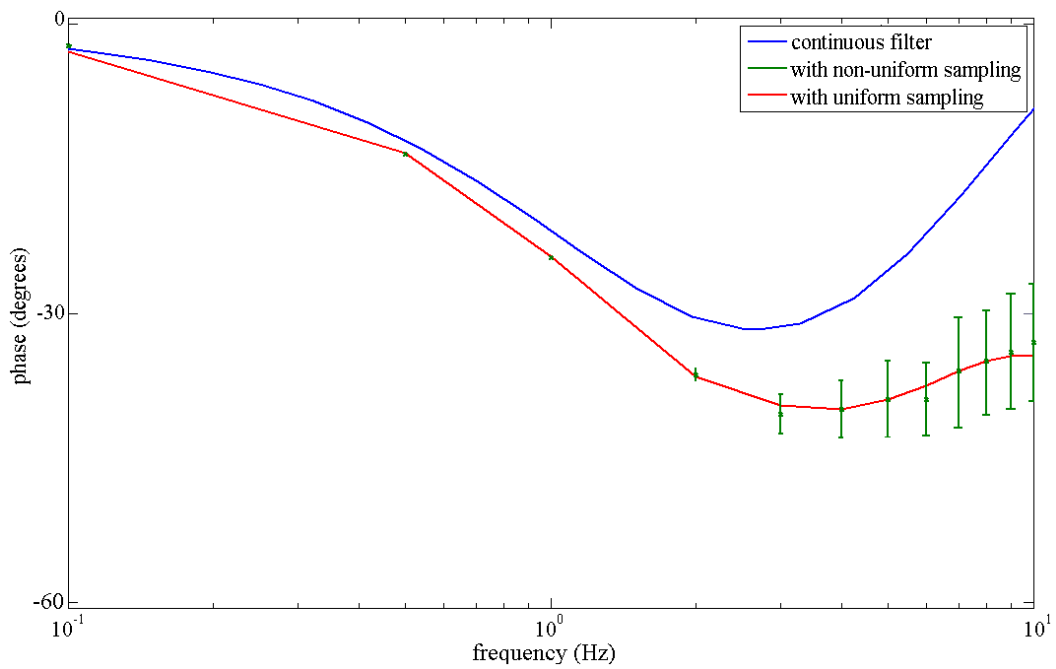


(b) The digital filter output to step input

Figure 3.28: The time response of the filter implementing sample rates according to a sin wave pattern i.e. $0.01s \leq T_n \leq 0.02s$

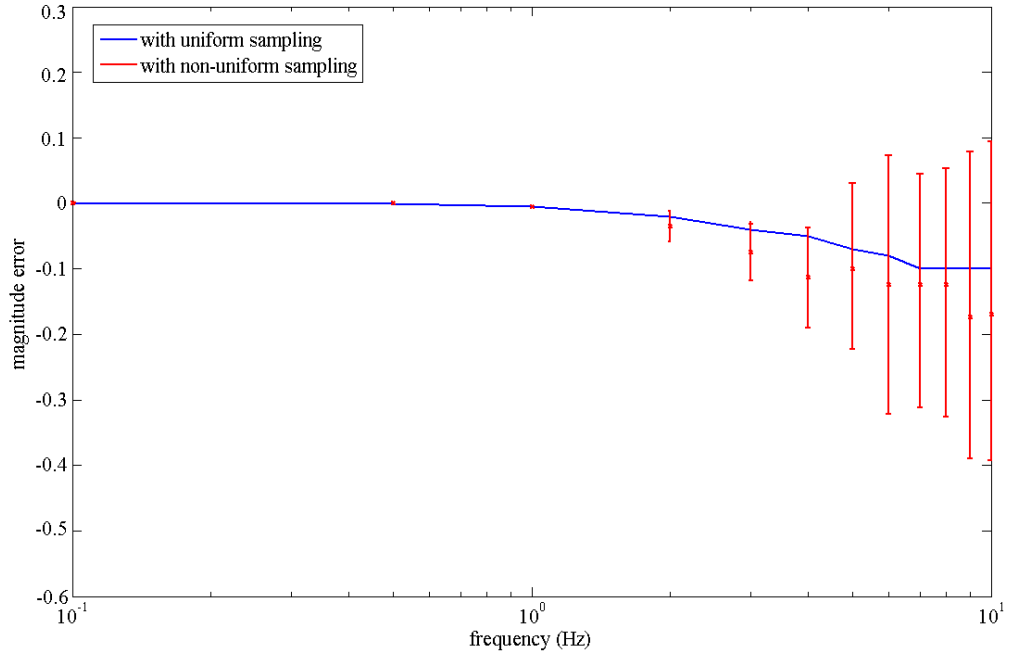


(a) Digital filter magnitude estimation

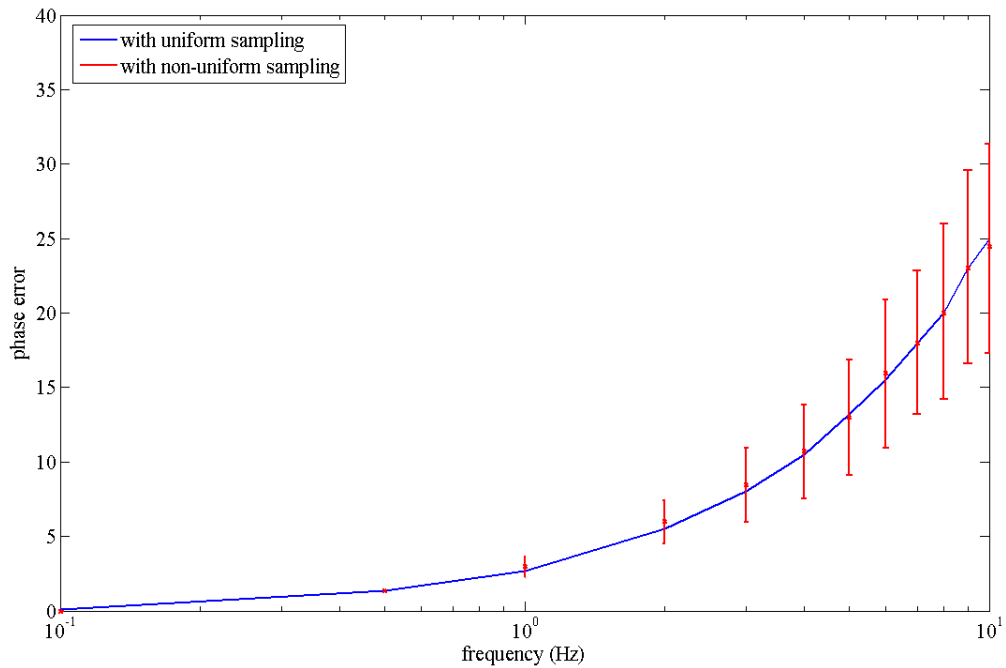


(b) Digital filter phase estimation

Figure 3.29: The frequency response of the filter implementing sample rates according to a sin wave pattern i.e. $0.01s \leq T_n \leq 0.02s$



(a) Digital filter magnitude estimation performance loss



(b) Digital filter phase estimation performance loss

Figure 3.30: The performance loss in the frequency response of the filter implementing sample rates according to a sin wave pattern i.e. $0.01s \leq T_n \leq 0.02s$

3.7 General conclusions

- A suitable digital filter structure is defined. The control algorithm is developed for implementing a non-uniform rate control system based on the modified canonic δ -filter structure. Moreover,
 - the control algorithm must be executed continuously in an infinite loop, and,
 - the sampling periods are predefined for the control program to select in each execution.
- The canonic z -filter structure was initially used for implementation, however, it was identified that it leads to the inclusion of undesirable transients in the output result.
- The modified canonic δ -transform and the direct- z filter structures are better at suppressing transients. This is largely due to way the internal variables are handled for processing.
- It is established that the transients are proportional to the amount of change in the sample rate. The severity of a transient signal depends on the filter input signal and the size of magnitude change in the filter coefficients.
- A combination of frequency domain and time domain analysis is used to obtain the dynamic response of a digital compensator. The technique, based on the Fourier analysis, can be used to evaluate the frequency characteristics under uniform and non-uniform sampling conditions. The Bode plot can now be obtained and be compared to observe the magnitude and phase response with that of the continuous-time system.

- From the frequency analysis of the non-uniform sampling schemes used, the magnitude plot seems to suffer from sudden gains. These are probably due to the reconfigurations or when the digital filter coefficients switch from one operational mode to another. These gains obviously depend on the amount of change in the sample rate.
- From the frequency analysis of the non-uniform sampling schemes used, the phase plot remains bounded in between the maximum and minimum sample rates being used in the experiment.
- The performance loss in the frequency characteristics are similar to the case with uniform sampling at low frequencies, i.e. it keeps increasing depending on the amount of variations induced in the sample frequency. This attribute remains the same for all non-uniform sampling distributions used. At higher frequencies however, the variations in the sample rates become much more evident largely due of the reconstruction aspects.
- Performing the analysis with an open-loop controller, it is concluded that varying the distribution of non-uniform samples seems to provide no added advantage¹⁶ in context to the phase response of the non-uniform rate control system.

This section applies the basic analysis methods involved in analysing the non-uniform rate control algorithm based on open-loop data. The next section will repeat the analysis with a much more complex controller and a DC motor model in a closed-loop.

¹⁶—at low frequencies.

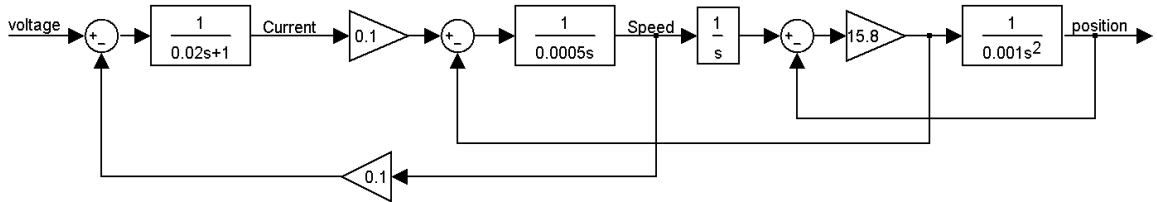


Figure 3.31: The DC motor block diagram

3.8 Practical example: closed-loop with DC motor

The following model example is taken from Wu, X (2005). The objective of the example is to control the position of a rotating load with some flexibility in the drive shaft. The block diagram of the DC motor model is show in Fig. 3.31.

The 4th order tracking controller includes a P+I, a phase advance and a notch filter to minimise the effects of resonance caused by the flexibility of the physical system. The continuous-time transfer function of the controller can be expressed as in Eq. 3.8.1:

$$H(s) = \frac{1 + 1000s}{1000s} \cdot \frac{1 + 0.1s}{1 + 0.01s} \cdot \frac{1 + 0.0025s^2}{1 + 0.005s + 0.0025s^2} \quad (3.8.1)$$

The step response of the closed-loop system is shown in Fig. 3.32.

Simulation setup

The overall control scheme is depicted in Fig. 3.33 and is implemented discretely with the uniform and non-uniform rate controller using the delta transform. The control algorithm implemented with the uniform controller is straightforward, i.e. the analogue signals are sampled and fed into the digital controller, that causes an update of the control signal to the motor at a

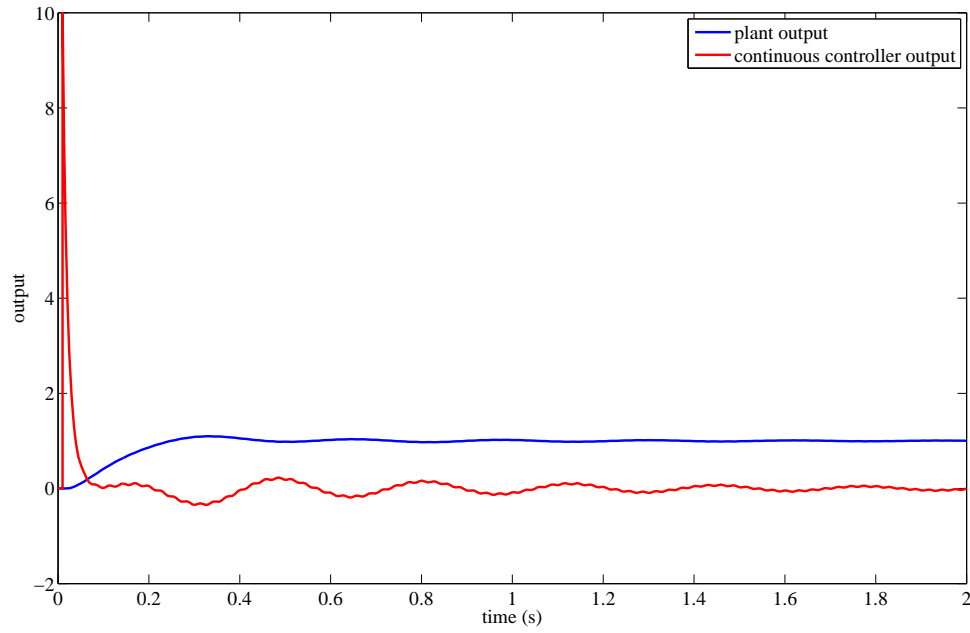


Figure 3.32: The step response with the continuous controller in a closed-loop

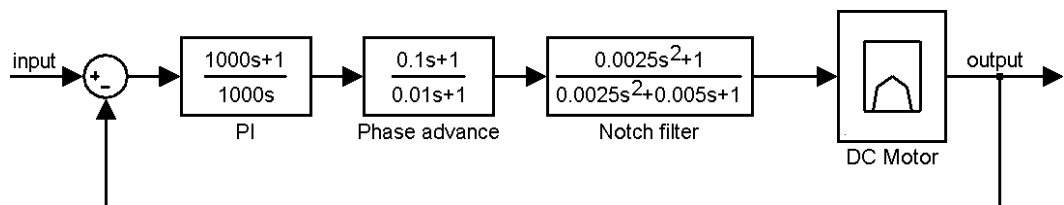


Figure 3.33: The overall control scheme

uniform rate. This operation is similar in the non-uniform rate equivalent, however, the main difference is that with the non-uniform rate controller, the parameters are continuously being attuned during run time, according to the sampling intervals, which are known before hand.

Therefore, in the analysis and design of non-uniform rate controllers, the following steps have to be considered:

- the plant process
- a set of sample periods¹⁷
- the closed-loop performance specifications

and the objectives are being to analyse if the non-uniform rate controller can map the performance of its continuous-time and the typical uniform rate counterpart's. The following closed-loop simulations include:

- uniform distribution of sample rates i.e. $T_n \sim U(0.01,0.02)$
- a truncated gaussian distribution i.e. $T_n \sim N(0.015,0.1^2)$
- dual sample rate (digital pattern) i.e. $T_n = 0.01\text{s}$ or $T_n = 0.02\text{s}$
- a sine wave pattern at 5Hz i.e. $0.01\text{s} \leq T_n \leq 0.02\text{s}$

¹⁷—which may or may not belong to some probability distribution.

Uniformly distributed sample rate

Consider the case when the non-uniform sample rates to be used, are generated such that all intervals have a uniform probability, where $T_n \sim U(0.01,0.02)$. The following simulations include the time response (in Figs.

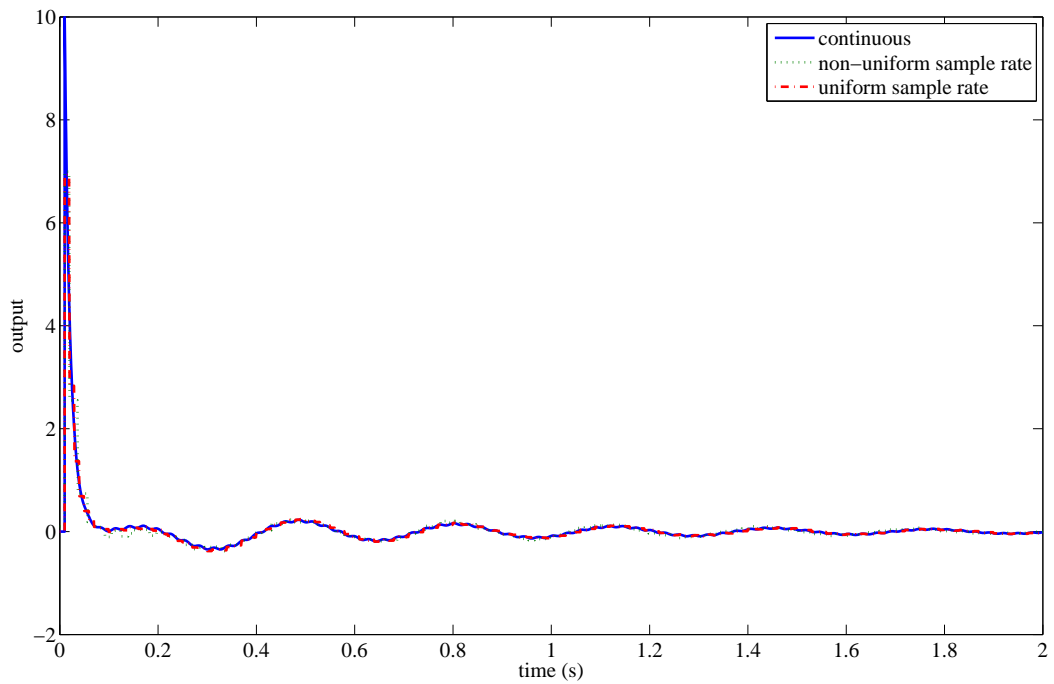


Figure 3.34: The digital controller output to a step input

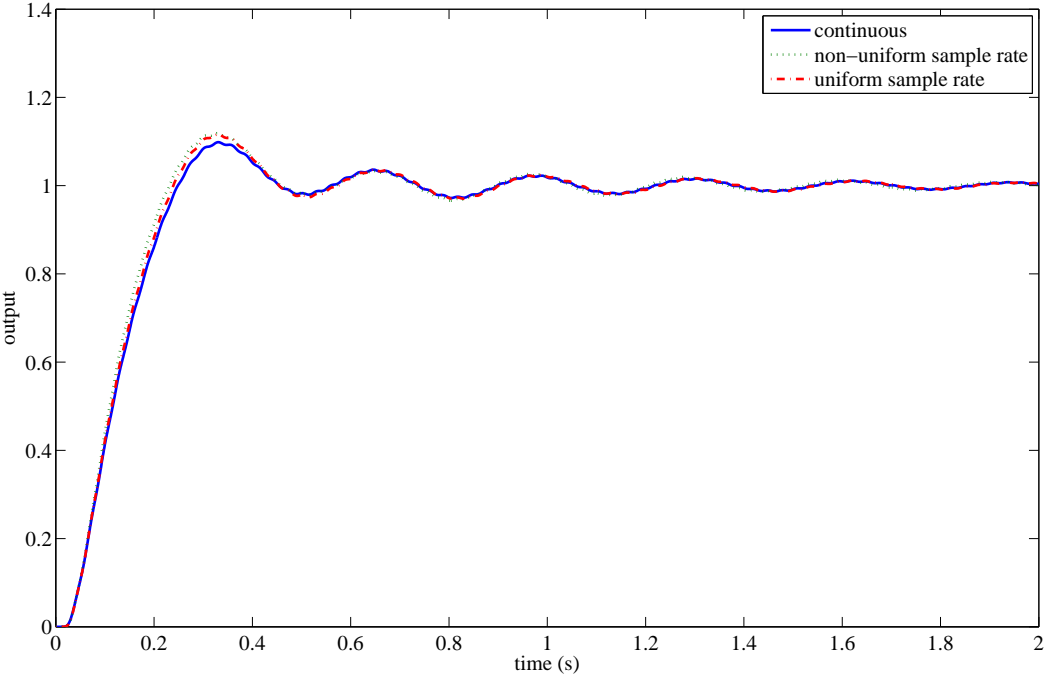
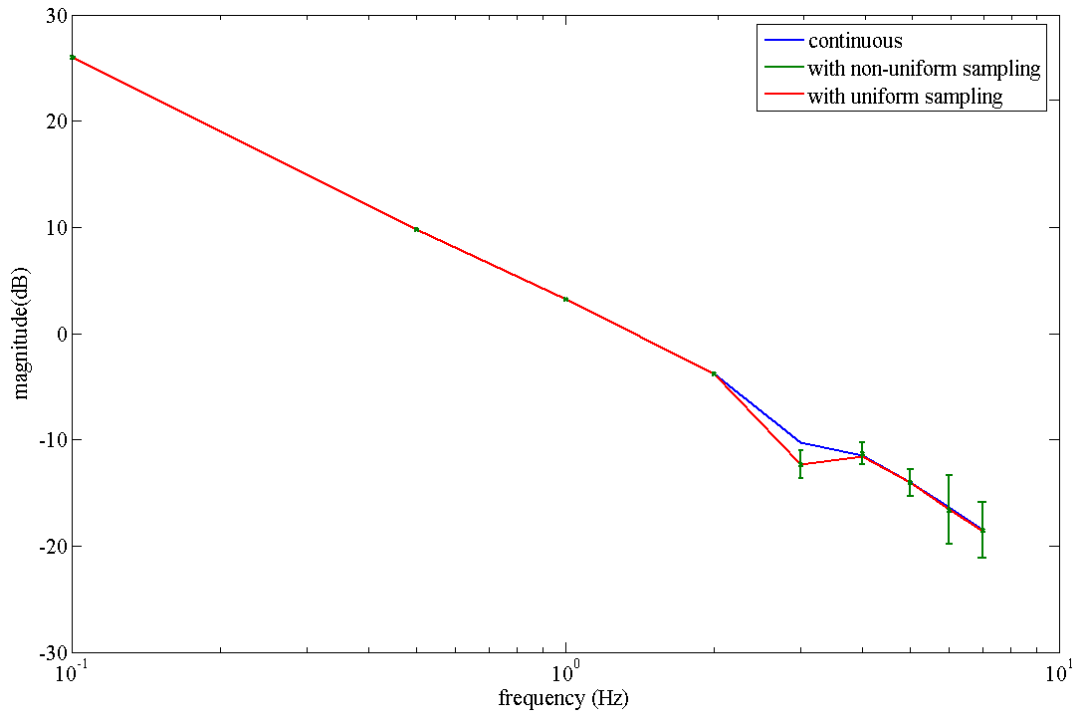
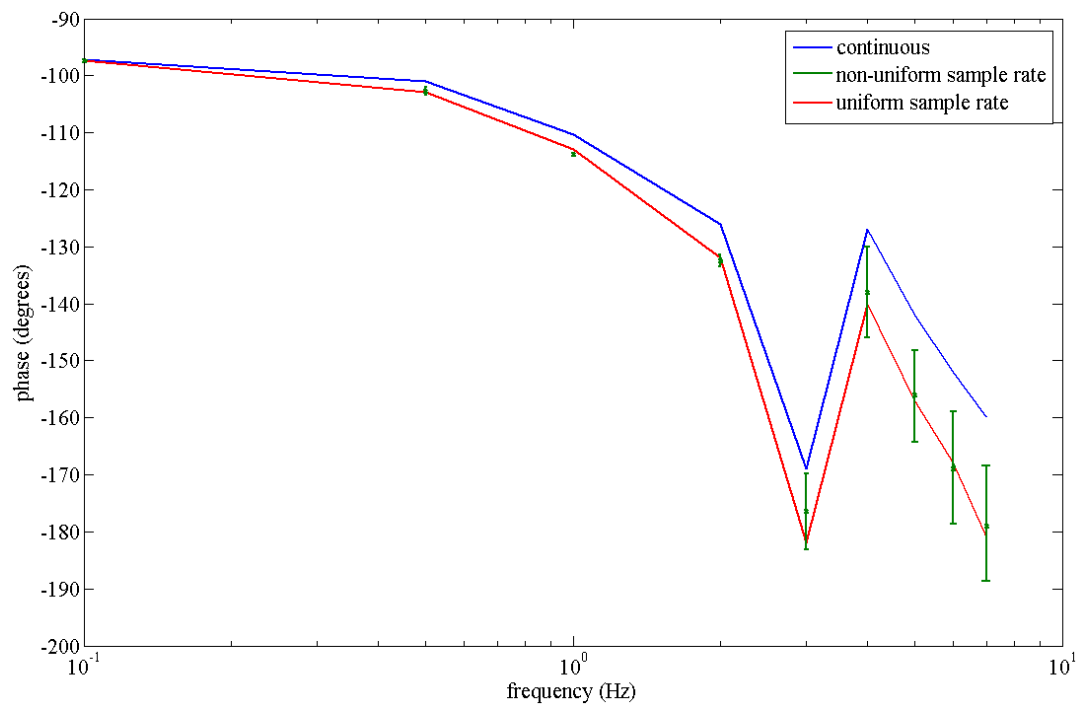


Figure 3.35: The plant output response to a unit step input



(a) The magnitude response



(b) The phase response

Figure 3.36: The frequency response of the closed-loop system implementing the digital controller with uniformly distributed sample rates i.e. $T_n \sim U(0.01,0.02)$

Truncated gaussian distribution

Consider the case when the non-uniform sample rates to be used, are generated such that most of the intervals are clustered around the mean sample rate, where $T_n \sim N(0.015, 0.1^2)$. The following simulations of the compensator include the time response (Figs. 3.37 and 3.38) and the frequency response

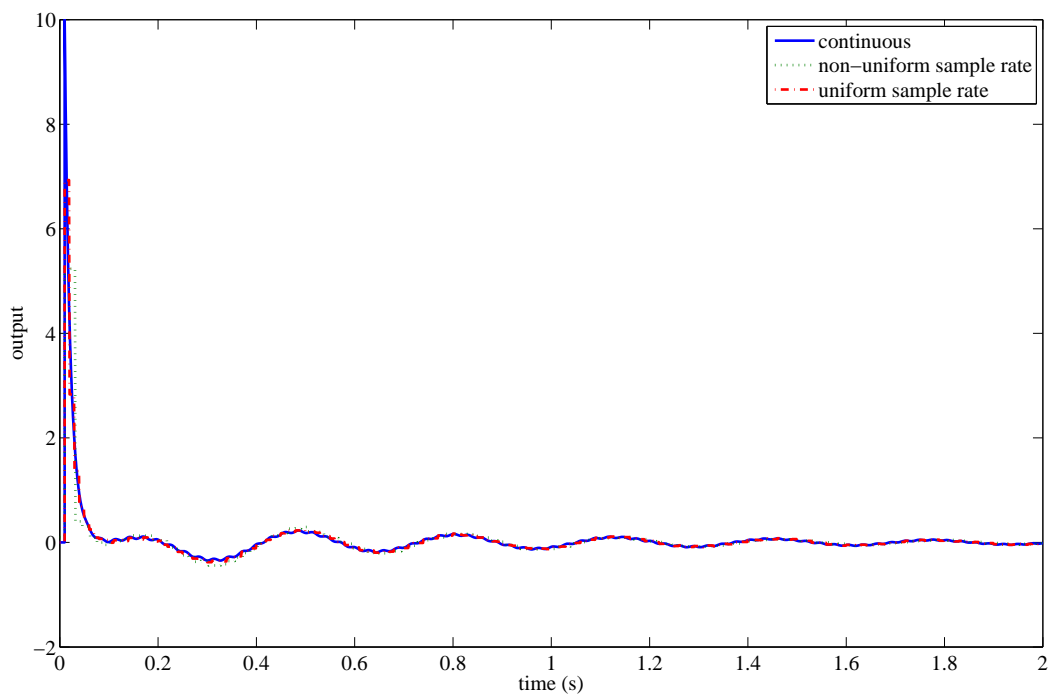


Figure 3.37: The digital controller output to a step input

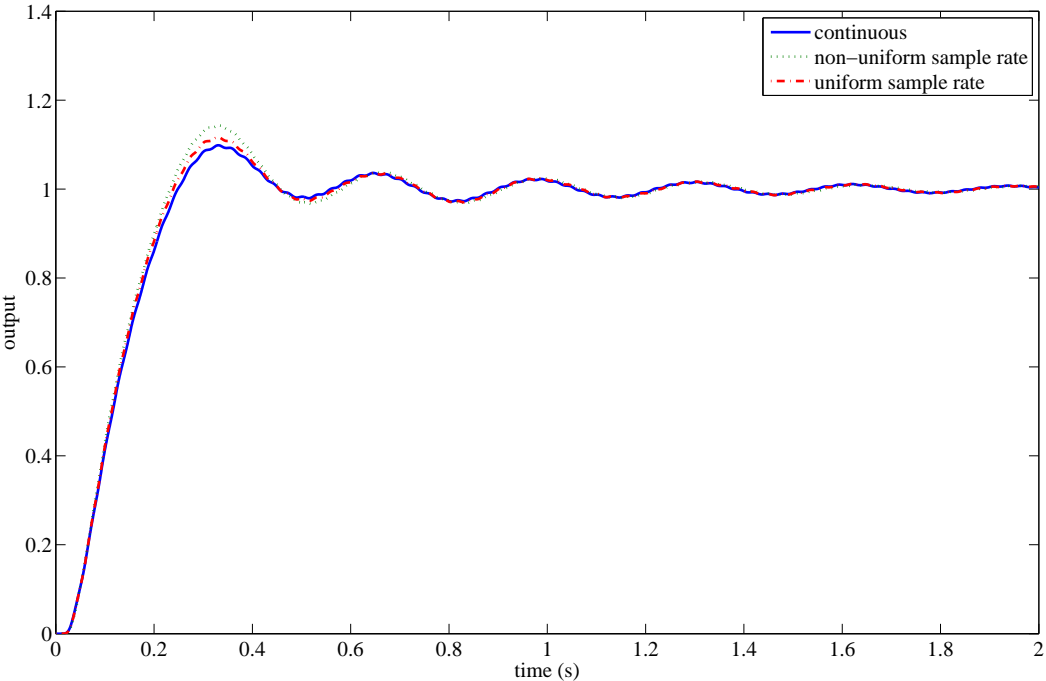
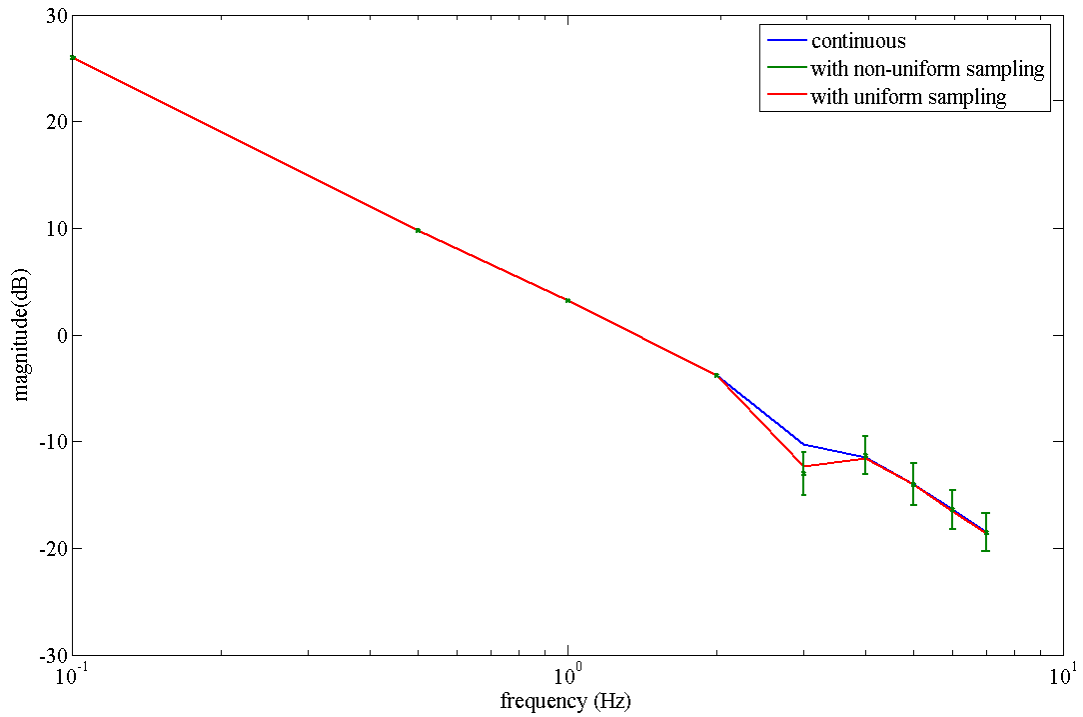
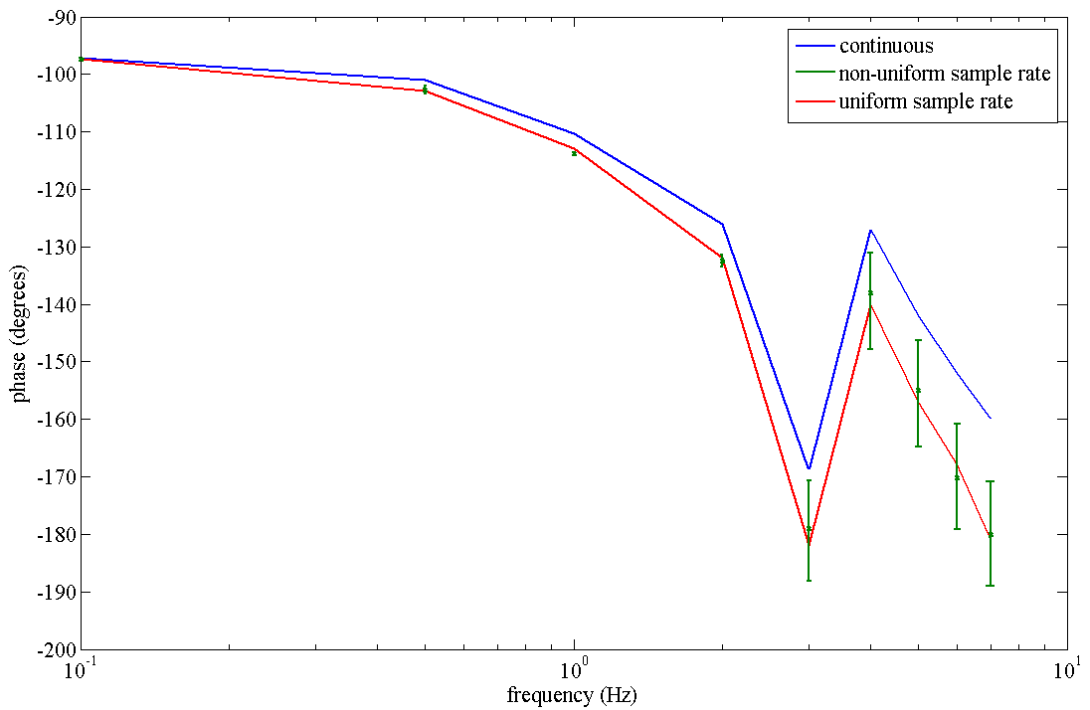


Figure 3.38: The plant output response to a unit step input



(a) The magnitude response



(b) The phase response

Figure 3.39: The frequency response of the closed-loop system implementing the digital controller with truncated gaussian distributed sample rates i.e. $T_n \sim N(0.015, 0.1^2)$

Dual sample rate

Consider the case when the non-uniform sample rates to be used, are generated according to a digital signal, where $T_n = 0.01\text{s}$ or $T_n = 0.02\text{s}$. The following simulations of the compensator include the time response (Figs. 3.40

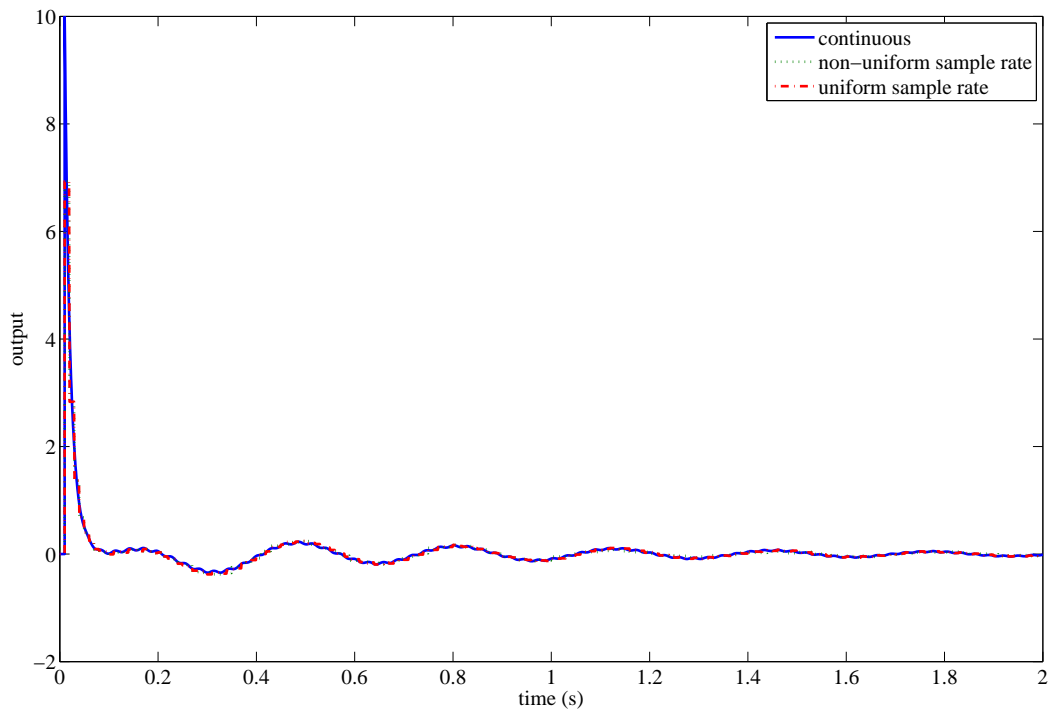


Figure 3.40: The digital controller output to a step input

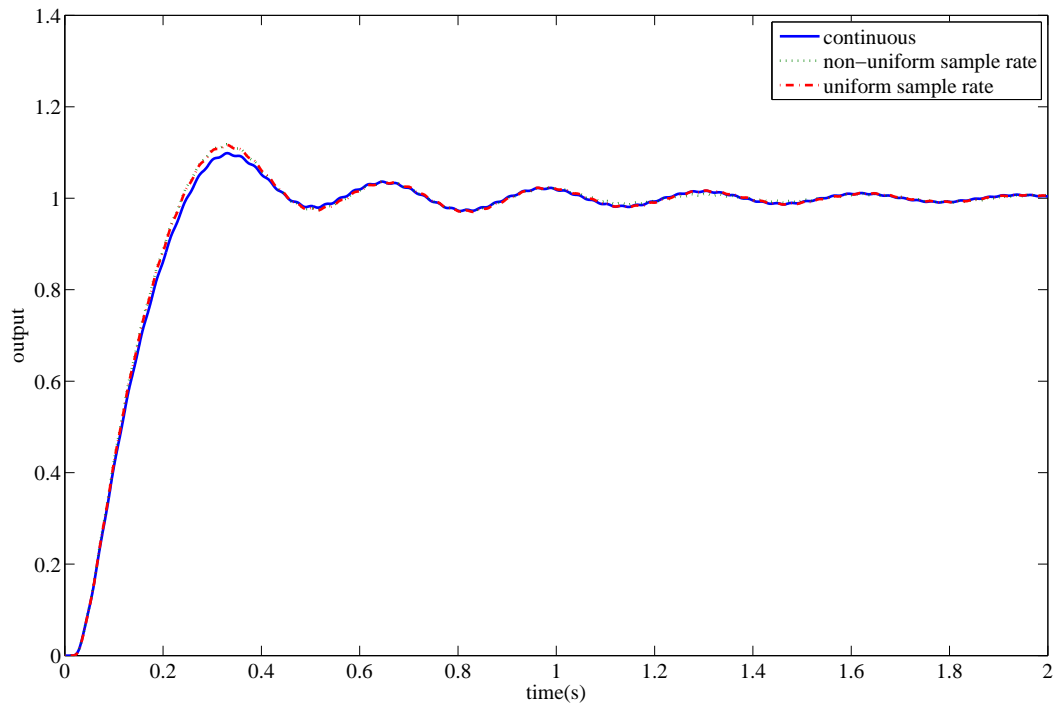
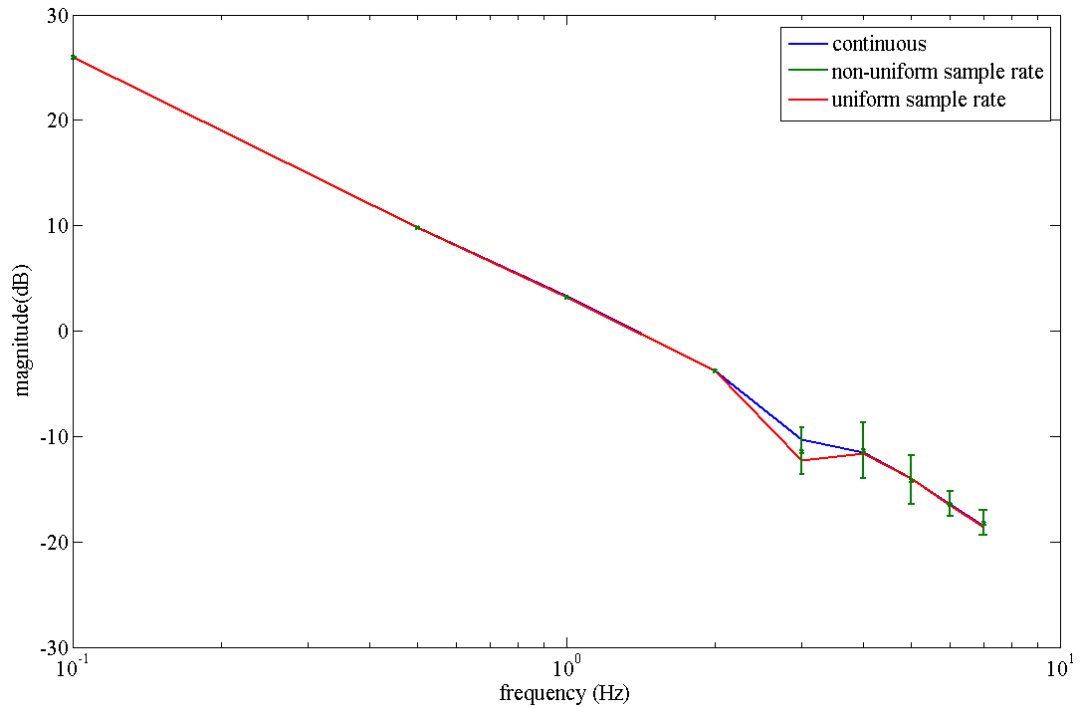
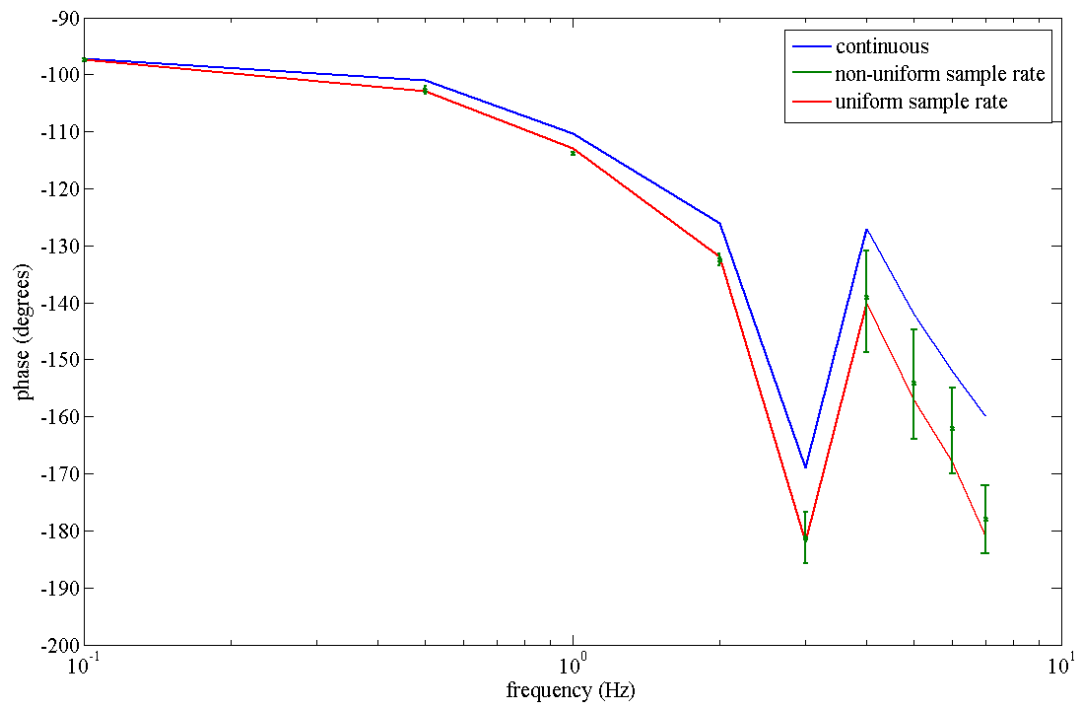


Figure 3.41: The plant output response to a unit step input



(a) The magnitude response



(b) The phase response

Figure 3.42: The frequency response of the closed-loop system implementing the digital controller with dual sample rate (digital pattern) i.e. $T_n = 0.01s$ or $T_n = 0.02s$

Sinusoid distributed samples

Consider the case when the non-uniform sample rates to be used, are generated according to a sinusoid signal pattern, where $0.01\text{s} \leq T_n \leq 0.02\text{s}$. The following simulations of the compensator include the time response (Figs. 3.43

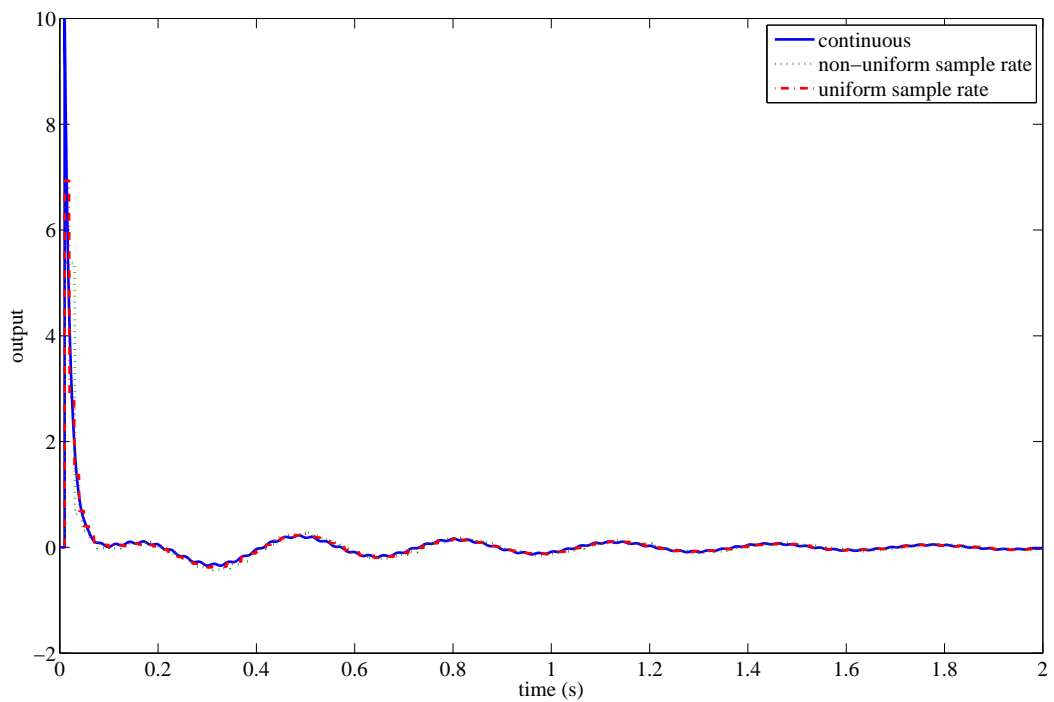


Figure 3.43: The digital controller output to a step input

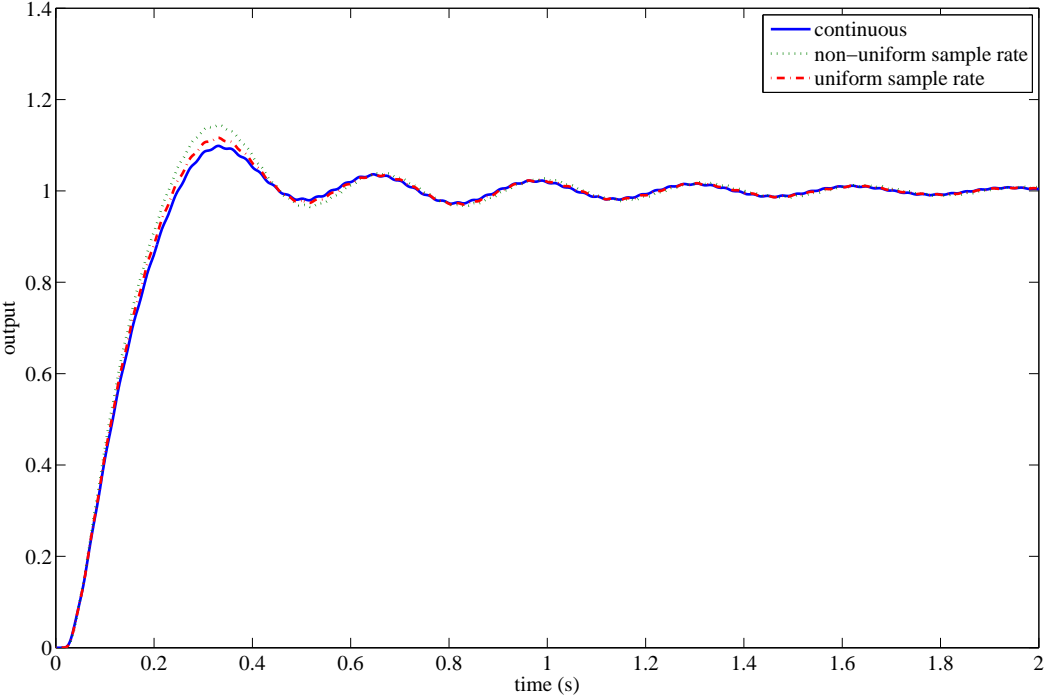
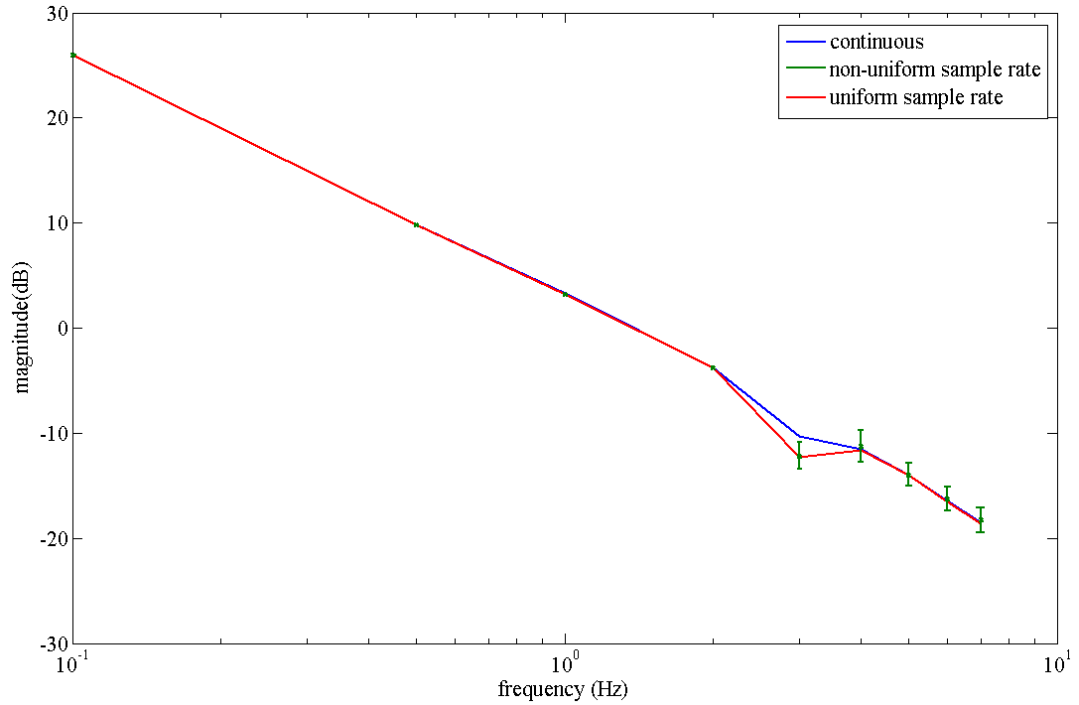
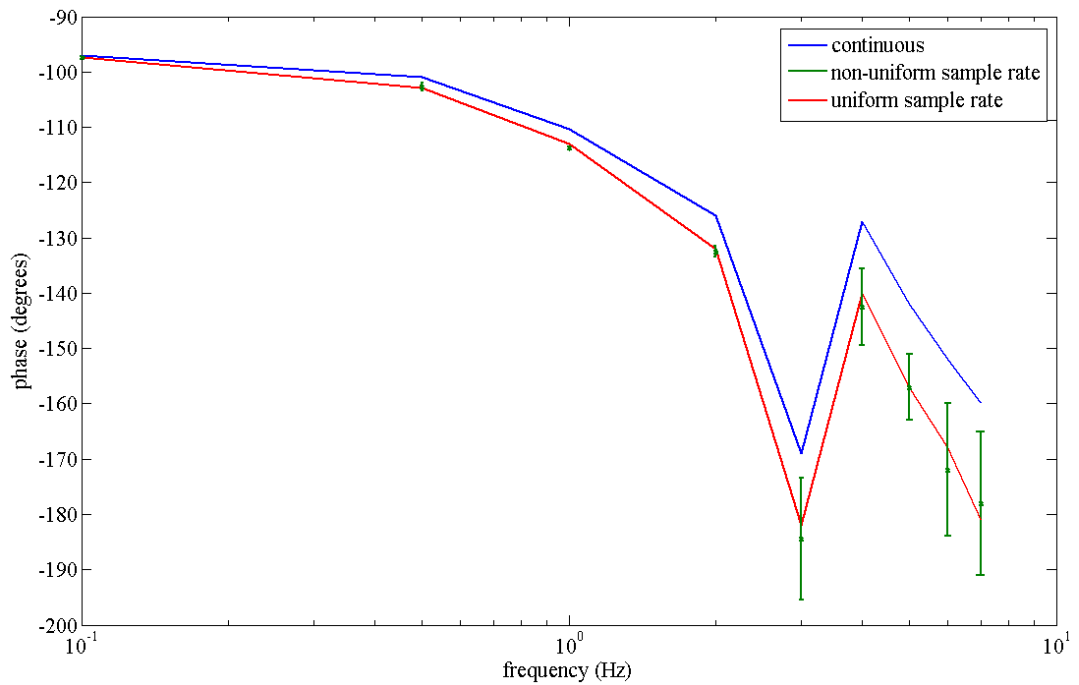


Figure 3.44: The plant output response to a unit step input



(a) The magnitude response



(b) The phase response

Figure 3.45: The frequency response of the closed-loop system implementing the digital controller with a sinusoid distribution i.e. $0.01s \leq T_n \leq 0.02s$

3.8.1 Discussion

The example provides an illustration of a relatively simple system which is controlled by using a non-uniform rate controller, designed using classical techniques. The simulation is implemented using the algorithm developed in this chapter. The frequency response of the control system is determined by using the Fourier analysis technique to estimate the magnitude and phase over time.

The exercise makes use of various sampling regimes to study the impact on the control performance. One apparent conclusion that can be deduced from the simulations carried out is that varying the sample rate does not have any benefit on the phase response of the system at low frequencies. In fact, as the operating frequency moves closer to the Nyquist limit, the variations become more evident¹⁸. Similar insights were not obvious before i.e. the more the variation in the sample rate distribution, the more the frequency response diverges at high frequencies. However, the simulation model provides a basis for very valuable testing of control systems with varying sampling periods prior to any practical implementation in terms of real hardware. This may particularly be of importance in safety critical applications such as flight control (Kopetz 1997).

¹⁸This is due the reconstruction process not having enough samples to reform the complete waveform.

3.9 Summary

This chapter examined various concepts for implementing time varying sample rates in digital control and proposed techniques for its analysis, design and evaluation. Control analysis is extended to the non-uniform sampling case and a controller design approach was presented based on the assumption of having predefined non-uniform sample rates. This *approach* is based on the same classic controller design methods used in control theory. It is interesting that the presented approach to the design of digital algorithms need not be confined to the case of uniform sampling; instead of specifying a single value for the sampling period at the design stage, several values can be specified. Then at each sample execution, the run time parameters can be adjusted according to specific sample period values.

This chapter further worked on identifying the limitations of using the digital implementation structures. Switching in between different sample rates introduced undesirable transients that can degrade the performance of the system. Subsequently, the direct- z and the modified canonic δ -transform were identified to be a better solution. Although the mean-square error analysis between the direct- z and the modified canonic δ revealed no distinction between the outputs produced by the two filter structures, the modified canonic δ structure is chosen for implementation in the remainder of this thesis.

Furthermore, an important technique for calculating the frequency response of a non-uniform rate control system was highlighted. A formal Fourier analysis can effectively be used on the system data to compute the magnitude and phase of a transfer function at various frequencies in the time domain. The technique presents an opportunity to calculate the frequency response characteristics of the continuous, uniform rate and non-uniform rate filter's

over time. For completeness, a performance evaluation criteria was also included that provides a measure of the error in the frequency response. It is noted that the absolute error keeps increasing at higher frequencies due to its association with the reconstruction process¹⁹. Such analysis can further be used to study the relationship/impact of switching between various sampling frequencies. Comparing the uniform sample rate and the average non-uniform sample rate results, it is concluded that non-uniform sampling does not provide any added advantage for improving the phase lag issues at low frequencies. The results are in fact very similar for all non-uniform sampling regimes (considered in this thesis), which may differ only at high frequencies due to the reconstruction aspects.

In digital control, most of the tools and techniques cited in literature are exclusively based on uniform sampling and hence the case for processing and analysing with non-uniform sampling becomes even more difficult. Nonetheless, the chapter has imparted methods of adapting digital controllers to suit non-uniform sample rates and evaluated their performance depending on various criterions. For completeness, a practical DC motor controller in a closed-loop has been used to demonstrate and examine the applicability of the control algorithm and analysis methods.

For the readers's convenience, the examples and demonstrations used in this chapter are summarised on page 130:

¹⁹There are not enough non-uniform samples available to reconstruct the complete signal.

	Sampling scheme	Topic	Implementation structure	Example/Analysis	Page
(i)	Constant sample rate	Algorithm validation	Canonic z -filter	B	64
(ii)	Uniformly distributed sample rate	Algorithm validation	Canonic z -filter	B	66
		Fourier coefficient analysis	Modified canonic δ -filter	C	87
		Bode plot	Modified canonic δ -filter	D	88
		Bode plot, performance evaluation	Modified canonic δ -filter	Open-loop	96
		Bode plot	Modified canonic δ -filter	Closed-loop	117
		Transient analysis	Canonic δ -filter		70
(iii)	Normally distributed sample rate	Transient analysis	Direct z -filter		75
		Bode, performance evaluation	Canonic z -filter	Open-loop	100
		Bode plot	Modified canonic z -filter	Closed-loop	120
(iv)	Dual sample rate	Transient phenomenon	Canonic z -filter	A	54
		Bode plot, performance evaluation	Modified canonic δ -filter	Open-loop	104
		Bode plot	Modified canonic δ -filter	Closed-loop	123
(v)	Sine wave distribution	Bode plot, performance evaluation	Modified canonic δ -filter	Open-loop	108
		Bode plot	Modified canonic δ -filter	Closed-loop	126

Chapter 4

Control algorithms for implementation

Chapter overview

The preceding chapter has presented the analysis and design methods of non-uniform rate digital controllers, mainly from an analytic and theoretical point of view. In this chapter, an equally important aspect in the software implementation of the non-uniform discrete-time equivalent is discussed. It outlines, in detail, the embedded software development in C/C++ and carries out two implementations which include an 8051 microcontroller and an FPGA board. The objectives of this chapter can be summarised as:

- To demonstrate and validate the proposed approach through the hardware implementation of the non-uniform sampling algorithm.
- To verify the control program: an overall structure of the program that implements the control algorithm is defined and programmed. This will determine the order of in which the initialisations, control loops and input sampling have to be implemented.
- To describe the process used to map the control algorithm directly onto a hardware structure.

§4.2 and onwards is concerned with hardware design aspects. These include the verification of the control algorithm on a microcontroller unit and definition of the Register Transfer Level (RTL) description to enable implementation of non-uniform sampling schemes with FPGAs. The concept of hardware-in-the-loop systems is discussed and an experiment is followed up with an industrial standard target controller.

4.1 Software considerations

§2.1.1 established that computational tasks¹ realised in control algorithms, are usually implemented by treating their execution times and periods as unchangeable parameters. On the other hand, controller design is primarily based on the continuous-time dynamics of the physical system being controlled.

Although, software and hardware considerations can be dealt with separately, there might be a strong interaction between the two. Such interactions occur not only at the stage of deciding upon the particular processor to be used, but also when it comes to the more detailed design aspects. In general, an integrated approach can be adopted that allows a certain tolerance for variations in the execution periods, as long as a change does not affect the critical control functions² (Marti, Fuertes, Fohler & Ramamritham 2001).

Forsythe & Goodall (1991) argue that there are three main components specific to the context of digital control; i.e. the overall structure of the software, the numerical routines and the programming language used for the actual coding of the software³. The discussions point out the fact that in digital

¹Such as coefficient calculations and operations.

²Such as stability.

³There are many other comprehensive texts devoted to provide an excellent guidance on this subject, see eg. Houps & Lamont (1985) and Kuo (1980).

systems, there are physical limitations due to the inherent discrete nature, *especially* on the sampling period. For instance, the sampling period of a controller is governed by the clock rate and how fast the numerical operations and instructions are executed by the digital processor. Therefore, in the case with a non-uniform sampling routine, the speed of execution might be relatively slow and this imposes an inherent upper limit on the fastest sampling instant based on the hardware chosen for implementation.

4.1.1 Algorithm design

The basic modules for the digital controller are typical i.e. the input data, the processing/filtering and the output data. Predominantly, there can be four routines that will implement the non-uniform rate controller, which are listed below:

Function	Description
MAIN	The executive routine that initialises the conversions and then enters an infinite loop calling the input, filtering, non-uniform sampling instances and the output
DELAY	Delays the filtering operation depending on the value stored in the register
NUS	This stores a set of sampling time instances that will be used by the DELAY routine. It loads the sampling periods into the register
FILTER	This is the digital controller. The routine is used to update the coefficients sets depending on the sample instance acquired by the NUS routine. It then performs the necessary filtering operations on the recalculated coefficients of the difference equation and updates the register's former values.

Note that the control algorithm requires the recursive execution of a set of instructions and the program will update its sample time variables in each MAIN execution. This will enable the filter coefficients to be recalculated for each sample period variation accordingly. However, even though the number of instructions in the control loop may be small in the case of implementing a simple controller, the overhead that manipulates the program counter may be relatively large.

Control program dataflow diagram

Fig. 4.1 depicts the control program flowchart, which summarises the controller program operation. The initialisation block initialises all the control variables. The sampling period is loaded from a look-up table that stores a number of sample rates that are being used during the implementation. Moreover, an internal clock keeps running in order to count until it is time to execute the control program. Once the sample time is reached, the program then enters the implementation stage where the input is sampled using the analogue-to-digital converter (ADC), to be processed by the control algorithm. After the necessary calculations, the output value will be provided to the digital-to-analogue converter (DAC) and the loop will keep repeating itself. The sample rate regulations depend on two blocks i.e. the '*update sample_time*' and '*sample_time reached?*'. Their operation is expanded in Fig. 4.2, which shows how the sampling setup works.

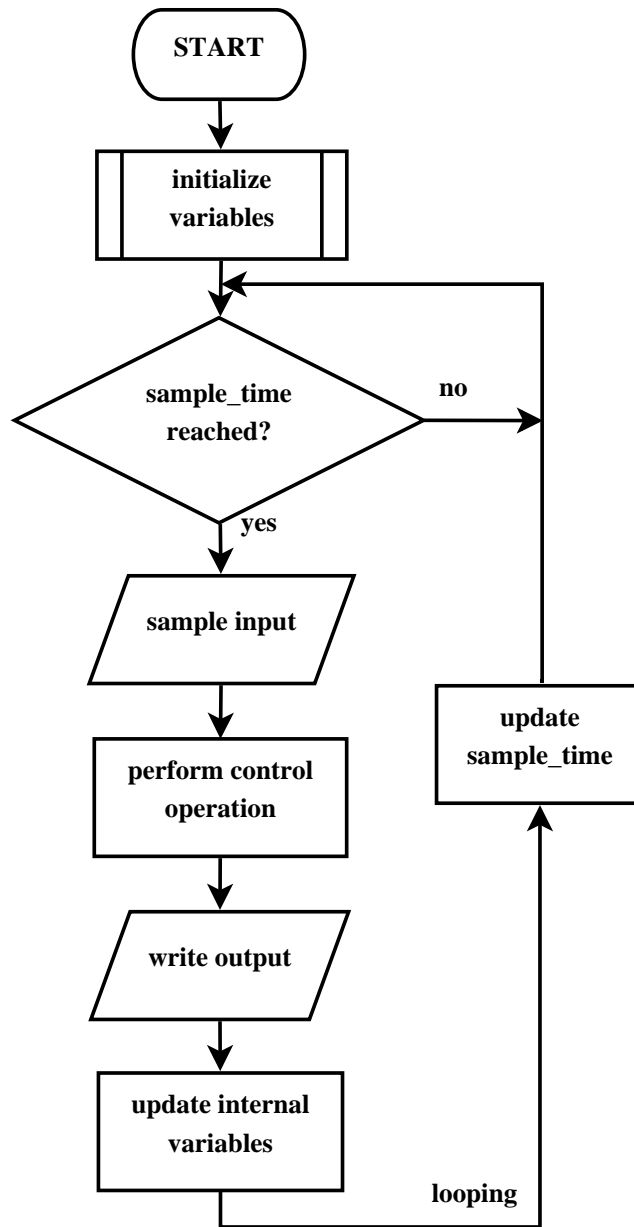


Figure 4.1: Control program dataflow diagram

Sample rate regulation dataflow diagram

The sample rate is regulated according to the dataflow shown in Fig. 4.2. The timer produces a sampling clock which is called *sample_clk*. Before the data can be loaded, the *sample_clk* is initialise at 0, and the sample period value is loaded in the variable *sample_time* from a look-up table. When the program is started, the counter keeps incrementing, and its value is compared to the *sample_time* value in every clock cycle. As soon as the counter reaches the *sample_time* data, the *sample_clk* counter is reset back to 0 and a new sample period value is loaded into the *sample_time* variable.

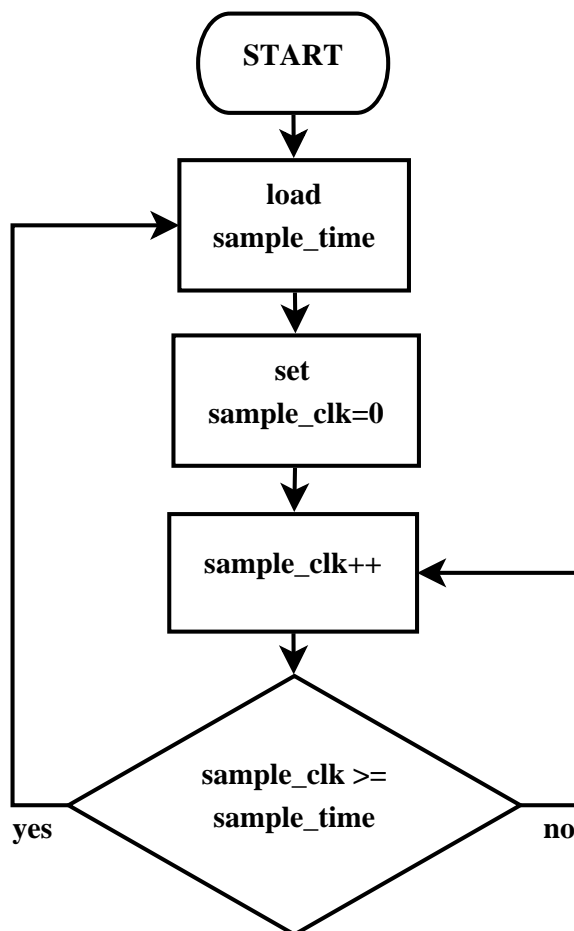


Figure 4.2: Sample rate regulation dataflow diagram

4.1.2 Embedded software development

The code to be implemented on the digital controller can be written in many programming languages and be executed in various types of hardware; for instance, C/C++ is commonly used by many control engineers for programming embedded systems (Svenk 2002). In this case, the Keil Development tools (Keil 2004) package has been used to design the application code for an 8051, a commonly-used microcontroller.

Consider a 2^{nd} order digital IIR filter, where the overall procedure is carried out by specifying functions that include a ranges of internal variables and operations. A C/C++ program can be used to list the actual computations and illustrate the real-time processes involved. Lst. 4.1 to 4.3 list the key program routines written⁴.

⁴—i.e. the MAIN, NUS and FILTER routines, respectively.

```
1 main() {
2     int i=1,time=0,count=0;           /*initialise variables*/
3     float y,u;
4     if (sample_clk>=sample_time)     /*wait for sample time*/
5     {
6         u=input_adc();                /*input data*/
7         y=filter(sample_time);        /*control operations*/
8         output_dac(y);                /*output data*/
9
10        if (i==6)
11            {i=1;}
12        else
13            {i++;}
14
15        sample_time=int_clk*nus(i);    /*new sample period*/
16        sample_clk = 0;                /*reset counter*/
17    }
18    sample_clk++;
19 }
```

Listing 4.1: The MAIN routine (includes the DELAY operation)

```
1 float nus (int i) {
2     float j;                           /*initialise variable*/
3     float ts[6]=[0.01,0.05,0.01,0.02,0.04,0.03];
4                                           /*stored sample times*/
5     j=ts(i);                             /*changing sample period*/
6     return j;                             /*return new value*/
7 }
```

Listing 4.2: The NUS routine

```

1 float filter (float ts, float in) {
2     float c0,c1,c2,r1,r2;
3     float p,q,r,d1,d2,v,w,x,u,y,ts_1;
4     float n0,n1,n2,m1,m2;           /*cont filter coeffs*/
5                                     /*recalculate coeffs*/
6     c0=(n0*ts*ts+2*n1*ts+4*n2)/(ts*ts+2*m1*ts+4*m2);
7     c1=(2*ts*ts*n0)/(ts*ts+2*m1*ts+4*m2)+(2*n0*ts_1*ts_1+4*
8         n1*ts_1)/(ts_1*ts_1+2*m1*ts_1+4*m2);
9     c2=(4*n0*ts*ts_1)/(ts*ts_1+m1*(ts+ts_1)+4*m2);
10    r1=(2*ts*ts)/(ts*ts+2*m1*ts+4*m2)+(2*ts_1*ts_1+4*
11        m1*ts_1)/(ts_1*ts_1+2*m1*ts_1+4*m2);
12    r2=(4*ts*ts_1)/(ts*ts_1+2*m1*ts+4*m2);
13
14    p=c0;
15    d1=r1;
16    q=c1/d1;
17    d2=r2/d1;
18    r=c2/(d1*d2);
19
20    v=in-w-x;
21    y=p*v+q*w+r*x;                 /*calculate output*/
22    x=x+d2*w;                       /*perform delta*/
23    w=w+d1*v;
24    ts_1=ts;                         /*store previous sample rate*/
25    return y;                         /*return output value*/
26 }

```

Listing 4.3: The FILTER routine of a 2nd order filter

Notes

- The program MAIN starts executing the standard initialisations and then enters an idle mode, waiting for the sample time.
- When the counter (*sample_clk*) reaches the sample time value, the program samples the input data and performs the filtering operations.
- The variable *int_clk* depends on the value of the clock crystal being used.
- The variable *sample_time* in Lst. 4.1 line 4 is initialised with a default sample period before entering the main program. It is also scaled by multiplying it with the internal clock value on line 15.
- The set of sample periods is stored in the NUS function variable *ts*[6], which as shown is an array with 6 sample values (although it can have more sample periods as required).
- An important operation is to pass the *sample_time* value to the FILTER function in order to recalculate the coefficient values; before performing the control operations
- At the very end, the routine outputs the data and resets the counter *sample_clk*. It then repeats the whole program again
- The equations can be modified for higher order filters, however, the overall approach presented will remain the same i.e. changing the sample period, recalculating new values, etc.

Computational requirements

As it is known that all of the steps involved in the required implementation consist largely of multiply or multiply-accumulate, which must be performed

within one sampling interval, it is important to consider other possibilities to reduce this burden. One option is to make use of stored coefficient values as look-up tables. If look-up tables are used, they would free the controller from the recalculation operations of the filter coefficients. On the other hand, if look-up tables are not used, the controller will need to do extra processing to adjust the coefficient values.

4.2 Microcontroller implementation

This section deals with the design and implementation of a microcontroller for tracking control of a real physical system or ‘plant’. The objective is to force the plant output to follow a given reference input with zero steady-state error. The plant of the digital control system under test is a linear actuator⁵, described in detail below.

4.2.1 Plant description

Fig. 4.3 illustrates the basic components of a moving coil electro-magnetic actuator. It comprises a moving coil wound round the centre pole of a magnetic assembly that produces a uniform magnetic field perpendicular to the current conducted in the coil. On providing a voltage, a current flows in the coil generating a force which is parallel to the direction of travel. This force causes the coil, and the rod which is mounted to it, to move. The force is proportional to the current in the coil, the number of turns, and the flux strength. End-stop ‘bumpers’ are set at each end of the travel to cushion any impact force that may occur.

⁵The moving coil actuator has been provided by SMAC UK Ltd. (SMAC 2004).

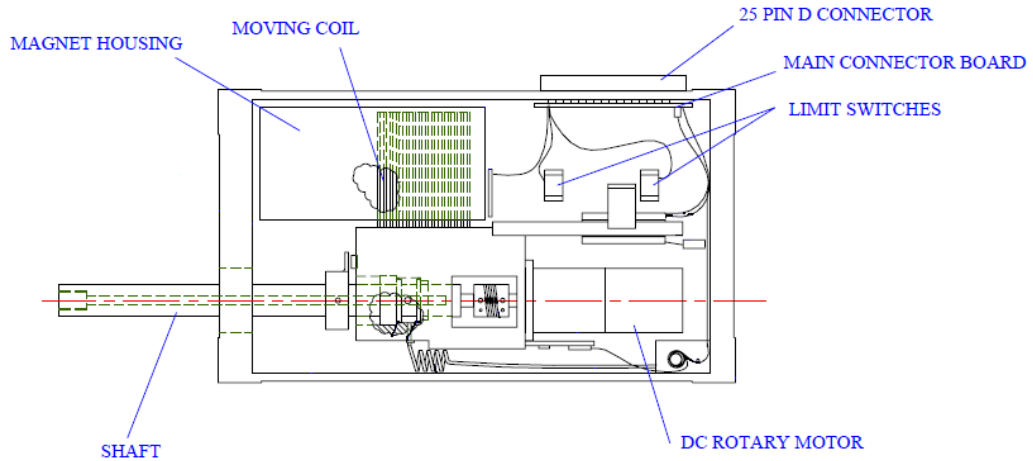


Figure 4.3: The SMAC actuator

4.2.2 Hardware realisation

Microcontrollers are often focused on integrating the peripherals needed to provide control within an embedded environment. Commonly, they incorporate the necessary components like the CPU, memory, timers, interfacing mechanisms, etc. These features allow them to be used as off-the-shelf solutions. Generally, they can provide an inexpensive programmable logic control. As a result, when connected in complex environments, they can be used to interpret inputs, communicate with other devices, and output to a variety of different devices. Such attributes add a great deal of flexibility in the development process. The control strategy described in §4.1.2 can practically be implemented by means of the standard 8051 8-bit microcontroller unit.

Fig. 4.4 gives a hardware description of the overall closed-loop system. The transducer output is measured and scaled to a $-5 \rightarrow +5V$ range on the ADC. This value is then subtracted from the reference signal by the microcontroller unit to produce an error signal. The error signal is then processed by the control algorithm begin implemented by the 8-bit microcontroller to produce a control signal. This signal is passed to the actuator through the DAC.

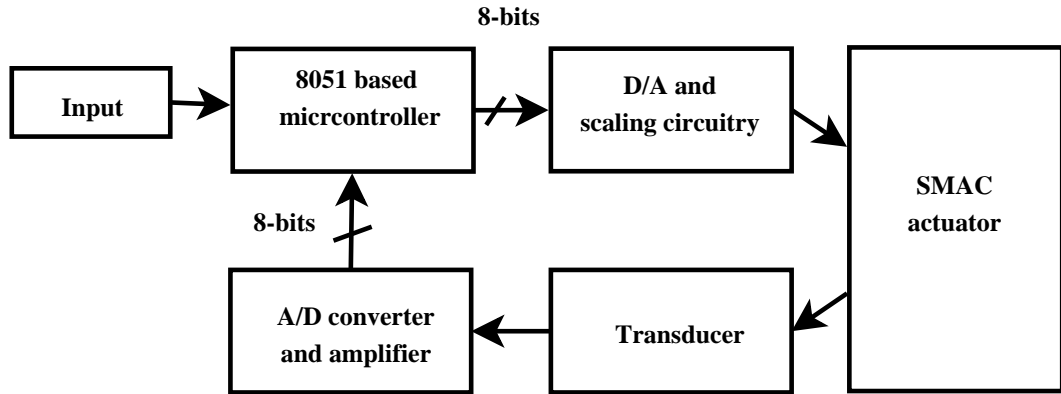


Figure 4.4: Block diagram of the actuator control system

4.2.3 Hardware simulation: open-loop

The controller under test belongs to §3.8 Eq. 3.8.1 on page 112. It is a 4th order controller consisting of a proportional-integral, phase advance and a notch filter. The transfer function is:

$$H(s) = \frac{1 + 1000s}{1000s} \cdot \frac{1 + 0.1s}{1 + 0.01s} \cdot \frac{1 + 0.0025s^2}{1 + 0.005s + 0.0025s^2}$$

The transfer function includes a notch filter — this would be used to control a resonant mode in the system which in fact does not exist with the actuator chosen, but the more complex 4th order controller is chosen anyway to demonstrate its applicability for a control algorithm with higher order controllers.

The implementation algorithm is initially simulated under the Keil development platform. The output results are shown with constant sampling in Fig. 4.5 and with non-uniform sampling in Figs. 4.6 and 4.7, for a step input.

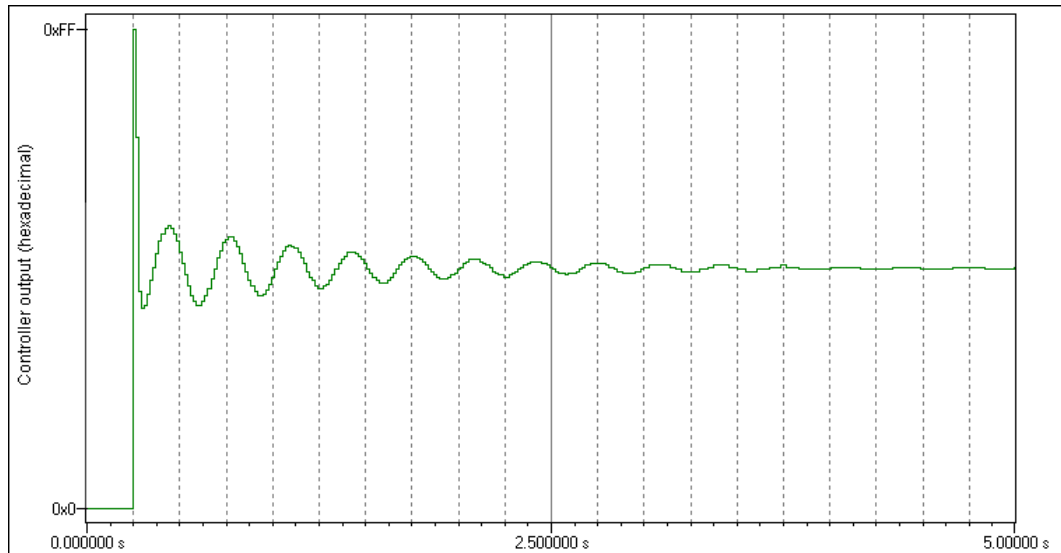


Figure 4.5: The response of the control signal to a step input, with constant sampling $T_n=0.015$ s

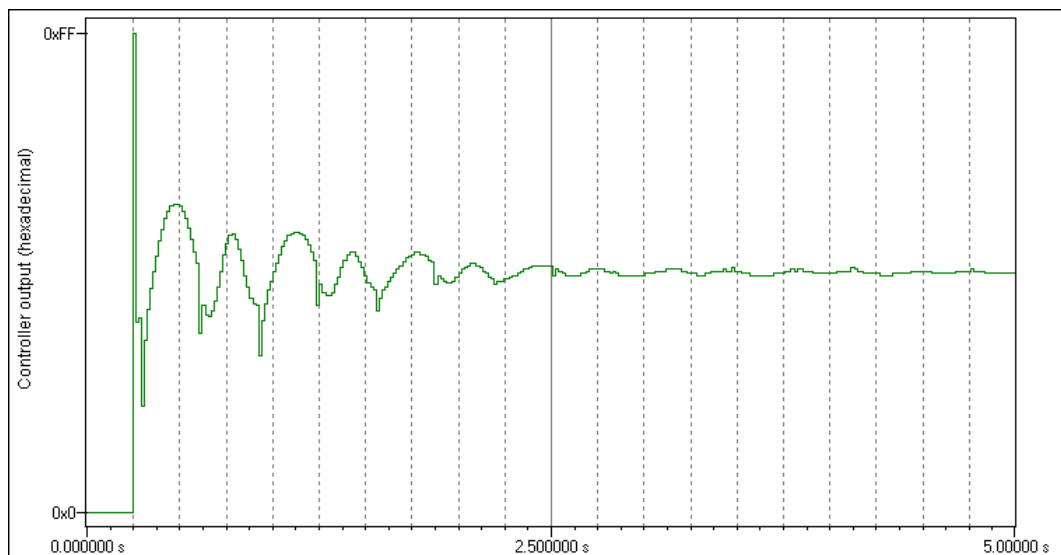


Figure 4.6: The response of the control signal to a step input, with non-uniform sampling period: dual rate sampling $T_n = 0.01$ s, 0.02 s, average sample rate= 0.015 s

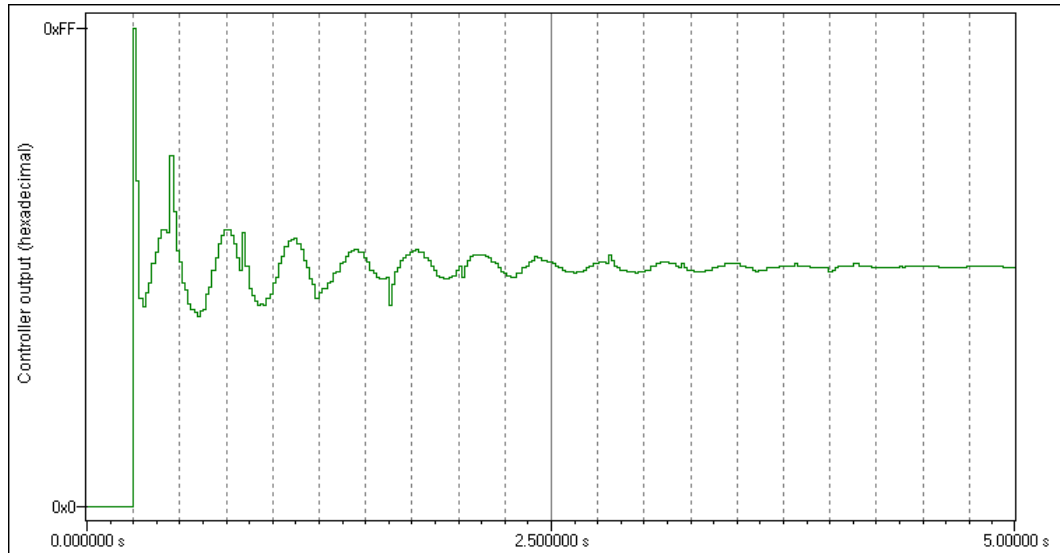


Figure 4.7: The response of the control signal to a step input, with non-uniform sampling period: uniformly distributed sample rates (Sawtooth wave pattern) $T_n \sim U(0.01, 0.02)$, average sample rate=0.015s

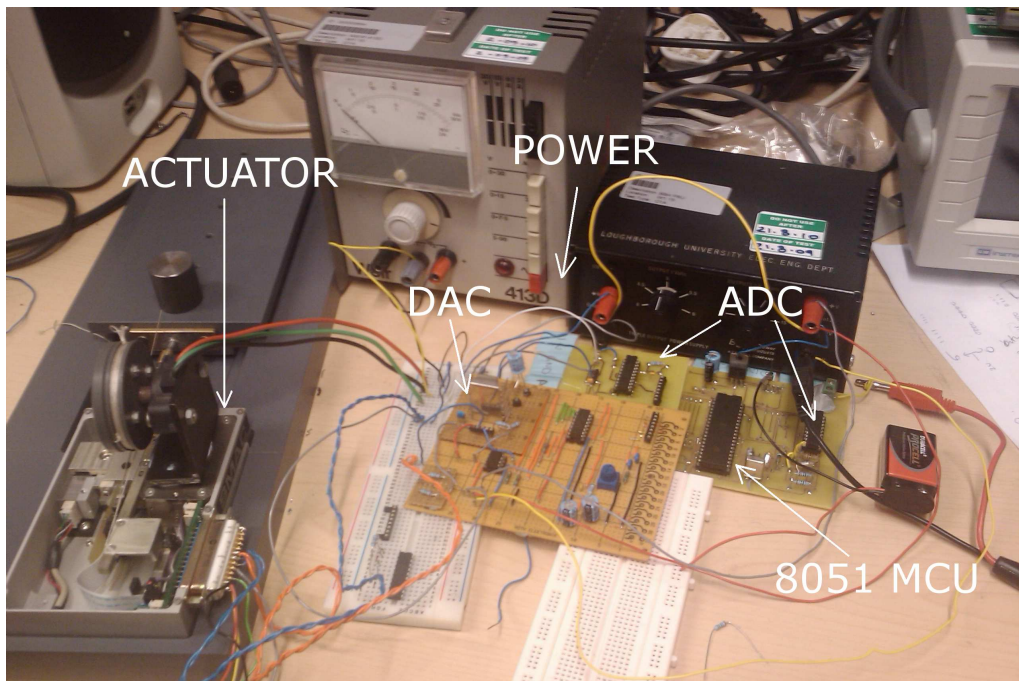


Figure 4.8: The experimental setup: includes the digital controller and the actuator.

The hardware simulations demonstrate that the non-uniform sample rate control algorithm performs similarly to simulations conducted in Chapter 3. Remnants of suppressed transients are visible in Figs. 4.6 and 4.7 at the point of sample rate change. These however, are not too problematic now since the implementation structure of the control algorithm is able to suppress them⁶.

4.2.4 Experimental Results: closed-loop with SMAC actuator

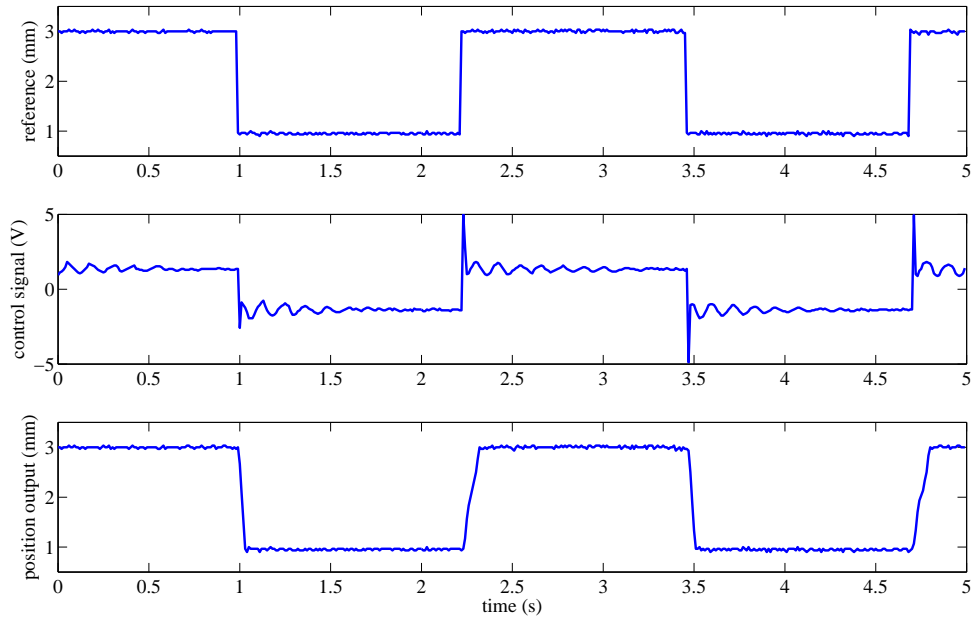
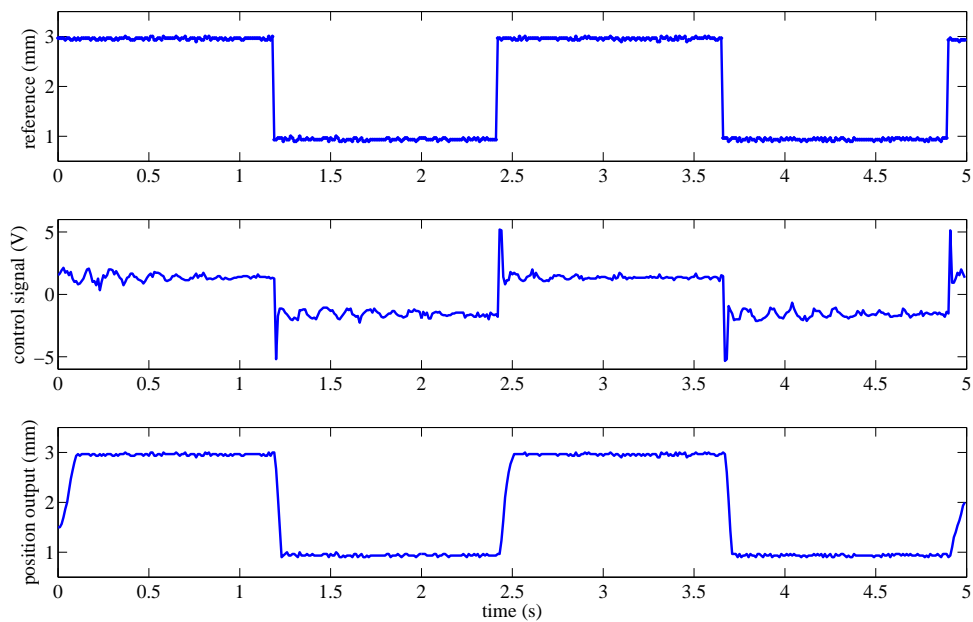
The following is a tracking performance experiment of the controller. It considers the output with respect to step input set-point changes. The experimental setup is shown in Fig 4.8.

Fig. 4.9 shows the result with constant sampling $T_n=0.015\text{s}$. The small dynamic variations at the nominally constant position sections of the output are the result of the limited (8-bit) position of the ADC converter. In addition, since the reference signal is generated externally and sampled, the result is slightly noisy. Figs. 4.10 and 4.11 show the outputs obtained with non-uniform sample rates: dual sampling (digital wave pattern) and uniformly distributed samples (sawtooth wave pattern), respectively. It can be seen that the output responses have very similar characteristics to those with uniform sampling.

Discussion

The closed-loop experiment with the SMAC actuator was used to validate the non-uniform control algorithm developed earlier in this thesis. It makes use of a uniform and two non-uniform sampling schemes (dual rate sampling and a sawtooth wave pattern, respectively) to track the reference input.

⁶—as discussed in §3.3.1.

Figure 4.9: With constant sampling period: $T_n = 0.015s$ Figure 4.10: With non-uniform sampling period: Dual rate sampling, $T_n = 0.01s, 0.02s$, average sample rate = $0.015s$

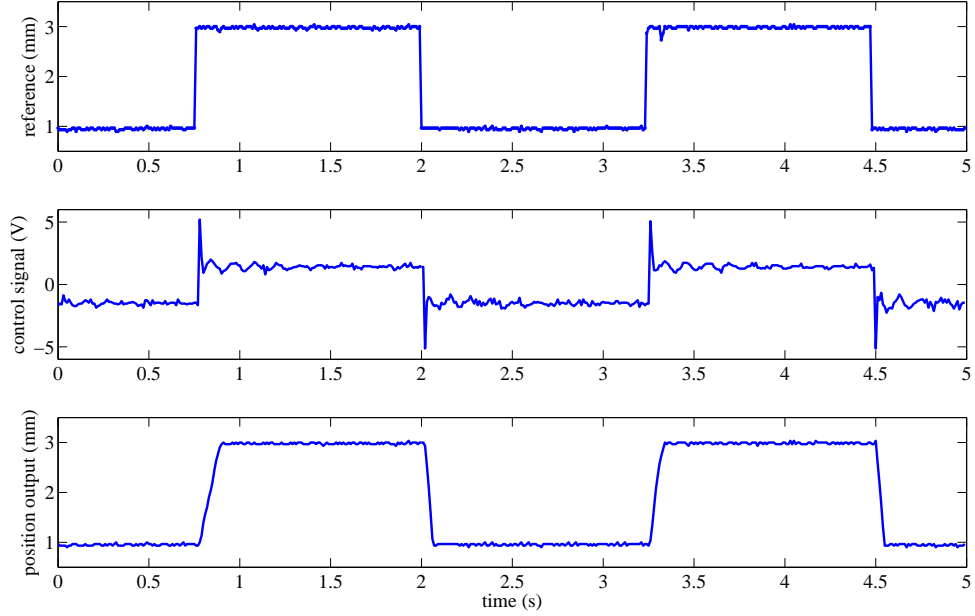


Figure 4.11: With non-uniform sampling period: uniformly distributed samples (sawtooth wave pattern), $T_n \sim U(0.01,0.02)$, average sample rate = 0.015s

The obtained results in Figs. 4.9, 4.10 and 4.11 can be quantified using a performance criteria such as the Integral of absolute error (IAE). The IAE, which weights all the errors equally, is generally used to evaluate a control system design and performance (Nise 2007). Its mathematical formula can be given as:

$$IAE = \int_0^T |e(t)|dt \quad (4.2.1)$$

where T is the integration time and $e(t)$ is the error signal.

Constant sample rate $T_n=0.015s$	Dual sample rate $T_n=0.01s,0.02s$	Sawtooth wave pattern $T_n=U(0.01,0.02)$
0.4842	0.5421	0.5808

Table 4.1: IAE with various sampling schemes

The IAE results have been tabulated in Table 4.1. The values are similar (or close to each other), with the sawtooth wave pattern sampling scheme

having the highest IAE value. This is probably because consecutive changes in the sampling rate can generate more transients effects, as demonstrated in §3.3.2. Nonetheless, the IAE can effectively be used as a criteria for performance assessments of the results achieved with non-uniform sampling.

The hardware results presented here support the simulation conclusion that the modified canonic δ structure is able to cope with non-uniform sample rates. In addition, the use of a 4th order controller indicates the feasibility of applying the approach to complex implementations. Other notable conclusions with context to the micro-controller implementation are:

- The non-uniform rate control algorithm can operate effectively with a uniform sample rate.
- Tracking performance is slightly degraded due to filter reconfigurations: The transient issue is inevitable, however, the filter structure can suppress the problem.
- Consecutive variations in the sample rate (such as the case with the sawtooth wave pattern) produces more transients due to the continuous change in the filter coefficients.

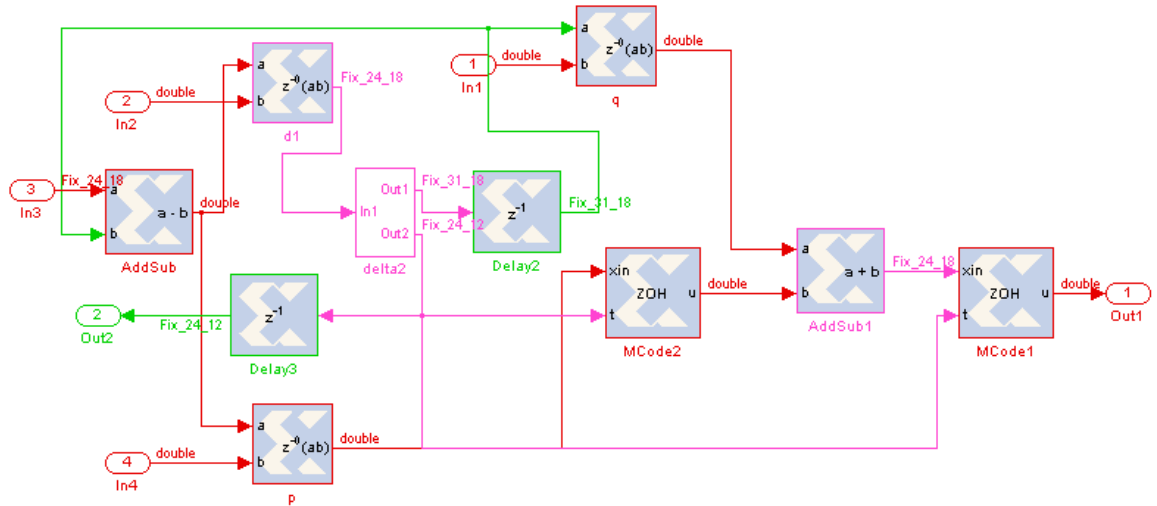


Figure 4.12: Simulink view of a 1st order non-uniform sample rate filter; implementation structure: modified canonic- δ

4.3 Implementation with programmable Logic

This section describes validation using an FPGA implementation but this time in a hardware-in-the-loop (HIL) environment rather than with a real physical system.

Until recent years, DSPs have been the best choice for enabling high-performance digital processing. However, due to their increasing speed and ability to perform mathematical operations, FPGAs are replacing DSPs in many processing applications. The modern FPGAs can have DSP processing cores embedded that perform the multiplication and accumulation (MAC) operations efficiently, without consuming any extra resources on the chip, although this facility is not needed for this study. Furthermore, the large number of programmable gates makes it possible to implement high performance systems on a single chip, reducing cost and space. Another advantage of FPGAs is the ability to re-configure. This allows a cheap medium to rectify design

errors and enables the same hardware to be used for a diverse range of applications. For the implementation of intentional non-uniform sampling schemes, the FPGAs are a suitable platform since they allow the execution of all specific tasks on a single chip, while performing operations in parallel.

In general, FPGAs contain programmable logic components called logic blocks, with a hierarchy of reconfigurable interconnects that can be configured to perform complex combinational functions. The FPGA configuration is generally specified using a hardware description language (HDL) which produces a technology-mapped netlist. This netlist can be fitted to the actual FPGA architecture for simulation and verification purposes.

The custom control algorithm has been designed using a MathWorks model-based design environment Simulink and the Xilinx System Generator design package can enable an accurate development of a Simulink model for generating a synthesizable HDL code. This Simulink plug-in for the Xilinx software provides the graphical development of FPGA designs and can simulate the implementation process on standard FPGA hardware.

4.3.1 RTL modelling of the non-uniform rate controller

This section describes the non-uniform rate controller design process using Xilinx's System Generator. The transfer function that is implemented is same one used earlier from Eq. 3.8.1 on page 112, i.e. consists of a notch, proportional-integral and phase advance:

$$H(s) = \frac{1 + 1000s}{1000s} \cdot \frac{1 + 0.1s}{1 + 0.01s} \cdot \frac{1 + 0.0025s^2}{1 + 0.005s + 0.0025s^2}$$

Design issues

The methodology for implementing Simulink models using System Generator involves several issues⁷, due to the underlying hardware characteristics of the available System Generator blocks (Murthy, Alvis, Shirodkar, Valavanis & Moreno 2008). Other significant issues include:

- Timing issues with algebraic loops: Since there is an element of feedback involved, the logic gates will induce a delay that might affect the stability of the system.
- Fixed point arithmetic: FPGA implementations expect fixed point arithmetics that must be defined during the hardware design phase. Therefore, it is necessary to determine the precision required against increased logic and potential delays associated with long word lengths.

Building and testing in XilinxTM System Generator

The system generator based system design of a 1st order filter is shown in Fig. 4.12. A word length of 24 bits is chosen, with 18 bits allocated to the fractional portion and the most significant bit as the sign bit. The non-uniform sample time coefficients are stored in a look-up table, avoiding the need of the additional maths required to recalculate them. These are updated appropriately depending on the current sample time value. As an example of the code, the script which generates the non-uniform sample rates is listed in Lst. 4.4. The script generates two sets of sample times (to form the digital signal sampling scheme) and the coefficient values (in Tables 4.2⁸, 4.3 and 4.4)

⁷—timing synchronisation, latencies associated with mathematical calculations conversion, fixed-point conversions.

⁸To implement an integrator (such as listed in Table 4.2), the equivalent δ -filter would have a pole at $\delta=0$. Having no denominator, of course the corresponding equations are very

are stored in the FPGA.

The complete schematic of the resulting 4th order filter is shown in Fig. 4.13. The output from the non-uniform rate controller designed using XilinxTM System Generator is shown in Fig. 4.14. The blocks ‘PLcoeffs’, ‘PAcoeffs’ and ‘NOTCHcoeffs’ contains the look-up tables for two sets of coefficients that depend on the sample rate variable, x . The continuous controllers are also included for comparison.

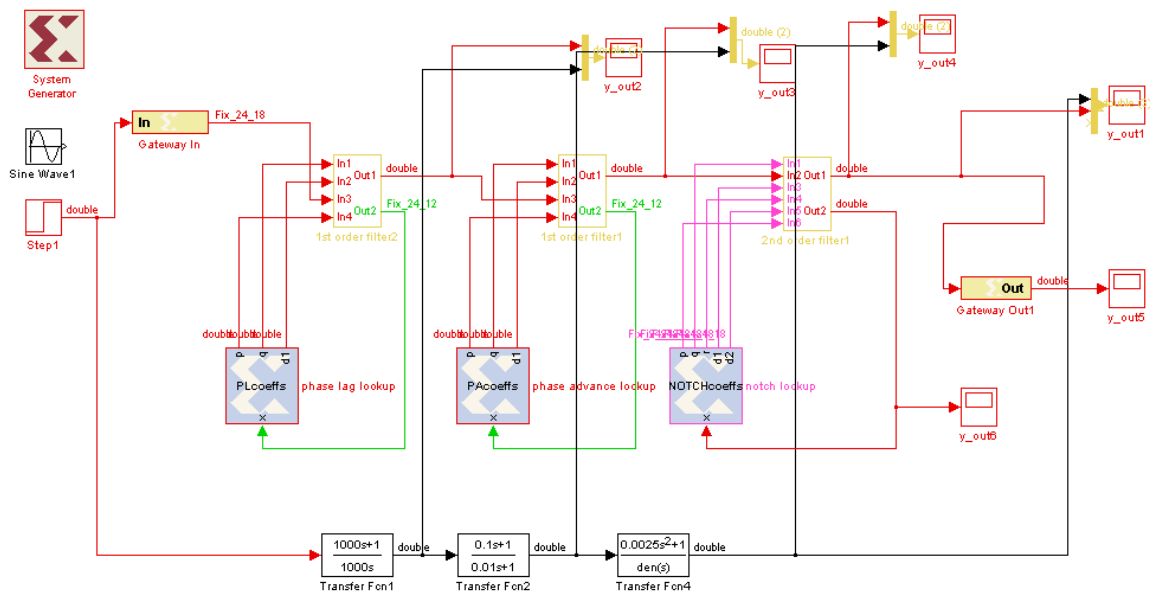


Figure 4.13: RTL view of a 4th order non-uniform rate controller

simple because there is no recursion.

```

1 function [u,t] = fcn(xin)
2
3     /*initialise variables*/
4     p={xlSigned, 24, 12, xlRound, xlWrap});
5     persistent ts, ts=xl_state(10,p); persistent a, a=xl_state(0,p);
6     persistent b, b=xl_state(10,p); persistent c, c=xl_state(10,p);
7     persistent i, i=xl_state(0,{xlUnsigned,8,0}); persistent temp,
8     temp=xl_state(0,p); persistent temp_1, temp_1=xl_state(0,p);
9     persistent pout, pout=xl_state(0,p);
10    persistent r1, r1 = xl_state(0, {xlUnsigned,20,10});
11
12    /*variable time step program*/
13    r1 = xfix(p,r1 +1);
14    if(r1>=(ts))
15        pout=temp;
16        a = xin;
17        r1=0;
18
19        if (i>=50)          /*changing the sample time*/
20            i=0;
21            if(ts==5)
22                ts=10;
23            else
24                ts=5;
25            end
26        else
27            i = xfix(p,i +1);
28        end
29
30        temp=a+pout;      /*delta operation*/
31        temp_1=temp;
32
33    end
34
35    u=temp_1;    /*output the value*/
36    t=ts;
37 end

```

Listing 4.4: The non-uniform (dual rate) sampling routine: T_n is 0.01s and 0.02s

```

1
2  /*initialise variables*/
3  p=({xlSigned, 20, 10, xlRound, xlWrap});
4  persistent ts, ts=xl_state(10,p); persistent a, a=xl_state(0,p);
5  persistent temp, temp=xl_state(0,p);
6  persistent r1, r1 = xl_state(0, {xlUnsigned,20,10});
7
8  /*main program*/
9  ts=t;
10 r1 = xfix(p,r1 +1);
11 if(r1>=(ts)) /*check if it is time to sample*/
12     a = xin;
13     r1=0;
14 end
15 u=a; /*output value*/
16 end

```

Listing 4.5: A ZOH routine with a non-uniform sampling output. The value of T_n is regulated by Lst. 4.4

Coefficients	$T_n = 0.02s$	$T_n = 0.01s$
c_0	1	1
c_1	0	0
r_1	0	0

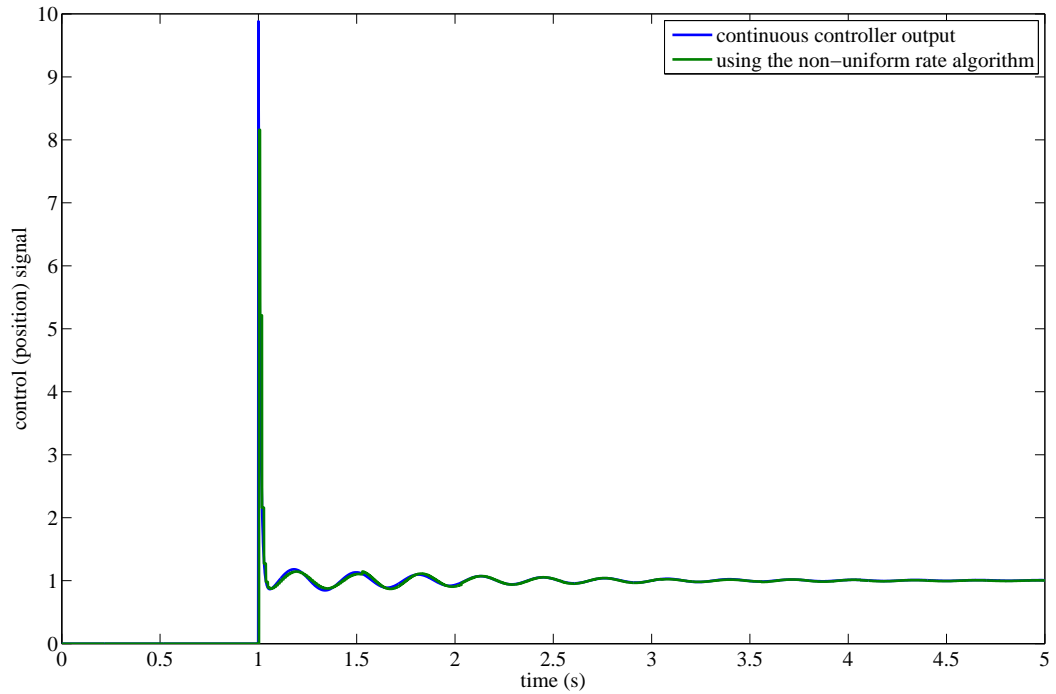
Table 4.2: PI coefficient parameters

Coefficients	$T_n = 0.02s$	$T_n = 0.01s$
p	7	8.2
q	1	1
d_1	0.667	0.4

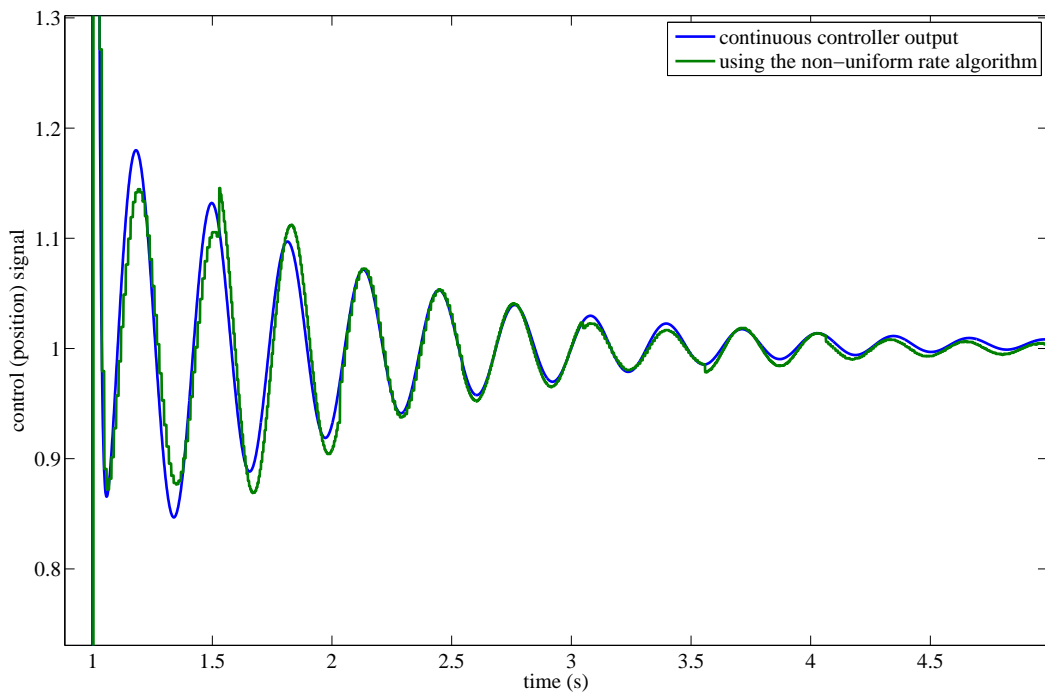
Table 4.3: Phase advance coefficient parameters

Coefficients	$T_n = 0.02s$	$T_n = 0.01s$
p	0.9902	0.995
q	0.6667	0.5
\bar{r}	1	1
d_1	0.0588	0.0199
d_2	0.6667	0.5

Table 4.4: Notch filter coefficient parameters



(a) Output from the non-uniform rate controller to a step input



(b) The zoomed illustration of Fig. 4.14a.

Figure 4.14: Open-loop output to a unit step input (in Matlab) from the non-uniform rate controller, designed using XilinxTM System Generator. The non-uniform sample rate: Dual rate with $T_n=0.01$ s, 0.02s.

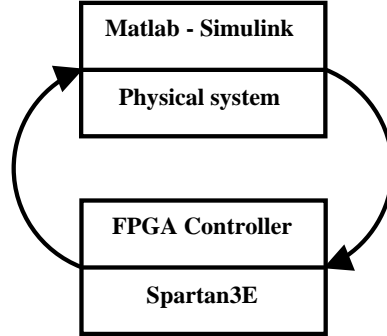


Figure 4.15: The HIL schematic block diagram of motor model

4.3.2 Hardware-in-the-loop simulation

In many cases, the most effective way to develop an embedded system is to connect the system with the actual plant. However, another option is to perform a Hardware-in-the-loop (HIL) simulation. HIL simulation is a technique that is used in the development and test of complex real-time embedded systems. It can provide an effective platform by adding the complexity of the plant under control to be tested without compromising factors such as cost, time and safety. Here, the non-uniform rate controller is implemented on the board with a dual sample rate with $T_n = 0.01s, 0.02s$. The HIL model in Simulink is the DC motor has been used earlier in Chapter 3, Fig. 3.31 on page 112. The Laplace transfer function for the system is:

$$H(s) = \frac{0.001 + 0.2501s + 0.25s^2 + 0.001s^3 + 0.0001s^4}{s + 0.11s^2 + 0.011s^3 + 0.0001s^4} \quad (4.3.1)$$

The output signals are then fed in to the physical system that is being simulated in a Matlab-FPGA based environment.

Prototyping board

The implementation is performed using the Xilinx Spartan3E Starter kit (Xilinx 2006). The kit, illustrated in Fig. 4.16, includes 500000 equivalent gates

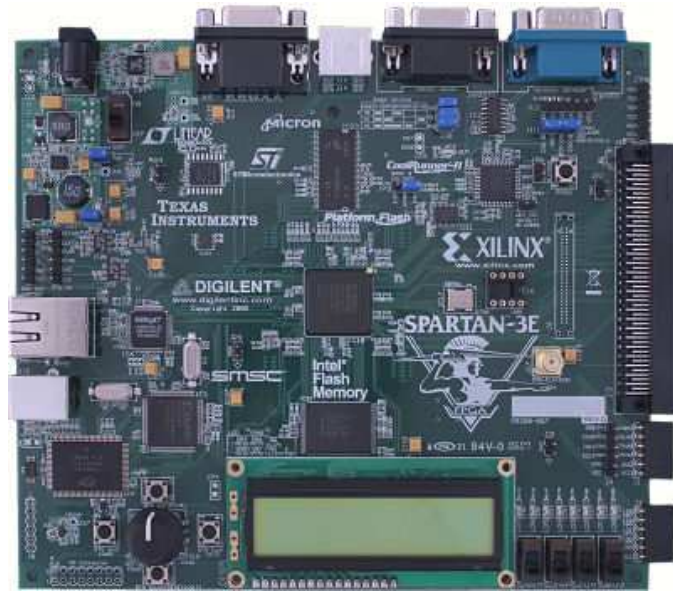


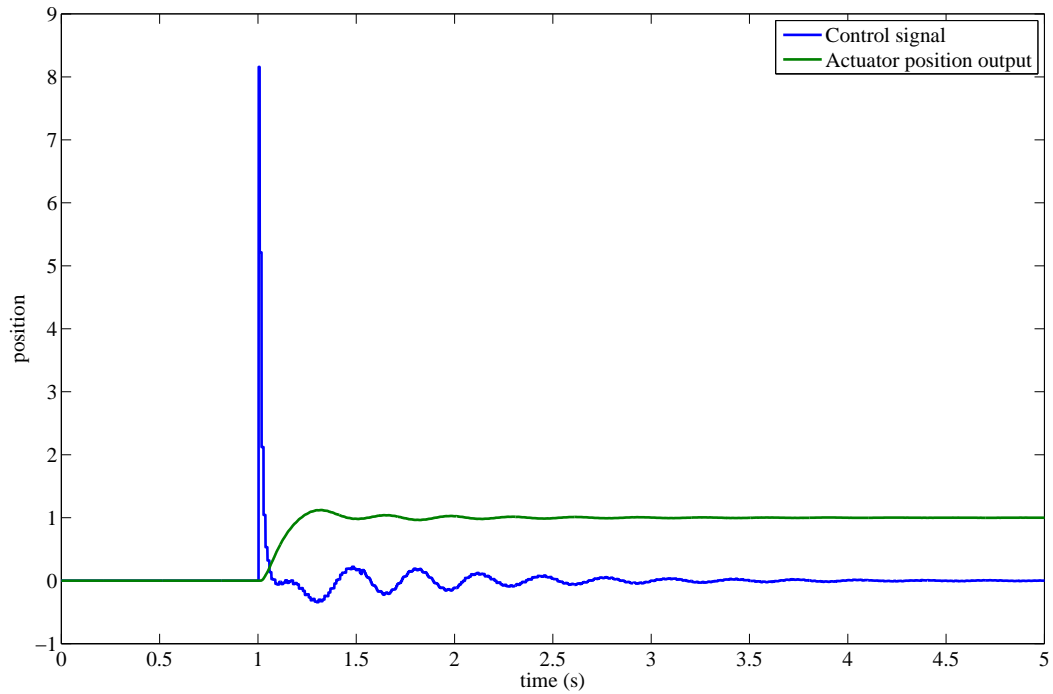
Figure 4.16: The Spartan3E Development kit

(XC3S5500E), a 50MHz crystal oscillator, asynchronous serial port with RS232 drivers, flash memory for bitstream storage and a USB port for configuring the FPGA and memory parameters. A PC is used to communicate with the board and perform the simulation.

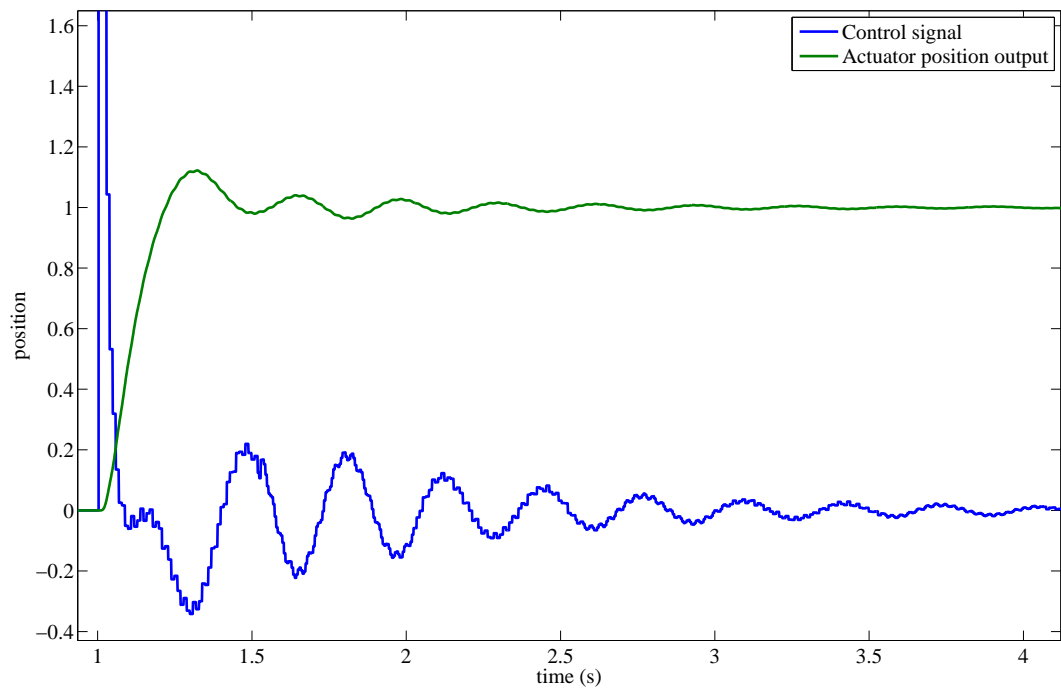
4.3.3 Hardware results: closed-loop with DC motor

The DC motor model has been used to verify the concept of treating non-uniform sampling schemes. The final verification can be made by implementing the hardware simulation of the controller. By selecting the prototype board, System Generator can generate a hardware co-simulation block which can be used to perform a hardware-in-the-loop simulation as in Fig. 4.17.

The System Generator plug-in enables the option of designing hardware descriptive models in a Matlab based environment. Utilizing the presented approach, the System Generator provides a graphical environment for simple conversions into HDL designs as compared to the any manual conversions.



(a) HIL simulation results with a step input



(b) The zoomed illustration of Fig. 4.17a.

Figure 4.17: Closed-loop output produced using the non-uniform rate control algorithm, with the DC motor model in a closed-loop. The non-uniform sample rate: Dual rate with $T_n=0.01\text{s}$, 0.02s .

After the non-uniform rate controller is mapped on the FPGA, a hardware-in-the-loop simulation is exercised. The HIL simulation is an important aspect for control engineering since the relevant theories can directly be applied to real-time situations and be put into practice.

4.4 Summary

This chapter was aimed at the software and physical implementation aspects of the non-uniform rate algorithm. Various non-uniform sampling schemes can efficiently be processed using simple microcontrollers families. The control program typically executes recursively, with the sampling frequency and controller coefficients being updated for the next sample period.

Moreover, this analysis is followed by an HDL implementation where a DC motor model is used to verify the practicality of the control algorithm for real-time control. The contributions are summarised as follows:

- The design of a well-defined control task to execute the controller formulation developed in Chapter 3.
- The implementation on basic microcontrollers proves an efficient and cost effective realization of the required processing functions.
- A Simulink plug-in was utilized to produce the hardware model of the non-uniform rate controller. The model was verified by successfully performing a HIL simulation.
- The implementation results demonstrate the efficacy of the proposed approach in practice.

Chapter 5

Conclusions

Chapter overview

This chapter summarises the work done and draws conclusions from the research. The objectives include:

- To summarise the aims and objectives of this research.
- To present the summary of results and conclusions mentioned in the previous chapters.
- To discuss the strengths and shortcomings of this work.
- To present a framework of potential future work on non-uniform-rate sampling in the context of control theory.

5.1 Summary of research

This thesis has amassed some theories and techniques from digital control, real-time signal processing and non-uniform sampling to deal with the problem of processing and analysing typical control system setups with non-uniform time sampling. From a practical standpoint, the main theme has been the design and analysis of control algorithms with intentional non-uniform sample rates.

The Introduction briefly discussed the problems experienced in signal processing, especially when implementing high performance control systems. It then explained the potential benefits of adopting to a non-uniform sampling regime, which is actively being researched for specific signal processing applications. It was hence established that this work is rather targeted towards the introduction of a novel approach for the control communities to appreciate, and the possibility of using intentional (or unintentional) variations in control systems to provide a cost effective solution.

The historical perspective reviewed in Chapter 2 provided a genuine argument regarding the rigidity of control engineers for adhering with uniform sampling throughout the vast developments in ongoing research. However, on accounting the successful achievements with non-uniform sampling schemes in signal processing applications and the potential benefits that can stem from them, this work becomes relevant, especially when factors like cost, performance, integration, easy development and power consumption can be improved through its successful realization.

Chapter 3 described the methodology and design technique that can be used to implement non-uniform rate control systems. The key developments have been the non-uniform rate discrete equivalent transfer functions that can be designed using the classical control theories. It is later established that such algorithms are prone to suffer from transient errors that occur due to the changes in the sample rate. Conversely, it is adjudged that the implementation structure plays an important role when dealing with non-uniform sampling, and the direct- z and the modified canonic- δ structures offer a better solution to resolve the problem. The identification of efficient controller formulation to implement the control algorithm using the proper implementation structure offers a number of advantages if compared with the traditional

canonic z approach. In addition, the motivation for using the non-uniform rate control algorithm is further justified in the MSE analysis presented on page 82. It is shown that not accounting for sample time variations results in a high MSE value as compared to the case that accounts for non-uniform sampling conditions. The Fourier transform was accepted to be the ultimate tool for evaluating the system response and to provide an acceptable solution for the frequency response characteristics. After the analysis, it was concluded that, as far as the phase lag issues of the system are concerned, the results of adopting a non-uniform sampling frequency are not very encouraging. It was shown (for the first time) that the variations in the sampling distribution does not provide any added advantage in the phase response, however the importance of the non-uniform rate implementation algorithm and its evaluation techniques is a plausible development for control literature. In addition, such techniques can be used to provide a basis for very valuable testing of control systems with varying sampling periods prior to any practical implementation in terms of real hardware. Constraints arising due to jitter sampling issues in real-time control can now be analysed thoroughly using the Fourier techniques highlighted and the frequency characteristics can be determined.

Chapter 4 addressed the hardware and software implementation aspects of the developed control algorithm and presented the general process of specifying the main components and an overall specification for implementation. In addition, the non-uniform rate control algorithm was validated using two hardware implementations: with a microcontroller control of an electromechanical magnetic positioning actuator and a HIL simulation of a DC motor. These showed that intentional non-uniform sampling can be applied and regulated using standard hardware technologies.

From the contributions of each chapter, it can finally be concluded that

the objectives of this research work have been fulfilled. By exploiting the requirements for implementing real-time control and by identifying the dependence of non-uniform sampling on the filter implementation structure, a method to implement the formulated control algorithm has been proposed. It is demonstrated that such a design can also be applied to basic microcontroller chips, which results in a low cost, numerically effective solution for real-time applications.

Thesis aims

The aim of this thesis has been to investigate the idea of non-uniform sampling and discuss some key questions that were mentioned at the outset, which have been reproduced here for convenience:

Q1 Can intentional non-uniform sampling administer any benefit in digital control applications?

A1 As far as the phase lag issue mentioned in §2.1.1 are concerned, No. With a non-uniform sample rate, the phase response at low frequencies remains bounded in between the maximum and minimum sample rate values being used in the process. At higher frequencies (or close to the Nyquist limit) the variations in the sampling scheme become apparent largely due to the reconstruction process not having enough samples to reform the complete waveform.

Q2 Given a continuous controller, how can its non-uniform rate discrete equivalent be designed using the z -transform?

A2 Yes, although this can be done only if the sample time instants are known in advance (or before the next sample instant). Then the transfer

function characteristics can be adjusted by recalculating its coefficients in the z -transform according to the changing sampling frequency. This approach is based on the same classic controller design methods used in control theory.

Q3 How to analyse the frequency response of a non-uniform rate controller?

A3 The Fourier analysis technique, highlighted in §3.4, can effectively be used to observe the frequency datum for uniform and non-uniform sampling conditions. Although there may be various techniques available from the signal processing communities, the chosen technique offers a combination of frequency domain and time domain analysis that plots the magnitude and phase of the signal in the time domain. This is relative to plotting Bode plots for digital control.

Q4 Can the non-uniform rate controller be implemented using standard hardware?

A4 Yes, the proposed control algorithm is implemented on an 8051 microcontroller and an FPGA starter board. The results show that control of a real system is possible and that the performance is no different than the fixed rate control.

Key features of the research

- Brief review of the role of non-uniform sampling in control applications.
- Design and development of non-uniform rate controllers using classical control theories. A flexible controller design approach is presented that relies on adjusting the controller parameters at run time according to the specific implementation timing behaviour.

- Identification of limitations in digital control with non-uniform sampling: limitations of the non-uniform rate control algorithm are demonstrated through examples:
 - the transient phenomenon is acknowledged to cause sudden amplifications in the output response of the digital filter, whenever the sample rate changes
 - it is noted that the severity of the transient signal depends on the size of magnitude change in the filter coefficients
 - the canonic z -filter and the canonic δ -filter structures are identified to be *unsuitable* for non-uniform sampling
 - the modified canonic δ -filter and the direct z -filter structures are able to suppress the transient phenomenon
- A Fourier analysis technique is used to analyze the frequency response of the non-uniform rate controller in the time domain:
 - a criterion is defined for performance evaluation
- Experiments with various sample rate distributions reveal no particular benefit when utilising non-uniform sampling in digital controllers. In addition, there is no significant detriment in the produced outputs (apart from the point where a very large rate change is imposed).
- The developed control algorithm is formally designed in C/C++ and is implemented on a standard 8051 microcontroller.
- The RTL modelling of the non-uniform rate controller is formulated by effectively using XilinxTM's System Generator.

- The implementations demonstrate the efficacy of the proposed approach in practice.

5.2 Future work

The concept of using non-uniform sampling as a tool has opened up a door of various possibilities in the engineering discipline. However, there will be some application-specific limitations that need to be discovered and hence a thorough design analysis will have to be carried out before the technique can make its way into a practical solution. It should be noted that uniform sampling is still the most preferable method to execute tasks due to its simplicity and scope, and this research study has not highlighted any distinct advantages of non-uniform sampling that will differentiate the choice of its suitability for an application. This work has brought forward some specific digital control issues to light and attempted to initiate and expand the scope of non-uniform sampling in this subject for research. Realising the importance of the ongoing topic, some elements of this work can be extended for future research endeavours:

- The non-uniform-rate control algorithm presented in §3.2.2 can regulate control systems while accommodating several sample rates. This is in particular interest to the application for analysing control systems for jitter impacts on control performance. Moreover, the idea could be extended to the design of time varying control systems and robust control.
- The generalized DFT presented in Appendix B can effectively be used by control engineers to estimate the spectral content of non-uniformly sampled data. The algorithm can further be improved by using higher order

numerical integration techniques to calculate the Fourier coefficients.

- The optimal strategy for eliminating/suppressing the transient phenomenon is still an open question. Selecting the correct implementation structure reduces the unwanted transients, however, they still exist and may have a significant effect in small signal analysis. Perhaps other filter structures (e.g. the lattice, all-pass based or wave structures, etc.) may offer better suppression capabilities as compared to the ones identified here.
- Non-uniform sampling can also be speculated for the theory of fault tolerant control (Bilinskis & Cain 1996, Mikelsons & Greitans 1996).
- The abstract of alias-free sampling can also be applied to the theory of system identification (See §5.2.1).

5.2.1 Possible application to system identification

In the authors opinion, system identification is an area where the potential benefits of non-uniform sampling are clear. Hence it is discussed in slightly more detail.

The ability to identify and analyze control systems in an effective and efficient manner is crucial for the success of any application in the engineering industry. If sufficient experimental data is available, the models can be constructed through the process of system identification, which can be applied to virtually any system and typically yields relatively simple models that can well describe the systems behavior within a defined operational regime. However, since the emergence of nonlinear sophisticated applications, the control system continually demands the development of increasingly complex means for

system identification. In such an environment it is important that the experimental data collected to be ‘*rich enough*’ to achieve a coherent specification model that accurately describes the overall system behavior, and is constantly synchronized with any changes to the system implementation (Landau, Landau & Zito 2006).

Dynamic systems in the physical world are naturally described in the continuous-domain. However, most system identification techniques are based on discrete-time models (Hugues & Liuping 2008), that are estimated from sampled data collected with a fixed sample rate, which may not be valid at other rates¹. Furthermore, in some situations, it may be difficult to obtain equidistantly sampled data and there is an absence of the appropriate non-uniform sampling analysis techniques. The problem is of importance as the case of non-uniformly sampled data occurs in several applications. Since most of the study in control theory has been targeted towards uniform sampling, the standard discrete-time linear, time invariant models might not be applicable for continuous identification. Moreover, issues such as inter-sample behavior and difficulties at high sampling frequencies are well known drawbacks of discrete-time models (Hugues & Liuping 2008).

One promising solution to this problem is to make use of digital alias-free signal processing techniques to identify continuous-time models that will match more closely to the actual system². Non-uniform sampling techniques can be used to generate a tentative continuous-time model, by utilising the randomization as a tool for improving the representation of the signals sampled for processing. On the other hand, a uniformly sampled signal will rapidly

¹—as discussed in §2.4 on page 38.

²It is worth mentioning that the concept of DASP does not use any sort of interpolation to compensate for the non-uniformity in between the sample instances, due to which it enables the simplification of complex designs.

become obsolete as the signal frequency increases, the DASP non-uniform sampling technique would enable the rapid and reliable processing of the signal even at low average sample rates³. A pragmatic solution will therefore satisfy the following properties:

- The algorithms must be able to adapt and process non-uniformly sampled data correctly by utilizing the appropriate numerical techniques.
- It should be able to accept whatever information is provided, and produce a reasonably accurate hypothesis of the underlying model that is a suitable generalization datum⁴.

³This is due to the preservation of the spectral contents of the sampled signal.

⁴Instead of expecting some notionally complete amount of initial information, the DASP approach can aid in generating a reasonable estimate of the target from a limited (or potentially missing) amount of input data.

Appendix A

Uniform sampling

A sample refers to a value or set of values at a point in time. In order to acquire a sample, a sampler has to be used, which is a subsystem or operator that extracts samples from continuous signal. A sampling process can convert a continuous analogue $x(t)$ into a discrete time representation $x[n]$.

The plain conceptualization of sampling is that of an Ideal sampler:

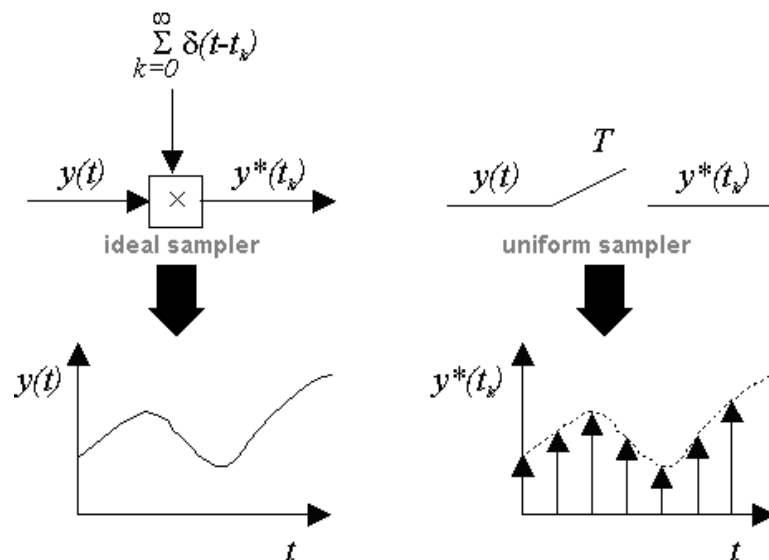


Figure A.1: The uniform sampling process

Let $x(t)$ be a continuous signal which is to be sampled, and that sampling is performed by measuring the value of the continuous signal every T seconds. Thus, the sampled signal $x[n]$ is given by:

$$x[n] = x(nT), \quad n = 0, 1, 2, \dots \quad (\text{A.0.1})$$

The sampling frequency or sampling rate f_s is defined as the number of samples obtained in one second, or $f_s = 1/T$. The sampling rate is measured in hertz or in samples per second. To reconstruct the original signal $x(t)$ completely and exactly, the Nyquist Sampling theorem needs to be followed, i.e. sample at least twice the maximum signal bandwidth. Since the time variable, T , in this case is a constant the sampling is uniform.

The phase lag

Generally, there is always a ‘half sample delay’ associated with uniform sampling. This means that there is a linear relationship of the sampling error with the sampling period¹. To illustrate the ‘half sample delay’ is to reconstruct part of the Fourier series to form the continuous equivalent as a staircase function. With a Zero-order-hold, the DAC converter can produce the output signal by holding the values constant between successive updated intervals (look at Fig. A.2). The laplace transfer function of the ZOH can be defined as (Middleton & Goodwin 1990):

$$ZOH(s) = \frac{1 - e^{-Ts}}{s}$$

So the phase delay introduced is approximately:

$$\phi_s \simeq 360f_0 \frac{1}{2f_s}$$

¹If the speed of sampling is increased by a factor of two, the phase lag error will be reduced by half.

where ϕ_s is the phase delay introduced by sampling. Wu (2005) took the additional effect of the computation time into consideration and derived the total phase lag to be:

$$\phi = \phi_s + \phi_c \simeq 360f_c\left(\frac{1}{2f_s} + T_c\right) \quad (\text{A.0.2})$$

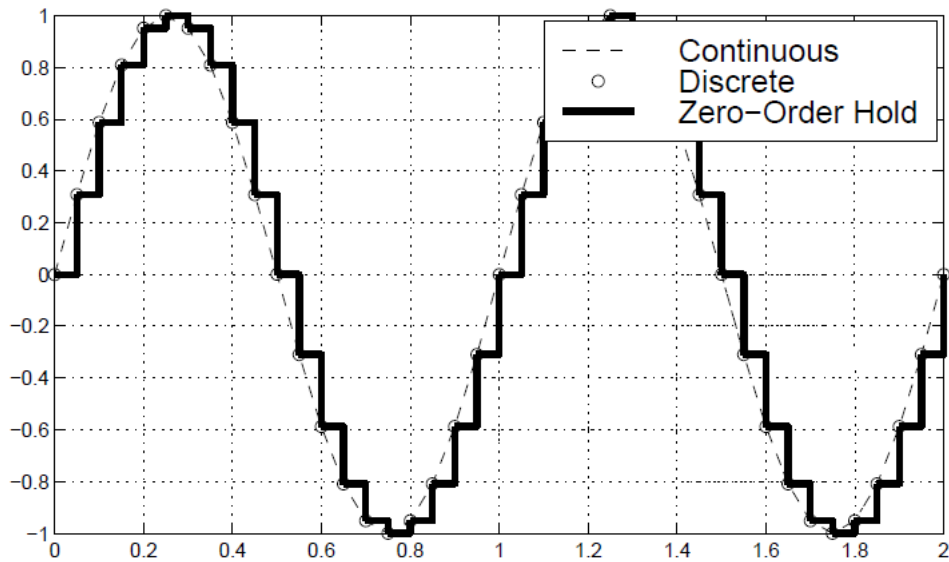
where T_c is the computation time, ϕ_c is the phase delay introduced by T_c and ϕ is the total phase delay. This can be expressed in somewhat more practical way. Let R be the ratio of the sampling frequency to the required bandwidth frequency, and K be the proportion of the sample period taken up by the computation. Then:

$$\begin{aligned} R &= \frac{f_s}{f_0} \\ K &= \frac{T_c}{T_s} = f_s T_c \\ \phi &= \frac{360(0.5 + K)}{R} \end{aligned}$$

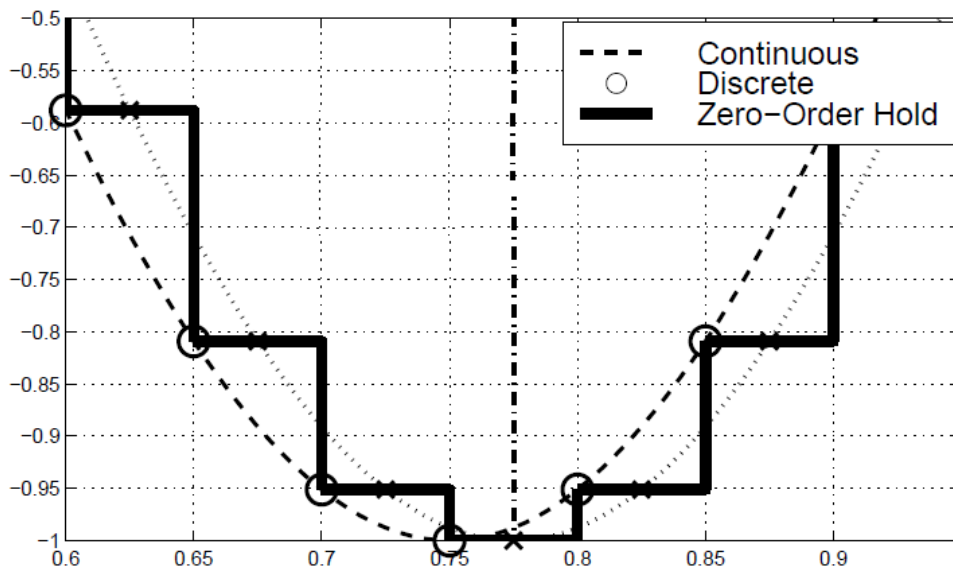
As the phase delay should be no more than 5° , the corresponding interrelationship between R and K to meet this requirement is given by:

$$R = 36 + 72K$$

which explains the high sampling requirements required by electromechanical applications for maintaining a 5° phase margin.



(a) The ZOH reconstruction



(b) Induced phase lag due to the ZOH

Figure A.2: The reconstruction process

Appendix B

Non-uniform sampling: Frequency estimation

This Appendix gives the method to estimate the Non-uniform Discrete Fourier Transform of a non-uniformly sampled signal¹.

The Fourier transform is basically a mapping from the time domain to the frequency domain. This transform defines the frequency content parameters of a signal. Consider the Discrete Fourier Transform $F(f_k)$ which, according to Ramirez (1992), can be defined as:

$$F(f_k) = \sum_{j=0}^{N-1} d(t_j) \exp\{2\pi f_k t_j i\} \quad (\text{B.0.1})$$

where

$$i = \sqrt{-1}$$

N is the total number of data points

f_k is the frequency

¹Despite the fact that the Fast Fourier transform is an effective technique to estimate the spectral contents of a sampled signal, it is limited in its scope with equally spaced samples.

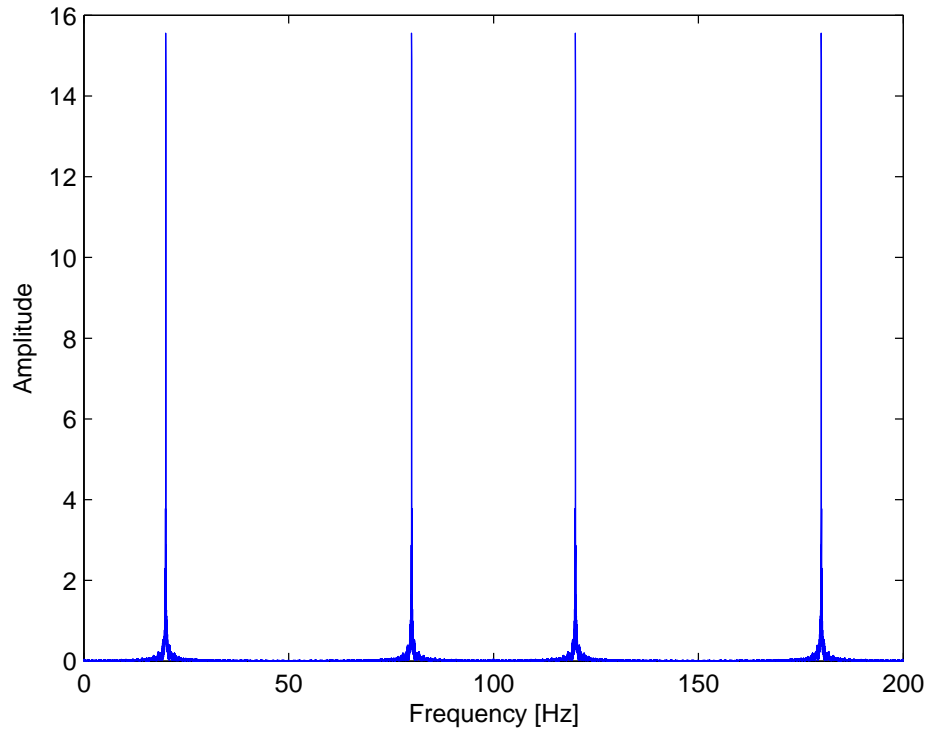


Figure B.1: Uniformly sampled data. Sampling below the Nyquist frequency causes aliases which can clearly be seen to have corrupted the signal

$t_j = j \Delta T$, $j \in \{0, 1, \dots, N-1\}$, are periodic time instances

$d(t_j)$ is a complex discretely sampled data.

To illustrate the use of the Eq. B.0.1 as a frequency estimation tool, a simulation is conducted with the signal data generated with a sample time of $\Delta T = 0.01s$. The Nyquist critical frequency will be one-half the inverse of the sample time:

$$f_{Nc} = \frac{1}{2T} = 50Hz \quad (\text{B.0.2})$$

With an input signal frequency of $80Hz^2$, the spectral plot generated is illustrated in Fig. B.1. Clearly, due to under-sampling, the spectral contents of the signal have been corrupted with false frequency images. For a much

²Since the input data is greater than the Nyquist frequency, aliasing will occur.

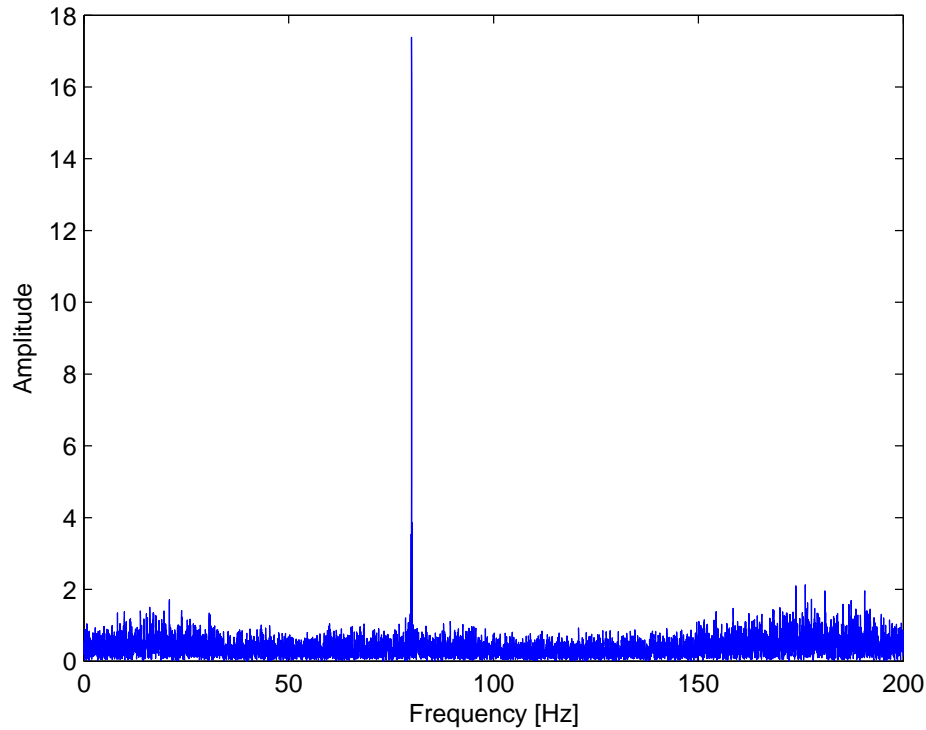


Figure B.2: Non-uniformly sampled data. Adding variations to the sampling scheme can mitigate the effects of aliasing

more in depth discussion on the topic, refer to Bretthorst (2008).

Consider the following generalisation of Eq. B.0.1:

$$F(f_k) = \sum_{j=0}^{N-1} d(t_j) \exp\{2\pi f_k t_j i\} \quad (\text{B.0.3})$$

where all the parameters are the same as before with the exception of t_g :

$t_g \in \{\dots, t_{k-1}, t_k, t_{k+1}, \dots\}$, are non-uniform time instances

The results are demonstrated using a simulation illustrated in Fig. B.2, The sampled signal is 80Hz being sampled with an average non-uniform sample rate of $0.01\text{s} \sim U(0.08, 0.012)$. In generating the times to acquire the samples, a uniform distribution is chosen³. It seems that when adding some randomness

³The discussion here pertains to all non-uniformly sampled data and not just to the pseudo-random sampling scheme.

in the sampling scheme, the aliases are converted into broadband noise which does not have the same implications as aliases.

Aliasing a general phenomenon and exists in both uniform and non-uniformly sampled data. It is the fact that all of the times may be expressed as an integer multiple of the smallest sample time, which is the primary reason for aliasing⁴. The addition of variations can help increase the operating bandwidth where the Nyquist frequency will depend on the smallest sample time parameter. Therefore, non-uniform sampling has an advantage over uniform sampling since it not how fast the sampler can sample data, but about how to use variations to measure frequencies with much larger bandwidths.

Improvements by numerical methods

Instead of using the Euler's approximation, the Fourier coefficients can further be improved by applying other sophisticated numerical integration rules⁵. Consider the following substitution where

$$y(t_i) = x(t_i)e^{-j2\pi k\Delta f t_i}$$

the result with trapezoidal integration can be expressed

$$X_d(k\Delta f) = \Delta t \sum_{n=0}^{N-1} [y(t_n) + y(t_{n+1})] \frac{(t_{n+1} - t_n)}{2} \quad (\text{B.0.4})$$

the result with Simpson's integration can be expressed

$$X_d(k\Delta f) = \Delta t \sum_{n=0}^{N-1} [y(t_n) + y(t_{n+1})] \frac{(t_{n+1} - t_n)}{2} \quad (\text{B.0.5})$$

⁴Constant sampling has the smallest possible bandwidth.

⁵Although the improvement in approximation will come at the cost of increased complexity of the expression.

Appendix C

Code implementation details

Matlab code

The following include the m-file codes for the:

- The conventional controller with the direct- z implementation structure.
- The non-uniform rate algorithm with the direct- z implementation structure.
- The non-uniform rate algorithm with the modified canonic δ implementation structure.
- Perform a non-uniform Discrete Fourier transform on a set of non-uniformly collected data samples.

```

17 out2s snoise wait ts sin sin_1 m temp v1 v2 q a0 a1 a2 b1 b2 global
18 input_1 input_2 output output_1 output_2 s s2 derivative outt ts_1
19 pseudo ii ptime time1 n0 n1 n2 m1 m2 time=x(1);
20
21 if (time==0)
22
23     m=0;temp=0;input_1=0;output=0;output_1=0;ii=1;ptime=0;time1=0;
24     input_2=0;output_2=0;wait=0;s=0;s2=0;outs=0;out2s=0;v1=0;v2=0;outt=0;
25     ts=0.01;
26     samplerate=ts;
27     n0=10;n1=0.35;n2=0.0025;m1=0.105;m2=0.0005;
28     % 2^{nd} phase lead-lag compensator
29     %n0=1;n1=0;n2=0.0025;m1=0.005;m2=0.0025;    % 2nd order notch
30 end
31
32 a0=((n0*ts*ts)+(2*n1*ts)+(4*n2))/((ts*ts)+(2*m1*ts)+(4*m2));
33 a1=((2*n0*ts*ts)-(8*n2))/((ts*ts)+(2*m1*ts)+(4*m2));
34 a2=((n0*ts*ts)-(2*n1*ts)+(4*n2))/((ts*ts)+(2*m1*ts)+(4*m2));
35 b1=((2*ts*ts)-(8*m2))/((ts*ts)+(2*m1*ts)+(4*m2));
36 b2=((ts*ts)-(2*m1*ts)+(4*m2))/((ts*ts)+(2*m1*ts)+(4*m2));
37
38 else
39     if (time>=samplerate)
40         time1=time;
41         ptime=samplerate;
42         samplerate=pseudo(ii+1);
43         ii=ii+1;
44
45         input=x(2);
46         %input data streaming in
47         ts=samplerate-ptime;
48         output = a0*input + a1*input_1 + a2*input_2 - b1*output_1 - b2*output_2;
49         output_2=output_1;output_1=output;input_2=input_1;input_1=input;
50
51     end
52 end
53 u=output;
54 end

```

Listing C.1: The conventional controller algorithm with the direct- z implementation structure

```

55 outs out2s snoise wait ts sin sin_1 m temp v1 v2 q a0 a1 a2 b1 b2
56 global input_1 input_2 output output_1 output_2 s s2 derivative outt
57 ts_1 pseudo ii ptime time1 n0 n1 n2 m1 m2
58
59 time=x(1);
60 if (time==0)
61     m=0;temp=0;input_1=0;output=0;output_1=0;ii=1;ptime=0;time1=0;
62     input_2=0;output_2=0;wait=0;s=0;s2=0;outs=0;out2s=0;v1=0;v2=0;outt=0;
63     pseudo=[0.01,0.015,0.02,0.03,0.035,0.05,...]; %predefined sample times
64     samplerate=pseudo(1);ts=pseudo(1);
65     %n0=1;n1=0;n2=0.0025;m1=0.005;m2=0.0025;    % 2nd order notch
66     n0=10;n1=0.35;n2=0.0025;m1=0.105;m2=0.0005;
    % 2^{nd} phase lead-lag compensator
67 end
68     if (time>=samplerate)
69         time1=time;
70         ptime=samplerate;
71         samplerate=pseudo(ii+1);
72         ii=ii+1;
73
74     a0=(n0*ts*ts+2*n1*ts+4*n2)/(ts*ts+2*m1*ts+4*m2);
75     a1=(n0*ts*ts-2*n1*ts-4*n2)/(ts*ts+2*m1*ts+4*m2)+
76         (n0*ts_1*ts_1-2*n1*ts_1-4*n2)/(ts_1*ts_1+2*m1*ts_1+4*m2);
77     a2=(n0*ts*ts-2*n1*ts+4*ts*n2)/(ts*ts+2*m1*ts+4*m2)
78     b1=(ts*ts-2*m1*ts-4*m2)/(ts+2*m1*ts+4*m2)+
79         (ts_1*ts_1+2*m1*ts_1-4*m2)/(ts_1*ts_1+2*m1*ts_1+4*m2);
80     b2=(ts*ts-2*m1*ts+4*m2)/(ts*ts+2*m1*ts+4*m2);
81
82     input=x(2);
    %input data streaming in
83
84     ts=samplerate-ptime;
85     output = a0*input + a1*input_1 + a2*input_2 - b1*output_1 - b2*output_2;
86     output_2=output_1;output_1=output;input_2=input_1;input_1=input;
87     end
88 end
89 u=output;
90 end

```

Listing C.2: The non-uniform rate algorithm with the direct- z implementation structure

```

91 samplerate ts sin sin_1 m temp q a0 a1 a2 b1 b2 input_1 input_2
92 output output_1 output_2 derivative pseudo ii snoise global wait u1
93 v1 y1 x1 w1 r1 p1 q1 d1 d2 ptime n0 n1 n2 m1 m2 w1_1 v1_1 x1_1 y1_1
94 y1_2 y1_3 ts_1 global u1i v1i y1i x1i w1i r1i p1i q1i d1i d2i n0i
95 n1i n2i m1i m2i w1_1i v1_1i x1_1i y1_1i y1_2i y1_3i global v1j y1j
96 w1j p1j q1j d1j w1_1j v1_1j y1_1j y1_2j y1_3j c02 c12
97
98 time=x(1); input=x(2);
99
100 if (time==0)
101     c02=0;c12=0;m=0;temp=0;wait=0;input_1=0;output=0;output_1=0;
102     u=0;ii=1;w1_1=0;v1_1=0;x1_1=0;y1_1=0;y1_2=0;y1_3=0;ts_1=0;
103
104     pseudo=[0.01,0.015,0.02,0.03,0.035,0.05,...]; %predefined sample times
105
106     u1i=0;v1i=0;y1i=0;x1i=0;w1i=0;r1i=0;p1i=0;q1i=0;d1i=0;d2i=0;w1_1i=0;v1_1i=0;
107     v1j=0;y1j=0;w1j=0;p1j=0;q1j=0;d1j=0;w1_1j=0;v1_1j=0;y1_1j=0;y1_2j=0;y1_3j=0;
108     ts=pseudo(1);samplerate=pseudo(1);snoise=0;x1_1i=0;y1_1i=0;y1_2i=0;y1_3i=0;
109     u1=0;v1=0;y1=0;x1=0;w1=0;r1=1;p1=0;q1=1;d1=0;d2=0;ptime=0;
110
111     %n0=1;n1=0;n2=0.0025;m1=0.005;m2=0.0025; % 2nd order notch
112     n0=10;n1=0.35;n2=0.0025;m1=0.105;m2=0.0005; % 2nd order phase lead-lag compensator
113 end
114 if (time>=samplerate)
115
116     ptime=samplerate;
117     samplerate=pseudo(ii+1);
118     ii=ii+1;
119     ts=samplerate-ptime;
120
121     c0=(n0*ts*ts+4*n2+2*n1*ts)/(ts*ts+2*m1*ts+4*m2);
122     c1=(2*ts*ts)/(ts*ts+2*m1*ts+4*m2)+(2*ts_1*ts_1+4*m1*ts_1)/(ts_1*ts_1+2*m1*ts_1+4*m2);
123     c2=(4*n0*ts*ts_1)/(ts*ts_1+2*m1*ts+4*m2);
124     r1=(2*ts*ts)/(ts*ts+2*m1*ts+4*m2)+(2*ts_1*ts_1+4*m1*ts_1)/(ts_1*ts_1+2*m1*ts_1+4*m2);
125     r2=(4*ts*ts_1)/(ts*ts_1+2*m1*ts+4*m2);
126
127     p=c0;
128     d1=r1;
129     q=c1/d1;
130     d2=r2/d1;
131     r=c2/(d1*d2);
132     u1=input;
133
134     v1=u1-w1-x1;
135     y1=((p1*v1)+(q1*w1)+(r1*x1));
136     x1=x1+d2*w1;
137     w1=w1+(d1*v1);
138
139     w1_1=w1; v1_1=v1; x1_1=x1;
140     y1_3=y1_2;y1_2=y1_1;y1_1=y1;
141
142     w1_1j=w1j; v1_1j=v1j;
143     y1_3j=y1_2j;y1_2j=y1_1j;y1_1j=y1j;
144     end
145 end
146 u=y1;
147 end

```

Listing C.3: The non-uniform rate algorithm with the modified canonic δ implementation structure

```
1 data=[5,1.2,3,4.5,1,...];
2
3 df=0.01; N=500;M=5/df;
4
5     for k=1:(M)
6         Sf(k)=complex(0,0);
7         for n=1:N
8             dt=(time(n+1)-time(n));
9             Sf(k)=Sf(k)+(data(n)*(exp(-i*2*pi*time(n)*k*df)));
10            Sf(k)=Sf(k)*1/N;
11        end
12
13    for k=1:M
14        F(k)=(k)*df;
15    end
16
17 mag=20*log10(abs(Sf)); %magnitude in dB
18 pha=180/pi*angle(Sf); %phase in degrees
```

Listing C.4: Calculating the non-uniform Discrete Fourier transform

Appendix D

Derivation of the non-uniform rate compensator coefficient values

Although the main text includes the equations to calculate the filter coefficient values, it is useful to know how they are derived for both the z and the δ -operators.

D.1 The z -filter

Transforming to the z -domain:

Consider the generalise Laplace equation of a 2^{nd} order filter:

$$H(s) = \frac{n_0 + n_1s + n_2s^2}{1 + m_1s + m_2s^2} \quad (\text{D.1.1})$$

The bilinear transform uses $s = \frac{2}{T} \frac{(z-1)}{(z+1)}$ so:

$$H(z) = \frac{n_0 + n_1 \left[\frac{2}{T} \frac{(z-1)}{(z+1)} \right] + n_2 \left[\frac{2}{T} \frac{(z-1)}{(z+1)} \right]^2}{1 + m_1 \left[\frac{2}{T} \frac{(z-1)}{(z+1)} \right] + m_2 \left[\frac{2}{T} \frac{(z-1)}{(z+1)} \right]^2} \quad (\text{D.1.2})$$

Since it is a 2^{nd} filter, there are 2 sample rates that must be considered in the analysis. Eq. D.1.2 can be rearranged to include the current sample rate, T_n , as well as the previous sample rate, T_{n-1} :

$$H(z) = \frac{n_0 + n_1 \left[\frac{2}{T} \frac{(z_1-1)}{(z_1+1)} \right] + n_2 \left[\frac{2}{T} \frac{(z_1-1)}{(z_1+1)} \frac{2}{T} \frac{(z_2-1)}{(z_2+1)} \right]}{1 + m_1 \left[\frac{2}{T} \frac{(z_1-1)}{(z_1+1)} \right] + m_2 \left[\frac{2}{T} \frac{(z_1-1)}{(z_1+1)} \frac{2}{T} \frac{(z_2-1)}{(z_2+1)} \right]} \quad (\text{D.1.3})$$

Eq. D.1.3 can be rearranged to give:

$$H(z) = \frac{z_1 z_2 [n_0 T^2 + 2n_1 T + 4n_2] + z_1 [n_0 T^2 + 2n_1 T - 4n_2] + z_2 [n_0 T^2 - 2n_1 T - 4n_2] + [n_0 T^2 + 2n_1 T + 4n_2]}{z_1 z_2 [T^2 - 2m_1 T + 4m_2] + z_1 [T^2 + 2m_1 T - 4m_2] + z_2 [T^2 - 2m_1 T - 4m_2] + [T^2 - 2m_1 T + 4m_2]} \quad (\text{D.1.4})$$

The z equivalent transfer function of Eq. D.1.1 is defined as:

$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2}} \quad (\text{D.1.5})$$

To achieve the result in the form of Eq. D.1.5, all the variables of Eq. D.1.4 are divided by $z_1 z_2 (T^2 - 2m_1 T + 4m_2)$. The coefficient values are now given as:

$$a_0 = \frac{n_0 T^2 + 2n_1 T + 4n_2}{T^2 + 2m_1 T + 4m_2}$$

$$a_1 = \frac{n_0 T^2 + 2n_1 T - 4n_2}{T^2 + 2m_1 T + 4m_2} z_2^{-1} + \frac{n_0 T^2 - 2n_1 T - 4n_2}{T^2 + 2m_1 T + 4m_2} z_1^{-1}$$

$$\begin{aligned}
a_2 &= \frac{n_0 T^2 - 2n_1 T + 4n_2}{T^2 + 2m_1 T + 4m_2} z_1^{-1} z_2^{-1} \\
b_1 &= \frac{T^2 + 2m_1 T - 4m_2}{T^2 + 2m_1 T + 4m_2} z_2^{-1} + \frac{T^2 - 2m_1 T - 4m_2}{T^2 + 2m_1 T + 4m_2} z_1^{-1} \\
b_2 &= \frac{T^2 - 2m_1 T + 4m_2}{T^2 + 2m_1 T + 4m_2} z_1^{-1} z_2^{-1}
\end{aligned} \tag{D.1.6}$$

By associating the current sample rate, T_n with z_1^{-1} , and the previous sample rate, T_{n-1} with z_2^{-1} , hence:

$$\begin{aligned}
a_0 &= \frac{n_0 \cdot (T_n)^2 + 2 \cdot n_1 \cdot T_n + 4 \cdot n_2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} \\
a_1 &= \frac{n_0 \cdot (T_n)^2 - 2 \cdot n_1 \cdot T_n - 4 \cdot n_2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} + \frac{n_0 \cdot (T_{n-1})^2 + 2 \cdot n_1 \cdot T_{n-1} - 4 \cdot n_2}{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} + 4 \cdot m_2} \\
a_2 &= \frac{n_0 \cdot T_n \cdot T_{n-1} - n_1 \cdot (T_n + T_{n-1}) + 4 \cdot n_2}{T_n \cdot T_{n-1} + m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2} \\
b_1 &= \frac{(T_n)^2 - 2 \cdot m_1 \cdot T_n - 4 \cdot m_2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} + \frac{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} - 4 \cdot m_2}{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} + 4 \cdot m_2} \\
b_2 &= \frac{T_n \cdot T_{n-1} - m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2}{T_n \cdot T_{n-1} + m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2}
\end{aligned} \tag{D.1.7}$$

D.2 The δ -filter

Transforming to the δ -domain:

Consider the generalise Laplace equation of a 2^{nd} order filter:

$$H(s) = \frac{n_0 + n_1 s + n_2 s^2}{1 + m_1 s + m_2 s^2} \tag{D.2.1}$$

The δ transform uses $s = \frac{2\delta}{T(2+\delta)}$ so:

$$H(\delta) = \frac{n_0 + n_1 \left[\frac{2\delta}{T(2+\delta)} \right] + n_2 \left[\frac{2\delta}{T(2+\delta)} \right]^2}{1 + m_1 \left[\frac{2\delta}{T(2+\delta)} \right] + m_2 \left[\frac{2\delta}{T(2+\delta)} \right]^2} \quad (\text{D.2.2})$$

Repeating the procedure carried out with the z -operator, the 2^{nd} δ filter with non-uniform sampling is expressed as:

$$H(\delta) = \frac{n_0 + n_1 \left[\frac{2\delta_1}{T(2+\delta_1)} \right] + n_2 \left[\frac{2\delta_1}{T(2+\delta_1)} \frac{2\delta_2}{T(2+\delta_2)} \right]}{1 + m_1 \left[\frac{2\delta_1}{T(2+\delta_1)} \right] + m_2 \left[\frac{2\delta_2}{T(2+\delta_2)} \right]^2} \quad (\text{D.2.3})$$

Eq. D.2.3 can be rearranged to give:

$$H(\delta) = \frac{\delta_1 \delta_2 [n_0 T^2 + 2n_1 T + 4n_2] + \delta_1 [2n_0 T^2 + 4n_1 T] + \delta_2 [2n_0 T^2] + [4n_0 T^2]}{\delta_1 \delta_2 [T^2 - 2m_1 T + 4m_2] + \delta_1 [2T^2 + 4m_1 T] + \delta_2 [2T^2] + [4T^2]} \quad (\text{D.2.4})$$

The δ equivalent transfer function of Eq. D.2.1 is defined as:

$$H(\delta) = \frac{c_0 + c_1 \delta^{-1} + c_2 \delta^{-2}}{1 + r_1 \delta^{-1} + r_2 \delta^{-2}} \quad (\text{D.2.5})$$

To achieve the result in the form of Eq. D.2.5, all the variables of Eq. D.2.4 are divided by $\delta_1 \delta_2 (T^2 - 2m_1 T + 4m_2)$. The coefficient values are now given as:

$$c_0 = \frac{n_0 T^2 + 2n_1 T + 4n_2}{T^2 + 2m_1 T + 4m_2}$$

$$c_1 = \frac{2n_0 T^2 + 4n_1 T}{T^2 + 2m_1 T + 4m_2} \delta_2^{-1} + \frac{2n_0 T^2}{T^2 + 2m_1 T + 4m_2} \delta_1^{-1}$$

$$c_2 = \frac{4n_0 T^2}{T^2 + 2m_1 T + 4m_2} \delta_1^{-1} \delta_2^{-1}$$

$$\begin{aligned}
r_1 &= \frac{2T^2 + 4m_1T}{T^2 + 2m_1T + 4m_2} \delta_2^{-1} + \frac{2T^2}{T^2 + 2m_1T + 4m_2} \delta_1^{-1} \\
r_2 &= \frac{4T^2}{T^2 + 2m_1T + 4m_2} \delta_1^{-1} \delta_2^{-1}
\end{aligned} \tag{D.2.6}$$

By associating the current sample rate, T_n with δ_1^{-1} , and the previous sample rate, T_{n-1} with δ_2^{-1} , hence:

$$\begin{aligned}
c_0 &= \frac{n_0 \cdot (T_n)^2 + 2 \cdot n_1 \cdot T_n + 4 \cdot n_2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} \\
c_1 &= \frac{2 \cdot n_0 \cdot (T_n)^2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} + \frac{2 \cdot n_0 \cdot (T_{n-1})^2 + 4 \cdot n_1 \cdot T_{n-1}}{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} + 4 \cdot m_2} \\
c_2 &= \frac{4 \cdot n_0 \cdot T_n \cdot T_{n-1}}{T_n \cdot T_{n-1} + m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2} \\
r_1 &= \frac{2 \cdot (T_n)^2}{(T_n)^2 + 2 \cdot m_1 \cdot T_n + 4 \cdot m_2} + \frac{2 \cdot (T_{n-1})^2 + 4 \cdot m_1 \cdot T_{n-1}}{(T_{n-1})^2 + 2 \cdot m_1 \cdot T_{n-1} + 4 \cdot m_2} \\
r_2 &= \frac{4 \cdot T_n \cdot T_{n-1}}{T_n \cdot T_{n-1} + m_1 \cdot (T_n + T_{n-1}) + 4 \cdot m_2}
\end{aligned} \tag{D.2.7}$$

Appendix E

Publications

The published work has been listed in this thesis for convenience. This includes two conference papers:

- M. S. Khan, R.M. Goodall and R. Dixon, Implementation of non-uniform sampling for alias-free processing in digital controls. UKACC International Conference on Control, 2008.
- M. S. Khan, R. Dixon, and R.M. Goodall, Design and analysis of non-uniform rate controllers. UKACC International Conference on Control, 2010.

Appendix F

Sampling models

Much of the theory in this appendix has been reported in Bilinskis (2007).

Randomizations in a sampling scheme can seriously affect the precision of signal processing applications. Therefore it is important to analyse the sampling model that can for the purpose of deliberate non-uniform sampling.

Normal distribution: jitter sampling

This type of sampling occurs naturally in many applications, e.g. imperfect sensors, clock delays, etc. It can be defined by:

$$t_k = kT + \tau_k, \quad T > 0, k = 0, 1, 2, \dots, \quad (\text{F.0.1})$$

where τ_k is a set of independent and identically distributed random variables with zero mean. The sample scheme has been illustrated in Fig. F.1, which depicts the probability density functions, $p(t)$, of time intervals $(t_k - t_0)$ for $k=1, 2, 3, \dots$. The sampling point density function is characterized by:

$$p(t) = \sum_{k=1}^{\infty} p_k(t) \quad (\text{F.0.2})$$

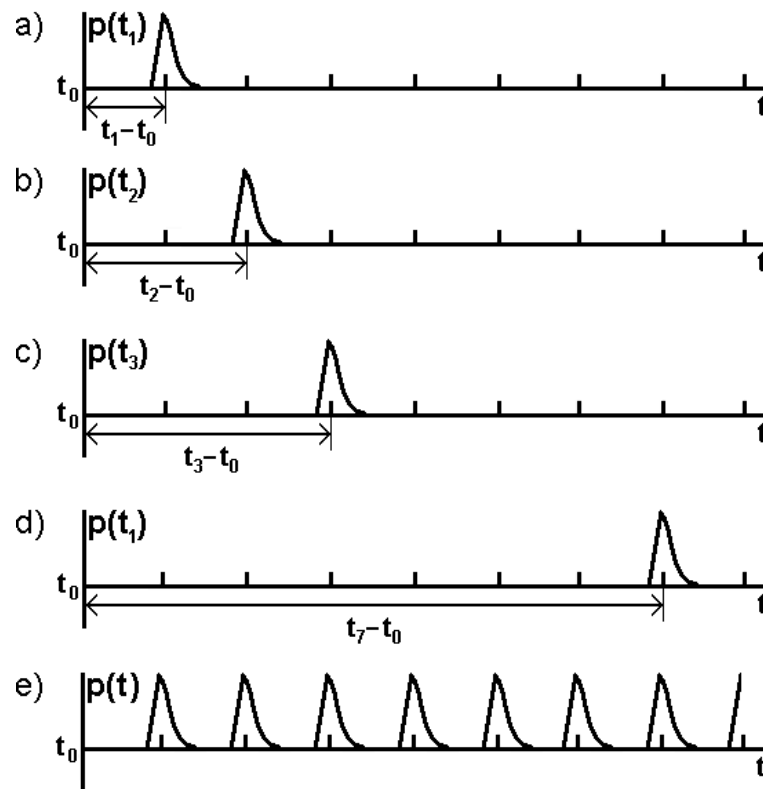


Figure F.1: Probability density functions for jitter sampling. (a), (b), (c), (d) are the functions of the time intervals $t_1 - t_0$, $t_2 - t_0$, $t_3 - t_0$ and $t_7 - t_0$ respectively. (e) concluding sampling point density function.

Note that as t increases, the peaks of the function do not decrease. This implies that when a signal, $x(t)$, is sampled at the instants t_k , some parts of the signal will be sampled with a higher probability than others. This is an undesirable characteristic and will lead to signal processing errors.

Uniform distribution: additive random sampling

This type of sampling can be defined as:

$$t_k = t_{k-1} + \tau_k, \quad k = 0, 1, 2, \dots, \quad (\text{F.0.3})$$

where τ is a random variable whose successive sampling intervals are statistically independent and identically distributed. Consider the time intervals $[0, t_k] = \tau_1 + \tau_2 + \dots + \tau_k$. These variables can be characterised by their probability density functions $p(t)$. Then:

$$\begin{aligned} p_1(t) &= p_\tau(t), \\ p_2(t) &= p_1(t) \star p_\tau(t), \\ &\dots \\ p_k(t) &= p_{k-1}(t) \star p_\tau(t), \end{aligned} \quad (\text{F.0.4})$$

where the \star denotes the convolution operation. When such a sampling scheme is used, the density function may vary within a wide boundary without worsening the sampling conditions since as t increases, the sampling point density function will tend to a constant level. Fig. F.2 illustrates the PDF of the additive random sampling scheme. It can be seen as a function that enables a signal to be sampled with an equal probability of all sampling instances.

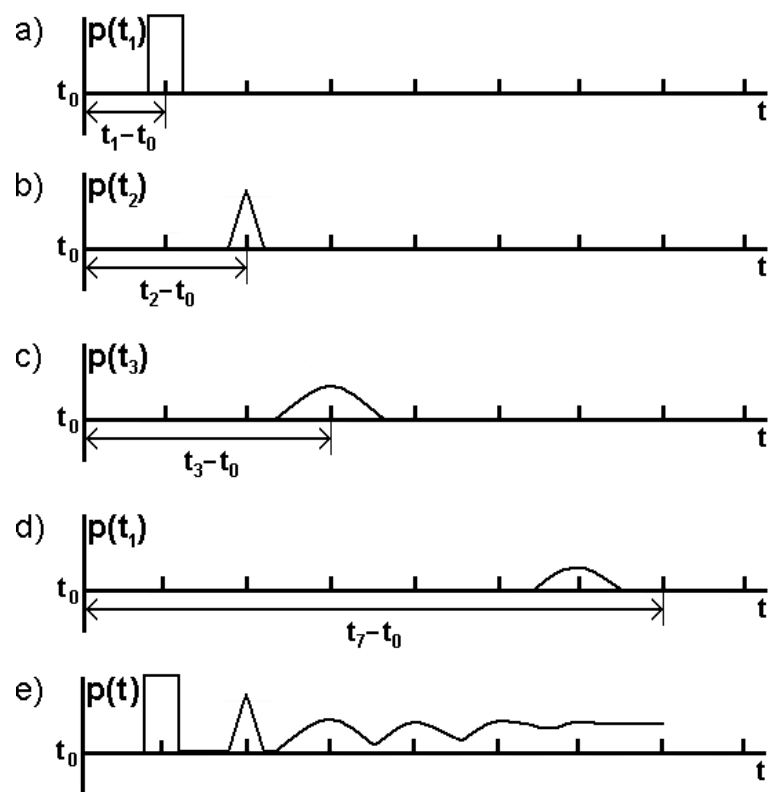


Figure F.2: Probability density functions for additive random sampling. (a), (b), (c), (d) are the functions of the time intervals $t_1 - t_0$, $t_2 - t_0$, $t_3 - t_0$ and $t_7 - t_0$ respectively. (e) concluding sampling point density function.

Bibliography

- Akira, S. (2006), ‘Filter coefficient setting technique for variable filtering process and variable sampling frequency conversion’.
- Albertos, P. & Crespo, A. (1999), ‘Real-time control of non-uniformly sampled systems’, *Control Engineering Practice*, **7**(4), 445–458.
- Albertos, P. & Salt, J. (1990), ‘Digital regulators redesign with irregular sampling’, *11 th. IFAC World. Congress (Preprints)* **8**, 157–161.
- Artyukh, Y., Bilinskis, I., Greitans, M. & Vedin, V. (1997), ‘Signal digitizing and recording in the dasp-lab system’, *Proc. Int. Workshop Sampling Applicat.* pp. 357–360.
- Artyukh, Y., Bilinskis, I., Sudars, K. & Vedin, V. (2008), ‘Multichannel data acquisition from sensor systems’, *Digital Signal Processing and its Applications // The 10-th International Conference and Exhibition.*
- Artyukh, Y., Medniks, I. & Vedin, V. (1997), ‘Virtual oscilloscope of the dasp-lab system’, *Workshop on Sampling Theory and Applications.*
- Arzen, K. E., Cervin, A., Eker, J. & Sha, L. (2000), An introduction to control and scheduling co-design, *in* ‘Decision and Control, 2000. Proceedings of the 39th IEEE Conference on’, Vol. 5, pp. 4865–4870. ID: 1.

- Astrom, K. J. & Bernhardsson, B. M. (2002), ‘Comparison of riemann and lebesgue sampling for first order stochastic systems’, *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on* **2**, 2011–2016.
- Balbastre, P., Ripoll, I. & Crespo, A. (2000), ‘Control task delay reduction under static and dynamic scheduling policies’, *Proc. 7th International Conference on Real-Time Computing Systems and Applications*.
- Beutler, F. J. (1970), ‘Alias-free randomly timed sampling of stochastic processes’, *IEEE Transactions on Information Theory* **16**(2), 147–152.
- Beutler, F. J. & Leneman, O. A. (1966), ‘Random sampling of random processes:stationary point processes’, *Information Control* **9**(4), 325–346.
- Bibian, S. & Jin, H. (2000), ‘Time delay compensation of digital control for dc switchmode power supplies using prediction techniques’, *Power Electronics, IEEE Transactions on* **15**(5), 835–842. ID: 1.
- Bilinskis, I. (2007), *Digital Alias-free Signal Processing*, John Wiley and Sons Ltd.
- Bilinskis, I. & Cain, G. D. (1996), ‘Digital alias-free signal processing tolerance to data and sensor fault’, *IEE Colloquium on Intelligent Sensors*.
- Bilinskis, I. & Mikelsons, A. (1990), ‘Application of randomized or irregular sampling as an anti-aliasing technique’, *Signal Processing V: Theories and Application, Elsevier Science Publishers: Amsterdam* pp. 505–508.
- Bilinskis, I. & Mikelsons, A. (1992), *Randomized Signal Processing*, Prentice Hall, London.
- Bretthorst, G. L. (2008), ‘Nonuniform sampling: Aliasing and bandwidth’, *American Astronomical Society, 197th AAS Meeting*.

- Cervin, A., Henriksson, D., Lincoln, B. & Arzen, K. E. (2002), ‘Jitterbug and truetype: Analysis tools for real-time control systems’, *2nd Workshop on Real-Time Tools*.
- Chaumette, F., Mansard, N. & Remazeilles, A. (2009), ‘Continuity of varying-feature-set control laws’, *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*.
- Clarke, D. W. & Maslen, S. P. (2007), ‘Discretising controllers with slow sampling’, *Control Theory and Applications, IET* **1**(3), 624–635.
- Colandairaj, J., Irwin, G. W. & Scanlon, W. G. (2007), ‘A co-design solution for wireless feedback control’, *IEEE International Conference on Networking, Sensing and Control* pp. 404–409.
- Cuenca, A., Garcia, P., Arzen, K. E. & Albertos, P. (2009), A predictor-observer for a networked control system with time-varying delays and non-uniform sampling, in ‘European Control Conference’, Budapest, Hungary.
- Dannenbergh, K. D. (1972), ‘Analysis and control of randomly sampled systems’, *PhD Thesis, Southern Methodist University*.
- Dorf, R. C. & Bishop, R. H. (2007), *Modern control systems*, 11 edn, Prentice Hall.
- Edwards, R. & Durkin, J. (1968), ‘Application of non-uniform quantization to closed-loop digital systems’, *Journal of Scientific Instruments (Journal of Physics E)* **2**(Series 2), 321–326.
- Eng, F. (2007), ‘Non-uniform sampling in statistical signal processing’, *PhD Thesis, Linköping University*.

- Feuer, A. & Goodwin, G. C. (1996), *Sampling in Digital Signal Processing and Control*, Birkhauser Boston Inc., Cambridge, MA, USA.
- Forsythe, W. & Goodall, R. M. (1991), *Digital Control: Fundamentals, Theory and Practice*, Macmillian Education LTD., Loughborough, UK.
- Freudenberg, J. S., Middleton, R. H. & Braslavsky, J. H. (1995), 'Inherent design limitations for linear sampled-data feedback systems', *International Journal of Control* **61**(6), 1387–1421.
- Goodall, R. M. (2001), 'Perspectives on processing for real-time control', *Annual Reviews in Control* **25**, 123–131.
- Goodall, R. M., Jones, S. & Cumplido-Parra, R. (1998), 'Digital filtering for high performance real-time control', *Digital Filters: An Enabling Technology (Ref. No. 1998/252)*, *IEE Colloquium on* pp. 7/1–7/5.
- Goodwin, G. C. & DeSouza, C. E. (1984), 'Intersample variances in discrete minimum variance control', *IEEE Transactions in Automatic Control* **AC-29**, 759–761.
- Goslin, G. R. (1996), 'Implementation dsp functions in fpgas to reduce cost and boost performance', *EDN*.
- Gran, R. & Berman, H. (1974), 'Design principles for digital autopilot synthesis', *Journal of Aircraft* **11**(7), 414–422.
- Houpis, C. H. & Lamont, G. B. (1985), *Digital Control Systems: Theory, hardware, software*, McGraw Hill.
- Hugues, G. & Liuping, W. (2008), *Identification of continuous-time models from sampled data*, Springer.
- Isermann, R. (1989), *Digital Control Systems*, Springer-Verlag NY.

- Johansson, H. & Lowenborg, P. (2004), ‘Reconstruction of nonuniformly sampled bandlimited signals using time-varying discrete-time fir filters’, *Proceedings of 12th European Signal Processing Conference (EUSIPCO '04)*.
- Johnson, J. M. (1974), ‘Digital filters using observers applied to icbm control system design’, *Journal of Spacecraft* **11**(7), 498–504.
- Jugo, J. & Arredondo, I. (2007), ‘Adaptive control for attenuating vibrations using non uniform sampling’, *9th IFAC Workshop on Adaptation and learning in control and signal processing*.
- Katznelson, Y. (1976), *An Introduction to Harmonic Analysis*, Dover Publications.
- Keil (2004), ‘uvision2 ide and debugger’, www.keil.com/uvision2.
- Kopetz, H. (1997), *Real-time systems: design principles for distributed embedded applications*, Springer.
- Kovacshazy, T., Peceli, G. & Simon, G. (2001), ‘Transient reduction in reconfigurable control systems utilizing structure dependence’, *IEEE Instrumentation and Measurement*.
- Kuo, B. (1980), *Digital control systems*, Rinehart and Winston, Inc, USA.
- Landau, I. D., Landau, Y. D. & Zito, G. (2006), *Digital control systems: design, identification and implementation*, Springer.
- Le, A. & McCann, R. (2007), Event-based measurement updating kalman filter in network control systems, in ‘Region 5 Technical Conference, 2007 IEEE’, pp. 138–141. ID: 1.

- Li, Z. & Fang, H. (2006), 'Fuzzy controller design for networked control system with time-variant delays', *Journal of Systems Engineering and Electronics* **17**(1), 172–176.
- Marti, P. (2002), 'Analysis and design of real-time control systems with varying control timing constraints'.
- Marti, P., Frigola, M. & Velasco, M. (2005), 'Bandwidth management for distributed control of highly articulated robots', *Robotics and Automation, Proceedings of the 2005 IEEE International Conference on* pp. 265–270.
- Marti, P., Fuertes, J. M. & Fohler, G. (2001), Minimising sampling jitter degradation in real-time control systems, in 'IV Jornadas de tiempo real'.
- Marti, P., Fuertes, J. M., Fohler, G. & Ramamritham, K. (2001), 'Jitter compensation for real-time control systems', *Real-Time Systems Symposium, IEEE Proceedings* pp. 39–48.
- Martin, R. J. (1998), 'Irregular sampled signals: Theories and techniques for analysis', *PhD Thesis, University of London*.
- Marvasti, F. (2001), *Non-uniform Sampling: Theory and Practice*, Springer.
- Masry, E. (1978), 'Alias-free sampling: an alternative conceptualization and its applications', *IEEE Transactions on Information Theory* **24**(3), 317–324.
- McCann, R., Gunda, A. K. & Damugatla, S. R. (2004), 'Improved operation of network control systems using lebesgue sampling', *Industry Applications Conference, 2004. 39th IAS Annual Meeting. Conference Record of the 2004 IEEE* **2**, 1211–1216.
- Middleton, R. H. & Goodwin, G. C. (1990), *Digital Control and Estimation: A Unified Approach*, Prentice Hall Professional Technical Reference.

- Mikelsons, A. & Greitans, M. (1996), ‘Fault detection based on spectral estimation of nonuniformly sampled signals’, *IEE Colloquium on Intelligent Sensors*.
- Miskowicz, M. (2007), ‘Asymptotic effectiveness of the event based sampling according to the integral criterion’, *Sensors* **7**, 16–37.
- Mitra, S. K. (2001), *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill.
- Murthy, S. N., Alvis, W., Shirodkar, R., Valavanis, K. & Moreno, W. (2008), ‘Methodology for implementation of unmanned vehicle control on fpga using system generator’, *Devices, Circuits and Systems, 2008. ICCDCS 2008. 7th International Caribbean Conference on*.
- Nilsson, J. (1998), ‘Real-time control systems with delays’, *PhD Thesis, Lund Institute of Technology*.
- Nise, N. S. (2007), *Control systems engineering*, John Wiley and Sons.
- Nyquist, H. (1928), ‘Certain topics in telegraph transmission theory’, *Transactions on American Institute of Electrical Engineers* **47**, 617–644.
- Papenfuss, F., Artyukh, Y., Boole, E. & Timmermann, D. (2003), ‘Nonuniform sampling driver design for optimal adc utilization’, *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, **4**, 516–519.
- Rabbath, C. A. & Hori, N. (2001), ‘A framework for discrete-time models of continuous-time systems’, *American Control Conference, 2001. Proceedings of the 2001* **4**, 2578–2583 vol.4. ID: 1.

- Ragazzini, J. & Franklin, G. (1958), *Sampled data control systems*, McGraw-Hill, NY.
- Rake, H. (1980), 'Step response and frequency response methods', *Automatica* **16**(5), 519–526.
- Sanchis, R., Sala, A. & Albertos, P. (1997), 'Scarce data operating conditions: process model identification', *International Federation of Automatic Control on System Identification* pp. 463–474.
- Scaechter, D. B. (1982), 'Hardware demonstration of a flexible bean control', *Journal of Guidance and Control* **5**(1), 48–53.
- Schneider, A. M., Kaneshige, J. T. & Groutage, F. D. (1991), 'Higher order s-to-z mapping functions and their application in digitizing continuous-time filters', *Proceedings of the IEEE* **79**(11), 1661–1674.
- Shannon, C. E. (1949), 'Communication in the presence of noise', *Proc. Institute of Radio Engineers* **37**(1), 10–21.
- Shapiro, H. S. & Silverman, R. A. (1960), 'Alias-free sampling of random noise', *SIAM Journal on Applied Mathematics* **8**(2), 245–248.
- Simon, G., Kovacs haz y, T. & Peceli, G. (2002), Transient management in reconfigurable control systems, Technical report, Budapest University of Technology and Economics.
- SMAC (2004), *Actuator data sheet - electric linear/rotary moving coil actuators lal90-50, 2004*.
- Sonnaillon, M. O. & Bonettot, F. J. (2007), Fpga implementation of a phase locked loop based on random sampling, in 'III Southern Conference on Programmable Logic'.

- Sonnaillon, M. O., Urteaga, R. & Bonettot, F. J. (2008), 'High-frequency digital lock-in amplifier using random sampling', *IEEE Instrumentation and Measurement* **57**(3), 616–621.
- Svenk, G. (2002), *Object-oriented programming: using C++ for engineering and technology*, Delmar Learning.
- Tarczynski, A., Valimaki, V. & Cain, G. D. (1997), 'Fir filtering of nonuniformly sampled signals', *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on* **3**, 2237–2240.
- Valimaki, V. & Laakso, T. I. (1998), 'Suppression of transients in variable recursive digital filters with novel and efficient cancellation method', *IEEE Transactions on Signal Processing* **46**(12), 3408–3414.
- Valimaki, V. & Tarczynski, A. (1996), 'Modifying fir and iir filters for processing signals with lost samples', *Proc. NOR SIG '96* **1**, 359–362.
- Valimaki, V., Laakso, T. I. & Mackenzie, J. (1995), 'Elimination of transients in time-varying allpass fractional delay filters with application to digital waveguide modeling', *Proc. Int. Computer Music Conference* pp. 303–306.
- Voulgaris, P. G. & Bamieh, B. (1993), 'Optimal h and h₂ control of hybrid multi-rate systems', *Systems and Control Letters* **20**(4), 249–261.
- Wang, G. (2008), 'Analysis of m-channel time-varying filter banks', *Digital Signal Processing* **18**(2), 127–147.
- Wittenmark, B., Nilsson, J. & Torngren, M. (1995), Timing problems in real-time control systems, *in* 'American Control Conference, 1995. Proceedings of the', Vol. 3, pp. 2000–2004. ID: 1.

- Wojtiuk, J. J. (2000), ‘Randomized sampling for radio design’, *PhD Thesis, University of South Australia*.
- Wu, X. (2005), ‘One-bit processing for real-time control’, *PhD Thesis, Loughborough University*.
- Xilinx (2006), ‘User guide for ug230 spartan-3e fpga starter kit board’.
- Zetterberg, L. H. & Zang, Q. (1988), ‘Elimination of transients in adaptive filters with application to speech coding’, *Signal Processing* **15**(4), 419–428.