

Analysis, Simulation and Design of Nonlinear RF Circuits

Tao Xu

A dissertation submitted in fulfilment of the requirements for the Degree of

Doctor of Philosophy

Research Supervisor: Dr. Marissa Condon

School of Electronic Engineering

Dublin City University

June 2009

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of *Doctor of Philosophy* is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed	·	(Candidate) ID No.:	56107625
Date:			

This dissertation is a gift to my Dad. I believe you are always looking at and proud of me in the other world.

Acknowledgements

I would like to thank my Mum, Lanping Ma and wife, Miao He. Without your support, I would not have achieved this.

I want to thank my supervisor, Dr. Marissa Condon for her amazing supervision. She always gave me help whenever I needed it.

Dr. Zhipeng Ye is another important person. I am grateful to him for the helpful discussions about the delta-sigma modulators.

I would also like to thank Dr. Georgi Grahovski for his constant friendliness, help and advice.

I would like to thank Biji, John, Diana, Marie, Patrick and Conor for their help throughout the years. Also, I would like to thank the technical staff whose friendliness was so much appreciated.

Contents

Co	i		iv	
Li	st of l	ts iv Figures ix Tables xv oduction 1 kground to Computer Aided Design (CAD) Algorithms 6 Numerical Integration Methods 6 Euler Methods 9 2.2.1 Forward Euler Method 9 2.2.2 Backward Euler Method 10 Trapezoidal Method 11 Runge-Kutta Method 12 Adams' Methods 14 2.5.1 Adams-Bashforth Method 15 Stiff Systems and Stability Definitions 15 Stiff Systems and Stability Definitions 16 é-Based Algorithms for Numerical Integration of Ordinary Different Equations (ODE) 18 Single-Step Padé-Based Algorithms 19		
Li	st of [Fables		iv ix xv 1 6 . 6 . 9 . 9 . 10 . 11 . 12 . 14 . 15 . 14 . 15 . 15 . 15 . 16 en- 18 . 19 . 19
1	Intr	oductio	n	1
2	Bac	kground	d to Computer Aided Design (CAD) Algorithms	6
	2.1	Nume	rical Integration Methods	6
	2.2	Euler I	Methods	9
		2.2.1	Forward Euler Method	9
		2.2.2	Backward Euler Method	10
	2.3	Trapez	zoidal Method	11
	2.4	Runge	-Kutta Method	12
	2.5	Adams	s' Methods	14
		2.5.1	Adams-Bashforth Method	14
		2.5.2	Adams-Moulton method	15
	2.6	Stiff S	ystems and Stability Definitions	15
	2.7	Conclu	usions	16
3	Pad tial 1	é-Based Equatio	Algorithms for Numerical Integration of Ordinary Differen- ons (ODE)	18
	3.1	Single	-Step Padé-Based Algorithms	19
		3.1.1	Padé-Taylor Method	19

CONTENTS

		3.1.2	Padé-Exponential Method	20
		3.1.3	Padé-Richardson Method	22
		3.1.4	Padé Approximation with Predictor-Corrector	24
	3.2	Multi-	Step Padé-Based Algorithms	25
		3.2.1	Explicit Multi-Step Padé Approximation	25
		3.2.2	Implicit Multi-Step Padé Approximation	28
	3.3	Applic	ation and Validation	28
		3.3.1	Classic Test System	28
		3.3.2	MESFET Amplifier	35
	3.4	Conclu	isions	37
4	Filo	n-Type	Numerical Quadrature for Highly-Oscillatory Systems.	38
	4.1	Backg	round to Highly Oscillatory Quadrature	38
	4.2	Filon-7	Type Numerical Approximation	40
	4.3	Examp	ole	42
	4.4	Numer	ical Results	47
		4.4.1	Amplitude Modulation	48
		4.4.2	Digital Modulation	51
	4.5	Compa	arison with the Envelope Simulation	53
	4.6	Conclu	isions	57
5	Sim	ulation	Techniques to Capture the Locking Range of an Injection-	
	Locl	ked Fre	quency Divider	58
	5.1	Introdu	action to the Injection-Locked Frequency Divider (ILFD)	58
	5.2	Backg	round to the LC-Oscillator Based ILFD	59
		5.2.1	Effect of Negative Resistance on LC Oscillators	59
		5.2.2	LC Oscillator with Cross-Coupled Negative Resistor	60
	5.3	Overvi	ew of the Existing Models of Injection-Locked Oscillators	62
		5.3.1	Adler's Model	62

CONTENTS

		5.3.2	Miller's Model	63
		5.3.3	Verma's Model	64
		5.3.4	Xu's Model	66
	5.4	Existir	ng Methods to Obtain the Locking Range of Oscillators	73
		5.4.1	Adler Approach	73
		5.4.2	Graphical Analysis of the Phase Characteristic	77
		5.4.3	Analysis with Harmonic Balance	79
	5.5	Novel	Use of the Warped Multitime Partial Differential Equation (WaMP	DE)
		to Dete	ermine the Locking Range	83
		5.5.1	The Use of the WaMPDE for Modelling the Injection-Locked	
			Oscillator	83
		5.5.2	The Determination of the Limit of the Locking Range	86
		5.5.3	Determination of the Locking Range from Use of the WaMPDE	
			in Conjunction with the Devil's Staircase	91
	5.6	A Nov	el Method Utilising the Multiple-Phase-Condition Envelope Fol-	
		lowing	g Method (MPCENV) to Determine the Locking Range	93
		5.6.1	Background to the Envelope Following Method	93
		5.6.2	Multiple-Phase-Condition Envelope Following method	94
	5.7	Numer	rical Results and Experimental Validation	97
		5.7.1	Experimental Equipments Setup	97
		5.7.2	Numerical Results	98
	5.8	Conclu	usions	101
6	Ana	lysis an	d Design of Digital Delta-Sigma Modulators (DDSM)	102
	6.1	Backg	round to Delta-Sigma Modulation	102
	6.2	Introdu	uction of Digital Delta-Sigma Modulators	106
	6.3	State-o	of-the-art Architectures of DDSMs	111
		6.3.1	The Conventional DDSMs	111

			6.3.1.1	The Conventional MASH	111
			6.3.1.2	The Conventional EFM	115
			6.3.1.3	The Sequence Length of Conventional DDSMs	117
		6.3.2	HK-MA	SH DDSMs	118
	6.4	Mathe	matical Ar	nalysis of the MASH DDSMs employing Multi-Moduli	122
		6.4.1	The Suit	ability of the Multi-Modulus MASH-DDSM	124
		6.4.2	The Effe	ect of the Multi-Moduli on the Modulator Sequence	
			Length		126
		6.4.3	A Mathe	matical Analysis of Higher-Order MM-EFMs	131
	6.5	The Pr	oposed Str	ructure of the MASH DDSM	134
	6.6	The Si	mulation I	Results	135
	6.7	Conclu	usions		138
7	Con	clusion	S		147
A	A-st	ability	and <i>L</i> -stal	bility of Padé approximations	150
B	Rich	nardson	extrapola	ntion and improvement of accuracy	152
С	ME	SFET a	mplifier d	etails	154
D	Sche	ematics	of the dio	de rectifier circuit used in ADS	159
E	Mod	lels of E	DDSMs us	ed in Simulink	161
F	List	of relev	ant publi	cations	164
Bi	ibliography 16		166		

Analysis, Simulation and Design of Nonlinear RF Circuits

Tao Xu

Abstract

The PhD project consists of two parts. The first part concerns the development of Computer Aided Design (CAD) algorithms for high-frequency circuits. Novel Padébased algorithms for numerical integration of ODEs as arise in high-frequency circuits are proposed. Both single- and multi-step methods are introduced. A large part of this section of the research is concerned with the application of Filon-type integration techniques to circuits subject to modulated signals. Such methods are tested with analog and digital modulated signals and are seen to be very effective. The results confirm that these methods are more accurate than the traditional trapezoidal rule and Runge-Kutta methods.

The second part of the research is concerned with the analysis, simulation and design of RF circuits with emphasis on injection-locked frequency dividers (ILFD) and digital delta-sigma modulators (DDSM). Both of these circuits are employed in fractional-N frequency synthesizers. Several simulation methods are proposed to capture the locking range of an ILFD, such as the Warped Multi-time Partial Differential Equation (WaMPDE) and the Multiple-Phase-Condition Envelope Following (MP-CENV) methods. The MPCENV method is the more efficient and accurate simulation technique and it is recommended to obviate the need for expensive experiments. The Multi-stAge noise Shaping (MASH) digital delta-sigma modulator (DDSM) is simulated in MATLAB and analysed mathematically. A novel structure employing multi-moduli, termed the MM-MASH, is proposed. The goal in this design work is to reduce the noise level in the useful frequency band of the modulator. The success of the novel structure in achieving this aim is confirmed with simulations.

List of Figures

2.1	The integral from t_n to t_{n+1} .	11
3.1	The comparison of the simulation results from Adams Moulton (—)	
	and exact analysis (**)	30
3.2	The comparison of the simulation results from Runge-Kutta $(-)$ and	
	<i>exact analysis (**).</i>	31
3.3	The comparison of the simulation results from Padé approximation (—	
) and exact analysis (**). \ldots \ldots \ldots \ldots \ldots	32
3.4	The comparison of the simulation results from Padé-based multi-step	
	predictor-corrector method (—) and exact analysis (**). \ldots .	34
3.5	Schematics of the MESFET amplifier.	36
3.6	The comparison of the simulation results from Adams-Moulton $(-)$	
	and Padé method (**).	36
4.1	Diode rectifier circuit	42
4.2	The exact solution of the amplitude-modulated equation with $I_0 = 1$,	
	C = 1, R = 1 and k = 40.	48
4.3	The solution for the amplitude-modulated case with the trapezoidal rule.	49
4.4	The solution for the amplitude-modulated equation with the third-order	
	explicit Runge-Kutta method	50

4.5	The solution for the amplitude-modulated case using the Filon-type	
	method, combined with waveform relaxation	50
4.6	The solution for the amplitude-modulated case with realistic values	
	using the Filon-type method combined with waveform relaxation	51
4.7	The exact solution of the digitally modulated equation.	52
4.8	A very small time sub-interval of Fig. 4.7	52
4.9	The solution of the digital-modulated equation by the trapezoidal rule.	54
4.10	The solution of the digital-modulated equation by third-order explicit	
	Runge-Kutta method	54
4.11	The solution of the digital-modulated equation by the Filon-type method,	
	combined with waveform relaxation	55
4.12	A Modulated signal and its simulated time-varying spectrum [1]	56
5.1	Decaying impulse responses of a tank	59
5.2	Addition of negative resistance to cancel tank losses.	60
5.3	Use of an active circuit to provide negative resistance	60
5.4	Cross-coupled CMOS LC Oscillator.	61
5.5	Adler's model for the injection-locked oscillator.	62
5.6	Miller-type model for the injection-locked oscillator	64
5.7	General model for the injection-locked frequency divider	65
5.8	Circuit schematic.	67
5.9	The simplified circuit.	67
5.10	The schematic used in PSpice to capture the driving point characteristics.	68
5.11	Driving point characteristics: Solid: Experimental; Dotted: Cubic Fit	
	To Experiment; Dashed: PSpice	69
5.12	Model of the free-running oscillator	70
5.13	I_D against V_{GS}	70

5.14 The loop enhanced model of the ILF.	D	71
5.15 Analog Multiplier as a phase detector	<i>)r.</i>	72
5.16 Vector diagram of the instantaneous	voltages.	74
5.17 Sample oscillator circuit [2]		74
5.18 Phase versus frequency for a simple	tuned circuit	75
5.19 (a)Conceptual oscillator.(b) Frequen	ncy shift due to additional phase	
shift. (c) Open-loop characteristics.(d)Frequency shift by injection	77
5.20 Phasor diagram		78
5.21 Phasor diagram with maximum angl	e between I_{inj} and I_T	79
5.22 General Circuit Model for ILFD		80
5.23 Relation between τ_1 , t and $\phi_1(t)$		85
5.24 Steady state: Non-locked oscillator.		88
5.25 Comparison of the norm of the Jacob	pian column when the oscillator is	
locked (continuous line) and non-loc	ked (dashed line)	88
5.26 Steady state: Locked oscillator		89
5.27 Local frequency and norm of Jacobia	an column (dashed line) vs. itera-	
tion number		90
5.28 Experimentally measured Devil's sta	iircase diagram showing lockings	
at $\omega_{ m inj}/\omega_{ m o}=2$ and $\omega_{ m inj}/\omega_{ m o}=4$		91
5.29 Forward-Euler-based envelope-follo	wing method	93
5.30 Backward-Euler-based envelope-foll	owing method	94
5.31 Experiment setup		97
5.32 The schematic of the LC-ILFD		98
5.33 The staircase obtained from simulati	on and experiment	99
5.34 The locking range of the LC-ILFD co	aptured from experiments	101
6.1 An example of 4-bit PCM		103

6.2	Block diagram of the delta-modulation process.	103
6.3	Derivation of Delta-Sigma Modulation from Delta Modulation	104
6.4	Block diagram of delta-sigma modulation	105
6.5	Spectrum of a signal	105
6.6	Spectrum of an oversampled signal	105
6.7	The Fractional-N Frequency Synthesizer	107
6.8	The architecture of an lth order single-loop delta-sigma modulator	108
6.9	Model of a dithered MASH DDSM	109
6.10	The power spectral density of a 9-bit MASH 1-1-1 DDSM [3]	110
6.11	MASH DDSM architecture	111
6.12	EFM1: First-order error-feedback modulator	113
6.13	The linear model of an EFM1	113
6.14	Higher-order error-feedback modulator.	116
6.15	Linear model of higher-order error-feedback modulator	116
6.16	The modified EFM1 used in the HK-MASH	118
6.17	The linear model of the HK-EFM1	120
6.18	The structure of the HK-MASH	121
6.19	MM-EFM1: The modified first-order error-feedback modulator used	
	in the MM-MASH	123
6.20	The MM-MASH consists of only MM-EFM1	125
6.21	Higher order error-feedback modulator.	132
6.22	Structure of the 2nd order error-feedback modulator.	132
6.23	The structure of the MM-MASH	134
6.24	The autocorrelation result for the 5-bit MM-MASH 1-1-1	136
6.25	The power spectral density of the dithered 5-bit conventional MASH	
	DDSM and non-dithered 5-bit MM-MASH.	139

6.26	The power spectral density of the dithered 6-bit conventional MASH	
	DDSM and non-dithered 6-bit MM-MASH	139
6.27	The power spectral density of the dithered 7-bit conventional MASH	
	DDSM and non-dithered 7-bit MM-MASH	140
6.28	The power spectral density of the dithered 8-bit conventional MASH	
	DDSM and non-dithered 8-bit MM-MASH	140
6.29	The power spectral density of the dithered 9-bit conventional MASH	
	DDSM and non-dithered 9-bit MM-MASH	141
6.30	The power spectral density of the dithered 10-bit conventional MASH	
	DDSM and non-dithered 10-bit MM-MASH	141
6.31	The power spectral density of the dithered 11-bit conventional MASH	
	DDSM and non-dithered 11-bit MM-MASH	142
6.32	The power spectral density of the 11-bit HK-MASH DDSM and 11-bit	
	<i>MM-MASH</i>	142
6.33	The power spectral density of different structures of 3rd order 5-bit	
	<i>MM-MASH</i>	143
6.34	The power spectral density of different structures of 3rd order 6-bit	
	<i>MM-MASH</i>	143
6.35	The power spectral density of different structures of 3rd order 7-bit	
	<i>MM-MASH</i>	144
6.36	The power spectral density of different structures of 3rd order 8-bit	
	<i>MM-MASH</i>	144
6.37	The power spectral density of different structures of 3rd order 9-bit	
	<i>MM-MASH</i>	145
6.38	The power spectral density of different structures of 3rd order 10-bit	
	<i>MM-MASH</i>	145

6.39	The power spectral density of different structures of 3rd order 11-bit	
	<i>MM-MASH</i>	146
C.1	Schematic of the MESFET amplifier	154
D.1	The diode rectifier circuit with amplitude modulated input signal	159
D.2	The diode rectifier circuit with digitally modulated input signal. \ldots	160
E.1	The model of EFM1 used in Simulink.	161
E.2	The model of HK-EFM1 used in Simulink	162
E.3	The model of EFM2 used in Simulink.	162
E.4	The model of MASH 1-1-1 used in Simulink	163
E.5	The model of MASH 1-2 used in Simulink.	163

List of Tables

1.1	The market value of the top 5 EDA companies	2
3.1	Total simulation time of different methods	35
5.1	The value of the selected components used in both experiment and	
	simulation.	99
5.2	Locking range, where M, E and W represent MPCENV, Experiment	
	and WaMPDE	100
6.1	The preferred initial conditions for the conventional DDSMs	117
6.2	The sequence length of the conventional DDSMs	117
6.3	The specifically-chosen a for various modulator word lengths	119
6.4	A comparison between the maximum sequence length of the conven-	
	tional MASH and the HK-MASH	119
6.5	Some sample moduli of the 3rd order MM-MASH	135
6.6	A comparison of the sequence lengths for the HK-MASH and MM-MASH	.137

Chapter 1 Introduction

Computer Aided Design (CAD) techniques are widely used in all branches of engineering from construction engineering to mechanical and electronic engineering. Construction and mechanical engineers draw both 2-Dimensional (2D) vector-based rendering and 3-Dimensional (3D) solid models with the aid of CAD software such as Auto CAD and Pro/E. Electronic engineers, on the other hand, use CAD software for for the design, analysis and optimisation of circuits and systems and also for drawing drafts such as Printed Circuit Boards (PCB) and circuit layouts. For electronic engineering applications, CAD software tools are termed Electronic Design Automation (EDA) tools. The earliest EDA tools were developed in Universities and are presented as open source. "Berkeley VLSI Tools Tarball" was one of the most popular software tools empoloyed by engineers to design VLSI in the 1970s [4]. In the 1980s, industry began developing EDA tools. A lot of traditional electronic companies began their research on EDA internally, such as Hewlett Packard, Tektronix and Intel. However, attracted by the bright future of EDA development, companies specifically for EDA development were also founded in this period. For example, Mentor Graphics was founded by managers from Tektronix, Daisy Systems was founded largely by designers from Intel, and Valid Logic Systems was founded by developers from Lawrence Livermore National Laboratory and Hewlett Packard [4]. Because of competition and mergers over the he next 30 years, only a few EDA companies are still in existence and these companies occupy the whole market. For example, Cadence and Mentor Graphics share most of the market for physical IC design, Synopsys occupies the logic synthesis area and Agilent provides good high-frequency simulation software. The market value of the top five EDA businesses is shown in Table 1.1 [4]. Note that the EDA division is only part of Agilent Technologies. The market value (\$11 billion) comes from all of its products such as EDA software (ADS), electronic measurement tools and life science applications.

 Table 1.1: The market value of the top 5 EDA companies.

Company	Location	Market Value (March 2009)
Agilent Technologies	Santa Clara, California	\$11 billion
Synopsys	Mountain View, California	\$2550 million
Cadence Design Systems	San Jose, California	\$990 million
Mentor Graphics	San Jose, California	\$410 million
Zuken Inc.	Yokohama, Japan	\$149 million

The development and advancement of CAD tools is extremely important in today's electronic and RF industry. The complexity, diversity and level of integration of electronic circuits has grown exponentially in recent times and CAD tools need to be redeveloped for these new and multifarious applications. In this thesis, two areas of CAD development shall be addressed. The first is on the numerical integration techniques that are employed. The second is on the modelling approaches for fractional-N frequency synthesisers and the components of such synthesisers. The basic numerical integration techniques of the earliest CAD tools are totally inappropriate for the advanced modulation formats in current communication applications. Hence, the thesis shall address the development of novel approaches that are suited to modulated signals and highly stiff systems. In Chapter 2, the background to numerical integration methods is introduced. Both of the advantages and disadvantages of these methods are described. Some important concepts are also explained in detail such as stability and stiffness.

In Chapter 3, novel Padé-based algorithms are proposed for the discrete-time integration of stiff non-linear differential equations. The basic single-step Padé-Taylor method and its form as a predictor corrector was introduced in [5]. However, in this thesis, a new condition is introduced to ensure that the method is always A- and Lstable. Furthermore, the use of a novel Padé-Exponential method and Richardson extrapolation are recommended for increasing efficiency and accuracy. Multi-step methods are also suggested. The efficacy of the methods is shown using two examples. The results are compared with those from the Adam-Moulton and Runge-Kutta techniques. The methods are suitable for application in any field of science requiring efficient and accurate numerical solution of stiff differential equations.

An effective Filon-type numerical integration scheme is introduced in Chapter 4. This novel technique is proposed for an efficient numerical solution of systems of highly oscillatory ordinary differential equations that arise in electronic systems subject to modulated signals. Olver [6] has shown that such a technique is beneficial if the system oscillates at a very high frequency. He also demonstrated that asymptotic expansion is helpful to bound the error to a small value. We extend his work and apply it to a rectifier circuit. The Filon-type method and waveform relaxation techniques are combined to solve nonlinear implicit systems of ODEs. The proposed method is compared with the traditional methods such as the trapezoidal rule and Runge-Kutta methods. This comparison shows that the proposed approach can be very effective when dealing with systems of highly oscillatory differential equations.

In many branches of science, the phenomenon of injection locking is of importance. This phenomenon occurs when the natural frequency of an oscillator changes to become identical to or an integer multiple of an external perturbing frequency. In

wireless communications, the phenomenon has been exploited for very beneficial purposes in applications such as frequency synthesis. In the feedback loop of a frequency synthesiser, a frequency pre-scaler is employed to divide the frequency by a fixed number. Injection Locked Frequency Dividers (ILFD) consume less power than static dividers [7] and hence are preferable for low-power wireless applications. Unfortunately, the bandwidth over which locking occurs for ILFDs is limited. However, they are usually employed in LC-VCO-based Phase Locked Loops (PLL) which have a limited tuning range. Consequently, the restriction on the bandwidth of the ILFDs is not an impediment to their usefulness. However, the ability to determine it to an adequate level of accuracy is an important requirement for the electronic design industry [7] [8]. The Devil's Staircase [9] was introduced as an experimental technique to measure the locking range of an ILFD. Since it requires expensive equipment and takes a long time, some analysis and simulation techniques were introduced to predict the locking range. For example, Ye introduced a method using expressions derived by Harmonic Balance analysis [8] to predict the locking range and Razavi [10] captured the locking range from phasor diagram.

In Chapter 5, several simulation techniques are proposed for the determination of the locking range of an Injection-Locked Frequency Divider (ILFD) or any general oscillator circuit. The first approach is based on the Warped Multi-time Partial Differential Equation (WaMPDE) [11]. This is a variant of the standard multi-time scale model [11]. For the determination of the locking range, three time scales are considered in the warped multi-time scale model. The first time scale is for the oscillator autonomous solution. The second time-scale is for the input signal to which the oscillator circuit synchronises when locking occurs. The third time scale is for the transient evolution of the system. The use of the warped multi-time scale model enables identification of the natural frequency of the ILFD which may then be compared with the input frequency to enable locking to be identified. The Multiple-Phase-Condition Envelope Following (MPCENV) technique is another technique that is proposed for the prediction of the locking range. This method is based on the envelope following simulation. The simulation is repeated to determine the output frequencies of an ILFD corresponding to the increasing input frequencies. The Devil's Staircase [9], a plot of ω_{inj}/ω_o against ω_{inj} , can be created from the obtained data. Then the locking range may be measured from the staircase.

As with the ILFD, the digital delta-sigma modulator (DDSM) is also employed in the feedback of an Fractional-*N* Frequency Synthesizers [12] [13]. However, it works as the controller of a static frequency divider in order to ensure that the division ratio is the required fractional value. In Chapter 6, the behaviour of the DDSMs shall be analysed mathematically. DDSMs have unwanted quantization noise in the desired frequency band. The goal is therefore to find a technique or a structure for DDSMs that reduces this noise. Multi-stAge noise Shaping (MASH) digital delta-sigma modulators (DDSM) shall be investigated with the goal being to obtain a structure that has the best noise performance.

It has been proved that a longer sequence length could reduce the noise effect in useful band [3]. Several methods are proposed to maximise the sequence length. Borkowski [14] obtains the maximum sequence length for the conventional DDSMs by setting the initial condition of registers. Hosseini [3] introduced a new structure to maximise the sequence length. We propose a novel architecture to further lengthen the sequence length [15] [16]. The multi-moduli technique is employed in this structure. Hence, it is termed the MM-MASH. Simulations shall confirm its superior noise performance to existing structures.

Chapter 2

Background to Computer Aided Design (CAD) Algorithms

2.1 Numerical Integration Methods

Electronic systems are modelled using ordinary differential equations (ODEs), differential algebraic equations (DAEs) or partial differential equations (PDEs). Thus the aim of circuit simulation is to find the numerical solution to these systems of equations. To start with, consider the following ODE:

$$y' = \frac{dy}{dt} = f(t, y(t)) \tag{2.1}$$

where f(t, y(t)) is a function of time t, and y is a variable such as capacitor voltage or inductor current. Newton, Leibniz and Euler found that there may be more than one solution that satisfies (2.1) [17]. Thus an initial value condition is required to make the solution unique [18]:

$$y(t_0) = y_0.$$
 (2.2)

If there are n variables, the system may be expressed as

$$y'_{1} = f_{1}(t, y_{1}, y_{2}, ..., y_{n}), \quad y_{1}(t_{0}) = y_{10}$$

$$y'_{2} = f_{2}(t, y_{1}, y_{2}, ..., y_{n}), \quad y_{2}(t_{0}) = y_{20}$$

$$\vdots \qquad \vdots$$

$$y'_{n} = f_{n}(t, y_{1}, y_{2}, ..., y_{n}), \quad y_{n}(t_{0}) = y_{n0}$$
(2.3)

(2.3) is usually written as

$$y'(t) = f(t, y(t)), \quad y(t_0) = y_0.$$
 (2.4)

where y is a vector-valued function and t is a scaler:

$$y = (y_1, y_2, ..., y_n)^{\mathrm{T}}$$
 (2.5)

$$f = (f_1, f_2, \dots, f_n)^{\mathrm{T}}.$$
 (2.6)

Hence, the PDEs for n variables are transformed to a system of ODEs.

In order to find the numerical approximation to (2.4), the differential equations are solved in a step by step manner and the techniques employed are termed step-by-step methods or difference methods. With this approach, a sequence of y(t) along the t coordinate are produced. The time sequence is defined as:

$$t_{1} = t_{0} + h$$

$$t_{2} = t_{0} + 2h$$

$$\vdots$$

$$t_{n} = t_{0} + nh$$

$$t_{n+1} = t_{0} + (n+1)h.$$
(2.7)

or alternatively, in the form:

$$t_{n+1} = t_n + h (2.8)$$

where *h* is the time-step size, $n \in \{0, 1, ..., N-1\}$ and *N* is the number of steps in the algorithm. A small time-step size results in a bigger *N*, which decreases the simulation speed. Consequently, there has been extensive research into the development of methods that achieve a balance between accuracy and efficiency for numerically integrating ODEs or PDEs. In this Chapter, for the purposes of this explanation, a fixed step size

shall be assumed. Then the approximate value of y(t) at different time states are

$$y_{0} = y(t_{0})$$

$$y_{1} = y(t_{1})$$

$$\vdots$$

$$y_{n} = y(t_{n})$$

$$y_{n+1} = y(t_{n+1}).$$
(2.9)

where y(t) represents all the variables in (2.1), y_n refers to the value of y(t) at time t_n , and y_0 is the initial value as introduced above.

There are two types of errors introduced in the numerical calculation: round-off error and truncation error. The round-off error results from the limit of the computer representation of a number. For example, an infinite decimal fraction, π , may be approximated as 3.14. On the other hand, the truncation error is caused by the algorithm itself. For instance, the most accurate Taylor series expansion has an infinite number of terms, while the computer calculation has a finite number of terms.

The numerical methods to solve ODEs and PDEs can be divided into two groups, explicit methods and implicit methods. An explicit method is of the form:

$$y_{n+1} = f(t_n, y_n, t_{n-1}, y_{n-1}, \dots, t_{n-k}, y_{n-k})$$
(2.10)

where $k \in \{0, 1, ..., n\}$. It uses the state of the system at the current time, t_n and y_n , and at previous time steps to define the system state at the later time y_{n+1} . The right hand side can be explicitly calculated. On the other hand, an implicit method involves y_{n+1} in both the left and right hand sides. It is defined as:

$$y_{n+1} = f(t_{n+1}, y_{n+1}, t_n, y_n, \dots, t_{n-k}, y_{n-k}).$$
(2.11)

The advantage of implicit methods is that they permit a larger time-step size owing to improved stability qualities. However, their evaluation is more difficult.

Explicit methods and implicit methods can be further subdivided into single-step and multi-step methods. Single-step methods require only one previous value of the function f(), i.e., y_n , to calculate y_{n+1} . On the contrary, multi-step methods use several previous values such as y_n , y_{n-1} and y_{n-2} .

The concept of stability is an important property of the numerical approximation. BIBO stability is what concerns us in this thesis. A bounded signal is any signal whose absolute value never exceeds a finite positive value, $B \in \Re$. If the output is bounded for all bounded inputs to the system, the system is termed BIBO stable system.

In this Chapter, the explicit and implicit numerical methods that are later used in this thesis for comparative purposes are briefly described. Detailed derivations and descriptions are given in numerous reference text-books e.g. [17] [18].

2.2 Euler Methods

2.2.1 Forward Euler Method

The Euler method is a first-order single-step explicit algorithm for numerically integrating the ordinary differential equations (ODEs).

Recall the ODE (2.1) introduced in Section 2.1:

$$y'(t) = f(t, y(t))$$
 (2.12)

where

$$y(t_0) = y_0$$

$$y(t_1) = y_1$$

$$\vdots$$

$$y(t_n) = y_n$$

(2.13)

The time sequence is defined as t_0 , $t_1 = t_0 + h$, $t_2 = t_0 + 2h$,..., $t_n = t_0 + nh$ and h is the time-step size. The differential y'(t) is approximated as in [19]:

$$y'(t) = \frac{y(t+h) - y(t)}{h}.$$
(2.14)

Substituting (2.14) into (2.12), it is obtained:

$$y(t+h) = y(t) + hf(t, y(t))$$
(2.15)

where t is assumed as the current time t_n . Thus (2.15) can be also expressed as

$$y_{n+1} = y_n + hf(t_n, y_n) \tag{2.16}$$

2.2.2 Backward Euler Method

The Backward Euler method is derived in a similar manner to the Forward Euler method. Instead of (2.14), the differential is expressed by

$$y'(t) = \frac{y(t) - y(t - h)}{h}.$$
(2.17)

Substitute (2.17) into (2.12), it yields

$$y(t) = y(t - h) + hf(t, y(t)).$$
(2.18)

It may also be expressed in the form:

$$y_n = y_{n-1} + hf(t, y_n). (2.19)$$

In order to compare it with the Forward Euler method, the Backward Euler method is usually expressed as:

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}). (2.20)$$

It is an implicit method and therefore, it is more stable than the Forward Euler method.

2.3 Trapezoidal Method

Sometimes the Trapezoidal method is also called the Trapezoidal Rule. The solution of the ODE (2.12) is obtained by approximating the area under the curve using a trapezoid.

$$y_{n+1} = y_n + \int_{t_n}^{t_{n+1}} f(t, y) dt.$$
 (2.21)

where

$$y_n = y(t_n) \tag{2.22}$$

$$y_{n+1} = y(t_{n+1}). (2.23)$$



Figure 2.1: *The integral from* t_n *to* t_{n+1} .

As seen in Fig. 2.1, the area of the trapezoid is obtained by:

$$\int_{t_n}^{t_{n+1}} f(t,y)dt = (t_{n+1} - t_n)\frac{f(t_{n+1}) + f(t_n)}{2}$$
$$= \frac{h}{2}(f(t_{n+1}) + f(t_n))$$
(2.24)

where $h = t_{n+1} - t_n$ is the time-step size. Hence, the approximation of y(t) is obtained as:

$$y_{n+1} = y_n + \frac{h}{2}(f(t_{n+1}) + f(t_n)).$$
(2.25)

Again, the Trapezoidal Rule is an implicit method.

2.4 Runge-Kutta Method

The Runge-Kutta method was developed around 1900 by the German mathematicians C. Runge and M.W. Kutta [20]. The basic idea of the Runge-Kutta method is that if the slopes at several points between t_n and t_{n+1} are found, the average value of them is assumed as the slope at t_n .

$$y_{n+1} = y_n + h \cdot (\text{average_slope}). \tag{2.26}$$

Mathematically, (2.26) is expressed as:

÷

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i k_i$$
(2.27)

where k_i represent the slopes at the points in the interval between y_n and y_{n+c_s} . They are defined as:

$$k_1 = f(t_n, y_n)$$
 (2.28)

$$k_2 = f(t_n + c_2 h, y_n + a_{21} h k_1)$$
(2.29)

$$k_3 = f(t_n + c_3h, y_n + a_{31}hk_1 + a_{32}hk_2)$$
(2.30)

$$k_s = f(t_n + c_s h, y_n + a_{s1} h k_1 + a_{s2} h k_2 + \dots + a_{s,s-1} h k_{s-1})$$
(2.31)

There are a lot of choices for the number of stages s and the coefficients a_{ij} , b_p and c_q , where $1 \le j < i \le s$, $p \in \{1, 2, ..., s\}$ and $q \in \{2, 3, ..., s\}$. To obtain specific values for the parameters, y_{n+1} is expanded in powers of h such that it agrees with the Taylor series expansion of the solution of the differential equation to a specified number of terms.

The Forward Euler method is the simplest one-stage Runge-Kutta with the corresponding coefficients:

$$s = 1 \tag{2.32}$$

$$b_1 = 1.$$
 (2.33)

Then the state y_{n+1} is obtained as:

$$y_{n+1} = y_n + hk_1. (2.34)$$

The classical fourth-order Runge-Kutta method (RK4) is the most popular one. It is:

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$
(2.35)

where

$$k_1 = f(t_n, y_n) \tag{2.36}$$

$$k_2 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right)$$
(2.37)

$$k_3 = f\left(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2\right)$$
(2.38)

$$k_4 = f(t_n + h, y_n + hk_3) \tag{2.39}$$

 k_1 is the slope at the beginning of the interval;

 k_2 is the slope at the midpoint of the interval. The slope k_1 is used to determine the value of y at the point $t_n + \frac{h}{2}$ using Euler's method;

 k_3 is again the slope at the midpoint, but now the slope k_2 is used to determine the *y*-value;

 k_4 is the slope at the end of the interval, with its y-value determined using k_3 . Then, the average slope is determined as:

$$slope = \frac{k_1 + 2k_2 + 2k_3 + k_4}{6} \tag{2.40}$$

Thus, y_{n+1} is determined from the previous state y_n , the time-step h and an estimated slope.

2.5 Adams' Methods

2.5.1 Adams-Bashforth Method

The Adams-Bashforth method [21] is an explicit multi-step method. The major advantage of the Adams-Bashforth method over the Runge-Kutta methods is that only one unknown function needs to be evaluated for each step. All the other functions such as $f(t_{k-1}, y_{k-1})$ and y_{k-1} have been obtained from the previous time-step.

The solution to (2.12) may be written as

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} f(t, y) dt.$$
 (2.41)

The function f is replaced by a Lagrange interpolation polynomial [22] at k points $t_n, t_{n-1}, ..., t_{n-k+1}$:

$$f_{poly}(t) = f_n + (t - t_n) f[t_n, t_{n-1}] + (t - t_n)(t - t_{n-1}) f[t_n, t_{n-1}, t_{n-2}] + \cdots + (t - t_n)(t - t_{n-1}) \cdots (t - t_{n-k+2}) f[t_n, t_{n-1}, \dots, t_{n-k+1}]$$
(2.42)

where

$$f[t_n, t_{n-1}, ..., t_{n-k+1}] = \frac{1}{t - t_{n-k+1}} (f[t_n, t_{n-1}, ..., t_{n-k+2}] - f[t_{n-1}, ..., t_{n-k+1}]).$$
(2.43)

As an example, the derivation of the 2nd-order Adams-Bashforth method, i.e., k = 2, is shown as below. First, the second-order Lagrangian polynomial is obtained

$$f_{poly}(t) = -\frac{1}{h}(t-t_i)f(t_{i-1}, y(t_{i-1})) + \frac{1}{h}(t-t_{i-1})f(t_i, y(t_i)).$$
(2.44)

Substituting (2.44) into (2.41) and integrating yields

$$y(t_{i+1}) = y(t_i) + \left[-\frac{1}{h} (t^2/2 - t_i t) f(t_{i-1}, y(t_{i-1})) + \frac{1}{h} (t^2/2 - t_{i-1} t) f(t_i, y(t_t)) \right]_{t_i}^{t_{i+1}}.$$
(2.45)

Then the final expression for the 2nd-order Adams-Bashforth method is obtained by inserting the limits.

$$y_{i+1} = y_i + \frac{h}{2} [3f(t_i, y_i) - 1f(t_{i-1}, y_{i-1})].$$
(2.46)

2.5.2 Adams-Moulton method

The Adams-Moulton method is an implicit multi-step method, which follows from the Adams-Bashforth method. The key difference between them is that with the Adams-Moulton method, the function f(t, y) is replaced by the polynomial at the (k+1) points $t_{n+1}, t_n, ..., t_{n-k+1}$. Then the polynomial function is

$$f_{poly}(t) = f_{n+1} + (t - t_{n+1})f[t_{n+1}, t_n]$$

+ $(t - t_{n+1})(t - t_n)f[t_{n+1}, t_n, t_{n-1}] + \cdots$
+ $(t - t_{n+1})(t - t_n)\cdots(t - t_{n-k+2})f[t_{n+1}, t_n, \dots, t_{n-k+1}]$ (2.47)

where

$$f[t_{n+1}, t_n, ..., t_{n-k+1}] = \frac{1}{t_{n+1} - t_{n-k+1}} (f[t_{n+1}, t_n, ..., t_{n-k+2}] - f[t_n, ..., t_{n-k+1}]).$$
(2.48)

The value of $f(t_{n+1}, y_{n+1})$ may be predicted using the Adams-Bashforth method before the evaluation of (2.47). This combined procedure is termed a predictor-corrector method.

2.6 Stiff Systems and Stability Definitions

A system of differential equations is said to be stiff when it is the interval of absolute stability rather than a requirement to keep the truncation error small that constrains the step-size. It generally arises when components of the solution of the differential equation system undergo changes of a similar magnitude over very different time intervals. Consequently, any ODE solver in an RF simulation package must be able to handle stiff systems in order to avoid the numerically instability and inefficiency.

In order to analyse the stability qualities of a numerical method, the methods are applied to the test equation:

$$y' = \lambda y \tag{2.49}$$

where λ is a constant. The solution is

$$y_{n+1} = \phi(\lambda)y_n \tag{2.50}$$

where ϕ is a function of λ . The numerical method is said to be A-stable, if the solution is stable for all $\lambda < 0$, i.e., the solution satisfies (??) for $\forall \lambda < 0$ [23]. A numerical method is L-stable if it satisfies the two conditions given below [24]:

- 1. It is A-stable.
- 2. It satisfies $\lim_{\lambda \to -\infty} \phi = 0$.

2.7 Conclusions

Since a small time-step size results in expensive and time-consuming computations, many CAD techniques have been developed to achieve a balance between stability and efficiency. The majority of circuit simulators such as SPICE [25] use implicit numerical integration techniques with adaptive time stepping. These methods have the advantages of good stability and accuracy control properties, but a solution to a set of non-linear algebraic equations is required at each time-step and this can be computationally expensive. An alternate approach is to employ predictor-corrector methods. The traditional predictor-corrector methods such as the Adams Moulton method employ polynomial extrapolation to provide an estimate of the solution at the next time step. The estimate is subsequently corrected using one or several iterations of an implicit formula. However, for stiff problems, the maximum allowable time-step that can be used with the traditional predictor-corrector techniques may be unacceptably small. Thus, novel numerical methods utilising the Padé approximation will be introduced in Chapter 3. With these methods, a larger time-step size h is permitted without deceasing the accuracy.

Chapter 3

Padé-Based Algorithms for Numerical Integration of Ordinary Differential Equations (ODE)

As mentioned in the previous chapter, there is an on-going need for improved numerical integration algorithms for solving stiff differential systems. To this end, several novel ODE solvers are proposed in this Chapter. In particular, several Padé-based algorithms [5] [26] [27] are proposed.

Recall the ODE system:

$$\frac{dy}{dt} = f(t, y(t)) \tag{3.1}$$

where

$$y_0 = y(t_0).$$
 (3.2)

In order to solve (3.1), a sequence of Padé approximations to $y_i(t)$ are provided:

$$y_i(t) = \frac{\sum_{j=0}^m a_j h^j}{\sum_{k=0}^n b_k h^k}, \quad m < n$$
(3.3)

where $b_0 = 1$ and $y_i(t)$ is the *i*th element of the y(t) vector. The order of this system is defined as m + n.

The Padé approximation (3.3) can be represented in the form of an m + nth order polynomial:

$$y_i(t) = c_0 + c_1 h + c_2 h^2 + \dots + c_{m+n} h^{m+n}.$$
 (3.4)

Equating (3.3) and (3.4),

$$\sum_{i=0}^{m+n} c_i h^i = \frac{\sum_{j=0}^m a_j h^j}{\sum_{k=0}^n b_k h^k}$$
(3.5)

where $i \in \{0, 1, ..., m + n\}$, $j \in \{0, 1, ..., m\}$ and $k \in \{0, 1, ..., n\}$. The coefficients a_j and b_k are obtained by equating the coefficients of h in (3.5). Once the coefficients a_j and b_k are obtained, the value for y_i at the next time step is obtained from (3.3).

The Padé-based algorithm is advanced in time by using the solution at time t as the initial condition for the next time-step. The use of Padé functions of the form of (3.3) with m < n results in an A-stable and L-stable method as shown in the Appendix A. The central element of the Padé algorithms is the determination of c_i . This shall be discussed in what follows.

3.1 Single-Step Padé-Based Algorithms

3.1.1 Padé-Taylor Method

To determine the coefficients c_i , consider a Taylor series for y(t). If (3.1) has an unique solution y(t), the *p*th order Taylor series of y(t) about t_n is

$$y(t) = y(t_n) + (t - t_n)y'(t_n) + \frac{1}{2!}(t - t_n)^2 y''(t_n) + \dots + \frac{1}{p!}(t - t_n)^p y^{(p)}(t_n) \quad (3.6)$$

where t is in the required interval [0, T]. By assuming that $t = t_{n+1}$ and $h = t_{n+1} - t_n$, (3.6) is then in the form of

$$y(t) = y(t_n) + hy'(t_n) + \frac{h^2}{2!}y''(t_n) + \dots + \frac{h^p}{p!}y^{(p)}(t_n)$$
(3.7)

Substituting (3.1) into (3.7) yields

$$y_{n+1} = y_n + hf(t_n, y_n) + \frac{h^2}{2!}f'(t_n, y_n) + \dots + \frac{h^p}{p!}f^{(p-1)}(t_n, y_n)$$
(3.8)

By comparing (3.8) and (3.4), the coefficients c_i are obtained as

$$c_{0} = y_{n}$$

$$c_{1} = f(t_{n}, y_{n})$$

$$c_{2} = \frac{f'(t_{n}, y_{n})}{2!}$$

$$\vdots$$

$$c_{m+n} = \frac{f^{(p-1)}(t_{n}, y_{n})}{p!}$$
(3.9)

Substituting (3.9) into (3.5), the coefficients a_j and b_k are found.

3.1.2 Padé-Exponential Method

In this method, the solution of (3.1) is formed as the composition of a Padé approximation and an exponential function as below:

$$y_i(t_{k+1}) = \frac{\sum_{i=0}^m a_i h^i}{\sum_{j=0}^n b_j h^j} + d_1 e^{d_2 h}$$
(3.10)

where $b_0 = 1$, $h = t_{k+1} - t_k$, m < n and $d_2 < 0$. The rationale for this choice is that the exponential term extracts the fast-varying part of the solution and that a lowerorder Padé expression suffices for the slow-varying part of the solution. The method is A-stable and gives an exact solution for the test function $e^{-\lambda t}$, $\lambda > 0$, if m < n and $d_2 < 0$. Let $y_i(t_{k+1})$ be approximated as the sum of a polynomial and an exponential function

$$y_i(t_{k+1}) = P_k(t_{k+1}) + d_1 e^{d_2 h}.$$
(3.11)
where

$$P_k(t_{k+1}) = c_0 + c_1 h + c_2 h^2 + \dots + c_L h^k.$$
(3.12)

Substituting (3.12) into (3.11) yields

$$y_i(t_{k+1}) = PE(t_{k+1})$$

= $c_0 + c_1h + c_2h^2 + \dots + c_Lh^L + d_1e^{d_2h}.$ (3.13)

To determine the coefficients, a system of L+3 algebraic equations is formed from y(t) and its derivatives $f^{(r)}(t)$, which is shown as below

$$y_i(t) = PE(t)$$

$$f_i(t) = PE'(t)$$

$$f'_i(t) = PE''(t)$$

$$\vdots$$

$$f_i^{(L+1)}(t) = PE^{(L+2)}(t).$$
(3.14)

From (3.14), the L + 3 unknowns in (3.13) are obtained, i.e., $c_0, c_1, ..., c_L$ and d_1, d_2 . From these, the coefficients a_i and b_j in (3.10) can be found from (3.5).

As an example, the first-order Padé-Exponential method is described here, i.e., m = 0, n = 1 and L = m + n = 1. Then (3.10) and (3.13) become

$$y_i(t_{k+1}) = \frac{a_0}{1+b_1h} + d_1 e^{d_2h}$$
(3.15)

$$y_i(t_{k+1}) = c_0 + c_1 h + d_1 e^{d_2 h}.$$
(3.16)

Thus by solving the equation

$$\frac{a_0}{1+b_1h} = c_0 + c_1h \tag{3.17}$$

it is obtained

$$a_0 = c_0$$
 (3.18)

$$b_1 = -\frac{c_1}{c_0}.$$
 (3.19)

Now (3.14) is in the form

$$y_{i}(t_{k+1}) = c_{0} + c_{1}h + d_{1}e^{d_{2}h}$$

$$f_{i}(t_{k+1}) = c_{1} + d_{1}d_{2}e^{d_{2}h}$$

$$f_{i}'(t_{k+1}) = d_{1}d_{2}^{2}e^{d_{2}h}$$

$$f_{i}''(t_{k+1}) = d_{1}d_{2}^{3}e^{d_{2}h}.$$
(3.20)

Hence, the solution of (3.18), (3.19) and (3.20) at h = 0 is

$$d_2 = \frac{f_i''(t_{k+1})}{f_i'(t)} \tag{3.21}$$

$$d_1 = \frac{f_i'(t_{k+1})}{d_2^2} \tag{3.22}$$

$$a_0 = y_i(t_{k+1}) - d_1$$

$$(3.23)$$

$$f_i(t_{k+1}) - d_1 d_2$$

$$(3.24)$$

$$b_1 = -\frac{f_i(t_{k+1}) - d_1 d_2}{y_i(t_{k+1}) - d_1}$$
(3.24)

Finally, the the value for y_{i+1} is obtained by substituting (3.21)–(3.24) into (3.15).

3.1.3 Padé-Richardson Method

In the Padé-Richardson method, one of the methods introduced above, i.e., the Padé-Taylor method or the Padé-Exponential method, is combined with Richardson extrapolation [28]. With this method, the accuracy is improved without having to evaluate higher-order derivatives of f in equation (3.1).

Let the initial approximation of the solution of (3.1) be given by $y(t_{k+1})$.

$$y_{ex}(t_{k+1}) = y(t_{k+1}) + O(h).$$
(3.25)

If the error O(h) is approximated by a polynomial of order L

$$O(h) = ch^{L} + O(h^{L+1})$$
(3.26)

(3.25) becomes

$$y_{ex}(t_{k+1}) = y(t_{k+1}) + ch^{L} + O(h^{L+1}).$$
(3.27)

By evaluating (3.27) with two different time-steps, h and h/2, the approximation of $y(t_{k+1})$ is expressed by two equations:

$$y_{ex}(t_{k+1}) = y_h(t_{k+1}) + ch^L + O(h^{L+1})$$
(3.28)

$$y_{ex}(t_{k+1}) = y_{h/2}(t_{k+1}) + c\left(\frac{h}{2}\right)^L + O(h^{L+1}).$$
(3.29)

Multiplying (3.29) by 2^L gives

$$2^{L}y_{ex}(t_{k+1}) = 2^{L}y_{h/2}(t_{k+1}) + ch^{L} + O(h^{L+1}).$$
(3.30)

Subtracting (3.28) from (3.30), it is obtained

$$(2L - 1)y_{ex}(t_{k+1}) = 2Ly_{h/2}(t_{k+1}) - y_h(t_{k+1}).$$
(3.31)

Hence,

$$y_{ex}(t_{k+1}) = \frac{2^L y_{h/2}(t_{k+1}) - y_h(t_{k+1})}{2^L - 1}.$$
(3.32)

Suppose that the two approximations, $y_h(t_{k+1})$ and $y_{h/2}(t_{k+1})$ are obtained with a Padé method of order L, the Padé approximation for the solution at t_{k+1} can be improved by forming:

$$\bar{y}_P(t_{k+1}) = \frac{2^L y_{h/2}(t_{k+1}) - y_h(t_{k+1})}{2^L - 1}.$$
(3.33)

The improved estimate in equation (3.33) combines two *L*th-order expansions and is thus computationally more expensive than a single *L*th-order expansion. For example, consider a first-order Padé method. If Richardson extrapolation is employed with two first-order methods, then the total computational effort is equivalent to a second-order expansion around a single point. (This is shown in the Appendix B for the specific case of the exponential function). However, the results in the next section show that the Richardson's Extrapolation in conjunction with Padé-based methods gives better results than a basic second-order expansion for the same time step. The conclusion is that a much larger time step may be employed to achieve a specific accuracy tolerance. Thus, savings are obtained from a computational viewpoint by utilising the Richardson extrapolation as although two function evaluations are required at each time step, there are fewer time steps involved.

3.1.4 Padé Approximation with Predictor-Corrector

With the predictor-corrector method, the accuracy of the Padé approximation is further increased and it is not necessary to calculate the higher-order derivatives. The basis to form a corrector stage for the single-step method is as follows: a Padé approximation of order p matches the first p + 1 coefficients (time-domain moments) of a Taylor series expansion. It also provides additional terms. Considering the test function e^{-t} . Its Taylor series about t = 0 is:

$$e^{-t} = 1 - h + \frac{h^2}{2} - \frac{h^3}{6} + \cdots$$
 (3.34)

A first-order Padé approximation for e^{-t} about t = 0 is given by:

$$y_{app} = \frac{1}{1+h}.$$
 (3.35)

This function matches the first two coefficients of a Taylor series expansion for e^{-t} . It also produces additional terms, the first of which is:

$$T_{3_P} = h^2. (3.36)$$

However, the correct third coefficient in a Taylor series expansion for e^{-t} as seen in equation (3.34) is:

$$T_{3_T} = \frac{h^2}{2}.$$
(3.37)

Now, a corrector is chosen so as to match T_3 for the particular test function, $y = e^{-t}$, without requiring a higher-order derivative. So for this test case, the corrector would be:

$$T_{corr} = -\frac{h^2}{2} = f(0)\frac{h^2}{2}.$$
(3.38)

where f(0) = -1. Bearing this in mind, the choice of corrector for a general singlestep first-order Padé method is selected as follows:

$$\hat{y}_i(t_{k+1}) = \bar{y}_i(t_{k+1}) + \frac{1}{2}h^2\bar{f}(t_{k+1}).$$
(3.39)

 $\hat{y}_i(t_{k+1})$ is the corrected estimate of $y_i(t_{k+1})$. $\bar{f}(t_{k+1})$ is the estimate of the first derivative of $y_i(t_{k+1})$. For higher-order methods, a similar procedure is followed:

$$\hat{y}_i(t_{k+1}) = \bar{y}_i(t_{k+1}) + \frac{1}{\alpha} h^{p+1} \bar{f}^{(p-1)}(t_{k+1}).$$
(3.40)

where $1/\alpha$ is the amount required to correct the (p + 2)th coefficient in a *p*th order Padé approximation for e^{-t} about t = 0.

3.2 Multi-Step Padé-Based Algorithms

3.2.1 Explicit Multi-Step Padé Approximation

To derive a multi-step formula, recall again:

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} f(y(t), t) dt.$$
(3.41)

In this case, a Padé approximation is used to interpolate the function, f, through m + n + 1 interpolation points starting from t_i and finishing at t_k . m + n is the order of the Padé approximation:

$$f(y,t) = \frac{\sum_{j=0}^{m} p_j (t-t_i)^j}{1 + \sum_{j=1}^{n} q_j (t-t_i)^j}.$$
(3.42)

The function f(y,t) can also be approximated by an (m+n)th order polynomial as below

$$f(y,t) = c_0 + c_1(t-t_i) + c_2(t-t_i)^2 + \dots + c_{m+n}(t-t_i)^{m+n}.$$
(3.43)

Then the values of f(y, t) at the different times are

$$f_{t_i} = \sum_{j=0}^{m+n} c_j (t_i - t_i)^j$$

$$f_{t_{i+1}} = \sum_{j=0}^{m+n} c_j (t_{i+1} - t_i)^j$$

$$\vdots$$

$$f_{t_k} = \sum_{j=0}^{m+n} c_j (t_k - t_i)^j$$

(3.44)

The coefficients c_j are obtained by solving (3.44). Then the coefficients p_j and q_j are determined by equating the Padé approximation in equation (3.42) with the series expansion in equation (3.43):

$$\frac{\sum_{j=0}^{m} p_j (t-t_i)^j}{1+\sum_{j=1}^{m} q_j (t-t_i)^j} = \sum_{j=0}^{m+n} c_j (t-t_i)^j$$
(3.45)

where $t = \{t_i, t_{i+1}, ..., t_k\}.$

Replacing f in equation (3.41) by the expression in equation (3.42) and performing the integration analytically yields a multi-step formula for $y(t_{k+1})$. This is an explicit formula for $y(t_{k+1})$. For example, if a first-order Pade approximation is employed with a time step of $h = t_{k+1} - t_k$, the resulting explicit formula is:

$$y(t_{k+1}) = y(t_k) + \int_{t_k}^{t_{k+1}} \frac{p_0}{1 + q_1(t - t_{k-1})} dt.$$
 (3.46)

In this case, m = 0 and n = 1.

$$y(t_{k+1}) = y(t_k) + \int_{h}^{2h} \frac{p_0}{1 + q_1 t} dt$$
(3.47)

which results in

$$y(t_{k+1}) = y(t_k) + \frac{p_0}{q_1} (\log(1 + 2hq_1) - \log(1 + hq_1)).$$
(3.48)

The first-order polynomial approximation is:

$$f(y,t) = c_0 + c_1(t - t_0)$$

= c_0 + c_1 t. (3.49)

Thus f at different times are

$$f(t_0) = c_0 + c_1 t_0 = c_0 \tag{3.50}$$

$$f(t_1) = c_0 + c_1 t_1 = c_0 + c_1 h.$$
(3.51)

Then the coefficients c_j are obtained:

$$c_0 = f(t_0)$$
 (3.52)

$$c_1 = \frac{f(t_1) - f(t_0)}{h}.$$
(3.53)

Substituting (3.52) and (3.53) into (3.45), when the order is 1, yields

$$\frac{p_0}{1+q_1t} = f(t_0) + \frac{f(t_1) - f(t_0)}{h}t.$$
(3.54)

The values of p_0 and q_1 are obtained by solving (3.54):

$$p_0 = f(t_0) \tag{3.55}$$

$$q_1 = -\frac{f(t_1) - f(t_0)}{f(t_0)h}.$$
(3.56)

By substituting (3.55) and (3.56) into (3.48), the Padé approximation to (3.1) is expressed as:

$$y(t_{k+1}) = y(t_k) - \frac{f^2(t_0)h}{f(t_1) - f(t_0)} \left(\log\left(1 + 2\frac{f(t_0) - f(t_1)}{f(t_0)}\right) - \log\left(1 + \frac{f(t_0) - f(t_1)}{f(t_0)}\right) \right).$$
(3.57)

3.2.2 Implicit Multi-Step Padé Approximation

In a similar manner to the derivation of the Adam's Moulton predictor-corrector method, a corrector for the multi-step method in equations (3.41)-(3.57) may be formed by interpolating f through points including t_{k+1} . When the resultant formula for f(t, y) is inserted in equation (3.41), the outcome is an implicit formula for $y(t_{k+1})$. The key difference from the explicit multi-step Padé approximation is that the implicit multi-step Padé approximation uses the following equations to find c_i :

$$f_{t_{i+1}} = \sum_{j=0}^{m+n} c_j (t_{i+1} - t_i)^j$$

$$\vdots$$

$$f_{t_k} = \sum_{j=0}^{m+n} c_j (t_k - t_i)^j$$

$$f_{t_{k+1}} = \sum_{j=0}^{m+n} c_j (t_{k+1} - t_i)^j$$
(3.58)

In order to solve (3.58), $f_{t_{k+1}}$ has to be predicted first. If the function f is nonlinear, then a nonlinear solver is required to solve for $y(t_{k+1})$ and this step can be computationally expensive. Hence, rather than solving the nonlinear equation set directly, the value for $y(t_{k+1})$ in the right-hand side of the implicit formula is replaced with its estimate from use of the explicit predictor formula–the explicit multi-step Padé approximation. This is the standard procedure in predictor-corrector methods such as the Adams-Moulton method.

3.3 Application and Validation

3.3.1 Classic Test System

The following well-known classic equation system [29] is chosen as the first illustrative example. Its analytical solution is known, equations (3.63) and (3.64) are an exact reference solution. In addition, since it constitutes a stiff system of differential equations, the accuracy and efficiency of new methods can be thoroughly investigated. The system equations are:

$$\frac{du}{dt} = 998u + 1998v \tag{3.59}$$

$$\frac{dv}{dt} = -999u - 1999v \tag{3.60}$$

where

$$u_0 = 1$$
 (3.61)

$$v_0 = 1.$$
 (3.62)

The exact analytical solution is given by:

$$u(t) = 4e^{-t} - 3e^{-1000t} \tag{3.63}$$

$$v(t) = -2e^{-t} + 3e^{-1000t}.$$
(3.64)

Figure 3.1 shows, superimposed on the analytical solution, the result computed with the standard third-order Adams-Moulton predictor-corrector method for a stepsize of 0.001s. Figure 3.2 illustrates the simulation results from the third-order Runge-Kutta method. Figure 3.3 shows the corresponding result computed with the basic third-order Padé single-step method of equation (3.3). As evidenced by these results, the new technique is in fact more accurate than the standard Adams-Moulton and Runge-Kutta method of the same order. To compare methods, the root mean squared (RMS) error is evaluated. The third-order Adams-Moulton technique yields an error of 1.53×10^{-1} for the given step size. The third-order Runge-Kutta method has an error of 6.65×10^{-2} for the same step size. The basic third-order Padé single-step method yields an error of 4.94×10^{-2} for the same step size.

When the combined third-order Padé and exponential function method is used, the error is dramatically reduced to 3.93×10^{-12} . When a first-order Padé and exponential

function is employed the error is 1.45×10^{-4} . Thus, the addition of the exponential term enables a specific error tolerance to be met with a lower-order Padé function.



Figure 3.1: The comparison of the simulation results from Adams Moulton (—) and exact analysis (**).

Finally, Richardson extrapolation is applied to the basic third-order Padé method. The error now is 1.4×10^{-3} . This goal of the Richardson extrapolation method is to eliminate the higher-order errors. When the method is applied to the third-order Padé



Figure 3.2: *The comparison of the simulation results from Runge-Kutta* (—) *and exact analysis* (**).



Figure 3.3: The comparison of the simulation results from Padé approximation (—) and exact analysis (**).

method, the aim is to eliminate fourth-order errors. However, the basic fourth-order Taylor series approximation results in an error of 2.01×10^{-2} . Thus, the Richardson extrapolation achieves greater accuracy and consequently, would enable the use of a significantly larger time-step for the same level of accuracy as a basic fourth-order method. Although use of Richardson extrapolation requires an extra function evaluation at each time-step, the time-step chosen can be considerably greater than that for the basic method to achieve comparable accuracy. Thus, the total number of function evaluations for a simulation is reduced. For example, if the time-step size is doubled when the Richardson extrapolation is applied to the basic third-order Padé method, the error is 1.3×10^{-2} .

All of the Padé-based methods result in an RMS error that is less than the Adams-Moulton method. This means that for a fixed accuracy tolerance, a larger step size can be employed if the new methods are used as the ODE solver.

Figure 3.4 shows the result computed with the Padé-based multi-step predictorcorrector method of Section 3.2. Again, a step size of h = 0.001 s is employed. The RMS error when first-order Padé approximants are used is 2.1×10^{-2} . Thus, Padébased predictor-corrector methods are effective and efficient when compared to the standard Adams Moulton method.



Figure 3.4: The comparison of the simulation results from Padé-based multi-step predictor-corrector method (—) and exact analysis (**).

The total simulation time of different methods, which simulate the example system (3.59) and (3.60) for 0.2 s with the same time-step size h = 0.001s, is illustrated in Table 3.1. The Padé-Exponential method costs the least time, while the 3rd-Order Adam-Moulton method is the slowest one. Note that the third-order Runge-Kutta method is not accurate enough, though it is faster than the Padé-Taylor method.

Table 5.1. Total simulation time of affected methods	
Numerical Integration Methods	Simulation Time (s)
Padé-Taylor	0.017
Padé-Exponential	0.0016
Padé-Exponential-Predictor-Corrector	0.0037
3rd-Order Adam-Moulton	0.041
3rd-Order Runge-Kutta	0.009

 Table 3.1: Total simulation time of different methods

3.3.2 MESFET Amplifier

As a second example to further confirm the utility of the Padé-based methods, Figure 3.5 shows a simple single-ended MESFET amplifier. The amplifier is described by ten non-linear differential equations which are stiff in nature. The complete details of the amplifier circuit and the model employed in it are given in the Appendix C. Figure 3.6 shows a short segment of the output voltage v(t) obtained with the fourth-order Adams-Moulton predictor-corrector. The input is a 2 GHz cosine wave.

A time-step of ~ 0.12 ps was required to solve the given system of equations with this technique. Figure 3.6 also shows the corresponding result obtained with the new Padé technique of Section 3.1.1 and the single-step corrector of Section 3.1.4. The new technique permits the use of a step-size ten times larger than the Adams Moulton technique, i.e., 1.2ps, for a comparable level of accuracy. This constitutes a remarkable saving in computing time.



Figure 3.5: Schematics of the MESFET amplifier.



Figure 3.6: *The comparison of the simulation results from Adams-Moulton* (—) *and Padé method* (**).

3.4 Conclusions

Several Padé-based algorithms are proposed to improve the accuracy and efficiency when solving the stiff systems. The Padé-Taylor method is the basic one which obtains the parameters by comparison with the Taylor series. The Padé-exponential method is used to increase the efficiency if the system has exponential components. The Richardson method is applied to Padé method to avoid the computation of higher-order derivatives. The predictor-corrector method is also used to further reduce the error. The simulation results for the examples in question show that Padé method is better than the Adams-Moulton technique in terms of both accuracy and efficiency.

Chapter 4

Filon-Type Numerical Quadrature for Highly-Oscillatory Systems.

4.1 Background to Highly Oscillatory Quadrature

High-frequency signals abound in Radio Frequency (RF) communication systems. This is a consequence of modulation: the imposition of a lower-frequency information signal onto a high-frequency carrier. The goal is to enable antennae of a manageable size to be employed for information transmission. Antennae of an impractical size would be required if modulation was not performed. In RF communication systems, signals in the MHz frequency range and higher are common. Furthermore, nonlinearities abound in RF transmission systems owing to the presence of solid-state amplifiers, mixers and so on [30].

Most RF systems involve a linear part and a nonlinear part with the linear part due to the presence of linear resistors, inductors and capacitors and the nonlinear part due to amplifiers, mixers or nonlinear and controlled resistors and capacitors. A typical equation system is of the form:

$$y' = Ay + g(t, y) \tag{4.1}$$

$$y_0 = y(0)$$
 (4.2)

where $t \ge 0$. The general solution of (4.1) may be written in the form:

$$y(t) = e^{tA}y_0 + \int_0^t e^{A(t-\tau)}g(\tau, y(\tau))d\tau$$
(4.3)

The recent explosion of developments in the RF and telecommunications industry has put pressure on circuit designers for faster simulations, faster designs and faster product output and the existing Computer Aided Design (CAD) tools have struggled to keep pace. In addition, the growing complexity of the modulation formats is rendering software tools unacceptably slow and consequently, unsatisfactory. There is therefore, an urgent need for a complete revamp and update of the fundamental numerical processes within these CAD packages taking into account the modern developments and formats.

Most numerical simulators of electric and electronic circuits, such as SPICE [25], as well as general-purpose solvers of ordinary differential equations (ODEs), like those in MATLAB, use either multi-step or Runge-Kutta methods. This is perfectly adequate for a great majority of ODEs in applications, yet falls woefully short for systems subject to modulated signals or RF oscillators. In this setting, traditional quadrature approaches can necessitate the use of minute step-sizes with the consequent outcome of great inefficiencies and often impractical simulation times.

Modulation is the process whereby information is transmitted at a high frequency to enable antennae of practical dimensions to be employed. In amplitude modulation (AM), the information signal (envelope) has a low frequency content relative to the carrier, $A_m \sin \omega_m t$. The resultant amplitude-modulated signal is therefore $A_c(1 + m \sin \omega_m t) \sin \omega_c t$, where $m = A_m/A_c$. Another variation of amplitude modulation is the Double-Sided Suppressed Carrier $A_m \sin \omega_m t \sin \omega_c t$. $\omega_c = 2\pi f_c$ is the carrier frequency in rad/s and A_c is its amplitude. In general AM, the information signal is in the kHz range, while the carrier signal is in the MHz or GHz range. In digital modulation, the information to be transmitted is a sequence of ones and zeros, termed bits. The amplitude, frequency or phase of a carrier signal is varied dependent on the bit value. For example, in Binary Phase Shift Keying (BPSK) the modulated signal is of the form $b(t) = cos(\omega_c t)x(t)$, where x(t) is +A or -A if a '1' or a '0' bit is to be transmitted, respectively. In digital technologies involving more complex formats, such as EDGE, the information/envelopes have bandwidths in the kHz range, while the carriers are 800MHz and 1800MHz. For evaluation of a biterror ratio of such an RF transmission, several information envelope time periods are required, but the step size is governed by the underlying carrier frequency which is significantly higher than the envelope frequency.

This chapter will address the issue of simulating ODEs involving very high frequencies and widely varying frequency ranges using Filon-type methods, applied in tandem with exponential integrators. This results in increased efficiencies for systems involving signal of widely varying frequencies. Similar techniques have been already investigated for different models of highly oscillatory ODEs in [31].

4.2 Filon-Type Numerical Approximation

The following equation

$$F[f] = \int_{a}^{b} f(t)G(t,\omega)dt$$
(4.4)

is assumed as a highly oscillatory integral: in particular, the oscillator G oscillates rapidly when the oscillatory parameter ω is large, i.e., $\omega \gg 1$, while f itself is nonoscillatory. A typical example of an oscillator is $G(x, \omega) = e^{i\omega g(x)}$ where g is some given smooth function. The calculation of (4.4) for large ω by classical methods (e.g. Gaussian quadrature) is prohibitively expensive, but such integrals can be calculated with relative ease using Filon-type integrators [32]. The idea behind Filon-type methods is to replace the function f in (4.4) by a polynomial interpolation. Specifically, let

$$a = c_1 < c_2 < \dots < c_q = b$$
 (4.5)

be given nodes, each c_k has a multiplicity $m_k \ge 1$. Thus, a polynomial p of degree $r = \sum_{k=1}^{q} m_k - 1$ is constructed as:

$$p^{(j)}(c_k) = f^{(j)}(c_k)$$
(4.6)

where $j = 0, 1, ..., m_k - 1$ and k = 1, 2, ..., q. The Filon-type method for the integral (4.4) is

$$Q_F[f] = \int_a^b p(t)G(t,\omega)dt$$
(4.7)

It is based on the assumption that the integral (4.4) can be calculated explicitly for function f.

The interpolating polynomial p can be written explicitly in the form:

$$p(t) = \sum_{k=1}^{q} \sum_{j=0}^{m_k - 1} \alpha_{k,j}(t) f^{(j)}(c_k)$$
(4.8)

where each $\alpha_{k,j}$ is the *cardinal polynomial of Hermite interpolation*, $\alpha_{k,j}^{(j)}(c_k) = 1$, otherwise $\alpha_{k,j}^{(i)}(c_l) = 0$ for all $i = 0, 1, ..., m_l - 1$, and l = 1, 2, ..., q. Therefore (4.7) can be written in the form

$$Q_F[f] = \sum_{k=1}^{q} \sum_{j=0}^{m_k - 1} b_{k,j}(\omega) f^{(j)}(c_k)$$
(4.9)

where

$$b_{k,j}(\omega) = \int_{a}^{b} \alpha_{k,j}(t) G(t,\omega) dt.$$
(4.10)

Then the suitable nodes and multiplicities are chosen by the Filon-type methods so as to increase the accuracy and efficiency.

4.3 Example

Here, the diode rectifier circuit shown in Fig. 4.1 is chosen as an example to verify the accuracy and efficiency of the Filon-type numerical quadrature, because its model is in the form of (4.1).



Figure 4.1: Diode rectifier circuit.

The governing equation for the circuit in Fig. 4.1 is

$$C\frac{dv(t)}{dt} + \frac{v(t)}{R} = I_0 \left[e^{k(b(t) - v(t))} - 1 \right]$$
(4.11)

where C is the capacitor value, R is the resistor value, I_0 and k are constants, b(t) is the input signal and the unknown v(t) is the voltage.

In order to modify (4.11) to be in the form of the variation constants formula (4.3), each side of (4.11) is multiplied by e^{-At} , where A = -1/RC.

$$e^{t/RC}\frac{dv(t)}{dt} + e^{t/RC}\frac{v(t)}{RC} = e^{t/RC}\frac{I_0}{C} \left[e^{k(b(t)-v(t))} - 1\right].$$
(4.12)

Since the left side

$$e^{t/RC}\frac{dv(t)}{dt} + e^{t/RC}\frac{v(t)}{RC} = \frac{d}{dt}(e^{t/RC}v(t))$$
(4.13)

(4.12) becomes

$$\frac{d}{dt}(e^{t/RC}v(t)) = e^{t/RC}\frac{I_0}{C} \left[e^{k(b(t)-v(t))} - 1\right].$$
(4.14)

If (4.14) is integrated from t_n to t_{n+1} , it is obtained

$$e^{t/RC}v(t)\big|_{t_n}^{t_{n+1}} = -RI_0e^{t/RC}\big|_{t_n}^{t_{n+1}} + \frac{I_0}{C}\int_{t_n}^{t_{n+1}} \exp\left(\frac{t}{RC} - kv(t) + kb(t)\right)dt.$$
 (4.15)

Thus

$$e^{t_{n+1}/RC}v_{n+1} - e^{t_n/RC}v_n = RI_0 \left[e^{t_n/(RC)} - e^{t_{n+1}/(RC)} \right] + \frac{I_0}{C} \int_{t_n}^{t_{n+1}} \exp\left(\frac{t}{RC} - kv(t) + kb(t)\right) dt.$$
(4.16)

By rearranging (4.16) with the time-step $h = t_{n+1} - t_n$, we get

$$v_{n+1} = v_n e^{-h/(RC)} + RI_0 \left[e^{-h/(RC)} - 1 \right] + \frac{I_0}{C} \int_{t_n}^{t_{n+1}} \exp\left(\frac{t - t_{n+1}}{RC} - kv(t) + kb(t)\right) dt$$
$$= v_n e^{-h/(RC)} + RI_0 \left[e^{-h/(RC)} - 1 \right] + \frac{I_0}{C} \Gamma[g]$$
(4.17)

where

$$g(t) = \exp\left(\frac{t - t_{n+1}}{RC} - kv(t) + kb(t)\right)$$
(4.18)

$$\Gamma[g] = \int_{t_n}^{t_{n+1}} g(t)dt.$$
(4.19)

If $b(t) = \varepsilon \cos \vartheta(t)$, then the integral in (4.17) is of the form

$$\Gamma[g] = \int_{t_n}^{t_{n+1}} f(t, v(t)) e^{z \cos \vartheta(t)} dt$$
(4.20)

where

$$z = k\varepsilon \tag{4.21}$$

$$f(t, v(t)) = \exp\left[\frac{t - t_{n+1}}{RC} - kv(t)\right].$$
 (4.22)

From [33],

$$e^{z\sin\vartheta} = \mathbf{I}_0(z) + 2\sum_{m=0}^{\infty} (-1)^m \mathbf{I}_{2m+1}(z)\sin(2m+1)\vartheta + 2\sum_{m=1}^{\infty} (-1)^m \mathbf{I}_{2m}(z)\cos 2m\vartheta$$
(4.23)

$$e^{z\cos\vartheta} = I_0(z) + 2\sum_{m=1}^{\infty} I_m(z)\cos m\vartheta$$
(4.24)

where \mathbf{I}_m is the mth modified Bessel function.

Consequently, the integral $\Gamma[g]$ can be written in the form

$$\Gamma[g] = \int_{t_n}^{t_{n+1}} f(t, v(t)) e^{z \cos \vartheta(t)} dt$$

= $I_0(z) \int_{t_n}^{t_{n+1}} f(t, v(t)) dt + 2 \sum_{m=1}^{\infty} I_m(z) \int_{t_n}^{t_{n+1}} f(t, v(t)) \cos(m\vartheta(t)) dt$ (4.25)

thereby expressing it as an infinite sum of integrals. However, the highly oscillatory integrals on the right of (4.25) are amenable to very rapid and efficient numerical calculation with Filon-type methods [32].

Applying the Filon-type method (4.7) to (4.25), it follows that:

$$\Gamma_F[g] = I_0(z) \int_{t_n}^{t_{n+1}} p(t)dt + 2\sum_{m=1}^{\infty} I_m(z) \int_{t_n}^{t_{n+1}} p(t)\cos(m\vartheta(t))dt$$
(4.26)

where p(t) interpolates the function f(t, v(t)).

As shown in [33]:

$$\mathbf{I}_m(z) = \mathbf{i}^{-m} \mathbf{J}_m(iz) \tag{4.27}$$

$$\mathbf{J}_m(z) = \frac{\left(\frac{ez}{2m}\right)^m}{\sqrt{2\pi m}} \tag{4.28}$$

it is true that

$$I_m(z) \sim \frac{1}{\sqrt{2\pi m}} \left(\frac{ez}{2m}\right)^m \tag{4.29}$$

where $z \in \Re$ and $m \gg 1$. Thus the infinite series in (4.26) converges very rapidly such that

$$\Gamma_F[g] = I_0(z) \int_{t_n}^{t_{n+1}} p(t)dt + 2\sum_{m=1}^N I_m(z) \int_{t_n}^{t_{n+1}} p(t)\cos(m\vartheta(t))dt$$
(4.30)

for a relatively small value of N.

Due to the fact that the unknown v(t) features inside the integral sign—the ODE is nonlinear. Hence, the variation of constants formula (4.17) is implicit. To this end, waveform relaxation (WR) is employed to solve it. Several waveform relaxation methods have been developed in the last few decades, see for example [34] [35] [36].

The standard form of WR for (4.3) is

$$y^{[0]}(t) = y_0$$

$$y^{[s]}(t) = e^{tA}y_0 + \int_0^t e^{A(t-\tau)}g(\tau, y^{[s-1]}(\tau))d\tau$$
(4.31)

where s = 1, 2, 3,

Combining the Filon-type quadrature with (4.31) results in the following iterative

scheme, that is executed for each time step.

$$\begin{aligned} v_{n+1}^{[0]} = v_0 \\ \tilde{v}^{[0]}(t) &= \frac{t_{n+1} - t}{h} v_n + \frac{t - t_n}{h} v_{n+1}^{[0]}(\equiv v_n) \\ v_{n+1}^{[1]} = v_n e^{-h/RC} + RI_0[e^{-h/RC} - 1] + \frac{I_0}{C} \int_{t_n}^{t_{n+1}} e^{kb(t)} p(t, \tilde{v}^{[0]}(t)) dt \\ \tilde{v}^{[1]}(t) &= \frac{t_{n+1} - t}{h} v_n + \frac{t - t_n}{h} v_{n+1}^{[1]} \\ v_{n+1}^{[2]} = v_n e^{-h/RC} + RI_0[e^{-h/RC} - 1] + \frac{I_0}{C} \int_{t_n}^{t_{n+1}} e^{kb(t)} p(t, \tilde{v}^{[1]}(t)) dt \\ \tilde{v}^{[2]}(t) &= \frac{t_{n+1} - t}{h} v_n + \frac{t - t_n}{h} v_{n+1}^{[2]} \\ \vdots \\ v_{n+1}^{[s]} = v_n e^{-h/RC} + RI_0[e^{-h/RC} - 1] + \frac{I_0}{C} \int_{t_n}^{t_{n+1}} e^{kb(t)} p(t, \tilde{v}^{[s-1]}(t)) dt \\ \tilde{v}^{[s]}(t) &= \frac{t_{n+1} - t}{h} v_n + \frac{t - t_n}{h} v_{n+1}^{[s]} \end{aligned}$$

$$(4.32)$$

Thus, the Filon nodes are $c_1 = t_n$ and $c_2 = t_{n+1}$, with unit multiplicities at both points: the linear polynomial p agrees with $f(t, \tilde{v}^{[i]}(t))$ at $t = t_n$ and t_{n+1} . The iteration is terminated once $\left| v_{n+1}^{[r]} - v_{n+1}^{[r-1]} \right| < tol$, hence we let $v_{n+1} = v_{n+1}^{[r]}$.

The next example to consider is the more complicated case of analogue amplitude modulation, whereby $b(t) = sin(\omega_1 t)sin(\omega_2 t)$. The first step is to rewrite b(t) in the form

$$b(t) = \frac{1}{2} [\cos(\tilde{\omega}_1 t) - \cos(\tilde{\omega}_2 t)]$$
(4.33)

where

$$\tilde{\omega}_1 = \omega_1 - \omega_2 \tag{4.34}$$

$$\tilde{\omega}_2 = \omega_1 + \omega_2. \tag{4.35}$$

This enables the integral $\Gamma[g]$ to be expressed as a product of two infinite series,

$$\begin{split} \Gamma[g] &= \int_{t_n}^{t_{n+1}} f(t, v(t)) \exp\left(\frac{1}{2}z\cos\tilde{\omega}_1 t - \frac{1}{2}z\cos\tilde{\omega}_2 t\right) dt \\ &= \left[I_0\left(\frac{1}{2}z\right)\right]^2 \int_{t_n}^{t_{n+1}} f(t, v(t)) dt \\ &+ 2I_0\left(\frac{1}{2}z\right) \sum_{m=1}^{\infty} I_m\left(\frac{1}{2}z\right) \int_{t_n}^{t_{n+1}} f(t, v(t)) \cos(m\tilde{\omega}_1 t) dt \\ &+ 2I_0\left(\frac{1}{2}z\right) \sum_{l=1}^{\infty} (-1)^l I_l\left(\frac{1}{2}z\right) \int_{t_n}^{t_{n+1}} f(t, v(t)) \cos(l\tilde{\omega}_2 t) dt \\ &+ 4 \sum_{m=1}^{\infty} \sum_{l=1}^{\infty} (-1)^l I_m\left(\frac{1}{2}z\right) I_l\left(\frac{1}{2}z\right) \int_{t_n}^{t_{n+1}} f(t, v(t)) \cos(m\tilde{\omega}_1 t) \cos(l\tilde{\omega}_2 t) dt. \end{split}$$
(4.36)

Following the same procedure as before, the Filon-type method corresponding to (4.37) is

$$\Gamma[g] = \left[I_0 \left(\frac{1}{2} z \right) \right]^2 \int_{t_n}^{t_{n+1}} p(t) dt + 2I_0 \left(\frac{1}{2} z \right) \sum_{m=1}^{\infty} \int_{t_n}^{t_{n+1}} p(t) \cos(m\tilde{\omega}_1 t) dt + 2I_0 \left(\frac{1}{2} z \right) \sum_{l=1}^{\infty} (-1)^l I_l \left(\frac{1}{2} z \right) \int_{t_n}^{t_{n+1}} p(t) \cos(l\tilde{\omega}_2 t) dt + 4 \sum_{m=1}^{\infty} \sum_{l=1}^{\infty} (-1)^l I_m \left(\frac{1}{2} z \right) I_l \left(\frac{1}{2} z \right) \int_{t_n}^{t_{n+1}} p(t) \cos(m\tilde{\omega}_1 t) \cos(l\tilde{\omega}_2 t) dt.$$
(4.37)

where p(t) is the interpolating polynomial of f.

4.4 Numerical Results

The proposed method will be illustrated with two input signals, one is amplitude modulated and the second is digitally modulated. The values of the parameters are specifically chosen to emphasise both the highly oscillatory nature of the responses and the important features of the algorithm presented before.

4.4.1 Amplitude Modulation

For this case, $b(t) = sin\omega_1 t sin\omega_2 t$, with $\omega_1 = 100$ rad/s and $\omega_2 = 10^7$ rad/s. There is a difference of five orders of magnitude between the two oscillation scales. The values are set as: $I_0 = 1A$, C = 1f, $R = 1\Omega$ and k = 40.



Figure 4.2: The exact solution of the amplitude-modulated equation with $I_0 = 1$, C = 1, R = 1 and k = 40.

Figure 4.2 depicts the 'exact' solution of the ODE—(this was formed using a numerical solution with an exceedingly small step size). It is clear that the solution is a slow-varying wave and the extraordinarily large frequency ω_2 is not visible. Nonetheless, the presence of high oscillation is enough to render traditional numerical methods inefficient. In Figs. 4.3–4.5 we exhibit the outcome of integrating the ODE in the interval [0, 0.2] with the constant step size $h = 2.5 \times 10^{-4}$ with three numerical methods.

In each case, the numerical solution is denoted by solid line while, for comparison, the exact solution from Fig. 4.2 features as a dashed line.

The standard approach to increase accuracy is to use a higher-order method. To this end, the trapezoidal rule is of order 2, while the Runge-Kutta method employed is of order 3. The improvement in accuracy is tangible but neither method gives satisfactory results.

In comparison, the Filon-type method from Fig. 4.5, using just linear interpolation, produces a result which is visually hardly distinguishable from the exact solution.

It should be noted that the waveform relaxation converged fairly rapidly and letting N = 7 in (4.37) our simulations produced perfectly satisfactory results.



Figure 4.3: The solution for the amplitude-modulated case with the trapezoidal rule.



Figure 4.4: *The solution for the amplitude-modulated equation with the third-order explicit Runge-Kutta method.*



Figure 4.5: The solution for the amplitude-modulated case using the Filon-type method, combined with waveform relaxation.

The values for the first example shown above are selected to emphasise various qualities of the Filon-method. Here, an example with realistic values will be implemented. $I_0 = 10^{-12}A$, $C = 10^{-6}f$, $R = 100\Omega$ and k = 1/0.0259. The input signal is the same as the one in the first example, i.e., $b(t) = sin\omega_1 t sin\omega_2 t$, with $\omega_1 = 100$ rad/s and $\omega_2 = 10^7$ rad/s. As seen from Fig. 4.6, the results from the Filon-method and exact solution match well.



Figure 4.6: *The solution for the amplitude-modulated case with realistic values using the Filon-type method combined with waveform relaxation.*

4.4.2 Digital Modulation

As an example of digital modulation, we have taken $b(t) = x(t) \cos \omega \pi t$, where $\omega = 4 \times 10^9$ rad/s, while x(t) is an alternating sequence of +1 and -1 with a bit period of 133 ns.¹ This is an example for the Binary Phase Shift Keying modulation technique. We let $I_0 = 100A$, $C = 10^{-4}f$, $R = 1\Omega$ and k = 1.

¹In a realistic model x(t) is random, rather than alternating. However, introducing stochastic component would have made the comparison of different methods considerably more difficult.



Figure 4.7: The exact solution of the digitally modulated equation.



Figure 4.8: A very small time sub-interval of Fig. 4.7

The exact solution of the digitally-modulated equation with the above parameters is displayed in Fig. 4.7 and 4.8. Fig. 4.8 zooms in on a small time window to show that the solution exhibits exceedingly rapid, small-amplitude oscillations. Such oscillations are invisible in less detailed plots but, defeat traditional ODE solvers or further decrease the step size.

In Figs. 4.9-4.11, we display the errors committed by three numerical methods, all with a constant step size $h = 2.5 \times 10^{-10}$, applied to the digital-modulation equation: the (second-order) trapezoidal rule, the standard explicit, third-order three-stage Runge-Kutta method and the Filon-type method with piecewise-linear approximation and waveform relaxation. The reason for the choice of the minute step size, roughly of the order of magnitude of $O(\omega^{-1})$, is that traditional methods require it. Even taking such a tiny step size, it is evident how the numerical solution (solid line) rapidly departs from the exact solution, denoted by a dashed line. All accuracy is lost in even such a short interval. In comparison, the Filon-type method produces an outcome visually indistinguishable from the exact solution. Of course, with Filon-type method, an accurate solution would be obtained with a larger step size: in methods designed using asymptotic principles the size of the step plays a minor role insofar as accuracy is concerned.

This Chapter presents a preliminary study into an alternative technique for numerical integration suitable for systems subjected to high-frequency signals. Numerical examples confirm the theoretical expectations of the significant potential of Filon-type methods.

4.5 Comparison with the Envelope Simulation

As stated in Chapter 1, Advanced Design System (ADS) from Angilent Technologies is the EDA tool which is often employed for the simulation of high frequency circuits.



Figure 4.9: The solution of the digital-modulated equation by the trapezoidal rule.



Figure 4.10: The solution of the digital-modulated equation by third-order explicit Runge-Kutta method.



Figure 4.11: *The solution of the digital-modulated equation by the Filon-type method, combined with waveform relaxation.*

Thus the diode rectifier circuit shown in Fig. 4.1 is simulated with the envelope simulator in ADS in order to compare it in terms of accuracy and efficiency to the Filon-type method.

The Envelope Transient Harmonic Balance technique (ETHB) [37] [38] is used in the envelope simulator. This is a mixed-mode simulation technique which performs the analysis in both the time- and the frequency-domain [1]. Briefly, the RF carrier signal is represented in the frequency domain and the modulation envelope is represented in the time domain.

Consider the fast oscillation signal shown as below:

$$u(t) = \sum_{k} U_k(t) e^{j2\pi f_k t}$$
(4.38)

where $U_k(t)$ and f_k represent the amplitudes and frequencies of the kth period of the envelope and t is the time scale. Note that $U_k(t)$ must vary slowly relative to f_k because when the bandwidth of is greater than $f_k/2$ then the sidebands of adjacent harmonics begin to overlap [39]. If there is an f_0 , which satisfies $f_k = K f_0$, then u(t) is one tone signal, and the corresponding analysis is a single carrier ETHB envelope simulation. Otherwise, u(t) is a multi-tone signal, and the corresponding analysis is a multi-carrier ETHB envelope simulation. As shown in Fig. 4.12 the output of the ETHB envelope simulation is a time-varying spectrum for time points t_1, t_2, \ldots The HB analysis is performed at each time points t_i . Also, the spectrum is converted to time domain waveforms in each clock cycle.



Figure 4.12: A Modulated signal and its simulated time-varying spectrum [1].

The circuit in Section 4.4 is simulated in ADS using the same values for the capacitance, resistance and input signal. The schematic is shown in Appendix D. However, ADS is NOT able to do the simulation with the same time-step stated in Section 4.4.1 and 4.4.2 as used in Filon-type method. The simulation can only be done in ADS when the time-step size is decreased to a suitable value. This means that the Filon-type method is more efficient than the ADS ETHB envelope simulation for the selected system.
4.6 Conclusions

We have presented a preliminary study into an alternative techniques for numerical integration of systems subject to high-frequency signals. We applied the basic theory to a rectifier diode circuit. Two cases are considered - when the input signal to the circuit is amplitude modulated and when it is digital modulated. Numerical results confirm the theoretical expectations of the significant potential of Filon-type methods in this setting. The simulation of the frequency-modulated signals will be addressed in future research.

Chapter 5

Simulation Techniques to Capture the Locking Range of an Injection-Locked Frequency Divider

5.1 Introduction to the Injection-Locked Frequency Divider (ILFD)

Phase-locked loops (PLL) are widely used in wireless communication applications such as frequency synthesis. The frequency divider (FD) is used in the feedback of a frequency synthesizer in order to divide the frequency by a fixed number. Frequency dividers are categorized into two groups, i.e. injection-locked (ILFD) and static [40] frequency dividers. Static dividers offer a wide-bandwidth but with a high power consumption. On the contrary, ILFDs consume less power but at the expense of a narrow locking range. One of the drawbacks to the use of injection locking as a method of frequency division is that there is a limited input bandwidth (locking range) over which frequency division occurs. Therefore, the locking range is one of the most important concerns in designing ILFDs. Hence, an efficient and accurate method for its determination from simulation is essential. Locking occurs when the oscillator locks into or tracks the phase and frequency of an injected signal. The output frequency is either equal to the input frequency or a submultiple or harmonic of it. Hence, the term "fre-

quency divider". In this work, the LC-oscillator based ILFD shall be considered so the first section of this chapter describes it. Then several existing modelling approaches for injection-locked oscillators are presented. The chapter proceeds to review some techniques for determining the locking ranges of oscillators. Finally, some novel approaches and numerical results for determination of locking ranges are presented.

5.2 Background to the LC-Oscillator Based ILFD

5.2.1 Effect of Negative Resistance on LC Oscillators

The concept of negative resistance is used to understand the operation of an oscillator. Figure 5.1 shows a model of a simple tank circuit.



Figure 5.1: Decaying impulse responses of a tank.

Consider the tank circuit being stimulated by a current impulse. The tank responds with decaying oscillatory behaviour because, in every cycle, some energy is lost in the form of heat dissipated in the resistor R_p . Now suppose that a resistor equal to $-R_p$ is placed in parallel with the tank, then since $R_p || - R_p = \infty$, the tank oscillates indefinitely, as illustrated in Fig. 5.2.

Thus, if a circuit exhibiting a negative resistance is placed in parallel with the lossy tank, then the combination may oscillate. Such a topology is called a one-port oscillator. In practice, a negative resistance is provided by using an active circuit, as



Figure 5.2: Addition of negative resistance to cancel tank losses.

shown in Fig. 5.3. The following quote is taken from [41], which remarks on the confusing decision of whether an oscillator is of the negative resistance type or not:

"As a final comment, it should be clear that many (if not all) oscillators may be considered as negative resistance oscillators since, from the point of view of the tank, the active elements cancel the loss due to the finite Q of the resonators. Hence, whether to call an oscillator a "negative resistance" oscillator type is actually more a philosophical decision than anything fundamental."



Figure 5.3: Use of an active circuit to provide negative resistance.

5.2.2 LC Oscillator with Cross-Coupled Negative Resistor

Due to their relatively good phase noise performance, ease of implementation, and differential operation, cross-coupled inductance–capacitance (LC) oscillators play an important role in high–frequency circuit design [42]. This topology uses both NMOS



Figure 5.4: Cross-coupled CMOS LC Oscillator.

and PMOS cross-coupled pairs to provide the negative resistance necessary to cancel the losses in the tank circuit.

As shown in Fig. 5.4, two cross-coupled CMOS inverters, consisting of M1 to M4, are used to generate the negative resistance needed to cancel the losses in the tank circuit. The resonant circuit comprising of R, L and C is connected in parallel with the negative resistance. The differential output of the oscillator is available between the nodes labeled V_{out1} and V_{out2} . A current source is used to provide a bias current I_o for the transistors.

5.3 Overview of the Existing Models of Injection-Locked Oscillators

5.3.1 Adler's Model

Adler [2] introduced the mechanism of injection-locking for a small injected signal. Figure 5.5 [43] shows a simple, popular model which can describe the injection-locked oscillator.



Figure 5.5: Adler's model for the injection-locked oscillator.

In this model, the two inputs v_i and v_o simply add before being operated on by the nonlinearity f. The nonlinearity is required both for amplitude stability and to enable frequency mixing. The linear filter $H(j\omega)$ rejects all frequencies far from the frequency of the free-running oscillator ω_o . The expression for the nonlinear device f is derived as a polynomial series, which is:

$$f(v_i + v_o) = \sum_{m=0}^{\infty} a_m \cdot (v_i + v_o)^m.$$
 (5.1)

By setting $v_o = V_o \cos(\omega_o t + \varphi)$ and $v_i = V_i \cos(\omega_i t)$, the products of $f(v_i + v_o)$ are obtained as:

$$products = f(v_i + v_o)$$
$$= \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} K_{m,n} \cos(n\omega_i t) \cos(m\omega_o t + n\varphi).$$
(5.2)

For some integers m and n, when $|m\omega_o \pm n\omega_i| = \omega_o$, the corresponding output terms in (5.2) will exist at ω_o , the frequency of the free-running oscillator. The output of f has a phase shift with respect to the input. For oscillation, Barkhausen's criterion must be satisfied. Hence, the phase shift contributed by $H(j\omega)$ must adjust so that the net phase around the loop is $2k\pi$. If the input frequency is too large, then this is not possible and locking fails. The locking range is the set of frequencies for which locking is possible.

5.3.2 Miller's Model

The injection-locked oscillator (ILO) can be modeled with a generalized Miller-type model [44] [45], as shown in Fig. 5.6. The filter $H(j\omega)$ is assumed to reject all frequencies except ω_o .

As seen from Fig. 5.6, there are two memoryless nonlinear functions, $f(v_o)$ and $g(v_i)$, in this model:

$$f(v_o) = \sum_{n=0}^{\infty} a_m \cdot v_o^m.$$
(5.3)

$$g(v_i) = \sum_{n=0}^{\infty} b_n \cdot v_i^n.$$
(5.4)



Figure 5.6: Miller-type model for the injection-locked oscillator

Set $v_o = V_o \cos(\omega_o t + \varphi)$ and $v_i = V_i \cos(\omega_i t)$. The output of the mixer is then obtained by a product of Fourier series for f and g.

$$product = f \cdot g$$
$$= \left[\sum_{n=0}^{\infty} B_n \cdot \cos(n\omega_i t)\right] \cdot \left[\sum_{m=0}^{\infty} A_m \cdot \cos(m\omega_o t + m\varphi)\right], \quad (5.5)$$

where the coefficients B_n are functions of the input amplitude V_i only, while A_m are functions of the output amplitude V_o . We can then determine which products lie at ω_o . This model has its limitations when predicting the sub- and superharmonic injection locking. In particular, the expression for the products is not accurate enough because the coefficients A_m and B_n may be functions of both V_i and V_o in practice.

5.3.3 Verma's Model

Verma [46] introduced a more general model for the injection-locked frequency divider shown in the block diagram in Fig. 5.7.

Here, $f(v_i, v_o)$ is assumed as a memoryless nonlinear function of both v_i and v_o . The linear filter $H(j\omega)$ filters out all the frequencies other than ω_o . The expression for f is given as:

$$f(v_i, v_o) = \sum_{m=0}^{\infty} a_m(v_i) \cdot v_o^m.$$
 (5.6)



Figure 5.7: General model for the injection-locked frequency divider.

Using a Taylor series expansion of v_i around a dc point V_{dc} , f is obtained as:

$$f(v_i, v_o) = \sum_{n=0}^{\infty} \frac{(v_i - V_{dc})^n}{n!} \cdot \left[\sum_{m=0}^{\infty} \frac{\partial^n}{\partial v_i^n} a_m(v_i) \Big|_{v_i = V_{dc}} v_o^m \right].$$
 (5.7)

It is assumed that the magnitude of the injection is weak compared to the static bias point, i.e. v_i is close to V_{dc} . In this case, only the terms with n < 2 in Eq. (5.7) are considered.

$$f(v_i, V_o) \cong \sum_{m=0}^{\infty} a_m(V_{dc}) \cdot v_o^m + (v_i - V_{dc}) \cdot \left[\sum_{m=0}^{\infty} \widetilde{a}_m(V_{dc}) \cdot v_o^m\right]$$
(5.8)

where

$$\tilde{a}_m(V_{dc}) = \left. \frac{\partial}{\partial_{v_i}} a_m(v_i) \right|_{v_i = V_{dc}}.$$
(5.9)

The coefficients a_m and their derivatives can be determined either from the analytical form of f or extracted by measuring the effect of slight perturbations on the nonlinearity about the bias point V_{dc} .

If both v_i and v_o are sinusoidal and of the form $v_i = V_{dc} + V_i \cos(\omega_i t)$ and $v_o = V_o \cos(\omega_o t + \varphi)$, the full output of (5.7) can be written as:

$$f(v_i, v_o) = \sum_{n=0}^{\infty} \frac{(V_i \cos(\omega_i t))^n}{n!} \cdot \left[\sum_{m=0}^{\infty} \frac{\partial^n}{\partial v_i^n} a_m \right|_{v_i = V_{dc}} (V_o \cos(\omega_o t + \varphi))^m \right].$$
(5.10)

The terms may be regrouped to express the bracketed quantity in (5.10) as a sum of harmonics of ω_o

$$f(v_i, v_o) = \sum_{n=0}^{\infty} \frac{(V_i \cos(\omega_i t))^n}{n!} \cdot \left[\sum_{m=0}^{\infty} \frac{\partial^n}{\partial v_i^n} A_m \right|_{v_i = V_{dc}} \cos(m\omega_o t + m\varphi) \right]$$
(5.11)

where the coefficient A_m is a function of all of the a_m and V_o . Assuming weak injection and simplifying (n < 2)

$$f(v_i, v_o) = \sum_{m=0}^{\infty} A_m \cos(m\omega_o t + m\varphi) + \frac{1}{2} \sum_{m=0}^{\infty} V_i \tilde{A}_m \cos[(m\omega_o \pm \omega_i)t + m\varphi],$$
(5.12)

where

$$\tilde{A}_m = \left. \frac{\partial A_m}{\partial v_i} \right|_{v_i = V_{dc}}.$$
(5.13)

Note that the first term in (5.12) represents the nonlinearity for a free-running oscillator while the second term shows the mixer products due to the presence of the injected signal.

5.3.4 Xu's Model

Xu's model [7] is a specialised model for an LC-oscillator based injection-locked frequency divider (LC-ILFD). This model proves that the LC-ILFD behaves like a Phase-Locked Loop (PLL) when it is locked.

The circuit schematic of the LC-ILFD under consideration has been described in [47], [48], and is shown in Fig. 5.8.

It can be simplified to the ideal circuit shown in Fig. 5.9. $N_{\rm R}$ represents the nonlinear resistor comprising the components in Fig. 5.8 other than $R_{\rm s}$, L and C.

When there is no injected signal, the state equations that describe this circuit are:

$$C\frac{dV_{\rm C}}{dt} = I_{\rm L} - f(V_{\rm C}) \tag{5.14}$$

$$L\frac{dI_{\rm L}}{dt} = -I_{\rm L}R_{\rm S} - V_{\rm C} \tag{5.15}$$



Figure 5.8: Circuit schematic.



Figure 5.9: The simplified circuit.

where $f(V_{\rm C})$ is the driving point characteristic of the non-linear resistor which can be obtained by experiment and/or by Spice simulation. The schematic used in PSpice to capture the driving point characteristics is illustrated in Fig. 5.10. The RLC tank is replaced by a DC source, whose value is increased from -9V to 9V during the simulation.



Figure 5.10: The schematic used in PSpice to capture the driving point characteristics.

The driving-point characteristics derived from Spice simulations and experiments are shown in Fig. 5.11, together with a cubic approximation. The cubic approximation is:

$$f(V_{\rm C}) = aV_{\rm C} + bV_{\rm C}^2 + cV_{\rm C}^2$$
(5.16)

where

$$c = -\frac{a}{V_{\rm DD}^2} \tag{5.17}$$

and the parameter $a(V_{\text{GS}})$ is a function of the bias voltage of the tail transistor M_6 , shown in Fig. 5.8. Note that the second harmonic term bV_{C}^2 is so small as to be ignored.



Figure 5.11: Driving point characteristics: Solid: Experimental; Dotted: Cubic Fit To Experiment; Dashed: PSpice.

Thus the cubic approximation (5.16) is expressed in the form:

$$f(V_{\rm C}) = aV_{\rm C} - \frac{a}{V_{\rm DD}^2}V_{\rm C}^3$$
(5.18)

This free-running oscillator can be viewed as a feedback system; its model is shown in Fig. 5.12. In this model, the RLC tank is represented by its transfer function $H(j\omega)$.

When a signal is injected into the circuit shown in Fig. 5.8, the parameter a is modified to:

$$a(V_{\rm GS} + v_{\rm GS}) \approx \hat{a}(V_{\rm GS}) + \left. \frac{d\hat{a}}{dV_{\rm GS}} \right|_{V_{\rm GS}} v_{\rm GS}, \tag{5.19}$$

where $V_{\rm GS}$ is the bias voltage of M_6 .

By varying $V_{GS_{M6}}$, a number of different driving point characteristics are found and thus a number of different cubic expressions for $f(V_{\rm C})$ [48]. Then several values for Aare extracted and plotted as shown in Fig. 5.13. The slope around the bias point, $\frac{d\hat{a}}{dV_{\rm GS}}$, can be estimated analytically after fitting a quadratic to the $I_{D_{M6}}-V_{GS_{M6}}$ characteristic.



Figure 5.12: Model of the free-running oscillator.



Figure 5.13: I_D against V_{GS} .

Substituting (5.19) into (5.18), the governing equations for the ILFD are obtained as:

$$C\frac{dV_{\rm C}}{dt} = I_{\rm L} - (A + daV_{\rm in})V_{\rm C} + \frac{A + daV_{\rm in}}{V_{\rm DD}^2}V_{\rm C}^3$$
(5.20)

$$L\frac{dI_{\rm L}}{dt} = -I_{\rm L}R_{\rm S} - V_{\rm C},\tag{5.21}$$

where A and da are used to represent $\hat{a}(V_{\text{GS}})$ and $\frac{d\hat{a}}{dV_{\text{GS}}}\Big|_{V_{\text{GS}}}$ in (5.19), respectively. Equation (5.20) may be rearranged by collecting the factors A and daV_{in} , to obtain:

$$C\frac{dV_{\rm C}}{dt} = I_{\rm L} - A\left(V_{\rm C} - \frac{V_{\rm C}^3}{V_{\rm DD}^2}\right) - daV_{\rm in}\left(V_{\rm C} - \frac{V_{\rm C}^3}{V_{\rm DD}^2}\right).$$
 (5.22)

The difference between (5.14) and (5.22) is the additional feedback terms $-daV_{in}\left(V_{C} - \frac{V_{C}^{3}}{V_{DD}^{2}}\right)$. The injection-locked system described by (5.22) can be modeled as shown in Fig. 5.14.

As evident from Fig. 5.14, the negative feedback is composed of two loops in this case. Both paths pass through the nonlinear device but the inner loop has a gain of A



Figure 5.14: The loop enhanced model of the ILFD.

and the outer loop has a gain of da. The nonlinear device is governed by the expression:

$$v_{\rm nd} = 1 - \frac{v_{\rm o}^2}{V_{\rm DD}^2}.$$
 (5.23)

The input signal, v_{in} , is first multiplied by the output of the outer loop to form I_{inj} . I_{inj} then combines with the output of the inner loop and the output is passed through the RLC tank. This model is specific to an LC-ILFD unlike previous models [46]. The RLC tank in the oscillator can be expressed as:

$$H(j\omega) = \frac{R_{\rm S} + j\omega L}{1 + j\omega R_{\rm S}C + (j\omega)^2 LC}.$$
(5.24)



Figure 5.15: Analog Multiplier as a phase detector.

As seen in Fig. 5.15, if the two inputs of the analog multiplier are $A \cos \omega t$ and $B \cos(\omega t + \phi)$, the output can be expressed as

$$AB\cos(\omega t)\cos(\omega t + \phi) = \frac{AB}{2}(\cos(\phi) - \cos(2\omega t + \phi)).$$
(5.25)

The average value of the output is

$$\overline{AB\cos(\omega t)\cos(\omega t + \phi)} = \frac{AB}{2}\cos(\phi).$$
(5.26)

Then the phase difference between the two input signals can be obtained. Thus an analog multiplier acts as a phase detector [41]. It can be concluded that the injection-locked frequency divider is equivalent from a mathematical perspective to a dual-loop phase-locked-loop (PLL) [7], when the injected frequency approximates the free-running oscillator frequency.

5.4 Existing Methods to Obtain the Locking Range of Oscillators

5.4.1 Adler Approach

This method was described in [2]. It is assumed that the free-running oscillation gives an angular frequency ω_0 . The injected signal has the frequency ω_1 which forces a beat frequency:

$$\Delta\omega_0 = \omega_0 - \omega_1. \tag{5.27}$$

Because of the effect of the injected signal, the output frequency changes to ω , and the instantaneous beat frequency is termed as:

$$\Delta\omega = \frac{d\alpha}{dt} \tag{5.28}$$

$$=\omega-\omega_1 \tag{5.29}$$

where as shown in Fig. 5.16, α is the angle between the input and output signal in the excited system.

As seen in Fig. 5.16 and Fig. 5.17, E_1 and E_g are the amplitude of the input and output signals in the excited system, E is the amplitude of the output signal of the free-running circuit. If the amplitude of the injected signal is much smaller than that of the free-running output signal, i.e. $E_1 \ll E$, then:

$$\varphi = \frac{E_1 \sin(-\alpha)}{E} = -\frac{E_1}{E} \sin \alpha.$$
(5.30)



Figure 5.16: Vector diagram of the instantaneous voltages.



Figure 5.17: Sample oscillator circuit [2].



Figure 5.18: Phase versus frequency for a simple tuned circuit.

Figure 5.18 shows the φ versus ω curve. The slope of it is

$$A = \frac{d\varphi}{d\omega}.$$
(5.31)

The angle between the free-running output signal and the resultant signal can be calculated as:

$$\varphi = A(\omega - \omega_0). \tag{5.32}$$

Now rearrange (5.32) and substitute (5.27) and (5.29):

$$\varphi = A(\omega - \omega_0)$$

= $A[(\omega - \omega_1) - (\omega_0 - \omega_1)]$
= $A(\Delta \omega - \Delta \omega_0).$ (5.33)

After substituting (5.30) on the left side and (5.28) on the right side,

$$-\frac{E_1}{E}\sin\alpha = A\left(\frac{d\alpha}{dt} - \Delta\omega_0\right).$$
(5.34)

Then

$$\frac{d\alpha}{dt} = -\frac{E_1}{E} \cdot \frac{1}{A}\sin\alpha + \Delta\omega_0.$$
(5.35)

The quality factor Q is defined as in [41]:

$$Q = \frac{\varphi}{2} \cdot \frac{\omega_0}{\omega - \omega_0}.$$
(5.36)

Merging (5.32) and (5.36),

$$A = \frac{2Q}{\omega_0}.$$
(5.37)

Substituting (5.37) into (5.35), yields:

$$\frac{d\alpha}{dt} = -\frac{E_1}{E} \cdot \frac{\omega_0}{2Q} \sin \alpha + \Delta \omega_0.$$
(5.38)

If the system is in steady state, $d\alpha/dt$ must be zero. Hence,

$$0 = -\frac{E_1}{E} \cdot \frac{\omega_0}{2Q} \sin \alpha + \Delta \omega_0 \tag{5.39}$$

or

$$\sin \alpha = 2Q \frac{E}{E_1} \cdot \frac{\Delta \omega_0}{\omega_0}.$$
(5.40)

Because $\sin \alpha$ can only assume values between +1 and -1, the right side of (5.40) must be within this range.

$$\left|2Q\frac{E}{E_1} \cdot \frac{\Delta\omega_0}{\omega_0}\right| \le 1.$$
(5.41)

So the locking range is

$$|\Delta\omega_0| \le \frac{E_1}{E} \cdot \frac{\omega_0}{2Q}.$$
(5.42)

For an RLC tank, the Q factor may be expressed as:

$$Q = R\sqrt{\frac{C}{L}}.$$
(5.43)

Hence, in an RLC circuit, the locking range is:

$$|\Delta\omega_0| \le \frac{E_1}{E} \cdot \frac{\omega_0}{2R} \cdot \sqrt{\frac{L}{C}}.$$
(5.44)

5.4.2 Graphical Analysis of the Phase Characteristic

In this second method, the locking range is determined graphically, as described in [10]. For example, consider Fig. 5.19, the frequency of the free-running oscillator is defined as $\omega_0 = 1/\sqrt{L_1C_1}$. (Neglecting all of the parasitics). As illustrated in Fig. 5.19(b), when a phase shift ϕ_0 is inserted into the system, the oscillator no longer oscillates at the frequency ω_0 . Thus, as shown in Fig. 5.19(c), the output frequency deviates from ω_0 to a new value ω_1 so as to cancel the effect of ϕ_0 and satisfy Barkhausen's criterion. If a current source I_{inj} is added to the drain of M_1 , as shown in Fig. 5.19(d), and if the frequency of I_{inj} is within the locking range, the output frequency of the system will be at the injected frequency ω_{inj} . The oscillator is then injection-locked.



Figure 5.19: (*a*)*Conceptual oscillator.*(*b*) *Frequency shift due to additional phase shift.* (*c*) *Open-loop characteristics.*(*d*)*Frequency shift by injection.*



Figure 5.20: Phasor diagram.

In order to determine the locking range, the phasor diagram for the oscillator subject to an external signal is shown in Fig. 5.20. ϕ_0 is the angle between the free-running output signal I_{osc} and the resultant output signal of the injected system I_T . θ denotes the phase difference between I_{osc} and I_{inj} . The relationship between them can be expressed as:

$$\sin \phi_0 = \frac{I_{inj}}{I_T} \sin \theta \tag{5.45}$$

$$=\frac{I_{inj}\sin\theta}{\sqrt{I_{osc}^2+I_{inj}^2+2I_{osc}I_{inj}\cos\theta}}.$$
(5.46)

The maximum value of $\sin \phi_0$ is

$$\sin\phi_{0,max} = \frac{I_{inj}}{I_{osc}} \tag{5.47}$$

when the condition

$$\cos\theta = -\frac{I_{inj}}{I_{osc}}.$$
(5.48)

is achieved.

As shown in Fig. 5.21, the angle between I_T and I_{inj} has to be $\pi/2$ so as to satisfy the condition (5.48). Hence, the maximum angle between I_{inj} and I_{osc} is $\pi/2 + \phi_0$.



Figure 5.21: Phasor diagram with maximum angle between I_{inj} and I_T .

A second-order parallel tank consisting of L, C, and R_P exhibits a phase shift of

$$\alpha = \frac{\pi}{2} - \tan^{-1} \left(\frac{L\omega}{R_P} \cdot \frac{\omega_0^2}{\omega_0^2 - \omega^2} \right).$$
(5.49)

Since
$$\omega_0^2 - \omega^2 \approx 2\omega_0(\omega_0 - \omega)$$
, $L\omega/R_P = 1/Q$, and $\pi/2 - \tan^{-1}x = \tan^{-1}(x^{-1})$,

$$\tan \alpha \approx \frac{2Q}{\omega_0} (\omega_0 - \omega). \tag{5.50}$$

From Fig. 5.21, $\tan \phi_0 = I_{inj}/I_T$ and $I_T = \sqrt{I_{osc}^2 - I_{inj}^2}$. It follows that

$$\omega_0 - \omega_{inj} = \frac{\omega_0}{2Q} \cdot \frac{I_{inj}}{I_{osc}} \cdot \sqrt{\frac{1}{1 - \frac{I_{inj}^2}{I_{osc}^2}}}$$
(5.51)

If the maximum value of $\omega_0 - \omega_{inj}$ is set as ω_L , the overall locking range is obtained as $\pm \omega_L$ around ω_0 . Here ω_L is "one-sided" locking range.

Note that the locking range depends on the amplitude of the injected signal, I_{inj} . If I_{inj} increases, the phase difference between I_{osc} and I_{inj} must become greater so as to maintain the angle between I_T and I_{osc} at ϕ_0 .

5.4.3 Analysis with Harmonic Balance

In this section, the locking range is obtained by analysing an ILFD using Harmonic Balance [8]. The general model of an ILFD is shown in Fig 5.22. $f(v_{inj}, v_o)$ represents



Figure 5.22: General Circuit Model for ILFD.

the nonlinear resistor while $H(j\omega)$ is the RLC tank.

The first step is to define the input and output voltages of the ILFD [49]:

$$v_{inj} = \frac{V_{inj}}{2} [e^{j(\omega_{inj}t+\phi)} + e^{-j(\omega_{inj}t+\phi)}]$$
(5.52)

$$v_o = \frac{V_o}{2} [e^{j\omega_{\text{out}}t} + e^{-j\omega_{\text{out}}t}].$$
(5.53)

A third-order approximation is derived for the nonlinear block $f(v_{inj}, v_o)$

$$f_{(v_{inj},v_o)} = a(v_{inj})v_o + b(v_{inj})v_o^2 + c(v_{inj})v_o^3$$
(5.54)

where the parameters a, b and c are the functions of v_{inj} which are shown as below:

$$a(v_{inj}) = A + dav_{inj},$$

$$b(v_{inj}) = B + dbv_{inj},$$

$$c(v_{inj}) = C + dcv_{inj}.$$

(5.55)

Then by substituting equations (5.52), (5.53) and (5.55) into equation (5.54), it is

obtained:

$$f_{o}(v_{inj}, v_{o}) = (A + \mathrm{d}av_{\mathrm{inj}})v_{o} + (B + \mathrm{d}bv_{\mathrm{inj}})v_{o}^{2} + (C + \mathrm{d}cv_{\mathrm{inj}})v_{o}^{3}$$

$$= \left(A + \mathrm{d}a\frac{V_{\mathrm{inj}}}{2}\left[e^{j(\omega_{\mathrm{inj}}t+\phi)} + e^{-j(\omega_{\mathrm{inj}}t+\phi)}\right]\right) \times \frac{V_{o}}{2}\left[e^{j\omega_{\mathrm{out}}t} + e^{-j\omega_{\mathrm{out}}t}\right]$$

$$+ \left(B + \mathrm{d}b\frac{V_{\mathrm{inj}}}{2}\left[e^{j(\omega_{\mathrm{inj}}t+\phi)} + e^{-j(\omega_{\mathrm{inj}}t+\phi)}\right]\right)$$

$$\times \frac{V_{o}^{2}}{4}\left[e^{2j\omega_{\mathrm{out}}t} + 2 + e^{-2j\omega_{\mathrm{out}}t}\right]$$

$$+ \left(C + \mathrm{d}c\frac{V_{\mathrm{inj}}}{2}\left[e^{j(\omega_{\mathrm{inj}}t+\phi)} + e^{-j(\omega_{\mathrm{inj}}t+\phi)}\right]\right)$$

$$\times \frac{V_{o}^{3}}{8}\left[e^{3j\omega_{\mathrm{out}}t} + 3e^{j\omega_{\mathrm{out}}t} + 3e^{-j\omega_{\mathrm{out}}t} + e^{-3j\omega_{\mathrm{out}}t}\right].$$
(5.56)

From the many terms generated in (5.56), the terms of interest are those that fall close to ω_{out} as the resonator $H(j\omega)$ attenuates the components far from ω_{out} . The bandpass filter response can be expressed as [50]

$$H(j\omega_0) = \frac{R_L H_0}{1 + j2Q\frac{\delta\omega}{\omega_0}}$$
(5.57)

where ω_0 is the free-running frequency, $\delta\omega$ is the one-sided locking range defined as

$$\delta\omega = \omega_{\rm out} - \omega_0. \tag{5.58}$$

 ${\cal Q}$ is the quality factor of the free-running LC tank, which is given by

$$Q = \frac{\omega_0 L}{R_L} \tag{5.59}$$

 H_0 is the coefficient to make the term $R_L H_0$ represent the net resistance across the LC tank. It can be found by using impedance transformation:

$$H_0 = Q^2 + 1. (5.60)$$

The following components at ω_{out} are generated. From av_o :

$$\frac{AV_o}{2} \left[e^{j(\omega_{\text{out}}t)} + e^{-j(\omega_{\text{out}}t)} \right]$$
(5.61)

From bv_o^2 : $\frac{\mathrm{d}bV_{inj}V_o^2}{8} \left(2e^{j(\omega_{\mathrm{inj}}t+\phi)} + e^{j(-\omega_{\mathrm{inj}}t-\phi+2\omega_{\mathrm{out}}t)} + 2e^{-j(\omega_{\mathrm{inj}}t+\phi)} + e^{j(\omega_{\mathrm{inj}}t+\phi-2\omega_{\mathrm{out}}t)}\right) \quad (5.62)$

From cv_o^3 :

$$\frac{3CV_o^3}{8}(e^{j\omega_{\text{out}}t} + e^{-j\omega_{\text{out}}t})$$
(5.63)

Next, these components pass through the resonator back to the input, which results

$$\frac{R_L H_o}{1+j\left(\frac{2Q\delta\omega}{\omega_0}\right)} \left[\frac{AV_o}{2}e^{j\omega_{\text{out}}t} + db\frac{V_{inj}V_{\text{out}}^2}{8}\left(2e^{j(\omega_{inj}t+\phi)} + e^{j(2\omega_{\text{out}}t-\omega_{\text{inj}}t-\phi)}\right) + \frac{3CV_o^3}{8}e^{j\omega_{\text{out}}t}\right] = \frac{V_o}{2}e^{j\omega_{\text{out}}t}$$
(5.64)

and

in

$$\frac{R_L H_o}{1+j\left(\frac{2Q\delta\omega_0}{\omega}\right)} \left[\frac{AV_o}{2}e^{-j\omega_{\text{out}}t} + \mathrm{d}b\frac{V_{inj}V_o^2}{8}\left(2e^{-j(\omega_{\text{inj}}t+\phi)} + e^{j(-2\omega_{\text{out}}t+\omega_{\text{inj}}t+\phi)}\right) + \frac{3CV_o^3}{8}e^{-j\omega_{\text{out}}t}\right] = \frac{V_o}{2}e^{-j\omega_{\text{out}}t} (5.65)$$

where $\frac{R_L H_o}{1+j(\frac{2Q\delta\omega}{\omega_0})}$ models the frequency selective block (an RLC tank in this analysis). Then $e^{j\omega_{\text{out}}t}$ is eliminated from both sides of Eq. (5.64)¹ and one gets:

$$\frac{R_L H_o}{1 + j(\frac{2Q\delta\omega}{\omega_0})} \left[\frac{AV_o}{2} + \mathrm{d}b\frac{V_{inj}V_{\mathrm{out}}^2}{4}e^{j\phi} + \mathrm{d}b\frac{V_{inj}V_{\mathrm{out}}^2}{8}e^{-j\phi} + \frac{3CV_{\mathrm{out}}^3}{8}\right] = \frac{V_o}{2} \tag{5.66}$$

Since elimination of the exponential terms on both sides of Eq. (5.64) and Eq. (5.65) gives the same result, in the following analysis, only the terms $e^{j\omega_{out}t}$ are considered.

 $e^{j\phi}$ is expanded and the top and bottom of the left-hand side of the equation are multiplied by the quantity $[1 - j2Q\frac{\delta\omega}{\omega_0}]$.

The resulting equations for the real and imaginary are as follows:

Real Part:

$$\left(\frac{R_L H_o}{1 + \left(\frac{2Q\delta\omega}{\omega_0}\right)^2}\right) \times \left[\frac{AV_o}{2} + \frac{3CV_o^3}{8} + \frac{\mathrm{d}bV_{inj}V_o^2}{8}\left(3\cos\phi + \sin\phi\left(\frac{2Q\delta\omega}{\omega_0}\right)\right)\right] = \frac{V_o}{2} \quad (5.67)$$

¹Note that in fundamental locking, $e^{j\omega_{\text{out}}t} = e^{j\omega_{\text{inj}}t}$

Imaginary Part:

$$\left(\frac{R_L H_o}{1 + (\frac{2Q\delta\omega}{\omega_0})^2}\right) \left(\frac{-2Q\delta\omega}{\omega_0}\right) \times \left[\frac{AV_o}{2} + \frac{3CV_o^3}{8} + \frac{3dbV_{inj}V_o^2}{8}\cos\phi\right] + \left(\frac{R_L H_o}{1 + (\frac{2Q\delta\omega}{\omega_0})^2}\right) \frac{dbV_{inj}V_o^2}{8}\sin\phi = 0 \quad (5.68)$$

The real and imaginary equations may be solved simultaneously to obtain the output voltage V_o and the locking range $\delta\omega$. Using the equation for the imaginary part, we can get the normalized locking range as follows:

$$\frac{2Q\delta\omega}{\omega_0} = \frac{\mathrm{d}bV_{inj}V_o\sin\phi}{4A + 3CV_o^2 + 3\mathrm{d}bV_{inj}V_o\cos\phi}$$
(5.69)

When $\phi = 90^{\circ}$, the maximum normalized locking range is:

$$\frac{2Q\delta\omega}{\omega_0} = \frac{dbV_{inj}V_o}{4A + 3CV_o^2} \tag{5.70}$$

From this equation, it can be seen clearly that the locking range can be enlarged by increasing V_{inj} and V_o .

Following a similar method, the locking range for other division ratios can be calculated as well. For example, if the division ratio is 2, $e^{j\omega_{inj}t} = e^{2j\omega_{out}t}$.

5.5 Novel Use of the Warped Multitime Partial Differential Equation (WaMPDE) to Determine the Locking Range

5.5.1 The Use of the WaMPDE for Modelling the Injection-Locked Oscillator

Consider a fairly general nonlinear circuit which is described by:

$$\dot{x}(t) = f(x(t)) + b(t)$$
 (5.71)

where b(t) is the excitation vector, x(t) are the state variables and f is a nonlinear function. Note that (5.71) follows from (2.1) except that the input signal b(t) is separated out.

The (p+1)-dimensional multivariate variables are defined as:

$$x(t) = \hat{x}(\tau_1, \tau_2, \dots, \tau_p, t)$$
 (5.72)

where the different time scales are assumed as the functions of real time:

$$\tau_1 = \tau_1(t)$$

$$\tau_2 = \tau_2(t)$$

$$\dots$$

$$\tau_p = \tau_p(t).$$

(5.73)

Differentiating x with respect to t, one gets:

$$\frac{dx}{dt} = \frac{d\hat{x}}{d\tau_1}\frac{d\tau_1}{dt} + \frac{d\hat{x}}{d\tau_2}\frac{d\tau_2}{dt} + \dots + \frac{d\hat{x}}{d\tau_p}\frac{d\tau_p}{dt} + \frac{d\hat{x}}{dt}.$$
(5.74)

Then the warped frequencies are introduced:

$$\omega_i = \frac{d\tau_i}{dt} \tag{5.75}$$

where $i \in \{1, 2, ..., p\}$.

Hence, (5.74) is rewrited to

$$\frac{dx}{dt} = \omega_1(t)\frac{d\hat{x}}{d\tau_1} + \omega_2(t)\frac{d\hat{x}}{d\tau_2} + \dots + \omega_p(t)\frac{d\hat{x}}{d\tau_p} + \frac{d\hat{x}}{dt}.$$
(5.76)

Then the (p + 1)-dimensional WaMPDE [11] corresponding to (5.71) is:

$$\sum_{i=1}^{p} \left(\omega_i(t) \frac{\partial \hat{x}}{\partial \tau_i} \right) + \frac{\partial \hat{x}}{\partial t} = f(\hat{x}) + \hat{b}(\tau_1, ..., \tau_p, t)$$
(5.77)

where $\tau_1, ..., \tau_p$ correspond to the warped time scales and t is the real time-scale. \hat{x} and \hat{b} are multivariate functions of the p + 1 time variables. Once the solution of (5.77) is found, (5.71) can be solved from (5.78) and (5.79).

Equation (5.72) can be represented in the form:

$$x(t) = \hat{x}(\phi_1(t), \phi_2(t), \dots, \phi_p(t), t)$$
(5.78)

where

$$\phi_i(t) = \int_0^t \omega_i(\tau_i) d\tau_i$$
(5.79)

The relationship between τ_1 , t and $\phi_1(t)$ for the case with p = 1 is shown in Fig. 5.23.



Figure 5.23: *Relation between* τ_1 , *t* and $\phi_1(t)$.

In the case of the ILFD, three dimensions are required. The equation becomes:

$$\omega_{\rm o}(t)\frac{\partial \hat{x}}{\partial \tau_1} + \omega_{\rm inj}\frac{\partial \hat{x}}{\partial \tau_2} + \frac{\partial \hat{x}}{\partial t} = f(\hat{x}) + b(t)$$
(5.80)

where τ_1 and τ_2 are the free-running oscillation time scale and the injected signal time scale, respectively, while t denotes the real time. Note that both τ_1 and τ_2 are warped time scales to enable the slow variation of the local and injected frequency. ω_0 is output frequency of the oscillator and ω_{inj} is the injected frequency.

The five-point centered difference formula [51] is used for the derivatives with respect to the warped time variables.

$$\frac{\partial \hat{x}_k}{\partial \tau_i} \approx \frac{x_{k-2} - 8x_{k-1} + 8x_{k+1} - x_{k+2}}{12h_i}$$
(5.81)

where h_i is the time-step in the τ_i scale and k denotes the index in the same time scale. The Backward Euler (BE) rule is used in the t time scale for the transient evolution. One useful characteristic of the BE rule is that it introduces artificial damping if the time step is large. This helps to attenuate undesired oscillations along the t dimension.

One phase condition is required here to obtain the local frequency. This is solved by making the derivative of one of the state variables equal to zero:

$$\frac{\partial \hat{x}_0}{\partial \tau_1} = 0. \tag{5.82}$$

The derivative of the solution has to be continuous for the state variable on which the phase condition is imposed. Then, the phase condition is guaranteed to be satisfied. When (5.81) is substituted into (5.80) and the BE rule is used, the resulting nonlinear algebraic equations are solved in conjunction with (5.82) using Newton's method [51] [52].

There are two methods introduced below, which utilise the WaMPDE to determine the locking range. One of them determines the limit of the locking range directly [53]. However, it has a relatively long simulation time. The second method [54] is proposed to improve this problem. It produces the Devil's staircase first, which will be introduced in Section 5.5.3. Then the locking range is measured from the staircase.

5.5.2 The Determination of the Limit of the Locking Range

The lower and upper limit of the locking range are determined separately. Here, the procedure to obtain the lower limit is described [53], and the upper limit can be determined with the similar method. To be specific, the work is concerned with the accurate determination of the **acquisition** locking range. To illustrate the important points of the method, results relating to a forced Van der Pol oscillator [53] are shown in Figs. 5.24-5.27.

The process to obtain the lower limit is started by running a transient simulation with the following initial condition: the free-running oscillation is set along τ_1 and no oscillations along τ_2 . The frequency of the excitation is set to a value known not to lock the oscillator. (5.80) along with the boundary conditions is then solved for several time-steps along t until the steady-state condition is achieved. Each step along t involves the calculation of a bi-dimensional steady-state problem with a grid with 15 points in each dimension, τ_1 and τ_2 . In order to minimize the number of time steps necessary to achieve the steady-state condition (along t), the time step size in this dimension is exponentially increased. When the difference between two steps along t is small enough, the analysis is switched from a 3-D transient analysis to a 2-D steady-state analysis by making the derivatives with respect to t equal to zero. Fig. 5.24 shows steady state solution when the oscillator is not locked. This procedure results in an autonomous solution along τ_1 , if it exists. At each time-step, but especially when solving for the steady-state condition, the Jacobian matrix that is employed in Newton's method is noted. If the column that corresponds to the local frequency is near zero, this implies the system is becoming independent of the local frequency because there are no more oscillations along τ_1 . Thus, an autonomous solution along τ_1 is no longer possible and only the forced oscillations are present. Locking has happened. Note that at this condition the Jacobian matrix is singular and so the Newton iterations are stopped. Thus, the locking condition can also be observed in Fig. 5.25. The norm of the Jacobian column approaches zero if the oscillator is locked. The steady state solution when the oscillator is locked is illustrated in Fig. 5.26.

After the steady-state solution is obtained (assuming no locking), the input frequency (ω_{in}) is progressively incremented. For each increment, a transient analysis is performed and followed by a steady-state analysis. If the oscillator is deemed locked, further analysis may be necessary to determine the starting point of the locking range to a certain degree of accuracy. The bisection method [51] is used in this work. There-



Figure 5.24: Steady state: Non-locked oscillator.



Figure 5.25: Comparison of the norm of the Jacobian column when the oscillator is locked (continuous line) and non-locked (dashed line).



Figure 5.26: Steady state: Locked oscillator.

fore, the frequency step by which ω_{in} is incremented is halved and the procedure is re-started with the initial conditions from the last frequency point at which there was no locking. This progressive halving of the frequency step increment is repeated until the lower limit of the locking range is attained to a certain tolerance.

Figure 5.27 shows the variation of the forcing frequency and the norm of the Jacobian column as a function of the iteration number. If no locking is detected, the frequency is increased. Each time that locking is detected, a lower frequency is tried and the initial conditions are reset to the last known oscillatory solution. This is the reason for the peaks observed in the norm of the Jacobian column.



Figure 5.27: Local frequency and norm of Jacobian column (dashed line) vs. iteration number.

5.5.3 Determination of the Locking Range from Use of the WaMPDE in Conjunction with the Devil's Staircase

The Devil's Staircase [9] is a method to visualize the locking range of an ILFD. Normally, the ILFD is considered as an oscillator with an injected external signal. In order to plot a Devil's Staircase, the frequencies of the oscillator and the injected signal must be varied relative to each other. In practice, it is easier and more accurate to adjust the injected frequency, ω_{inj} , automatically. The output frequency of the ILFD, ω_o , is then the only unknown variable. The Devil' Staircase [9] is obtained by plotting ω_{inj}/ω_o against ω_{inj} , as shown in Fig. 5.28.

The locking range can be measured from the Devil's Staircase diagram [9]. From the staircase diagram in Fig. 5.28, it is clear that there are lockings (flat regions) at division ratios of 2 and 4, as predicted experimentally in [7] for the LC-oscillator based ILFD.



Figure 5.28: Experimentally measured Devil's staircase diagram showing lockings at $\omega_{inj}/\omega_o = 2$ and $\omega_{inj}/\omega_o = 4$.

From the experimental results on an ILFD performed in [9] and as shown in Fig. 5.28, it is noted that the relationship between ω_{inj}/ω_o and ω_{inj} is approximately linear between the locking intervals [54] (the ILFD locks at multiples of its natural frequency - the *n*th locking range is described by $\omega_{inj}/\omega_o = n$). During the locking intervals, the slope is obviously zero. Consequently, two simulations are performed with two carefully selected input frequencies. The two selected ω_{inj} are known not to lock the ILFD and to be below the lower limit of the particular *n*th locking range. In other words, (5.80) is solved twice to obtain the values of ω_o corresponding to two values of the manually picked ω_{inj} . From this, an estimate of the start of the *n*th locking range can be obtained. For instance, the start of the divide-by-two locking range is when $\omega_{inj}/\omega_o = 2$. The slope of the line connecting the two points below the start is determined as:

$$m_{\rm below} = \frac{\left(\frac{\omega_{\rm inj2}}{\omega_{\rm o2}}\right) - \left(\frac{\omega_{\rm inj1}}{\omega_{\rm o1}}\right)}{\omega_{\rm inj2} - \omega_{\rm inj1}}.$$
(5.83)

The slope can be also calculated by

$$m_{\rm below} = \frac{2 - \left(\frac{\omega_{\rm inj1}}{\omega_{\rm o1}}\right)}{\omega_{\rm start} - \omega_{\rm inj1}}.$$
(5.84)

where ω_{start} is the lower limit of the locking interval when $\omega_{\text{inj}}/\omega_{\text{o}} = 2$. Hence, the lower limit of the locking range is

$$\omega_{\text{start}} = \omega_{\text{inj1}} + \frac{2 - \left(\frac{\omega_{\text{inj1}}}{\omega_{\text{o1}}}\right)}{m_{\text{below}}}$$
(5.85)

The upper limit of the locking range can be obtained with a similar procedure. In this case, the two input frequencies are selected to determine the slope between the divide by 2 and the divide by 4 locking ranges.

$$2\omega_n < \omega_{\rm inj1} < \omega_{\rm inj2} < 4\omega_n \tag{5.86}$$

where ω_n is the natural frequency of the oscillator.
5.6 A Novel Method Utilising the Multiple-Phase-Condition Envelope Following Method (MPCENV) to Determine the Locking Range

5.6.1 Background to the Envelope Following Method

The circuit solution is assumed to be composed of fast oscillations whose amplitude and frequency vary much more slowly than the oscillations themselves. Let the period of the fast oscillation be T. In the case of oscillators, this will vary slowly. Let T_{env} be the envelope time-step over which the response of the system can be extrapolated.



Figure 5.29: Forward-Euler-based envelope-following method.

The Forward-Euler-based envelope-following method is illustrated in Fig. 5.29. Firstly, an accurate transient simulation is performed from x_0 for one cycle of the fast oscillation T. This brings one to point x_1 . If there are (m-1)T cycles between x_1 and x_{end} , the following equation is obtained:

$$\frac{x_1 - x_S}{mT} = \frac{x_L - x_S}{T}$$
(5.87)

However, the Forward-Euler-based envelope-following method suffers from stability problems, when the period of the envelope T_{env} is too large. To overcome this, the Backward Euler method is preferred. This will be described in Section 5.6.2.



Figure 5.30: Backward-Euler-based envelope-following method.

5.6.2 Multiple-Phase-Condition Envelope Following method

Here, a novel transient envelope following method, MPCENV, is proposed to determine the output frequencies corresponding to different input frequencies [55]. Consider Fig. 5.30. Let x_0 and x_1 be the state at $t_0 = t_S + T_{env}$ and $t_1 = t_S + T_{env} + T$, respectively, where t_S is the ending time of the last envelope step. Using the implicit Backward-Euler method for stability purposes, the envelope following process is described by:

$$\frac{x_1 - x_0}{T} = \frac{x_0 - x_{\rm S}}{T_{\rm env}}$$
(5.88)

where $x_S = x(t_S)$ is known from a previous step, and x_1 is determined using the trapezoidal integration method from t_0 to t_1 . This means that x_1 depends on x_0 and T. Note that apart from the circuit variables, there are two extra unknowns, T and T_{env} , since the period of the oscillator is always changing, and T_{env} has to vary in order to remain equal to an integer number of periods T. To solve for the extra unknowns, two further equations are required [56]:

$$\begin{cases} \frac{dx_{0l}}{dt} = 0\\ \frac{dx_{1l}}{dt} = 0 \end{cases}$$
(5.89)

5.6 A Novel Method Utilising the Multiple-Phase-Condition Envelope Following Method (MPCENV) to Determine the Locking Range

where *l* denotes the l_{th} state variable. The two derivative-based phase conditions (5.89) ensure that x_{0l} and x_{1l} are the peaks or troughs of a fast cycle. In practice, value-based constraints are better for numerical handling of certain circuits such as the ILFD:

$$\begin{cases}
 x_{0l} = c \\
 x_{1l} = d
\end{cases}$$
(5.90)

where c and d are constants.

Equations (5.88) and (5.90) are reorganized as a matrix and solved using the Newton– Raphson method [57]:

$$F = \begin{bmatrix} f_1(x_0, T, T_{env}) \\ f_2(x_0, T, T_{env}) \\ f_3(x_0, T, T_{env}) \end{bmatrix}$$
$$= \begin{bmatrix} (x_1 - x_0)T_{env} - (x_0 - x_s)T \\ x_{0l} - c \\ x_{1l} - d \end{bmatrix}$$
(5.91)
$$= 0.$$

If the circuit has n state variables, this system consists of n + 2 equations with n + 2 unknowns.

The Jacobian matrix corresponding to (5.91) is given by:

$$J = \begin{bmatrix} \frac{df_{1}}{dx_{0}}, & \frac{df_{1}}{dT}, & \frac{df_{1}}{dT_{env}} \\ \frac{df_{2}}{dx_{0}}, & \frac{df_{2}}{dT}, & \frac{df_{2}}{dT_{env}} \\ \frac{df_{3}}{dx_{0}}, & \frac{df_{3}}{dT}, & \frac{df_{3}}{dT_{env}} \end{bmatrix}$$
$$= \begin{bmatrix} \frac{\partial x_{1}}{\partial x_{0}} T_{env} - (T_{env} + T)I_{n}, & \frac{\partial x_{1}}{\partial T} T_{env} - (x_{0} - x_{s}), & x_{1} - x_{0} \\ & I_{n}|_{l}, & 0, & 0 \\ & \frac{\partial x_{1}}{\partial x_{0}}|_{l}, & \frac{\partial x_{1}}{\partial T}|_{l}, & 0 \end{bmatrix}$$
(5.92)

where I_n is an identity matrix of size $n \times n$, $I_n|_l$, $(\partial x_1/\partial x_0)|_l$ and $(\partial x_1/\partial T)|_l$ are the *l*th row of I_n , $\partial x_1/\partial x_0$ and $\partial x_1/\partial T$, respectively.

5.6 A Novel Method Utilising the Multiple-Phase-Condition Envelope Following Method (MPCENV) to Determine the Locking Range

In this implementation, both $\partial x_1/\partial x_0$ and $\partial x_1/\partial T$ are derived using the trapezoidal integration method, as introduced in [58]. Set x_r to be the state at t_r , where $t_0 \leq t_{r-1} < t_r \leq t_1$. Then

$$\frac{dx_r}{dx_0} = \left(1 - \frac{h}{2}\frac{\partial f(x_r)}{\partial x_r}\right)^{-1} \left(1 + \frac{h}{2}\frac{\partial f(x_{r-1})}{\partial x_{r-1}}\right) \frac{dx_{r-1}}{dx_0}$$
(5.93)

where f(x) is the expression to represent the derivative of the circuit variables: $\dot{x} = f(x)$. The term dx_1/dx_0 can be found by repeatedly evaluating (5.93) from t_0 to t_1 with $dx_0/dx_0 = I$, where I is an $n \times 1$ matrix with all ones.

In a similar manner, dx_1/dT can be found by solving

$$\frac{dx_r}{dT} = \left(1 - \frac{h}{2}\frac{\partial f(x_r)}{\partial x_r}\right)^{-1} \times \left[\left(1 + \frac{h}{2}\frac{\partial f(x_{r-1})}{\partial x_{r-1}}\right)\frac{dx_{r-1}}{dT} + \frac{x_r - x_{r-1}}{T}\right]$$
(5.94)

starting from $dx_0/dT = 0$.

Then the system in (5.91) can be solved using the Newton-Raphson method [57]:

$$Z_{\rm new} = Z - J^{-1}F \tag{5.95}$$

where Z_{new} and Z represent the current and previous states of all the variables, i.e., $Z = \begin{bmatrix} x_0 & T & T_{\text{env}} \end{bmatrix}^T$. In the case of the ILFD, x_0 represents the capacitance voltage and the inductance current, i.e., $x_0 = \begin{bmatrix} V_C & I_L \end{bmatrix}^T$.

As described in [9], the Devil's Staircase is a plot of ω_{inj}/ω_0 against ω_{inj} . For simulation purposes, the injected frequency, ω_{inj} is increased from the minimum ω_{inj} with a fixed frequency step-size. ω_0 is then determined from the MPCENV solution as:

$$\omega_0 = \frac{2\pi}{T}.\tag{5.96}$$

5.7 Numerical Results and Experimental Validation

5.7.1 Experimental Equipments Setup

Here we describe our experimental technique [9] to measure the locking range of ILFD. In order to measure a Devil's Staircase, we must vary the frequencies of the oscillator and the injected signal relative to one another. In practice, it is easier and more accurate to vary the injected frequency automatically. Hence we fix the natural frequency of the ILFD, f_n , and adjust the frequency, f_{inj} , of the injected signal.



Figure 5.31: Experiment setup

An automated measurement system has been developed which includes a personal computer, a function generator, a frequency counter and the ILFD circuit [9]. The experimental setup is shown in Fig. 5.31. The driving voltage V_{inj} and the driving frequency ω_{inj} are produced by a precision frequency generator (Agilent 33220A) connected to an IEEE-488 bus. The frequency of the driven oscillator (ω_o) is obtained through the use of a precision counter (Agilent 53131A) which is also under IEEE-488 control.

5.7.2 Numerical Results

The LC oscillator-based ILFD (LC-ILFD) is selected as an example. It was introduced in Section 5.3.4. The schematic is shown again in Fig. 5.32.



Figure 5.32: The schematic of the LC-ILFD.

The governing equations are:

$$C\frac{dV_C}{dt} = I_L - (A + daV_{inj})V_C + \frac{A + daV_{inj}}{V_{DD}^2}V_C^3$$

$$L\frac{dV_C}{dt} = -I_L - V_C$$
(5.97)

where A and da are the coefficients obtained from the negative resistance characteristic [7] shown in Fig. 5.11 and Fig. 5.13. The value of the selected components¹ is shown in Table 5.1.

The Devil's Staircases obtained by simulation and experiment are shown in Fig. 5.33. The widths of the locking ranges agree when ω_{inj}/ω_o is an even number [7] i.e., 2 and 4, as shown in Fig. 5.28 and Fig. 5.33. The staircase from MPCENV is almost the

 $^{{}^{1}}V_{DD}$ is supplied by a dry battery.

 Table 5.1: The value of the selected components used in both experiment and simulation.

C	L	R_S	V_{DD}	A	da
100 pF	$220\mu H$	3.06Ω	9V	-2.5mA/V	$-0.1mA/V^{2}$



Figure 5.33: The staircase obtained from simulation and experiment.

same as that from experiment, while the one from use of the WaMPDE described in Section 5.5.3 has an appreciably bigger difference. However, it should be accepted that this is a very basic method. It could be used to obtain an initial estimate.

		$\omega_{\rm inj}/\omega_{\rm o}=2$	
Vinj	M (Mrad/s)	E (Mrad/s)	W (Mrad/s)
1 V	0.63	0.62	0.63
1.5 V	0.88	0.92	0.87
		$\omega_{\rm inj}/\omega_{\rm o}=4$	
	M (Mrad/s)	E (Mrad/s)	W (Mrad/s)
1 V	1.1	1.04	1.12
1.5 V	1.48	1.51	1.49

 Table 5.2: Locking range, where M, E and W represent MPCENV, Experiment and

 WaMPDE

Table 5.2 shows the locking ranges captured from the staircases. The difference between the MPCENV method and experimental results is less than 6%, while that between the full WaMPDE method described in Section 5.5.2 and experimental results is almost 8%. Therefore, it is sufficiently accurate to predict the locking range when designing ILFDs of this type. The locking range captured from experiments is illustrated in Fig. 5.34. As seen from it, if the amplitude of the injected signal is larger, then the locking range is larger.



Figure 5.34: The locking range of the LC-ILFD captured from experiments.

5.8 Conclusions

Injection-locked frequency dividers (ILFDs) are normally used as the prescaler in the feedback of a Phase-Locked-Loop(PLL). In comparison with the conventional static frequency divider, they consumes less power, but their locking range is smaller. Hence, its accurate determination is essential in design work. The WaMPDE method is used to determine the acquisition range for locking. However, it requires quite a long simulation time when implemented in full. An improved method is proposed to reduce the simulation time. However, it is less accurate. The MPCENV method is another suggested method. It follows the envelope following technique. It is used to enable a simple plotting of the Devil's Staircase from which the locking range can be determined. With this method, the simulation speed is increased and the error is reduced to 6%.

Chapter 6

Analysis and Design of Digital Delta-Sigma Modulators (DDSM)

6.1 Background to Delta-Sigma Modulation

The delta-sigma modulation (DSM) technique has become popular in very-large-scale integration (VLSI) circuits in recent years, although it was first introduced in 1962 [59]. Its main use is in data converters and in frequency synthesis. Examples include analog-to-digital converters (ADC), digital-to-analog converters (DAC), direct digital frequency synthesizers (DDFS) and fractional-N frequency synthesizers.

Delta-sigma modulation is based on delta modulation (DM), which is the simplest form of the differential pulse-code modulation (DPCM). A simple example of pulse-code modulation (PCM) is shown in Fig. 6.1: As seen from Fig. 6.1, a signal, represented by the sine wave, is sampled and quantized at fixed intervals. In other words, the signal is discrete and can be expressed by digital words. For example, the quantized values of it are 7, 9, 11, 12, 13, 14, 14, 15, 15, 15, 14,... These may then be expressed in binary format as 0111, 1001, 1011, 1100, 1101, 1110, 1111, 1111, 1111, 1110,..., which can be used for digital signal processing. DPCM quantizes the difference between successional samples of a continuous-time signal into an n-bit binary word. Delta-modulation is 1-bit DPCM.



Figure 6.1: An example of 4-bit PCM.



Figure 6.2: Block diagram of the delta-modulation process.

The block diagram of the delta-modulation process is illustrated in Fig. 6.2. The demodulation part is the inverse procedure of the delta-modulation. It converts the digital data stream to a smooth analog signal.

Integration is a linear operation and hence has the following property

$$\int a + \int b = \int (a+b) \tag{6.1}$$

Consequently, the integrator in the demodulator can be moved to the position between the input analog signal and the modulator, as shown in Fig. 6.3. Furthermore, the two integrators in Fig. 6.3 (b) can be merged into one integrator as shown in Fig. 6.4. This resultant system is the so-called delta-sigma modulator [59].



Figure 6.3: Derivation of Delta-Sigma Modulation from Delta Modulation.



Figure 6.4: Block diagram of delta-sigma modulation.



Figure 6.5: Spectrum of a signal.



Figure 6.6: Spectrum of an oversampled signal.

Oversampling is one of the most popular techniques used in delta-sigma modulation. The most attractive benefit of oversampling is that it reduces noise. Figure 6.5 illustrates the concept. f_m and f_s represent the bandwidth and the sampling frequency, respectively. In signal processing, if the sampling frequency is higher than twice of the bandwidth, i.e., $f_s \ge 2f_m$, this process is termed oversampling. If the sampling frequency is N times of the bandwidth, the noise power in the useful band is reduced to 1/N. That means the signal-noise ratio (SNR) is increased by a factor of N. The technique of moving the noise power to higher frequencies is termed noise shaping. The quantization noise is pushed to higher frequencies out of the useful band in a delta-sigma modulator. Consequently, it can be simply filtered by a low-pass filter.

6.2 Introduction of Digital Delta-Sigma Modulators

Digital delta-sigma modulators (DDSM) perform delta-sigma modulation on a digital input and produce a digital output. They take an n_0 -bit input signal and produce an m-bit output signal. This results in a quantization error that is often modelled as an addition of white noise. The modulator filters the spectrum such that its noise power is concentrated at higher frequencies away from the signal band. They are widely used in consumer entertainment and communication equipment such as Fractional-N Frequency Synthesizers, cellular telephones and MP3 players. Fractional-N Frequency Synthesizers [12] [13] in general, are employed as Local Oscillators (LO) to generate exactly the required frequencies. A Fractional-N Frequency Synthesizer is shown in Fig. 6.7. It consists of a phase/frequency detector, a charge pump, a loop filter and a voltage controlled oscillator in the forward path. The feedback path includes an injection-locked frequency divider (ILFD) [7] [8] [53] [54] [55] which is chosen to be the pre-scaler due to its low power consumption. The static frequency divider is controlled by a fully-digital delta-sigma modulator (DDSM). It ensures that the division ratio is the required fractional value. The input to the DDSM is a constant digital word

X. This is a high resolution value and it sets the fractional division ratio. The output is a low resolution sequence which controls the static divider. The spectrum of the output contains the desired input DC tone and the undesired quantization noise.



Figure 6.7: The Fractional-N Frequency Synthesizer.

Delta-sigma modulators (DSM) are attractive as they push most of the quantisation noise to higher frequencies and hence out of the useful frequency band, which is so-called noise shaping as introduced in Section 6.1. Thus the quantisation noise power, obtained from the DDSM appears at frequencies out of the pass-band of the fractional-N frequency synthesizer and can simply be filtered without affecting the signals in the frequency synthesizer.

DDSMs fall into two categories [60]: single-loop delta-sigma modulators and Multi-stAge noise SHaping (MASH) delta-sigma modulators. The architecture of an *l*th order single-loop delta-sigma modulator is shown in Fig. 6.8. It consists of *l* stages of 1st order modulators (DSM1) introduced in Section 6.1. x[n] and y[n] represent the n_0 -bit input and *m*-bit output binary word, respectively. The input of the *i*th stage of DSM1 is the output of the (i-1)th DSM1 where $i \in \{2, 3, ..., l\}$., The input of the 1st stage is the difference between x[n] and y[n]. The sum of the output of the *l* DSM1s is quantized by a quantizer to produce the output of the complete modulator. The ad-



Figure 6.8: The architecture of an lth order single-loop delta-sigma modulator.

vantage of this arrangement is that it has the choice of either a single-bit or a multi-bit output, i.e., m = 1 or $m \ge 2$. The components receiving only a single-bit data stream require less hardware than those receiving multi-bit data. However, the disadvantage of the single-loop DSM is that it is prone to instability because of the feedback loop inherent within this architecture [60]. In contrast, the MASH DDSM has has no *n*th order feedback and 1st order stages have been proven to be stable so that it is unconditionally stable [60]. However, now there is one quantizer in the DSM1 of each stage instead of one quantizer after the last DSM1. This structure results in a multi-bit output y[n], i.e., $m \ge 2$. The architecture and characteristics of MASH DDSMs will be described in detail later in this Chapter.

The DDSM is a finite state machine. Hence, when the input is a constant, the output is always a repeating pattern (a cycle) [14] [61]. In particular, the quantization noise is a periodic sequence. The period of the cycle is termed the sequence length. Its length depends on the input, the initial conditions of the DDSM registers and the DDSM architecture. Short sequence lengths result in unwanted frequency components or spurs from quantisation noise in the output frequency spectrum [62]. Hence, much research has been done into detecting and removing the occurrence of a short sequence length [63].



Figure 6.9: Model of a dithered MASH DDSM.

There are two classes of technique for whitening quantization noise, stochastic and deterministic. The dithering [64] [65] method is one of the most common stochastic approaches employed. It uses a random dither sequence to disrupt the periodic cycle and thereby effectively increase the sequence length. However, it requires extra hardware and inherently introduces additional noise in the the useful frequency band. The structure of a dithered MASH DDSM is illustrated in Fig. 6.9. d[n] is an 1-bit pseudo-random dither sequence. The probability of d[n] = 0 and d[n] = 1 is set manually to be equal.

$$P(d[n] = 0) = P(d[n] = 1) = 50\%.$$
(6.2)

The block F(z) represents a filter after the dither input. The power spectral density of the output sequence of a 9-bit MASH 1-1-1¹ DDSM is shown in Fig. 6.10 for three cases:

- 1. F(z) = 0, no dither
- 2. F(z) = 1, normal dither
- 3. $F(z) = (1 z^{-1})$, shaped dither

¹MASH m - n - l means that it consists of an *m*th order error feedback modulator (EFM*m*), an *n*th order EFM (EFM*n*) and an *l*th order EFM (EFM*l*), cascaded.



Figure 6.10: The power spectral density of a 9-bit MASH 1-1-1 DDSM [3].

As seen in Fig. 6.10, the spectrum from the MASH DDSM without dither has strong tonal behaviour, while that from the dithered MASH DDSM is much smoother, because the dither lengthens the sequence length. The filter F(z) is used to improve the noise performance at low frequencies.

Recently, some deterministic design methodologies have been proposed to maximise the sequence length. Borkowski [14] gives a guaranteed minimum and maximum period obtained by setting the initial condition of the registers. Hosseini [3] introduced a digital delta-sigma modulator structure termed the HK-MASH with a very long sequence and the period of such a sequence is derived by mathematical analysis [66] [67] [68]. They will be described in Section 6.3. A novel design methodology for the MASH DDSM employing multi-moduli [15] will be proposed in this Chapter. The new structure is termed the MM-MASH. It can produce a longer sequence length than that of HK-MASH.

6.3 State-of-the-art Architectures of DDSMs

6.3.1 The Conventional DDSMs

There are two classes of structure for digital delta-sigma modulators: Multi-stAge noise SHaping (MASH) and error feedback modulator (EFM). Normally, a MASH modulator is comprised of several EFMs. The conventional structures of both of them are introduced in this Section.



Figure 6.11: MASH DDSM architecture.

6.3.1.1 The Conventional MASH

The architecture of an *l*th order MASH digital delta-sigma modulator (DDSM) is illustrated in Fig. 6.11. It contains *l* first-order error-feedback modulators (EFM1). x[n] and y[n] are an n_0 -bit input digital word and an *m*-bit output, respectively. The relationship between them is

$$mean(y) = \frac{X}{M} \tag{6.3}$$

where X is the decimal number corresponding to the digital sequence x[n] [69], i.e., $x[n] = X \in \{1, 2, ..., M - 1\}$, and M is the quantizer modulus which is set as 2^{n_0} in the conventional DDSM. mean(y) is the desired fraction in a fractional-N frequency synthesizer.

The model of the EFM1 is shown in Fig. 6.12. This is a core component in the make-up of the MASH digital delta-sigma modulator (DDSM). The rectangle Z^{-1} represents the register which stores the error e[n] and delays it for one time sample. $Q(\cdot)$ is the quantization function:

$$y[n] = Q(u[n]) = \begin{cases} 1, & u[n] \ge M \\ 0, & u[n] < M \end{cases}$$
(6.4)

where

$$u[n] = x[n] + e[n-1].$$
(6.5)

In order to simplify the analysis, a linear model of the EFM1 is introduced here. As seen in Fig. 6.13, an additive quantization noise source $e_{as}[n]$ is used to take the place of the quantizer. The expression for the linear model can be written in z-domain as below:

$$Y(z) = U(z) + E_{as}(z) \tag{6.6}$$

$$U(z) = X(z) + E(z)z^{-1}.$$
(6.7)

Substituting (6.7) into (6.6),

$$Y(z) = X(z) + E(z)z^{-1} + E_{as}(z).$$
(6.8)

Consider the expression for the error e[n] in the z-domain:

$$E(z) = U(z) - Y(z).$$
 (6.9)

By merging (6.6) and (6.9), it follows that

$$E(z) = -E_{as}(z).$$
 (6.10)



Figure 6.12: EFM1: First-order error-feedback modulator.



Figure 6.13: *The linear model of an EFM1.*

Combining (6.10) and (6.8),

$$Y(z) = X(z) + (1 - z^{-1})E_{as}(z).$$
(6.11)

Thus it can be concluded that the EFM1 output consists of the input sequence and the output quantization error. The output quantization error is the quantizer error $E_{as}(z)$ which is shaped by a filter $(1 - z^{-1})$. From (6.11), it follows that for the conventional EFM1, the signal transfer function (STF) and the noise transfer function (NTF) are:

$$STF = 1 \tag{6.12}$$

$$NTF = 1 - z^{-1}. (6.13)$$

They represent a measure of the output signal versus the input signal and versus the noise in a system, respectively.

In a MASH DDSM, there is a cascade of several EFM1, as shown in Fig. 6.11. The error of each stage $(e_i[n])$ is fed to the next stage and the carry-out of each stage $(y_i[n])$ is fed to the noise cancellation network to eliminate the intermediate quantization noises [3]. Thus the spectrum of the final output depends only on the input signal and the shaped quantization noise power of the last stage.

$$Y(Z) = X(z) + (1 - z^{-1})^{l} E_{asl}(z)$$
(6.14)

where $E_{asl}(z)$ is the quantization error of the last stage. Then the signal transfer function and the noise transfer function for a conventional MASH are:

$$STF = 1 \tag{6.15}$$

$$NTF = (1 - z^{-1})^l. (6.16)$$

6.3.1.2 The Conventional EFM

The structure of an *l*th order EFM (EFM*l*) is illustrated in Fig. 6.14. The symbols $R_1, R_2, ..., R_l$ and $A_1, A_2, ..., A_l$ represent the initial conditions and gains of the *i*th registers, respectively, where $i \in \{1, 2, ..., l\}$. The expressions governing the linear model of it as shown in Fig. 6.15 are

$$Y(z) = U(z) + E_{as}(z)$$
 (6.17)

$$U(z) = X(z) + \sum_{i=1}^{i} A_i z^{-i} E(z)$$
(6.18)

$$E(z) = U(z) - Y(z).$$
 (6.19)

From the equations above, it is obtained

$$Y(z) = X(z) + (1 - \sum_{i=1}^{l} A_i z^{-i}) E_{as}(z)$$
(6.20)

where

$$E_{as}(z) = -E(z).$$
 (6.21)

Thus the signal transfer function and the noise transfer function for an lth order EFM are

$$STF_{EFM} = 1 \tag{6.22}$$

$$NTF_{EFM} = 1 - \sum_{i=1}^{l} A_i z^{-i}.$$
(6.23)

Since the higher-order EFMl is used as the delta-sigma modulator, it should have the same noise transfer function as a MASH composed of a cascade of first-order EFMs given by (6.16). Then

$$1 - \sum_{i=1}^{l} A_i z^{-i} = (1 - z^{-1})^l.$$
(6.24)

Thus the value of the gains A_i is obtained from:

$$\sum_{i=1}^{l} A_i z^{-i} = 1 - (1 - z^{-1})^l.$$
(6.25)



Figure 6.14: *Higher-order error-feedback modulator.*



Figure 6.15: Linear model of higher-order error-feedback modulator.

6.3.1.3 The Sequence Length of Conventional DDSMs

The sequence length of the two types of conventional DDSMs, the MASH and the EFM, is controlled by the initial conditions of the registers, which are represented by the symbols $\{R_1, R_2, ..., R_l\}$ shown in Fig. 6.11 and Fig. 6.14. Borkowski [14] proposed the so-called preferred initial conditions, which are shown in Table 6.1. If the DDSMs work with these initial conditions, the guaranteed minimum sequence length is dependent only on the modulator bus width n_0 and is not affected by the DC input any more. The guaranteed minimum and maximum sequence lengths for these structures have been found from simulations [14], and are as shown in Table 6.2. The modulator period varies between the guaranteed minimum and the maximum sequence length dependent on the value of the input. Note that the sequence length of an *l*th order MASH and an *l*th order EFM is the same. It is found that the conventional MASH modulator period will always be approximately equal to the maximum sequence length if the quantizer modulus, M, is set as a prime number [66] [67] [68].

Modulator Order	MASH	EFMl
2	R_1 odd	$R_1 + R_2$ odd
3	R_1 odd	$R_1 + R_3 \text{ odd}$
4	R_1 odd	$R_1 + R_2 + R_3 + R_4$ odd
5	R_1 odd	$R_1 + R_5$ odd

Table 6.1: The preferred initial conditions for the conventional DDSMs.

Table 6.2: *The sequence length of the conventional DDSMs.*

Modulator Order	Guaranteed Minimum Period	Maximum Period
2	2^{n_0-1}	2^{n_0+1}
3	2^{n_0+1}	2^{n_0+1}
4	2^{n_0+1}	2^{n_0+2}
5	2^{n_0+2}	2^{n_0+2}

6.3.2 HK-MASH DDSMs

The HK-MASH DDSM is the current MASH that produces the longest sequence length. It has two advantages. First, it produces a much longer sequence length when compared to the conventional MASH. Secondly, its maximum sequence length can be achieved independent of the initial conditions described in Section 6.3.1 and independent of all constant inputs.



Figure 6.16: The modified EFM1 used in the HK-MASH.

The architecture of the modified EFM1 used in the HK-MASH (HK-EFM1) is illustrated in Fig. 6.16. The only difference between it and the conventional EFM1 in Fig. 6.12 is the presence of the feedback block az^{-1} . The effect of this block is that, unlike the conventional DDSM, the quantizer is not reset to 0 again, but to a, when it overflows. Equivalently, the effective quantizer mudulus is changed to M - a from M. a is a specifically-chosen small integer to make (M - a) the maximum prime number below 2^{n_0} [3]. The value of a chosen in [3] is shown in Table 6.3.

The sequence length of an *l*th order HK-MASH is determined by the following formula [3]:

$$N = (2^{n_0} - a)^l. ag{6.26}$$

n_0	a
5, 7, 13, 17, 19	1
6, 9, 10, 12, 14, 20, 22, 24	3
8, 18,25	5
11, 21	9
16, 23	15
15	19

Table 6.3: *The specifically-chosen a for various modulator word lengths.*

The maximum sequence length of an *l*th order HK-MASH is compared with that of the conventional MASH DDSM in Table 6.4. The maximum sequence length of the HK-MASH is almost $(2^{n_0})^{l-1}$ times that of the conventional MASH DDSM. Note that, since the input does not affect the period of HK-MASH output, the guaranteed minimum sequence length of it is the same as its maximum sequence length.

MASH and the HK-MASH. Modulator Order *l* Conventional MASH HK-MASH

Table 6.4: A comparison between the maximum sequence length of the conventional

2	2^{n_0+1}	$(2^{n_0} - a)^2 \approx (2^{n_0})^2$
3	2^{n_0+1}	$(2^{n_0} - a)^3 \approx (2^{n_0})^3$
4	2^{n_0+2}	$(2^{n_0} - a)^4 \approx (2^{n_0})^4$
5	2^{n_0+2}	$(2^{n_0} - a)^5 \approx (2^{n_0})^5$

The linear model of the HK-EFM1 is illustrated in Fig. 6.17. As with the linear model of conventional EFM1, the quantizer in the HK-EFM1 is replaced by an additive noise source $(e_{as}[n])$. Since all of the signals shown in Fig. 6.17 are normalized to 1 in order to do the linear analysis, the coefficient a is modified to α :

$$\alpha = \frac{a}{2^{n_0}}.\tag{6.27}$$

The expression for the output y[n] in terms of the input x[n] and the quantizer error



Figure 6.17: The linear model of the HK-EFM1.

 $e_{as}[n]$ in the z-domain is:

$$Y(z) = U(z) + E_{as}(z)$$
(6.28)

$$E(z) = U(z) - Y(Z).$$
 (6.29)

Thus it follows:

$$E(z) = -E_{as}(z).$$
 (6.30)

After summing the input and the feedback,

$$U(z) = X(z) + \alpha z^{-1} Y(z) + z^{-1} E(z).$$
(6.31)

Substituting (6.30) into (6.31),

$$U(z) = X(z) + \alpha z^{-1} Y(z) - z^{-1} E_{as}(z).$$
(6.32)

From (6.28), it follows

$$U(z) = Y(z) - E_{as}(z).$$
 (6.33)

Substituting it into (6.32),

$$Y(z) - E_{as}(z) = X(z) + \alpha z^{-1} Y(z) - z^{-1} E_{as}(z).$$
(6.34)

Then (6.32) can be modified to

$$Y(z) = \frac{X(z)}{1 - \alpha z^{-1}} + \frac{1 - z^{-1}}{1 - \alpha z^{-1}} E_{as}(z).$$
(6.35)

Therefore, the signal transfer function (STF) and the noise transfer function (NTF) of the HK-EFM1 are

$$STF_{HK} = \frac{1}{1 - \alpha z^{-1}}$$
 (6.36)

$$NTF_{HK} = \frac{1 - z^{-1}}{1 - \alpha z^{-1}}.$$
(6.37)

In comparison with the STF and NTF of the conventional EFM1, (6.15) and (6.16), there is a pole at $z = \alpha$. However, the value of α is such that the pole is close to the origin in the z-plane and hence does not affect the in-band operation of the modulator to any significant level.



Figure 6.18: The structure of the HK-MASH.

The structure of the HK-MASH is shown in Fig. 6.18. The difference between the HK-MASH and the conventional MASH DDSM is that the HK-MASH utilises the HK-EFM1 introduced above instead of the conventional EFM1. Using (6.35), the expression for the first stage of HK-EFM1 in a HK-MASH is obtained:

$$Y_1(z) = \frac{X(z)}{1 - \alpha z^{-1}} + \frac{1 - z^{-1}}{1 - \alpha z^{-1}} E_{as1}(z).$$
(6.38)

Combining (6.30), (6.38) is modified to

$$Y_1(z) = \frac{X(z)}{1 - \alpha z^{-1}} - \frac{1 - z^{-1}}{1 - \alpha z^{-1}} E_1(z).$$
(6.39)

Similarly, for the other stages of HK-EFM1 are expressed as:

$$Y_2(z) = \frac{E_1(z)}{1 - \alpha z^{-1}} - \frac{1 - z^{-1}}{1 - \alpha z^{-1}} E_2(z)$$
(6.40)

$$\frac{\vdots}{Y_l(z)} = \frac{E_{l-1}(z)}{1 - \alpha z^{-1}} - \frac{1 - z^{-1}}{1 - \alpha z^{-1}} E_l(z).$$
(6.41)

From Fig. 6.18, it is derived that

$$Y(z) = Y_1(z) + Y_2(z)(1 - z^{-1}) + \dots + Y_l(z)(1 - z^{-1})^{l-1}.$$
 (6.42)

Substituting (6.39)–(6.41) into (6.42),

$$Y(z) = \frac{X(z)}{1 - \alpha z^{-1}} - \frac{(1 - z^{-1})^l}{1 - \alpha z^{-1}} E_l(z).$$
(6.43)

Alternatively,

$$Y(z) = \frac{X(z)}{1 - \alpha z^{-1}} + \frac{(1 - z^{-1})^l}{1 - \alpha z^{-1}} E_{asl}(z).$$
(6.44)

6.4 Mathematical Analysis of the MASH DDSMs employing Multi-Moduli

A novel structure for the MASH DDSMs which employ Multi-Moduli (MM-MASH) is proposed in this section. The advantage of this structure will be that it increases the modulator period. It is proposed that the modulus of each quantizer is set as a different value from each other. Note that each quantizer has only one modulus. Furthermore, all of the moduli are co-prime numbers. The difference between a prime number and co-prime numbers [70] is stated as follows:

1. A prime number is a natural number which has exactly two divisors: 1 and itself.

2. If the greatest common divisor of any two numbers is 1, they are co-prime numbers. They do NOT have to be prime numbers, for example 8 and 9 are co-prime but 8 or 9 are not prime numbers.



Figure 6.19: MM-EFM1: The modified first-order error-feedback modulator used in the MM-MASH.

The structure of the modified EFM1 used in the MM-MASH (MM-EFM1) is shown in Fig. 6.19. The key difference between the proposed MM-MASH and the existing structures described in Section 6.3 is that the quantizer modulus of each EFM1 is different from each other. The quantizer modulus of the *i*th stage EFM1 in an *l*th order MASH-DDSM is denoted by M_i , where $i \in \{1, 2, ..., l\}$. It shall first be shown that the MM-MASH is an accurate modulator and that the use of a different modulus for each stage has not affected its output. Then the effect of the multi-moduli on the sequence length shall be investigated mathematically. In Section 6.4.1 and 6.4.2, a MM-MASH consisting of only first-order EFMs (MM-EFM1s) is mathematically analysed. The use of higher-order EFMs in the MM-MASH (MM-EFM) will be investigated in Section 6.4.3.

6.4.1 The Suitability of the Multi-Modulus MASH-DDSM

In a fractional-N frequency synthesizer, the static frequency divider, shown in Fig. 6.7, is controlled by the average value of the delta-sigma modulator output, mean(y). For example, if the static frequency divider divides the output frequency of the frequency synthesizer by the factor 7.89, the average delta-sigma modulator output should be 0.89, i.e., mean(y) = 0.89. The goal of this section is to show that in an MM-MASH, mean(y) is affected only by the quantizer modulus of the first stage EFM1, M_1 , and is independent of the moduli in other stages. With this being true, having a multi-modulus architecture does not affect the accuracy of the digital delta-sigma modulator. Hence, it is a suitable digital delta-sigma modulator. So it is required to prove:

$$mean(y) = \frac{X}{M_1}.$$
(6.45)

Proof. The structure of MM-MASH consists of only MM-EFM1 is shown in Fig. 6.20. As seen in Fig. 6.20, at the output of the last adder,

$$v_{l-1}[1] = y_{l-1}[1] + y_l[1] - y_l[0]$$
(6.46)

$$v_{l-1}[2] = y_{l-1}[2] + y_l[2] - y_l[1]$$
(6.47)

$$v_{l-1}[N] = y_{l-1}[N] + y_l[N] - y_l[N-1]$$
(6.48)

where N is assumed as the sequence length of the complete MASH delta-sigma modulator. Adding all of the above equations yields:

:

$$\sum_{k=1}^{N} v_{l-1}[k] = \sum_{k=1}^{N} y_{l-1}[k] + \sum_{k=1}^{N} y_{l}[k] - \sum_{k=0}^{N-1} y_{l}[k]$$
(6.49)

where the period of y_l is assumed as N_l . As seen from Fig. 6.20, the output of the MASH modulator is obtained by simply summing and/or subtracting the output of each EFM1. Hence, the period of the MASH DDSM is the least common multiple of



Figure 6.20: The MM-MASH consists of only MM-EFM1.

the sequence length of each stage. In other words, N is a multiple of N_i , where N_i is the period of the *i*th stage EFM1 and $i \in \{1, 2, ..., l\}$. It follows that

:

$$\sum_{k=1}^{N} y_l[k] = \sum_{k=0}^{N-1} y_l[k].$$
(6.50)

Thus (6.49) becomes:

$$\sum_{k=1}^{N} v_{l-1}[k] = \sum_{k=1}^{N} y_{l-1}[k].$$
(6.51)

Similarly, each output of the other adders is obtained as:

$$\sum_{k=1}^{N} v_{l-2}[k] = \sum_{k=1}^{N} y_{l-2}[k]$$
(6.52)

$$\sum_{k=1}^{N} v_2[k] = \sum_{k=1}^{N} y_2[k]$$
(6.53)

$$\sum_{k=1}^{N} y[k] = \sum_{k=1}^{N} y_1[k].$$
(6.54)

Each side of (6.54) may be expressed as:

$$\sum_{k=1}^{N} y[k] = N \cdot mean(y) \tag{6.55}$$

$$\sum_{k=1}^{N} y_1[k] = K \sum_{k=1}^{N_1} y_1[k]$$
(6.56)

where N_1 is the sequence length of y_1 , K is an integer and $N = K \cdot N_1$. Since y_1 is the output of a first-order delta-sigma modulator EFM1,

$$\sum_{k=1}^{N_1} y_1[k] = N_1 \cdot mean(y_1)$$

= $N_1 \cdot \frac{X}{M_1}$. (6.57)

On substitution of (6.57) into (6.56), the right-hand side of (6.54) becomes

$$\sum_{k=1}^{N} y_1[k] = K \cdot N_1 \cdot \frac{X}{M_1}$$
$$= N \cdot \frac{X}{M_1}.$$
(6.58)

By substituting (6.55) and (6.58) into (6.54), the average value of the MASH DDSM output y is determined as:

$$mean(y) = \frac{X}{M_1}.$$
(6.59)

6.4.2 The Effect of the Multi-Moduli on the Modulator Sequence Length

It is required to prove that the sequence length of the MASH modulator depends on the product of all the quantizer moduli. The expression for the *l*th order MASH DDSM sequence length is:

$$N = \frac{M_1 \cdot M_2 \cdot \ldots \cdot M_l}{\lambda} \tag{6.60}$$

where λ is a parameter to make N the least common multiple of the sequence length of each stage N_i .

In addition, if the following two conditions are satisfied:

Cond1: X and M_1 are co-prime numbers

Cond2: $\{M_1, M_2, ..., M_l\}$ are different co-prime numbers

then the sequence length of the MASH DDSM attains the maximum value:

$$N_{max} = M_1 \cdot M_2 \cdot \ldots \cdot M_l. \tag{6.61}$$

Proof. In the first-stage EFM1 shown in Fig. 6.12,

$$e_{1}[1] = u[1] - y_{1}[1]M_{1}$$

= X + e_{1}[0] - y_{1}[1]M_{1} (6.62)

$$e_1[2] = X + e_1[1] - y_1[2]M_1$$
(6.63)
$$\vdots$$

$$e_1[N_1] = X + e_1[N_1 - 1] - y_1[N_1]M_1$$
(6.64)

where $e_1[0]$ is the initial condition of the register. The sum of all of the above equations is:

$$\sum_{k=1}^{N_1} e_1[k] = N_1 X + \sum_{k=0}^{N_1-1} e_1[k] - \sum_{k=1}^{N_1} y_1[k] M_1.$$
(6.65)

Since in the steady state, the first EFM1 is periodic with a period N_1 [67],

$$\sum_{k=1}^{N_1} e_1[k] = \sum_{k=0}^{N_1 - 1} e_1[k].$$
(6.66)

Hence, (6.65) may be modified to

$$\sum_{k=1}^{N_1} y_1[k] = \frac{N_1 X}{M_1} \tag{6.67}$$

In practice, the input DC X is set as $0 < X < M_1$. So in order to make the right side of (6.67) an integer, the minimum nonzero solution of N_1 has to be:

$$N_1 = \frac{M_1}{\lambda_1} \tag{6.68}$$

where λ_1 is the greatest common divisor of M_1 and X. If M_1 and X are co-prime numbers, λ_1 equals to 1.

If the process of (6.62)–(6.67) is repeated with the second EFM1, the sum of its output, which has a period N_2 , is obtained as:

$$\sum_{k=1}^{N_2} y_2[k] = \frac{\sum_{k=1}^{N_2} e_1[k]}{M_2}.$$
(6.69)

If the relationship between the sequence length of the first and second stages is

$$N_2 = K_1 N_1 \tag{6.70}$$

(6.69) becomes

$$\sum_{k=1}^{N_2} y_2[k] = \frac{\sum_{k=1}^{K_1 N_1} e_1[k]}{M_2}.$$
(6.71)

Since e_1 is periodic with a sequence length N_1 [3],

$$\sum_{k=1}^{N_2} y_2[k] = \frac{K_1 \sum_{k=1}^{N_1} e_1[k]}{M_2}$$
(6.72)

where

$$\sum_{k=1}^{N_1} e_1[k] = N_1 \cdot mean(e_1).$$
(6.73)

Recalling (6.68),

$$\sum_{k=1}^{N_1} e_1[k] = \frac{M_1 \cdot mean(e_1)}{\lambda_1}.$$
(6.74)
On substitution of (6.74) into (6.72), the following expression is obtained:

$$\sum_{k=1}^{N_2} y_2[k] = \frac{K_1 \cdot M_1 \cdot mean(e_1)}{\lambda_1 \cdot M_2}$$
(6.75)

Normally, $mean(e_1)$ is a decimal fraction. However, if both sides of (6.74) are multiplied by λ_1 , the result is:

$$\lambda_1 \sum_{k=1}^{N_1} e_1[k] = M_1 \cdot mean(e_1).$$
(6.76)

Since $\lambda_1 \sum_{k=1}^{N_1} e_1[k]$ has to be an integer, $M_1 \cdot mean(e_1)$ is always an integer.

Then the minimum solution of K_1 so that the right-hand-side of (6.75) is an integer is obtained as:

$$K_1 = \frac{\lambda_1 M_2}{\lambda_2} \tag{6.77}$$

where λ_2 is the greatest common divisor of $\lambda_1 M_2$ and $M_1 mean(e_1)$. Substituting (6.68) and (6.77) into (6.70), the sequence length of the second stage is:

$$N_2 = \frac{M_1 M_2}{\lambda_2}.\tag{6.78}$$

If M_1 and M_2 are co-prime numbers, the greatest common divisor of $\lambda_1 M_2$ and $M_1 mean(e_1)$ is λ_1 , i.e., $\lambda_2 = \lambda_1$. Hence,

$$N_2 = \frac{M_1 M_2}{\lambda_1}.$$
 (6.79)

When X and M_1 are also co-prime numbers, λ_1 equals to 1. Thus the maximum sequence length for y_2 is obtained as:

$$N_{2_max} = M_1 M_2. (6.80)$$

Continuing in this manner, the sequence length of the *i*th effective stage EFM1 in an *l*th order MASH modulator is:

$$N_i = \frac{M_1 M_2 \dots M_i}{\lambda_i} \tag{6.81}$$

where $i \in \{1, 2, 3, ..., l\}$ and λ_i is the maximum common divisor of $\lambda_{i-1}M_i$ and $M_1M_2...M_{i-1}mean(e_{i-1})$. Note that when i = 1, $mean(e_0) = X$ and $\lambda_0 = M_0 = 1$.

If $\{M_1, M_2, ..., M_i\}$ are co-prime numbers, M_i and $(M_1 \cdot M_2 \cdot ... \cdot M_{i-1})$ have to be co-prime numbers as well. Thus $\lambda_i = \lambda_{i-1}$. Since M_{i-1} and $(M_1 \cdot M_2 \cdot ... \cdot M_{i-2})$ are also co-prime numbers, $\lambda_{i-1} = \lambda_{i-2}$. Repeating this manner, it is follows that

$$\lambda_i = \lambda_{i-1} = \dots = \lambda_1. \tag{6.82}$$

Then the sequence length of the *i*th EFM1 becomes

$$N_i = \frac{M_1 M_2 \dots M_i}{\lambda_1} \tag{6.83}$$

where λ_1 is the greatest common divisor of X and M_1 . In practice, if the input X and M_1 are set as co-prime numbers, the maximum sequence length of the *i*th stage EFM1 is:

$$N_{i_max} = M_1 M_2 \dots M_i. ag{6.84}$$

Since N is the least common multiple of $\{N_1, N_2, ..., N_l\}$, as was stated in Section 6.4.1, the sequence length of the MASH DDSM is obtained as

$$N = \frac{M_1 \cdot M_2 \cdot \ldots \cdot M_l}{\lambda} \tag{6.85}$$

where λ is the least common multiple of $\{\lambda_1, \lambda_2, ..., \lambda_l\}$.

When $\{M_1, M_2, ..., M_l\}$ are co-prime numbers, (6.82) is true. Then

$$N = \frac{M_1 \cdot M_2 \cdot \ldots \cdot M_l}{\lambda_1}.$$
(6.86)

In addition, if X and M_1 are co-prime numbers as well, λ_1 becomes 1. Thus the maximum sequence length is:

$$N_{max} = M_1 \cdot M_2 \cdot \ldots \cdot M_l. \tag{6.87}$$

6.4.3 A Mathematical Analysis of Higher-Order MM-EFMs

The structure of an *m*th order MM-EFM (MM-EFM*m*) is illustrated in Fig. 6.21. It is similar to the conventional EFM except that the quantizer modulus is M_i instead of M, since the modulus in each EFM is assumed to be different from each other. The symbols $R_1, R_2, ..., R_m$ and $A_1, A_2, ..., A_m$ represent the initial conditions and gains of the *i*th registers, respectively, where $i \in \{1, 2, ..., m\}$. The noise transfer function is

$$NTF = \frac{(Z-1)^m}{Z^m}.$$
 (6.88)

The value of the gains A_i are:

$$\sum_{i=1}^{m} A_i Z^{-i} = 1 - NTF.$$
(6.89)

Since the EFM2 is the most popular higher-order EFM in practice [14], the calculation of its sequence length shall be selected as an example. The structure of EFM2 is shown in Fig. 6.22. The coefficients $[A_1, A_2]$ are obtained from (6.89) as [2, -1]. The *p*th stage of the MASH DDSM is considered. Thus the state variable *u* is dependent on the output of previous stage:

$$u_p[1] = e_{p-1}[1] + 2R_1 - R_2.$$
(6.90)

Using (6.90), the error is expressed as

$$e_p[1] = u_p[1] - M \cdot y_p[1]$$

= $e_{p-1}[1] + 2R_1 - R_2 - M_p \cdot y_p[1]$ (6.91)

After several time steps, the values of the registers are changed from their initial conditions to e_p , since steady-state has been established. Then

$$e_p[2] = e_{p-1}[2] + 2e_p[1] - R_2 - M_p \cdot y_p[2]$$
(6.92)

$$e_p[3] = e_{p-1}[3] + 2e_p[2] - e_p[1] - M_p \cdot y_p[3]$$

$$\vdots$$
(6.93)

$$e_p[N_p] = e_{p-1}[N_p] + 2e_p[N_p - 1] - e_p[N_p - 2] - M_p \cdot y_p[k].$$
(6.94)



Figure 6.21: Higher order error-feedback modulator.



Figure 6.22: Structure of the 2nd order error-feedback modulator.

The sum of all of the above equations (6.91)-(6.94) is

$$\sum_{k=1}^{N_p} e_p[k] = \sum_{k=1}^{N_p} e_{p-1}[k] + 2 \sum_{k=0}^{N_{p-1}} e_p[k] - \sum_{k=-1}^{N_{p-2}} e_p[k] - M_p \cdot \sum_{k=1}^{N_p} y_p[k]$$
(6.95)

Since e_p is periodic with the period N_p [66] in the steady state,

$$\sum_{k=1}^{N_p} e_p[k] = \sum_{k=0}^{N_{p-1}} e_p[k] = \sum_{k=-1}^{N_{p-2}} e_p[k]$$
(6.96)

Then (6.95) becomes

$$\sum_{k=1}^{N_p} y_p = \frac{1}{M_p} \sum_{k=1}^{N_p} e_{p-1}.$$
(6.97)

If the relationship between the sequence length of the *p*th and (p-1)th stage EFM is

$$N_p = K \cdot N_{p-1} \tag{6.98}$$

(6.97) is modified to

$$\sum_{k=1}^{N_p} y_p = \frac{K}{M_p} \sum_{k=1}^{N_{p-1}} e_{p-1}.$$
(6.99)

In comparison with (6.72), it is obtained that the sequence length of the EFM2 is same as that of the EFM1, if they both constitute the *p*th stage in a *l*th stage DDSM [16]. In other words and the crucial point, the sequence length of a MASH DDSM does not depend on the order level, but on the number of EFMs. For example, the period of MM-MASH 1-2 is

$$N = \frac{M_1 \cdot M_2}{\lambda} \tag{6.100}$$

where λ is a parameter to make N the least common multiple of N_1 and N_2 . If the two conditions stated in Section 6.4.2 are satisfied, the maximum sequence length is

$$N = M_1 \cdot M_2. (6.101)$$

This is same as the period of the MM-MASH 1-1.

6.5 The Proposed Structure of the MASH DDSM

The novel structure for the MASH digital delta-sigma modulator (DDSM) employing multi-moduli (MM-MASH) shall now be confirmed in this section. The structure of a MM-MASH consisting of l stages of MM-EFM is illustrated in Fig. 6.23. Note that the block MM-EFM represents either a first-order or a higher order MM-EFM. The crucial points to note about this structure are:

- 1. Every MM-EFM has a different modulus
- 2. Each MM-EFM still only has ONE quantizer modulus.



Figure 6.23: The structure of the MM-MASH.

 M_1 is set as a prime number around 2^{n_0} . This is to make X and M_1 always coprime numbers and therefore satisfy the first condition **Cond1** stated at the begin of Section 6.4.2. This condition must be satisfied to maximise the sequence length of the MASH DDSM and to make the sequence length independent of the value of input. In order to maintain the modulator output accuracy, the value of the input DC X is adjusted to

$$X = M_1 \cdot mean(y) \tag{6.102}$$

where mean(y) is the required output to control the static frequency divider in a fractional-N frequency synthesizer.

In an *l*th order MM-MASH, there are *l* co-prime numbers around 2^{n_0} that need to be found in order to satisfy the second condition **Cond2**. The higher the modulator order, the greater the difficulty in finding suitable values for these moduli. Fortunately, the most popular MASH DDSM in modern communication systems is MASH 1-1-1 [14]. Note that all of the quantizer moduli should be chosen no bigger than 2^{n_0} to avoid necessitating extra hardware. Some quantizer moduli chosen by the author for MM-MASH 1-1-1 are given in Table 6.5.

Word length	M_1	M_2	M_3
5 bit	31	32	29
6 bit	61	64	63
7 bit	127	128	125
8 bit	251	256	255
9 bit	509	512	511
10 bit	1021	1024	1023
11 bit	2039	2048	2047

Table 6.5: Some sample moduli of the 3rd order MM-MASH.

6.6 The Simulation Results

All of the models of the EFM1 and MASH DDSM are built and simulated in **Simulink**. They are shown in Appendix E. All of the simulations in this section are performed with the input constant X = 1. The simulations confirm that the average value of the output mean(y) equals to $\frac{X}{M_1}$. The sequence length of the MASH DDSM is determined using the autocorrelation function [14], which is defined as

$$R_{xx}(m) = \frac{1}{N} \sum_{n=1}^{N-m+1} x(n)x(n+m-1)$$
(6.103)

where m = 1, 2, ..., N and N is the number of samples. As a typical example, Fig. 6.24 shows that the sequence length of a 3rd-order 5-bit MM-MASH is 28768 and this equals $M_1 \cdot M_2 \cdot M_3$ as given in Table 6.5. The sequence length is only 64 from a 3rd-order 5-bit conventional MASH. The sequence lengths of the HK-MASH and the MM-MASH are compared in Table 6.6, since the HK-MASH has the longest sequence length of current structures of DDSMs. The MM-MASH achieves a longer sequence when the word lengths are 6, 8, 9, 10 and 11. Thus the choice between MM-MASH and HK-MASH depends on the word length. However, the MM-MASH produces a longer sequence length, if $M_1 \neq 2^{n_0} - 1$, where M_1 is the biggest prime number below 2^{n_0} as described in Section 6.5. Hence, it has a greater probability of producing a longer sequence length.



Figure 6.24: The autocorrelation result for the 5-bit MM-MASH 1-1-1.

Word length	HK-MASH	MM-MASH	Difference
5 bit	32768	28768	-4000
6 bit	0.227×10^6	0.246×10^6	$+0.019 \times 10^{6}$
7 bit	2.048×10^{6}	2.032×10^6	-0.016×10^{6}
8 bit	15.81×10^{6}	16.39×10^{6}	$+0.58 \times 10^{6}$
9 bit	131.87×10^{6}	133.17×10^{6}	$+1.3 \times 10^{6}$
10 bit	1.06×10^{9}	1.07×10^9	$+10 \times 10^{6}$
11 bit	8.48×10^{9}	8.55×10^9	$+70 \times 10^{6}$

Table 6.6: A comparison of the sequence lengths for the HK-MASH and MM-MASH.

The ultimate goal in the novel design is to improve the quality of the power spectrum, i.e to reduce noise power in the useful band. Hence, the power spectrum is now examined. The power spectral density for 5-11 bit MASH DDSMs are compared and in all cases the performance of the MM-MASH is better. As an example, the power spectral density [71] of the MM-MASH 1-1-1 and dithered conventional MASH 1-1-1 are compared in Figs. 6.25-6.31. The dithering is applied to the DDSM by adding a 1-bit pseudorandom dither sequence to the LSB of the dc input [65]. It is evident from the figures that the MM-MASH is significantly more effective than the conventional MASH DDSM at lower frequencies.

Also, the power spectral density of the 11-bit MM-MASH 1-1-1 and 11-bit HK-MASH 1-1-1 are compared in Fig. 6.32, since it produces the biggest sequence length difference as shown in Table 6.6. The noise power performance of MM-MASH is a little better than that of HK-MASH. Thus the noise power difference results from the MM-MASH and HK-MASH with a smaller word length can be even neglected. However, MM-MASH is a better choice in high-end products, since they desire the noise to be the lower the better.

The power spectral density [71] of the MM-MASH 1-1-1 and MM-MASH 1-2 is compared in Fig. 6.33-6.39. Both of the DDSMs are 3rd order. It is evident from the figure that the MM-MASH 1-1-1 has a better noise performance than the MM-MASH 1-2. This is expected as the sequence length of the former is longer than that of the latter as shown in Section 6.4.3. However, the MM-MASH 1-2 has the advantage of less hardware requirements and so a balance between hardware cost and noise performance is required in the selection of the most suitable structure.

6.7 Conclusions

Digital delta-sigma modulators have been widely used for years. The benefit of their use is that the quantisation noise is pushed to the higher frequencies out of the useful bandwidth. Then the noise can be simply eliminated by a lowpass filter. Now with DDSMs, the output sequence length is periodic and short. This is disadvantageous as it results in spurs in the useful spectrum. To avoid this, one needs to whiten the quantization noise. This can be done by dithering. However, in order to avoid the extra noise and hardware requirements resulting from the dithering method, several novel DDSM structures have been proposed to lengthen the sequence length. In this chapter, it has been proved that MM-MASH produces the maximum sequence length for most input word lengths. Expressions for the sequence length of the modulator are derived. The simulation results confirm the results. Higher-order EFMs are also investigated as previous research indicates that they yield hardware savings [72]. However, it has been proved in this thesis that they are not able to increase the sequence length. Hence, as a final note, a balance must be made between the noise performance and the hardware requirements.



Figure 6.25: The power spectral density of the dithered 5-bit conventional MASH DDSM and non-dithered 5-bit MM-MASH.



Figure 6.26: The power spectral density of the dithered 6-bit conventional MASH DDSM and non-dithered 6-bit MM-MASH.



Figure 6.27: The power spectral density of the dithered 7-bit conventional MASH DDSM and non-dithered 7-bit MM-MASH.



Figure 6.28: The power spectral density of the dithered 8-bit conventional MASH DDSM and non-dithered 8-bit MM-MASH.



Figure 6.29: The power spectral density of the dithered 9-bit conventional MASH DDSM and non-dithered 9-bit MM-MASH.



Figure 6.30: The power spectral density of the dithered 10-bit conventional MASH DDSM and non-dithered 10-bit MM-MASH.



Figure 6.31: The power spectral density of the dithered 11-bit conventional MASH DDSM and non-dithered 11-bit MM-MASH.



Figure 6.32: The power spectral density of the 11-bit HK-MASH DDSM and 11-bit MM-MASH.



Figure 6.33: The power spectral density of different structures of 3rd order 5-bit MM-MASH.



Figure 6.34: The power spectral density of different structures of 3rd order 6-bit MM-MASH.



Figure 6.35: The power spectral density of different structures of 3rd order 7-bit MM-MASH.



Figure 6.36: *The power spectral density of different structures of 3rd order 8-bit MM-MASH.*



Figure 6.37: The power spectral density of different structures of 3rd order 9-bit MM-MASH.



Figure 6.38: The power spectral density of different structures of 3rd order 10-bit MM-MASH.



Figure 6.39: The power spectral density of different structures of 3rd order 11-bit MM-MASH.

Chapter 7 Conclusions

Several novel CAD algorithms are proposed in this dissertation. The Padé-based approximation is proposed for the numerical integration of stiff differential equations. Illustrative examples have shown that the Padé-based techniques permit a significant increase in step-size when compared to a traditional predictor-corrector such as the Adams-Moulton technique. Both single-step and multi-step methods are proposed. Multi-step methods have the advantage of obviating the need for the evaluation of high-order derivatives. All of the techniques are particularly suitable for highly nonlinear systems as there is no need for the solution of a non-linear set of algebraic equations at each time step.

The Filon-type methods are explored for systems subjected to high-frequency signals. Numerical examples confirm the significant potential of Filon-type methods in this setting. Futher research is required into issues of implementation and software design, as well as detailed comparison with existing software for realistic differential equations originating in RF and communications engineering.

Two simulation strategies have been presented for the determination of the locking range of an ILFD. The first strategy involves the use of the WaMPDE in conjunction with the bisection method to identify external frequencies to which the oscillator locks. The WaMPDE enables identification of the natural frequency of the oscillator. Hence, by examining the Jacobian matrix used in the transient evolution to steady state, it is possible to ascertain when the system becomes independent of the local frequency and can then be deemed locked. In addition, a technique for estimating the locking ranges may be obtained using linear extrapolation if an accurate result is not necessary. Results confirm the efficacy of the approaches. The second technique is based on the Multiple-Phase-Condition Envelope Following (MPCENV) algorithm. The periods of the carrier and envelope are set as variables. In order to solve for them, two extra conditions are required. The output frequencies of the ILFD corresponding to the different injected frequencies are then obtained. Thus the Devil's staircase can be plotted from the obtained data. Results for an LC-oscillator based ILFD confirm its efficacy. Computer simulation for the determination of the locking range is advantageous in avoiding the need for time-consuming experiments. It also greatly aids in design work involving ILFDs and their use as lower power frequency dividers in PLLs for wireless systems.

A novel design methodology for a MASH DDSM aimed at increasing the sequence length is introduced as well. Increasing the sequence length improves the noise performance of DDSMs. The proposed method employs different moduli in each stage of the EFM1. It is proven that the multi-modulus architecture is suitable because the output of the MASH modulator is only dependent on the quantizer modulus of the first stage EFM1 and independent of the others. Hence, having different moduli does not affect the output of the modulator. The expressions for the sequence length of the EFM1 of each stage and for the complete MASH DDSM are derived. There are two conditions given that must be satisfied to yield the maximum modulator period. A novel structure for the MASH digital delta-sigma modulator employing the multi-moduli (MM-MASH) is proposed. In most cases, the MM-MASHs produce an increased sequence length when compared to that of the current best DDSM, HK-MASH. Both of the output accuracy and the predicted sequence length of the MM-MASH are validated by simulation. The power spectral density spectra confirm that the proposed modulator architecture is, in most cases, more effective than the HK-MASH method at moving noise from the lower frequencies.

Future work will involve investigation of ring-oscillator based injection-locked frequency dividers. In particular, expressions for the locking range and the conditions for locking will be explored. Deficiencies in previous work such as that by Lai [73] will be addressed.

Appendix A

A-stability and L-stability of Padé approximations

In order to examine the stability of Padé approximations, consider the following standard test differential equation:

$$y' = \lambda y \tag{A.1}$$

where $\lambda < 0$. Now, consider the second-order Padé approximation where h is the time-step and y_k is the estimate of the solution y at time t_k :

$$y_{k+1} = \frac{1 + (h\lambda/3)}{1 - (2h\lambda/3) + ((h\lambda)^2/6)} y_k.$$
 (A.2)

The amplification factor is:

$$R(\lambda h) = \frac{1 + (h\lambda/3)}{1 - (2h\lambda/3) + ((h\lambda)^2/6)}.$$
 (A.3)

As $\forall h\lambda < 0$, it is obtained

$$|R(\lambda h)| < 1. \tag{A.4}$$

Thus the method in equation (A.2) is A-stable. Similarly, higher-order Padé methods are A-stable $\forall h\lambda < 0$.

In addition, the Padé method in (3.1) is *L*-stable if m < n. A numerical method is *L*-stable if, in addition to *A*-stability, when applied to the scalar test equation (A.1) with $\Re(\lambda) < 0$, $|R(\lambda h)| \to 0$ as $\Re(\lambda h) \to -\infty$ [23]. Examining equation (3.3), it is evident that $\lim_{\Re(\lambda h)\to-\infty} |R(\lambda h)| = 0$. This is true for higher-order methods if m < n and $\forall h\lambda < 0$.

Appendix B

Richardson extrapolation and improvement of accuracy

Consider the test equation:

$$y' = \lambda y \tag{B.1}$$

where $\lambda < 0$. Also, consider the first-order Padé approximation:

$$y_{k+1} = \frac{1}{1 - h\lambda} y_k. \tag{B.2}$$

Let y_h be the estimate of y_{k+1} obtained with a time-step of h. Let $y_h/2$ be the estimate obtained using a time-step of h/2. The second-order error in y_h is $E_h = \lambda^2/2$. The second-order error in $y_{h/2}$ is $E_{h/2} = \lambda^2/8$. Now consider:

$$y_{ex} = y_h + E_h h^2 + O(h^3)$$
 (B.3)

$$y_{ex} = y_{h/2} + E_{h/2}h^2 + O(h^3).$$
 (B.4)

From (B.3) and (B.4), the second-order errors may be eliminated by evaluating:

$$y_{ex} = \frac{4y_{h/2} - y_h}{3} + O(h^3).$$
(B.5)

In general, for an (L - 1)th-order method, Richardson extrapolation eliminates *L*th errors as follows:

$$y_{ex} = \frac{2^L y_{h/2} - y_h}{2^L - 1} + O(h^{L+2}).$$
(B.6)

Appendix C

MESFET amplifier details



Figure C.1: Schematic of the MESFET amplifier.

$$\frac{dv_{gs}}{dt} = \frac{1}{C_{gs}} \left[\frac{v_{gd} - v_{gs} + v_{ds} + R_{gdi}(i_s - i_d - i_{gs} - i_{gd})}{R_{gsi} + R_{gdi}} \right]$$
(C.1)

$$\frac{dv_{gd}}{dt} = \frac{1}{C_{gd}} \left[\frac{v_{gs} - v_{gd} - v_{ds} + R_{gsi}(i_s - i_d - i_{gs} - i_{gd})}{R_{gsi} + R_{gdi}} \right]$$
(C.2)

$$\frac{dv_{ds}}{dt} = \frac{1}{C_{ds}} \left[i_s - \frac{v_{gd} - v_{gs} + v_{ds} + R_{gdi}(i_s - i_d - i_{gs} - i_{gd})}{R_{gsi} + R_{gdi}} - i_{gs} - i_{ds} - \frac{v_{ds} - v_{xx}}{R_x} \right]$$
(C.3)

$$\frac{dv_{xx}}{dt} = \frac{1}{C_x} \left[\frac{v_{ds} - v_{xx}}{R_x} \right]$$
(C.4)

Let:

$$v_{dx} = \frac{L_g L_s (v_d - R_d i_d) + L_d L_s (v_g - v_{gd} - R_g i_g) + L_d L_g (R_s i_s + v_{ds})}{L_g L_s + L_d L_s + L_d L_g}$$
(C.5)

$$\frac{di_s}{dt} = \frac{1}{L_s} (v_{dx} - v_{ds} - R_s i_s)$$
(C.6)

$$\frac{di_d}{dt} = \frac{1}{L_d} (v_d - v_{dx} - R_d i_d) \tag{C.7}$$

$$\frac{dv_g}{dt} = \frac{(C_{pgd} + C_{pd})[(e_g - v_g)G_s - i_{lg} - i_g] + C_{pgd}(-G_l v_o - i_{ld} - i_d)}{C_{pg}C_{pd} + C_{pgd}(C_{pg} + C_{pd})}$$
(C.8)

$$\frac{dv_d}{dt} = \frac{C_{pgd}[(e_g - v_g)G_s - i_{\lg} - i_g] + (C_{pgd} + C_{pd})(-G_l v_o - i_{ld} - i_d)}{C_{pg}C_{pd} + C_{pgd}(C_{pg} + C_{pd})}$$
(C.9)

$$\frac{di_{\rm lg}}{dt} = \frac{v_g - R_{gg}i_{\rm lg} - V_{GG}}{L_{bg}} \tag{C.10}$$

$$\frac{di_{1g}}{dt} = \frac{v_d - R_{dd}i_{1g} - V_{DD}}{L_{bd}}$$
(C.11)

$$i_{gs} = \text{IS}_{-}\text{GS}\left[e^{\frac{q_{\text{Vgs}}}{NkT}} - 1\right]$$
 (C.12)

$$i_{gd} = \text{IS}_{-}\text{GS}\left[e^{\frac{q(v_{gs}-v_{gd})}{NkT}} - 1\right]$$
(C.13)

The Curtice-Ettenberg model is employed for the MESFET shown in Fig. C.1. Its details are as follows:

$$v_1 = \begin{cases} v_{gs}(1 + \beta(v_{out0} - v_{ds})), & v_{ds} \ge 0\\ v_{gd}(1 + \beta(v_{out0} - v_{ds})), & v_{ds} < 0 \end{cases}$$
(C.14)

$$i_{ds0} = \begin{cases} A_0 + v_1(A_1 + v_1(A_2 + v_1A_3)), & v_1 \ge v_{PMAX} \\ A_0 + v_{PMAX}(A_1 + v_{PMAX}(A_2 + v_{PMAX}A_3)), & v_1 < v_{PMAX} \end{cases}$$
(C.15)

where

$$v_{PMAX} = \frac{-2A_2 + \sqrt{4A_2^2 - 12A_1A_3}}{6A_3} \tag{C.16}$$

If $v_{ds} > 0$ and $i_{ds0} < 0$

$$i_{ds} = 0 \tag{C.17}$$

else

$$\chi_{2} = -2\gamma v_{ds}$$

$$z_{1} = e^{\chi_{2}}$$

$$i_{ds} = i_{ds0} \frac{1 - z_{1}}{1 + z_{1}}$$
(C.18)

Let $\phi = F_C \cdot V_{BI}$. Then the capacitances C_{gs} and C_{gd} are given as:

$$C_{gs} = \begin{cases} \frac{C_{gs0}}{\sqrt{1 - \frac{v_{gs}}{V_{BI}}}} & v_{gs} < \phi \\ \frac{C_{gs0} \left(1 - 1.5F_C + 0.5\frac{v_{gs}}{V_{BI}}\right)}{\left(1 - F_C\right)^{\frac{3}{2}}} & v_{gs} \ge \phi \end{cases}$$
(C.19)

$$C_{gd} = \begin{cases} \frac{C_{gd0}}{\sqrt{1 - \frac{v_{gd}}{V_{BI}}}} & v_{gd} < \phi \\ \frac{C_{gd0} \left(1 - 1.5F_C + 0.5 \frac{v_{gd}}{V_{BI}} \right)}{(1 - F_C)^{\frac{3}{2}}} & v_{gd} \ge \phi \end{cases}$$
(C.20)

The following parameters are used:

$$G_S = 0.02s \tag{C.21}$$

$$G_l = 0.02s \tag{C.22}$$

$$L_{bg} = 100nH \tag{C.23}$$

$$L_{bd} = 100nH \tag{C.24}$$

$$R_{gg} = R_{dd} = 0 \tag{C.25}$$

$$C_{pg} = 0.15 pF \tag{C.26}$$

$$C_{pd} = 0.15 pF \tag{C.27}$$

$$L_g = 0.35nH \tag{C.28}$$

$$L_d = 0.35nH \tag{C.29}$$

$$L_s = 0.35nH \tag{C.30}$$

$$R_g = 7.21W \tag{C.31}$$

$$R_d = 5.0686W$$
 (C.32)

$$R_s = 3.6953W$$
 (C.33)

$$C_{pgd} = 0.01 pF \tag{C.34}$$

$$R_{gdi} = 5.5W \tag{C.35}$$

$$R_{gsi} = 3.18W \tag{C.36}$$

$$T = 300.15K$$
 (C.37)

$$IS_GS = 1.0 \times 10^{-14} A \tag{C.38}$$

$$IS_{-}GD = 0.0$$
 (C.39)

$$N = 1.0$$
 (C.40)

$$V_{BI} = 1.5V$$
 (C.41)

$C_{gs0} = 0.7243 pF$ (C.42)

$$C_{gd0} = 0.0197 pF \tag{C.43}$$

$$F_C = 0.5$$
 (C.44)

$$C_x = 0.19 pF \tag{C.45}$$

$$R_x = 800W \tag{C.46}$$

$$C_{ds} = 0.1037 pF$$
 (C.47)

$$b = 0.05396/V$$
 (C.48)

$$g = 7.680735/V \tag{C.49}$$

$$V_{out0} = 19.996182V \tag{C.50}$$

$$A_0 = 0.17229A \tag{C.51}$$

$$A_1 = 0.093461 A/V \tag{C.52}$$

$$A_2 = -0.053499A/V^2 \tag{C.53}$$

$$A_3 = -0.028237A/V^3 \tag{C.54}$$

Appendix D

Schematics of the diode rectifier

circuit used in ADS



Figure D.1: The diode rectifier circuit with amplitude modulated input signal.



Figure D.2: The diode rectifier circuit with digitally modulated input signal.

Appendix E

Models of DDSMs used in Simulink



Figure E.1: The model of EFM1 used in Simulink.



Figure E.2: The model of HK-EFM1 used in Simulink.



Figure E.3: The model of EFM2 used in Simulink.



Figure E.4: The model of MASH 1-1-1 used in Simulink.



Figure E.5: The model of MASH 1-2 used in Simulink.

Appendix F

List of relevant publications

Journal Papers:

- [27] M. Condon, E. Dautbegovic and T. Xu, "Novel Padé-based algorithms for numerical integration of ODEs," *COMPEL: The International Journal for Computations and Mathematics in Electrical and Electronic Engineering*, vol. 27, issue 6, 2008, pp. 1402-1417.
- [74] M. Condon, A. Deaño, A.Iserles, K. Maczynski and T. Xu, "On highly oscillatory problems arising in electronic engineering," Accepted by COMPEL: The International Journal for Computations and Mathematics in Electrical and Electronic Engineering.

Conference Papers:

[16] T. Xu and M. Condon, "Comparative Study of the MASH Digital Delta-Sigma Modulators," in *Proc. IEEE Ph.D. Research in Microelectronics and Electronics*, Cork, Ireland, July 2009, pp. 196-199.
- [15] T. Xu and M. Condon, "Design methodology for a maximum sequence length MASH digital delta-sigma modulator," in *Proc. World Congress on Engineering*, London, U.K., July 2009.
- [55] T. Xu and M. Condon, "Accurate simulation of the Devil's staircase of an injectionlocked frequency divider," in *Proc. International Conference on Scientific Computing in Electrical Engineering*, Espoo, Finland, Sept. 2008, pp. 105-106.
- [54] T. Xu and M. Condon, "An effective method for the determination of the locking range of an injection-locked frequency divider," in *Proc. Emerging Trends in Wireless Communications*, Dublin, Ireland, April 2008, pp. 47-50.
- [53] C. E. Christoffersen, M. Condon and Tao Xu, "A new method for the determination of the locking range of oscillators," in *Proc. European Conference on Circuit Theory and Design*, Seville, Spain, Aug. 2007, pp. 575-578.
- [7] T. Xu, Z. Ye and M. P. Kennedy, "Mathematical analysis of injection-locked frequency dividers," in *Proc. International Symposium on Nonlinear Theory and its Applications*, Bologna, Italy, Sept. 2006, pp. 639-642.
- [8] Z. Ye, T. Xu and M. P. Kennedy, "Locking range analysis for injection-locked frequency dividers," in *Proc. International Symposium on Circuits and Systems*, Island of Kos, Greece, May 2006, pp. 4070-4073.

Bibliography

- Circuit Envelope Simulation, Agilent Technologies, Santa Clara, California, Aug. 2005.
- [2] R. Adler, "A study of locking phenomena in oscillators," *Proc. of the IEEE*, vol. 61, no. 10, pp. 1380–1385, Oct. 1973.
- [3] K. Hosseini and M. P. Kennedy, "Maximum sequence length mash digital deltasigma modulators," *IEEE Trans. on Circuits and Systems-I: Regular Papers*, vol. 54, no. 12, pp. 2628–2638, Dec. 2007.
- [4] *Electronic design automation*. [Online]. Available: http://en.wikipedia.org/wiki/ Electronic_design_automation
- [5] M. Condon, E. Dautbegovic, and T. J. Brazil, "An efficient numerical algorithm for the transient analysis of high-frequency non-linear circuits," in *European Microwave Conference*, Milan, Italy, Oct. 2002, pp. 1–4.
- [6] S. Olver, "Numerical approximation of highly oscillatory integrals," Ph.D. dissertation, University of Cambridge, Nov. 2007.
- [7] T. Xu, Z. Ye, and M. Kennedy, "Mathematical analysis of injection-locked frequency dividers," in *Proc. International Symposium on Nonlinear Theory and its Applications*, Bologna, Italy, Sep. 2006, pp. 639–642.

- [8] Z. Ye, T. Xu, and M. P. Kennedy, "Locking range analysis for injection locking frequency dividers," in *Proc. International Symposium on Circuits and Systems*, Island of Kos, Greece, May 2006, pp. 4070–4073.
- [9] D. O'Neill, D. Bourke, and M. P. Kennedy, "The devil's staircase as a method of comparing injection-locked frequency divider topologies," in *Proc. European Conference on Circuit Theory and Design*, vol. 3, Cork, Ireland, Sep. 2005, pp. 317–320.
- [10] B. Razavi, "A study of injection locking and pulling in oscillators," *IEEE Journal of Solid-Stagte Circuits*, vol. 39, no. 9, pp. 1415–1424, Sep. 2004.
- [11] O. Narayan and J. Roychowdhury, "Analysing oscillators using multitime PDEs," *IEEE Trans. on Circuits and Systems*, vol. 50, no. 7, pp. 894–903, July 2003.
- [12] R. Gu and S. Ramaswamy, "Fractional-N phase locked loop design and applications," in *Proc. The 7th International Conference on ASIC*, Guilin, China, Oct. 2007, pp. 327–332.
- [13] Z. Ye, W. Chen, and M. P. Kennedy, "A novel dual-loop multi-phase frequency synthesizer," in *Proc. European Conference on Circuit Theory and Design*, Seville, Spain, Aug. 2007, pp. 567–570.
- [14] M. J. Borkowski, T. A. D. Riley, J. Häkkinen, and J. Kostamovaara, "A practical Δ - Σ modulator design method based on periodical behavior analysis," *IEEE Trans. on Circuits and Systems-II: Express Briefs*, vol. 52, no. 10, pp. 626–630, Oct. 2005.
- [15] T. Xu and M. Condon, "Design methodology for a maximum sequence length mash digital delta-sigma modulator," in *Proc. World Congress on Engineering*, London, U.K., July 2009.

- [16] —, "Comparative study of the mash digital delta-sigma modulators," in *Proc. IEEE Ph.D. Research in Microelectronics and Electronics*, Cork, Ireland, July 2009.
- [17] M. K. Jain, *Numerical Solution of Differential Equations*, 2nd ed. New Delhi: Wiley Eastern Limited, 1984.
- [18] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations I*. Springer-Verlag, 1987.
- [19] R. Bronson, Differential Equations, E. J. Bredensteiner, Ed. McGraw-Hill, 2003.
- [20] J. Butcher, Numerical Methods for Ordinary Differential Equations, 2nd ed. John Wiley and Sons, 2008.
- [21] F. Bashforth and J. C. Adams, *Theories of Capillary Action*. London: Cambridge University Press, 1883.
- [22] K. A. Atkinson, An Introduction to Numerical Analysis, 2nd ed. John Wiley and Sons, 1988.
- [23] G. Dahlquist, "A special stability problem for linear multistep methods," *BIT*, vol. 3, pp. 27–43, 1963.
- [24] B. L. Ehle, "On padé approximations to the exponential function and *a*-stable methods for the numerical solution of initial value problems," in *Report 2010*. University of Waterloo, 1969.
- [25] L. W. Nagel, "Spice2, a computer program to simulate semiconductor circuits," University of California, Berkeley, CA., Tech. Rep. ERL- M520, 1975.

- [26] E. Dautbegovic, "Transient simulation of complex electronic circuits and systems operating at ultra high frequencies," Ph.D. dissertation, Dublin City University, Dublin, Ireland, Jan. 2005.
- [27] M. Condon, E. Dautbegovic, and T. Xu, "Novel padé-based algorithms for numerical integration of odes," *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 27, no. 6, pp. 1402–1417, 2008.
- [28] C. F. Gerald and P. O. Wheatley, *Applied Numerical Analysis*, 6th ed. Harlow, England: Addison Wesley Longman, 1999.
- [29] M. Stabrowski, "An efficient algorithm for solving stiff ordinary differential equations," *Simulation Practice and Theory*, vol. 5, no. 4, pp. 333–344, May 1997.
- [30] M. C. Jeruchim, P. Balaban, and K. S. Shanmugan, Simulation of Communication Systems, Modeling, Methodology and Techniques, 2nd ed. New York: Kluwer Academic/Plenum Publishers, 2000.
- [31] M. Khanamirian, "Quadrature methods for systems of highly oscillatory odes," DAMTP, University of Cambridge, Cambridge, UK, Tech. Rep. NA2008/01, 2008.
- [32] A. Iserles and S. P. Nørsett, "Efficient quadrature of highly oscillatory integrals using derivatives," *Proc. Royal Soc. A*, vol. 461, pp. 1383–1399, 2005.
- [33] M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions*. New York: Dover Publications, 1965.
- [34] U. Miekkala and O. Nevanlinna, "Iterative solution of systems of linear differential equations," *Acta Numerica*, vol. 5, p. 259307, 1996.

- [35] O. Nevanlinna, "Remarks on picardlindeloff iteration," *BIT*, vol. 29, p. 328346, 1989.
- [36] S. Vandewalle, *Parallel multigrid waveform relaxation for parabolic problems*. Stuttgart, Germany: B.G. Teubner GmbH, Dec. 1993.
- [37] E. Ngoya and R. Larcheveque, "Envelope transient analysis: A new method for the transient and steady-state analysis of microwave communications circuits and systems," in *Digest IEEE MTT-S International Microwave Symposium*, vol. 3, San Francisco, CA, USA,, June 1996, pp. 1365–1368.
- [38] D. Sharrit, "New method of analysis of communication systems," in *Proc. IEEE MTT-S WMFA: Nonlinear CAD Workshop*, San Francisco, CA, USA,, June 1996.
- [39] A Users Guide to Envelope Following Analysis Using Spectre RF, Cadence Design Systems, San Jose, CA, USA, April 2007.
- [40] U. Singh and M. Green, "Dynamics of high-frequency cmos dividers," in *Proc. IEEE International Symposium on Circuits and Systems*, vol. 5, Scottsdale, Arizona, USA, May 2002, pp. 421–424.
- [41] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*. Cambridge University Press, 1998.
- [42] A. Buonomo and A. L. Schiavo, "Modelling and analysis of differential VCOs," *International Journal of Circuit Theory and Applications*, vol. 32, pp. 117–131, 2004.
- [43] H. R. Rategh and T. H. Lee, "Superharmonic injection-locked frequency dividers," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 813–821, June 1999.

- [44] R. L. Miller, "Fractional-frequency generators utilizing regenerative modulation," *Proc. of IRE*, vol. 27, pp. 446–457, July 1939.
- [45] R. J. Betancourt-Zamora, S. Verma, and T. H. Lee, "1-GHz and 2.8GHz injectionlocked ring oscillator prescalers," in *Symp. VLSI Circuits Dig. Tech. Papers*, Kyoto, Japan, June 2001, pp. 47–50.
- [46] S. Verma, H. Rategh, and T. H. Lee, "A unified model for injection-locked frequency dividers," *IEEE Journal of Solid-State Circuits*, vol. 38, no. 6, pp. 1015– 1027, June 2003.
- [47] S.-M. Wu and W.-L. Chen, "A 5.8-GHz CMOS VCO with injection-locked frequency divider for IEEE 802.11a application," in *Proc. European Conference on Circuit Theory and Design*, Cracow, Poland, Sep. 2003.
- [48] D. O'Neill, D. Bourke, Z. Ye, and M. P. Kennedy, "Accurate modeling and experimental validation of an injection-locked frequency divider," in *Proc. European Conference on Circuit Theory and Design*, vol. 3, Cork, Ireland, Sep. 2005, pp. 409–412.
- [49] H. P. Moyer and A. S. Daryoush, "A unified analytical model and experimental validations of injection-locking processes," *IEEE Trans. on Microwave Theory and Techniques*, vol. 38, no. 4, pp. 493–499, Apr. 2000.
- [50] B. Razavi, "Analysis, modeling and simulation of phase noise in monolithic voltage-controlled oscillators," in *Proc. IEEE Custom Integrated Circuits Conference*, May 1995, pp. 323–326.
- [51] R. L. Burden and J. D. Faires, *Numerical Analysis*, 5th ed. Boston: PWS Publishing Company, 1993.

- [52] E. L. Allgower and K. Georg, Numerical Continuation Methods. New York: Springer-Verlag, 1990.
- [53] C. E. Christoffersen, M. Condon, and T. Xu, "A new method for the determination of the locking range of oscillators," in *Proc. European Conference on Circuit Theory and Design*, Seville, Spain, Aug. 2007, pp. 575–578.
- [54] T. Xu and M. Condon, "An effective method for the determination of the locking range of an injection-locked frequency divider," in *Proc. Emerging Trends in Wireless Communications*, Dublin, Ireland, Apr. 2008, pp. 47–50.
- [55] —, "Accurate simulation of the devil's staircase of an injection-locked frequency divider," in *Proc. International Conference on Scientific Computing in Electrical Engineering*, Espoo, Finland, Sep. 2008, pp. 105–106.
- [56] T. Mei and J. Roychowdhury, "A time-domain oscillator envelope tracking algorithm employing dual phase conditions," *IEEE Trans. on Computer-Aided Design* of Integrated Circuits and Systems, vol. 27, no. 1, pp. 59–69, Jan. 2008.
- [57] H. P. Langtangen, Computational Partial Differential Equations: Numerical Methods and Diffpack Programming. Springer, 1999.
- [58] T. J. Aprille and T. N. Trick, "Steady-state analysis of nonlinear circuits with period inputs," *Proc. of IEEE*, vol. 60, no. 1, pp. 108–114, Jan. 1972.
- [59] H. Inose, Y. Yasuda, and J. Marakami, "A telemetering system by code modulation, delta-sigma modulation," *IRE Trans. on Space, Electronics and Telemetry*, vol. 8, pp. 204–209, Sept. 1962.
- [60] A. Reddy, "Noise shaping with sigma-delta modulators in fractional-N synthesizers," in *Proc. IEEE Intl. Workshop on Radio-Frequency Integration Technology*, Singapore, Dec. 2007, pp. 329–332.

- [61] M. Kozak and I. Kale, "Rigorous analysis of delta-sigma modulators for fractional-n pll frequency synthesis," *IEEE Trans. on Circuits and Systems-I: Regular Papers*, vol. 51, no. 6, pp. 1148–1162, June 2004.
- [62] D. Reefman, J. Reiss, E. Janssen, and M. Sandler, "Description of limit cycles in sigma-delta modulators," *IEEE Trans. on Circuits and Systems-I: Regular Papers*, vol. 52, no. 6, pp. 1211–1223, June 2005.
- [63] J. Reiss and M. Sandler, "Detection and removal of limit cycles in sigma delta modulators," *IEEE Trans. on Circuits and Systems-I: Regular Papers*, vol. 55, no. 10, pp. 3119–3130, Nov. 2008.
- [64] S. Pamarti, J. Welz, and I. Galton, "Statistics of the quantization noise in 1-bit dithered single-quantizer digital delta-sigma modulators," *IEEE Trans. on Circuits and Systems-I: Regular Papers*, vol. 54, no. 3, pp. 492–503, Mar. 2007.
- [65] S. Pamarti and I. Galton, "Lsb dithering in mash delta-sigma d/a converters," *IEEE Trans. on Circuits and Systems-I: Regular Papers*, vol. 54, no. 4, pp. 779– 790, Apr. 2007.
- [66] K. Hosseini and M. P. Kennedy, "Mathematical analysis of a prime modulus quantizer mash digital delta-sigma modulator," *IEEE Trans. on Circuits and Systems-II: Express Brief*, vol. 54, no. 12, pp. 1105–1109, Dec. 2007.
- [67] —, "Mathematical analysis of digital mash delta-sigma modulator for fractional-N frequency synthesizers," in Proc. IEEE Ph. D. Research in Microelectronics and Electronics, Otranto (Lecce), Italy, June 2006, pp. 309–312.
- [68] K. Hosseini, M. P. Kennedy, and C. McCarthy, "Calculation of sequence lengths in mash 1-1-1 digital delta sigma modulators with a constant input," in *Proc.*

IEEE Ph. D. Research in Microelectronics and Electronics, Bordeaux, France, July 2007, pp. 13–16.

- [69] M. J. Borkowski and J. Kostamovaara, "Spurious tone free digital delta-sigma modulator design for dc inputs," in *Proc. IEEE International Symposium on Circuits and Systems*, Kobe, Japan, May 2005, pp. 5601–5604.
- [70] R. E. Crandall and C. Pomerance, *Prime numbers: a computational perspective*. Springer, 2001.
- [71] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*. Upper Saddle River: NJ: Prentice-Hall, 1999.
- [72] L. Sun, T. Lepley, F. Nozahic, A. Bellissant, T. Kwasniewski, and B. Heim, "Reduced complexity, high performance digital delta-sigma modulator for fractionaln frequency synthesis," in *Proc. of the 1999 IEEE International Symposium on Circuits and Systems*, vol. 2, Orlando, FL, USA, July 1999, pp. 152–155.
- [73] X. Lai and J. Roychowdhury, "Capturing oscillator injection locking via nonlinear phase-domain macromodels," *IEEE Trans. Microwave Theory Tech.*, vol. 52, no. 9, p. 22512261, Sep. 2004.
- [74] M. Condon, A. Deano, and A. Iserles, "On highly oscillatory problems arising in electronic engineering," *To appear in Math. Modelling & Num. Anal.*, 2009.