UNIVERSITÉ DE LIMOGES

ÉCOLE DOCTORALE Science – Technologie – Santé

FACULTÉ DES SCIENCES ET TECHNIQUES

XLIM - Département C2S2

Année : 2009 Thèse N° 8-2009

Thèse

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Discipline : Électronique des Hautes Fréquences et Optoélectronique

présentée et soutenue par

Bo Wang

le 19 Mars 2009

Techniques de modélisation et de simulation pour la vérification précise de PLLs à facteur de division entier

Thèse dirigée par Edouard NGOYA

JURY :

Raymond QUÉRÉ	Professeur, Université de Limoges	Président
Almudena SUAREZ	Professeur, Université de Cantabria, Espagne	Rapporteur
Francis CALMON	Maître de Conférence HDR, INSA de Lyon	Rapporteur
Dominique DALLET	Professeur, ENSEIRB, Bordeaux	Examinateur
Jean-Christophe NALLATAMBY	Maître de Conférence, Université de Limoges	Examinateur
Arnaud SOURY	Ingénieur, Agilent Technologies, EEsof EDA Division	Examinateur
Edouard NGOYA	Directeur de recherche CNRS, XLIM	Examinateur
Jean ROUSSET	Ingénieur de recherche CNRS, XLIM	Examinateur
Cédric PUJOL	Ingénieur, Agilent Technologies, EEsof EDA Division	Invité

"C'est le temps que tu as perdu pour ta rose qui fait ta rose si importante."

Antoine de SAINT-EXUPERY

To the memory of my grandparents
To my parents
To Huiling

Acknowledgements

The work presented in this thesis is realized under the close cooperation between the laboratory XLIM UMR CNRS n°6172 and Agilent-EEsof GoldenGate team initially Xpedion Design System, and the outcome of this thesis would not have been possible without the help and support of many people.

First I would like to thank to my advisor Mr. Edouard NGOYA for his support and guidance, and especially for his availability and numerous suggestions which have been invaluable for me throughout these 3 years. I am grateful to him for having shared so much of time and his insights into this thesis work. Also I have learned a lot from his rigorous scientific attitude. Here I am also thankful to Mr. Jean Rousset for his help and encouragement. He has taught me a lot about the simulation techniques. I thank him also for helping me solve many problems, which is indispensable for the implementation of this work.

I would like to thank Agilent-EEsof for its interest on this research, for funding and technical support of this thesis work. Concerning the engineers whom I had opportunities to work with, I'd like thank to Mr. George Estep for the facilities he has provided to me, which serve as the basis of my thesis work. I am also thankful to Mr. Arnaud Soury and Mr. Cédric Pujol for their kind help and encouragement.

I would like express my gratitude to Mr. Francis Calmon, my master advisor, who has led me to the world of the simulation and the PLL. I am very grateful for his kindness and encouragement.

It has been a great privilege to work in the XLIM lab and C2S2 department. First, I'd like to thank Rémond QUERE, responsible of C2S2, for his confidence and encourage-

ment. I am also thankful to all my colleagues in the laboratory for creating a comfortable and pleasant environment: Tibo, Audrey, FX, Abderrezak, Zhour, Vincent, Christophe, David, Grégoire, Alaa, Alaaeddine, Mouhamad, Turkan, Hassan, Ji, Ben, Philippe, Fabien, Jérôme... A special thank to Tibo and Audrey for their kind suggestions, jokes and spiritual support during the preparation of this thesis. I cherish the time spent with you all.

Also, I won't forget the help from the secretary Ms. Marie-Claude Rouge, and I am grateful for her kindness, availability and good humor that ease my thesis life.

Finally, I would like thank to my parents and my brother, who have been always ready to support. I am specially thankful to my maternal grandmother, who watched me grow up, guided me and devoted her love and a lot of time on me. Also, I am deeply grateful to Huiling, for her tender love, profound comprehension and encouragement all along the most difficult time. I hope this dissertation could be as beautiful as her love.

Contents

Table of Contents I

List of Figures V

List of Tables XI

General introduction 1

Chapter 1 : Fundamentals of the PLL 5

1.1 Introduction 7
1.2 PLL basics 7
1.2.1 Principles 7
1.2.2 Applications 8
1.2.3 Integer-N PLL and fractional-N PLL 11
1.3 Building blocks of the PLL 13
1.3.1 VCO 13
1.3.2 Divider 14
1.3.3 Phase detector 16
1.3.3.1 PD 16
1.3.3.2 PFD 18
1.3.3.3 Charge Pump (CP) 23
1.3.3.4 Summary of phase detector circuits 24
1.3.4 Loop filter 25
1.4 Idealized description of the PLL: linear model 27
1.4.1 Transfer function of the PLL 27
1.4.2 Noise transfer function 31
1.4.3 Description of two typical PLL structures 32
1.5 Conclusion 35
1.6 Bibliography 37

Chapter 2 : Steady-state Analysis of the PLL . . . 41

2.1 Introduction 43
2.2 Steady-state of the PLL 43
2.3 General steady-state analysis method 44
2.3.1 Time domain integration 45
2.3.2 Time domain Shooting method 45
2.3.3 Harmonic balance 46
2.3.4 RF circuit Simulators 47
2.4 Existing PLL steady-state analysis method 47
2.4.1 Brute force transistor-level simulation of the PLL 48
2.4.2 Controlled brute force transistor-level simulation of the PLL 49
2.4.3 Hierarchical simulation of the PLL 52
2.5 Proposed method 53
2.5.1 Fundamentals of the proposed method 53
2.5.2 Algorithm presentation 56
2.5.2.1 Introduction 56
2.5.2.2 Simplified algorithm 62
2.5.3 Algorithm description 64

2.5.3.1 Initial estimates . . . 65
2.5.3.2 VCO simulation . . . 66
2.5.3.3 Divider simulation . . . 67
2.5.3.4 Inner loop process . . . 67
2.5.3.5 Outer loop process . . . 70
2.5.3.6 Discussion on the algorithm termination conditions . . . 71
2.5.4 Numerical application – Step by step simulation of a PLL . . . 72
2.6 Conclusion . . . 81
2.7 Bibliography . . . 83

Chapter 3 : Noise Analysis of the PLL . . . 85
3.1 Introduction . . . 87
3.2 Noise in the electronic circuits . . . 87
3.2.1 Random noise and deterministic noise . . . 88
3.2.2 Basic definitions of the random noise . . . 88
3.2.3 Power spectral density of the noise . . . 90
3.2.4 White noise and colored noise . . . 90
3.2.5 Cyclo-stationary noise . . . 91
3.2.6 AM noise and PM noise . . . 92
3.3 Noise analysis of the PLL at the transistor level . . . 94
3.3.1 Conversion matrix . . . 94
3.3.2 Noise analysis based on envelope transient technique . . . 96
3.3.2.1 Introduction of envelope transient technique . . . 96
3.3.2.2 Noise analysis using envelope transient technique . . . 97
3.3.3 Monte Carlo method . . . 98
3.4 Noise modeling and analysis at the block level . . . 99
3.4.1 Introduction . . . 99
3.4.2 Behavioral modeling of the PLL blocks . . . 99
3.4.2.1 VCO . . . 102
3.4.2.2 PFD+CP . . . 105
3.4.2.3 Divider . . . 107
3.4.2.4 Low-pass filter . . . 109
3.4.3 Deterministic noise modeling . . . 109
3.4.4 Overall block-level noise simulation . . . 111
3.4.5 Simulation results . . . 112
3.4.5.1 Noise characterization of the various blocks . . . 112
3.4.5.2 Simulation results of PLL phase noise . . . 114
3.4.5.3 Simulation results of PLL deterministic noise . . . 115
3.4.5.4 Simulation time comparison . . . 117
3.5 Conclusion . . . 117
3.6 Bibliography . . . 118

Chapter 4 : Dynamic performances analysis of the PLL . . . 121
4.1 Introduction . . . 123
4.2 Principles . . . 123
4.2.1 Tracking process / Acquisition process . . . 123

4.2.2 Lock time . . . 125
4.3 Existing simulation methods . . . 126
4.3.1 PFD modeling . . . 126
4.3.1.1 Average duty cycle model . . . 126
4.3.1.2 Uniform time step sampling model . . . 128
4.3.2 VCO and VCO+divider modeling . . . 128
4.4 Proposed method . . . 129
4.4.1 VCO+divider modeling . . . 129
4.4.1.1 Model derivation . . . 129
4.4.1.2 Model implementation . . . 132
4.4.1.3 Model verification . . . 132
4.4.2 PFD+CP modeling . . . 135
4.4.2.1 Model derivation . . . 135
4.4.2.2 Model identification . . . 137
4.4.2.3 Model implementation and verification . . . 142
4.4.2.4 Numerical application of proposed modeling approach . . . 145
4.5 Conclusion . . . 147
4.6 Bibliography . . . 148

General conclusion . . . 149

Appendix $: Previous work on VCO noise modeling . . . 154
A.1 Phase noise of oscillator . . . 155
A.1.1 Impulse response to perturbation . . . 156
A.1.2 Noise representation . . . 158
A.2 Existing models . . . 159
A.2.1 Leeson model . . . 159
A.2.2 Hajimiri model . . . 161
A.2.3 Demir model . . . 164
A.3 Bibliography . . . 165
Glossary . . . 167
Publications and communications related to this work . . . 169

List of Figures

1.1 Basic structure of the PLL . . . 7
1.2 Application of the frequency synthesizer: GSM900 . . . 9
1.3 Diagram of the frequency synthesizer . . . 9
1.4 Application of the PLL: frequency demodulation . . . 10
1.5 Static voltage - frequency relationship of the VCO . . . 14
1.6 Frequency divider with the integer and programmable division ratio . . . 15
1.7 Characteristic of the mixer as phase detector . . . 17
1.8 Characteristic of the XOR circuit as phase detector . . . 17
1.9 Characteristic of the sequential phase detector . . . 18
1.10 PFD+CP+Filter . . . 18
1.11 Topology and the ideal behavior of the PFD . . . 19
1.12 State diagram of the PFD: ideal case . . . 19
1.13 Characteristic of the PFD: ideal case . . . 20
1.14 Characteristic of the PFD: dead zone . . . 20
1.15 Modified topology of PFD and the behavior of PFD . . . 21
1.16 State diagram of the PFD : with delay element . . . 22
1.17 Illustration of the PFD hysteresis . . . 22
1.18 Hysteresis of the PFD . . . 22
1.19 Average net duty cycle in the case of large frequency error . . . 23
1.20 Charge pump and the time sequence of the signals . . . 24
1.21 Loop filter . . . 26
1.22 Block diagram of the PLL . . . 28
1.23 Diagram of the PLL and the noise sources . . . 31
1.24 Noise transfer function for the different blocks . . . 31
1.25 Frequency response of the close-loop gain for the type II 2nd order PLL . . . 34
1.26 Frequency response of G_{err} for the type II 2nd order PLL . . . 34
1.27 Bode plots for the type II 2nd order PLL and the type II 3rd order PLL . . . 35
2.1 Discretization of the basic component: the capacitor . . . 45
2.2 Harmonic balance method . . . 47
2.3 Evolution of the VCO control voltage . . . 49
2.4 Controlled brute force transistor-level simulation of the PLL . . . 50
2.5 Start of the oscillation of the VCO with an external stimulation . . . 51
2.6 Hierarchical simulation with the idealized block loading conditions . . . 52
2.7 Diagram of the PLL . . . 53
2.8 Interconnection of the blocks and the definition of the signals . . . 54
2.9 Simulation of a block with the input signal and the load impedance . . . 54
2.10 Mechanism of the voltage substitution . . . 55
2.11 Block diagram of the PLL for the proposed algorithm . . . 56
2.12 Block diagram of the PLL with arbitrary cut . . . 57
2.13 Procedure of the proposed algorithm . . . 60
2.14 Simplified procedure of the proposed algorithm . . . 63
2.15 VCO: search for the nominal control voltage from the voltage-frequency curve . . . 65
2.16 VCO simulation bench . . . 66
2.17 Divider simulation bench . . . 67
2.18 Resolution of the inner loop equation . . . 67

2.19 Synthesis of PFD/CP load impedance profile by means of an ideal DC voltage source . . . 68
2.20 PFD+CP simulation bench in the initial iteration . . . 68
2.21 Relationship between the DC output current of CP and $\Delta\phi_L/N$ . . . 69
2.22 Relationship between DC output current of CP and $\Delta\phi_L/N$ - 2 periods . . . 69
2.23 PFD+CP simulation bench in the second and the following iterations . . . 70
2.24 Relationship between the DC output voltage of CP and $\Delta\phi_L/N$ - 2 periods 70
2.25 Loop filter simulation . . . 71
2.26 Verification of the new method . . . 72
2.27 VCO input-output static characteristic . . . 73
2.28 VCO output voltage $v_{out}(1,t)$ . . . 73
2.29 Divider output voltage $v_{out}(2,t)$ . . . 73
2.30 Relation $\overline{i_{cp}(t)}$ vs. phase shift $\Delta\phi_L/N$ . . . 74
2.31 Relation $\overline{v_{out}(3,t)}$ vs. phase shift $\Delta\phi_L/N$ . . . 75
2.32 Evolution of the input impedance of the VCO: $Z_{in}(1,k)$ . . . 75
2.33 Evolution of the input impedance of the divider: $Z_{in}(2,k)$ . . . 76
2.34 Evolution of the input impedance of the PFD/CP: $Z_{in}(3,k)$ . . . 76
2.35 Evolution of the input impedance of the filter: $Z_{in}(4,k)$ . . . 77
2.36 Evolution of the loop filter output voltage spectrum: $V_{out}(4,k)$ . . . 77
2.37 Nominal phase difference: new method . . . 78
2.38 Zoom of Fig. 2.37 . . . 78
2.39 Nominal phase difference: brute force simulation . . . 79
2.40 Zoom of Fig. 2.39 . . . 79
2.41 Influence of the number of harmonics (NH) on the relation $\overline{v_{out}(3,t)}$ vs. $\Delta\phi_L/N$: NH is sufficient when $NH = 768$ and $NH = 512$, and NH is NOT sufficient when $NH = 256$. . . . 81
3.1 Phase noise and the deterministic noise of the PLL . . . 88
3.2 Comparison between the white noise and the colored noise . . . 91
3.3 Correlation in cyclo-stationary noise . . . 92
3.4 Double sideband and single sideband . . . 93
3.5 Decomposition of AM noise and PM noise . . . 93
3.6 Model for the noise analysis . . . 95
3.7 Principle of envelope transient . . . 97
3.8 Model of a N-port block . . . 100
3.9 Model of a N-port block using a voltage/current formalism . . . 100
3.10 Norton equivalent circuit with noiseless circuit, noise sources and small signal perturbations at the block access . . . 101
3.11 VCO block: two port network . . . 102
3.12 VCO block: Norton equivalent circuit . . . 102
3.13 VCO model: I/V at the input and phase at the output . . . 103
3.14 VCO equivalent network . . . 104
3.15 PFD+CP block . . . 105
3.16 PFD+CP block: voltage domain small-signal equivalent network . . . 105
3.17 PFD+CP model: phases at the input and I/V at the output . . . 106
3.18 PFD+CP equivalent network . . . 107
3.19 Divider block: two-port network . . . 108
3.20 Divider equivalent network . . . 108

3.21 Cause of the deterministic noise: current mismatch . . . 109
3.22 PFD+CP equivalent network with deterministic source . . . 110
3.23 Block-level PLL model for noise and small perturbation analysis . . . 111
3.24 Internal VCO phase noise PSD: $S_{n_o}(f)$ . . . 112
3.25 Internal divider phase noise PSD: $S_{n_d}(f)$ . . . 113
3.26 Internal PFD/CP current noise PSD: $S_{n_{cp}}(f)$ . . . 113
3.27 Phase noise of the PLL: Proposed method and brute force transistor-level simulation . . . 114
3.28 Noise contribution of each block: proposed method . . . 115
3.29 Discrete PSD of the CP current: $S_{d_{cp}}$ . . . 115
3.30 Deterministic noise: proposed model . . . 116
3.31 Deterministic noise: brute-force transistor-level simulation . . . 116
4.1 Relationship between the different frequency ranges . . . 124
4.2 Lock time definition . . . 125
4.3 Model of Jess Chen . . . 127
4.4 Definition of the input and output signals of VCO+divider block . . . 129
4.5 v_c vs. VCO output frequency $f_o(v_c)$ . . . 130
4.6 v_c vs. divider output spectrum coefficients - real part $Re\left(V_k^d(v_c)\right)$ . . . 131
4.7 v_c vs. divider output spectrum coefficients - imaginary part $Im\left(V_k^d(v_c)\right)$ . . . 131
4.8 PLL simulation with the proposed VCO+divider model . . . 132
4.9 Lock time simulation: waveform of f_{ref} and f_{div} (First jump of f_{ref}: 264MHz → 271.75MHz, second jump of f_{ref}: 271.75MHz → 264MHz) . . . 133
4.10 Sweep f_{ref} to measure capture range and hold range . . . 133
4.11 Lower boundaries of hold and capture range: (f_{ref} upward: 245MHz → 250MHz, f_{ref} downward: 250MHz → 245MHz) - C1: lower capture boundary, H1: lower hold boundary . . . 134
4.12 Upper boundaries of hold and capture range: (f_{ref} downward: f_{ref} 282MHz → 277MHz, f_{ref} upward: 277MHz → 283MHz) - C2: upper capture boundary, H2: upper hold boundary . . . 134
4.13 PFD+CP input and output signals . . . 135
4.14 PFD+CP model . . . 136
4.15 Input and output signals of PFD+CP: $F_{ref} < F_{div}$, $F_{ref} = Q \cdot F_{diff}$ with $Q = 4$ . . . 138
4.16 Waveform of i_{cp}: $F_{ref} < F_{div}$, $F_{diff} = 2MHz$ and $F_{ref} = 19 \cdot F_{diff}$ . . . 138
4.17 Input and output signals of PFD+CP: $F_{ref} < F_{div}$, $F_{ref} = \rho \cdot F_{diff}$ with $\rho = 3.8$ . . . 139
4.18 Evolution of the envelope of i_{cp} when the PLL approaches the locked state 140
4.19 Input and output signals of PFD+CP: $F_{ref} > F_{div}$, $F_{div} = Q \cdot F_{diff}$ with $Q = 4$ . . . 141
4.20 Model identification bench: $F_{ref} < F_{div}$ . . . 142
4.21 Model identification bench: $F_{ref} > F_{div}$ . . . 142
4.22 Schematic to verify the PFD+CP model . . . 143
4.23 i_{cp} : Comparison between PFD+CP model simulation and PFD+CP circuit simulation, $f_{ref} = 38MHz$, $f_{div} = 40MHz$ . . . 143
4.24 Zoom of Fig. 4.23 . . . 144
4.25 i_{cp}: Comparison between PFD+CP model simulation and PFD+CP circuit simulation, $f_{ref} = f_{div} = 38MHz$ . . . 144

4.26 PLL simulation with the configuration CF3 145
4.27 Lock time simulation - CF1: full transistor-level; CF2: Mixed transistor / block-level; CF3: Full block level 146
4.28 Lower boundaries of hold and capture range - CF1: full transistor-level; CF2: Mixed transistor / block-level; CF3: Full block level; C1: lower capture boundary, H1: lower hold boundary 146
4.29 Upper boundaries of hold and capture range - CF1: full transistor-level; CF2: Mixed transistor / block-level; CF3: Full block level; C2: upper capture boundary, H2: upper hold boundary 147
A.1 Influence of the phase noise on the output spectrum of the oscillator 155
A.2 Influence of the phase noise on the performance of the receiver 156
A.3 Impulse response of the phase 157
A.4 Limit cycle of the oscillator 157
A.5 Definition of $\mathcal{L}(\Delta\omega)$ 159
A.6 Typical curve of the oscillator phase noise predicted by Leeson model . . . 160
A.7 (a) Schematic of an oscillator with noise, (b) Schematic of a Colpitts oscillator 161
A.8 Noise conversion: from noise source to the voltage output of the VCO . . 162
A.9 Noise conversion in the Hajimiri model 163
A.10 Noise conversion 164

List of Tables

1.1 Applications of the wide band PLL and narrow band PLL 11
1.2 Example of a frequency synthesizer realized by an integer-N PLL 12
1.3 Example of a frequency synthesizer realized by a fractional-N PLL 12
2.1 RF circuit simulators 47
2.2 Brute force transistor-level simulation. N: Division ratio; NH: Number of harmonic 49
2.3 List of simulation time of each step 80
2.4 Comparison of the simulation time 80
3.1 Deterministic noise: comparison between the proposed method and the brute force simulation 116
3.2 Noise analysis time of the proposed method and the transistor-level simulation 117
4.1 Simulation time comparison 135
4.2 Simulation time comparison between three methods 145

General introduction

Over the last decade, more and more wireless communication devices enter our life, such as mobile phones, Bluetooth, Wi-Fi, WiMAX, RFID, etc. At the same time the researchers and engineers continue to work on these devices to improve the data transfer rate and minimize their size. These devices often operate at a high frequency from several hundreds MHz to tens of GHz with an integration of thousands of transistors.

In these wireless communication devices, a radio frequency (RF) transceiver is indispensable to realize the high speed data transfer, and a stable periodic signal in the RF transceiver is essential to provide a timing basis for the purpose of frequency synthesis, frequency demodulation, etc. Among all the techniques that can provide that timing basis, phase lock technique is for now the most widely used means.

This phase lock technique, realized by phase-locked loop (PLL) which is basically a close-loop frequency control system, dates from 1930s. At that time, PLL was first used to extract the audio signal from the modulated carrier. Although this synchronous technique was very efficient, the realization of a such circuit at the era of electronic tube was very expensive. The massive use of PLL circuit begins with the development of the monolithic integrated circuit. At the same time, the phase locked principle has extended from AM and FM demodulation to many synchronization applications such as clock recovery, frequency synthesis, clock generation in microprocessor, DTMF/FSK decoder, etc.

Recently more and more PLL circuits are implemented in a high-level integration SoC (System-on-Chip) structure to achieve a smaller size. The design and verification of such circuits need a rigorous simulation and optimization method. The actual commonly used simulation methods are based on two types of algorithms: time domain analysis (shooting method) and frequency domain analysis (Harmonic balance). The former is suitable for the analog low-frequency signal and the digital signal, and the latter is preferable for the RF high-frequency continuous wave signal. The PLL circuits, however, covers all these types of signals: analog low-frequency signal (in loop filter, charge pump), digital signal (in frequency divider, phase-frequency detector) and RF continuous wave signal (in VCO). It therefore poses a great challenge for the existing analysis methods. Moreover, the PLL constitutes a stiff system with widely separating time constants spreading from hundreds of Hz to tens of GHz, which make the traditional analysis methods more difficult and sometimes impractical.

Generally, the verification of the PLL circuit will involve the following main aspects:

- Large signal steady state response
- Noise characteristics (phase noise, deterministic noise)
- Dynamic performances (lock time, hold range, capture range, etc)

The conventional approach is to analyze them through a transistor-level simulation, which comprises all the implementation details of the PLL. However, this approach may encounter very huge difficulties due to the complexity of the PLL circuit. Consequently, all figure-of-merits of the PLL are not fully verified, and the time spent on the verification can be extremely long. Therefore, an appropriate block-level simulation approach is often necessary, which is aimed to bypass some implementation details of the PLL circuit while retaining the main characteristics.

This thesis is involved in this context. It is aimed to bring forward some new algorithms and models which allow an accurate and fast analysis and verification of the PLL characteristics.

The thesis is organized as follows:

Chapter 1 presents fundamentals of the PLL circuit and its basic characteristics, which constitute the objects for the analysis techniques presented in the subsequent chapters.

Chapter 2 deals with the large-signal steady-state response of the PLL. First, the problem of the transistor-level verification is presented: long simulation time, large memory consummation. After introducing the principal existing methods, we present a new method to compute the large-signal steady-state solution in a piecewise and iterative way. The piecewise flexibility introduced by the new method, along with a careful consideration of the block interface conditions, provides very efficient and accurate solutions for the PLL steady-state response.

Chapter 3 is dedicated to the noise analysis of the PLL. We first introduce the noise characteristics of the PLL and the existing analysis methods. Then, based on the new steady-state method of chapter 2, we have proposed a new modeling for the PLL building blocks, which allows the computation of the random phase noise and the deterministic noise of the PLL rapidly and accurately.

Chapter 4 deals with the simulation of the PLL dynamic performances. The difficulties here is how to retain the main characteristics of the PLL without losing the simulation precision. Once again, based on the piecewise steady-state method proposed in chapter 2, we propose a modeling methodology of the various blocks, which allows an accurate and fast simulation of the important dynamic characteristics of the PLL. A comparison with the transistor-level simulation has shown an enormous speed-up factor with virtually the same accuracy and detail level.

In the end, we make a summary on the main contributions of this thesis work, and present the perspectives.

Chapter 1 :

Fundamentals of the PLL

1.1 Introduction

This chapter is focused on the fundamentals of the PLL: its operations, performances, building blocks and basic characteristics, which define the basis for the modeling and the simulation of the PLL in the following chapters.

At the beginning of the chapter, some fundamental notions on the PLL and its applications are introduced, and key characteristics of the PLL are described. Next, the main building blocks of the PLL are presented: VCO, loop filter, frequency divider, and especially phase/frequency detector. Then, a brief simplified description of the PLL operation is given with the help of the transfer function, from which some important concepts are derived, such as type and order of the PLL, phase margin, etc. In the end, two typical PLL structures and their characteristics are presented.

1.2 PLL basics

In this section, an overview of the PLL basics is given concerning the principles of the PLL, its applications and two kinds of the PLLs: integer-N PLL and fractional-N PLL.

1.2.1 Principles

The PLL is a non-linear feed-back system in which the output signal is forced to track the input signal frequency and phase. A basic PLL has three components, as showed in Fig. 1.1.

- Phase detector (PD)
- Low Pass Filter (LFP)
- Voltage-Controlled Oscillator (VCO)

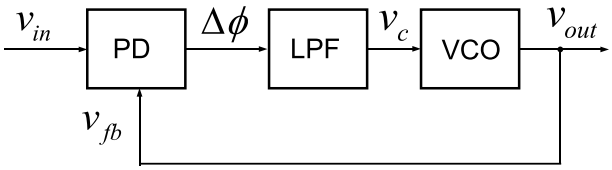


FIG. 1.1: *Basic structure of the PLL*

The operation of the PLL is described as follows [1] [2]: PD compares the phases of its two signals; the reference signal v_{in} and the feedback signal v_{fb} produced by the VCO;

and then generates a signal representing the phase difference $\Delta\phi$ of the two input signals, called the phase error. The phase error is then filtered by the low-pass filter, and applied to the input of the VCO. The input voltage of the VCO v_c adjusts the VCO frequency in the direction of reducing the phase error $\Delta\phi$ until it becomes zero or a small constant value. When the transient stage dies away, we say that the PLL is in the locked state.

Based on the above description, we can observe two different states of the PLL: the locked state and the unlocked state. In the locked state, the signals in the PLL are periodic, and the output frequency is constant. While in the unlocked state, the PLL changes its frequency and phase continuously, and always searches for a stable equilibrium.

Accordingly, we use two different types of characteristics to describe the performances of the PLL: unlocked and locked state characteristics. The unlocked state is characterized by the following characteristics, called the dynamic performances:

- Lock time: time needed for the PLL to return to the locked state (Ref. Chapter 4)
- Capture range: frequency range within which the PLL can lock on from an unlocked state (Ref. Chapter 4)
- Hold range: frequency range within which the PLL can maintain its phase tracking (Ref. Chapter 4)

The locked state is characterized by the following performances:

- Phase noise (in the frequency domain) or jitter (in the time domain): phase perturbation at the output of the PLL (Ref. Chapter 3)
- Deterministic noise or Spur: perturbation at the output of the PLL occurring at the reference frequency and its harmonic frequencies (Ref. Chapter 3)
- Phase margin: parameter describing the stability of the PLL (Ref. Chapter 1)

The details of these characteristics of the PLL will be discussed throughout the rest of this thesis.

1.2.2 Applications

The PLL is widely used in many fields. One of the most important applications is the frequency synthesis. The frequency synthesizer can generate a signal whose frequency is a multiple of a reference frequency [3]. In the emission-reception chain, the frequency synthesizer is often used as a LO, showed in Fig. 1.2.

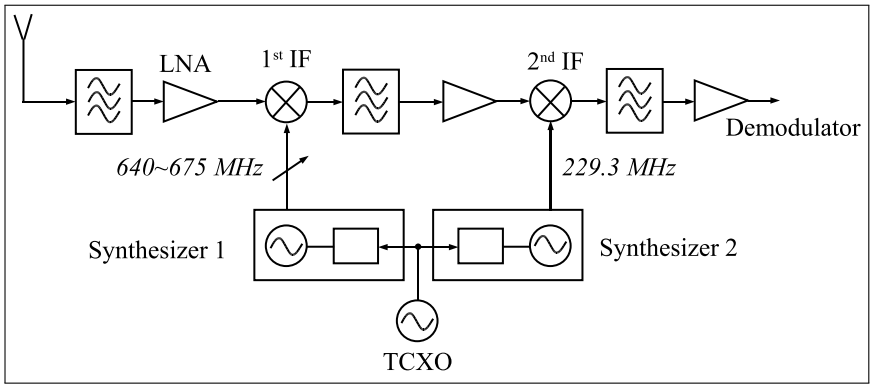


FIG. 1.2: *Application of the frequency synthesizer: GSM900*

The reference clock TCXO produces a stable but low frequency called reference frequency at the input of the synthesizer. The synthesizer then delivers a clock signal whose frequency is a multiple of the reference frequency.

To generate a high frequency at the output of the synthesizer, a frequency divider is added in the feedback loop, as shown in Fig. 1.3.

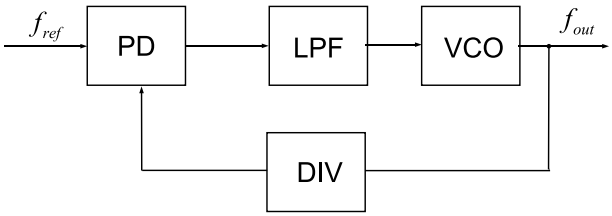


FIG. 1.3: *Diagram of the frequency synthesizer*

Hence, the output frequency of the synthesizer can be described by the following equation:

$$f_{out} = n \cdot f_{ref} \quad (\mathrm{n} \geq 1) \tag{1.1}$$

The division ratio may be fixed or programmable, integer or fractional, depending on the requirement of the design. The division ratio n may take large value; this ranges from 2 to several thousands.

The input of the synthesizer is often realized by a crystal oscillator, for example, TCXO (Temperature Compensated Crystal Oscillator), which can generate a very stable clock signal with little noise. However, its output frequency is generally very low compared with LO frequency needed for RF applications.

Recently, new techniques are developed for frequency synthesis, like DLL (Delay lock loops) [4] and DDS (Direct Digital Synthesizer) [5]. We will not consider these mostly digital technique in this thesis.

For the frequency synthesizer, a very low phase noise is highly required, because the

phase noise brought into the emission-reception chain can dramatically deteriorate the performance of the system. At the same time, the dynamic performance of the synthesizer is very crucial too, because if the communication channel changes, the synthesizer should jump from one frequency to another specified frequency within a short time interval.

For the other applications of the PLL, the key characteristic could be either phase noise or dynamic performances. For example, for a frequency hopping system which switches its carrier rapidly among many frequency channels, the most critical characteristic is the lock time. Yet for a satellite communication, where the signal noise ratio or SNR of the input signal is very small, one requires the PLL to generate the smallest noise possible. In this case, the specification about the phase noise will be the emphasis of the design.

Frequency demodulation is another application of the PLL. Illustrated in Fig. 1.4, the output voltage of the filter represents in fact the input frequency of the PLL. Therefore, if there is any frequency variation at the input of PD, PLL should react rapidly and adjust the VCO frequency to track the input frequency. Hence the PLL should have a good dynamic performance.

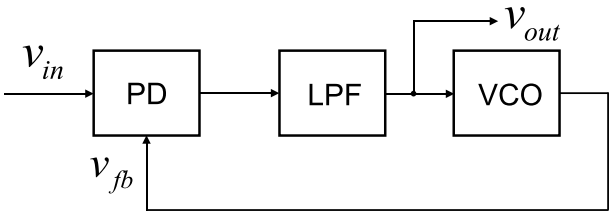


FIG. 1.4: *Application of the PLL: frequency demodulation*

The other applications of the PLL include the clock and data recovery circuit, the phase demodulation, the processor clock generation, etc.

The PLLs can be classified into wide band PLLs and narrow band PLLs [2] [6], depending on the loop bandwidth. As we will see from chapter 3 and 4, the loop bandwidth of the PLL has important impacts on the phase noise as well as on the dynamic performances of the PLL.

The loop bandwidth affects the contributions to the output noise of the various blocks of the PLL; the intrinsic phase noise of the VCO is largely attenuated in the loop bandwidth and kept unchanged outside the bandwidth; on the contrary, the noise of the divider/PD is much filtered outside the loop bandwidth. As a result, by altering the bandwidth of the PLL one can optimize the noise performance of the PLL.

With regard to the dynamic performance, a large bandwidth allows reducing the lock time and accelerating the response to the input variation.

In the table 1.1, the advantages and the disadvantages of the wide band and narrow band PLL are compared, and their appropriate applications are also indicated.

Performances	**Applications**	**Narrow Band**	**Wide band**
Short lock time	Frequency hopping, Frequency demodulation	-	+
Attenuate the noise from the input or within the bandwidth	Communication by satellite where the SNR (Signal noise ratio) is small	+	-
Attenuate the noise outside the bandwidth	VCO with a poor noise spectrum outside the bandwidth	-	+
Fine and precise frequency synthesis	Frequency synthesis with numerous communication channels, GSM	+	-

TAB. 1.1: *Applications of the wide band PLL and narrow band PLL*

For the purpose of the stability, the bandwidth of the PLL is often designed to be 1/10 of the reference frequency or smaller, which directly impacts on the PLL dynamic performances.

As a result, it's very difficult to satisfy the phase noise specification, the stability and the dynamic performances at the same time. On one side, the narrow bandwidth allows eliminating the perturbation within the bandwidth, i.e., the noise from the reference signal and the PD/divider, whereas it also limits the dynamic performance of the PLL and increases the lock time. On the other side, the wide bandwidth allows the PLL to track closely the reference variation, but it introduces more perturbation from the phase detector, producing ripples at the VCO input. A compromise between the noise and the dynamic performance can be found with a fractional PLL, whose division ration is no longer an integer, but a fraction, as will be introduced in the following section.

1.2.3 Integer-N PLL and fractional-N PLL

The PLL with an integer division ratio is referred to as an integer-N PLL, while the PLL with a fractional division ratio is called a fractional-N PLL or simply a fractional PLL.

To see how a fractional-N PLL can reduce both the lock time and the phase noise of the PLL, first take an example of an integer-N PLL based frequency synthesizer, whose output frequency is:

$$f_{out} = N \cdot f_{ref} \tag{1.2}$$

With an integer-N PLL, the channel spacing has to be an integer multiple of the reference frequency. To vary the output frequency f_{out} from 800 MHz to 806 MHz with a

$2MHz$ channel spacing, the maximal reference frequency is $f_{ref} = 2MHz$. The division ratio N should then change from 400 to 403, as shown in the table 1.2. The bandwidth of the PLL will be 200 kHz, if it's designed to be 1/10 of the reference frequency.

Channel	**1**	**2**	**3**	**4**
f_{out} (MHz)	800	802	804	806
N(Integer)	400	401	402	403
f_{ref} (MHz)	2	2	2	2

TAB. 1.2: *Example of a frequency synthesizer realized by an integer-N PLL*

On the other hand, if one uses a fractional-N PLL to realize the same channel spacing, we have more choice for the reference frequency and division ratio. An example is showed in the table 1.3, where the reference frequency is set to $8MHz$ and the division ratio varies from 100 to 100.75.

Channel	**1**	**2**	**3**	**4**
f_{out} (MHz)	800	802	804	806
N(Fractional)	100	100.25	100.5	100.75
f_{ref} (MHz)	8	8	8	8

TAB. 1.3: *Example of a frequency synthesizer realized by a fractional-N PLL*

Consequently, some interesting results can be found. First of all, the increase of the reference frequency (by 4 times) leads to an increase in the bandwidth, which is favorable for the short lock time. Secondly, as we will see from the chapter 3, the noise contribution from the PD and the divider is proportional to N. In this example, N decreases 4 times, meaning that the noise contribution of the PD or the divider will reduce by 12 dB. Thirdly, as we explained before, the VCO noise contribution within the bandwidth can be well attenuated. So the increase of the loop bandwidth will allow more VCO noise to be eliminated. Finally, as we can see from the above table, the frequency resolution is no longer equal to the reference frequency, but a fraction of the reference frequency. Hence, with the fractional division a finer frequency resolution can be realized at lower division ratio and higher reference frequency.

The divider of the fractional-N can be realized by a common divider plus an accumulator [6] [7] . A prescaler is used to change its modulo between N and $N + 1$. So at the output of the divider, the average division ratio is a real number between N and $N + 1$, depending on how many times N appears. If the probability for N is p $(0 < p < 1)$, then the average division ratio can be expressed like below:

$$n = N \cdot p + (N + 1) \cdot (1 - p) = N + f \tag{1.3}$$

where f is a fraction.

However, the sudden change of the division ratio between N and $N+1$ causes quantification noise. Some techniques are proposed to compensate quantification noise, but the result is not always satisfying. A better solution is to use the $\Sigma\Delta$ converter, which can eliminate the quantification noise by a noise shaping mechanism. The details of the $\Sigma\Delta$ converter can be found in many bibliographies [8] [9] [10].

1.3 Building blocks of the PLL

After introducing the basic concepts of the PLL, we look into the structure of the PLL, and briefly describe the main building blocks in the PLL: VCO, frequency divider, phase detector and loop filter.

1.3.1 VCO

A VCO is an essential element of the PLL, which generates an output signal whose frequency is variable in terms of the input voltage. The output voltage of VCO can be roughly as below [1] [11] [12]:

$$y(t) = A\cos(\omega_0 t + \phi(t)) = A\cos\left(\omega_0 t + \int_0^t g(\tau, V_{in})\, d\tau\right) \tag{1.4}$$

where A is the amplitude, V_{in} is the VCO input voltage, and ω_0 is the free angular frequency when $V_{in} = 0$. The function $g(\tau, V_{in})$ represents the non-linear relationship between V_{in} and the output frequency of the VCO, as illustrated in Fig. 1.5. In the ideal case, the function $g(\tau, V_{in})$ is linear and time-invariable, which can be expressed like below:

$$g(V_{in}, \tau) = K_{vco} V_{in} \tag{1.5}$$

where K_{vco} (rad/s/V) is the static frequency sensitivity.

Considering the equations (1.4) and (1.5), we have:

$$\phi(t) = \int_{-\infty}^{t} K_{vco} V_{in}(\tau)\, d\tau \tag{1.6}$$

Through Laplace transform of the equation (1.6), we get the transfer function of the VCO:

$$\frac{\phi(s)}{V_{in}(s)} = \frac{K_{vco}}{s} \tag{1.7}$$

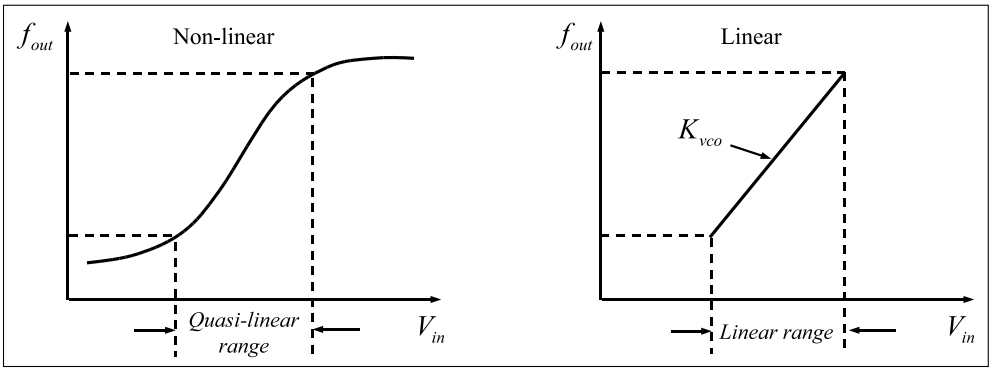


FIG. 1.5: *Static voltage - frequency relationship of the VCO*

According to the equation (1.6), the phase of the VCO depends not only on the actual value of the input voltage, but also on its previous value.

The characteristics of the VCO may include:

- Phase noise
- Frequency sensitivity
- Tuning frequency range
- Output signal power
- Pulling: frequency sensitivity with the output load variation
- Pushing: frequency sensitivity with the source voltage variation
- Power consumption

To avoid the pulling effect and generate a proper output power, a buffer circuit is inserted at the output of the VCO circuit [13]. The buffer circuit has a good reverse isolation and a high impedance. It can also isolate the VCO from the on-chip noise.

In general, there are two types of VCOs: those using a resonator to select the oscillation frequency (e.g., LC-VCO) and those operating on the relaxation principle (e.g., ring VCO). The details of the realization of the VCO can be found in the bibliographies [11] [12] [14].

1.3.2 Divider

If the division ratio is an integer, the VCO output frequency can be expressed by $f_{vco} = f_{ref} \cdot N$. To realize this, a divider with the variable modulo is often used. [2] [7]

As illustrated in Fig. 1.6, the divider is composed of 3 parts:

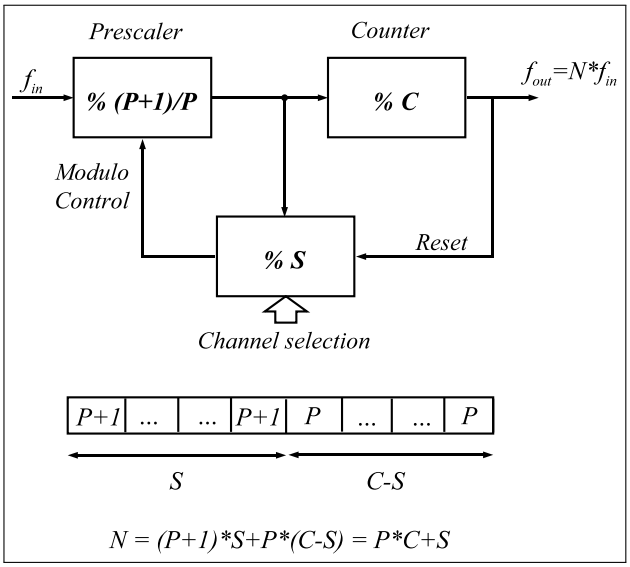


FIG. 1.6: *Frequency divider with the integer and programmable division ratio*

- A prescaler which divides the input frequency by P or $P+1$ according to the control signal
- A counter C which divides the frequency by C
- A counter S which selects the frequency channel and divides the frequency by S, where S is determined by the input digital signal and can vary from 0 to the maximal number of channel $S_{\max}$. $(S < C)$

The behavior of this divider is briefly described as follows: when the modulo of the prescaler is set to be $P + 1$, the counter C and the counter S accumulate and it spends $(P + 1) \cdot S$ cycles to make the counter S becomes full first. Then the modulo is switched to P , and it takes another $P \cdot (C - S)$ cycles to make the counter C become full. In total, a complete divider output cycle contains N input cycles with N equal to:

$$N = (P + 1) S + P (C - S) = P \cdot C + S \tag{1.8}$$

With this programmable integer divider, the VCO frequency vary from $P \cdot C \cdot f_{ref}$ to $(P \cdot C + S_{\max}) \cdot f_{ref}$.

To drive the succeeding circuit, a buffer circuit is added at the output stage of the divider [15]. The buffer circuit is generally realized by an emitter-follower , whose input impedance is very high [16].

1.3.3 Phase detector

An ideal phase detector can produce a signal whose magnitude is proportional to the phase difference of the input signals. The signal representing the phase difference can be a voltage or a current, depending on the structure of the phase detector.

In general, there exist two types of structure for the phase detector:

- PD (Phase detector): the phase difference is represented by the average output voltage of the PD.
- PFD (Phase Frequency Detector) + CP (Charge pump): the phase difference is represented by the average output current of the CP.

Next, we introduce respectively these two structures of phase detector and their advantages and disadvantages.

1.3.3.1 PD

A PD can be realized by an analog circuit (e.g., mixer), a digital XOR circuit, or a RS latch circuit, which are briefly described below. [1] [7] [11] [17]

A. PD with Mixer

If the two inputs of the PD are sinusoidal, they can be expressed as $v_1 = V_1 \cos(\omega t + \phi_1)$ and $v_2 = V_2 \cos(\omega t + \phi_2)$. If one neglects the high-frequency component of the output signal which is afterward eliminated by a low-pass filter, the output of the mixer can be approximated as:

$$V = v_1 \cdot v_2 \approx \frac{kV_1V_2}{2} \cos(\phi_1 - \phi_2) = \frac{kV_1V_2}{2} \cos(\Delta\phi) \tag{1.9}$$

When $\Delta\phi$ lies in the vicinity of $\pm\pi/2$, the relationship between the output V and the phase difference $\Delta\phi$ becomes nearly linear, as showed in Fig. 1.7. This means the two input signals of the PD should have a phase shift of $\pm\pi/2$ to make the PLL locked, hence

$$V = \frac{kV_1V_2}{2} (\Delta\phi \pm \pi/2) \tag{1.10}$$

B. PD with XOR circuit

As known, the output of the XOR circuit is high if the states of the two inputs are different. The characteristic of the XOR circuit as phase detector is illustrated in Fig. 1.8:

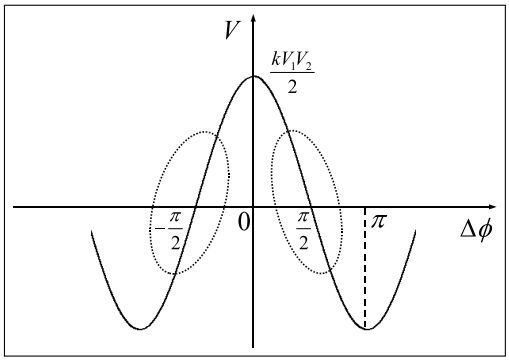


FIG. 1.7: *Characteristic of the mixer as phase detector*

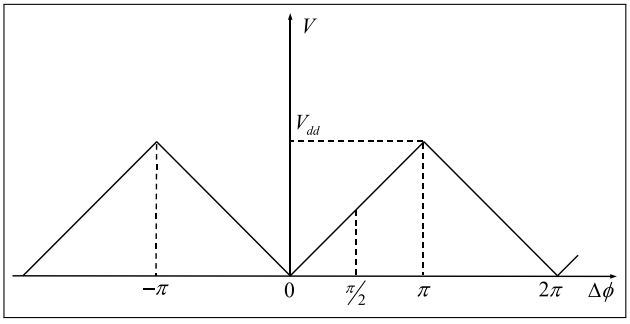


FIG. 1.8: *Characteristic of the XOR circuit as phase detector*

Hence, the gain for this phase detector is:

$$K_D = V_{dd}/\pi \tag{1.11}$$

The linear range of this characteristic is between 0 and π and the equilibrium point is located at $\Delta\phi = \pi/2$.

C. PD with a RS latch circuit

The most simple sequential phase detector can be realized by the RS latch. The latch works on the signal edge instead of state. A valid edge at the input of the latch leads to a state change at the output of the circuit. Therefore, the relationship between the phase difference and the output is showed in Fig. 1.9. The curve is linear in the range $[0, 2\pi]$. [18]

Therefore, the gain of this phase detector is:

$$K_D = V_{dd}/2\pi \tag{1.12}$$

Compared with the XOR circuit, the linear range of the sequential circuit is larger ($[0, 2\pi]$ vs. $[0, \pi]$). The equilibrium point locates at the middle of the linear range where $\Delta\phi = \pi$.

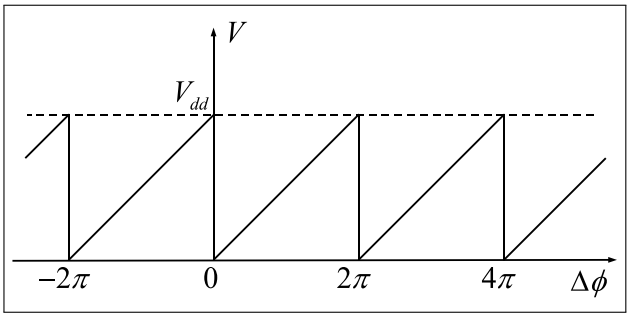


FIG. 1.9: *Characteristic of the sequential phase detector*

1.3.3.2 PFD

The structure of the PFD+CP is depicted in Fig. 1.10. In the next two sections, we describe roughly the PFD and the charge pump. The use of a charge pump along with the passive filter forms an type II PLL, which has advantages such as large hold range, zero static phase error, etc. More details can be found in [2] [19].

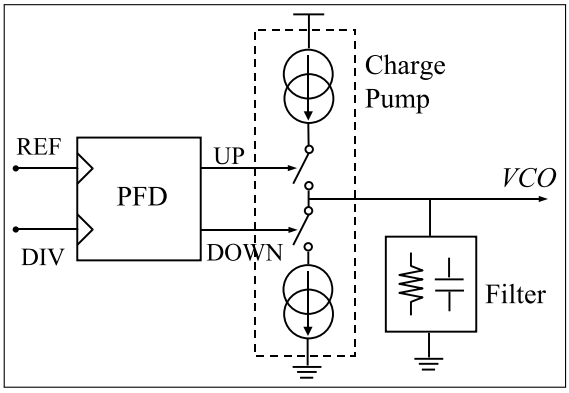


FIG. 1.10: *PFD+CP+Filter*

The topology of a typical PFD is illustrated in Fig. 1.11. The circuit includes two D-latches and one AND gate. The D-latch changes its logic state along with the edge of the input signal but it's not sensitive to the duty cycle of the input signal. The input terminal D is connected to the logic 1.

The behavior of the PFD follows some simple principles: A valid REF signal edge sets the signal UP to high state; A valid DIV signal edge sets the signal DOWN to high state; When UP and DOWN are at the high state simultaneously, the PFD is reset to zero; Two consecutive REF or DIV signal edges will maintain the output states of UP or DOWN.

As a result of the above principles, the pulse width of one output signal (UP or DOWN) is proportional to the phase error, and another output signal is a glitch.

The duty cycle of the signal UP or DOWN is the ratio of the pulse width to the period of the signal, denoted as d_{up} and d_{dn} ($0 < d_{up} < 1$, $0 < d_{dn} < 1$). It's necessary to define

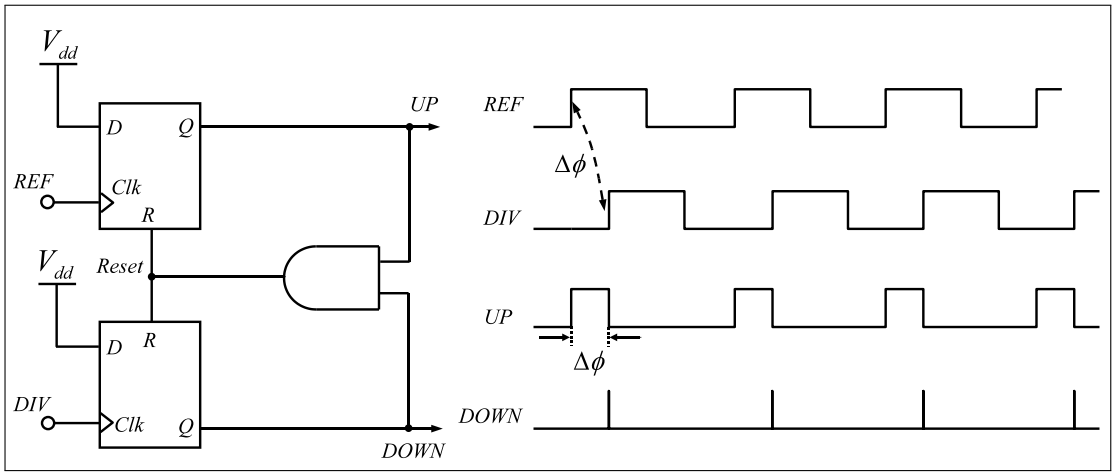


FIG. 1.11: *Topology and the ideal behavior of the PFD*

a net duty cycle d: $d = d_{up} - d_{dn}$ $(-1 < d < 1)$. As we can see, the phase difference in the range $[-2\pi, 2\pi]$ can be represented directly by the net duty cycle multiplied by 2π.

$$\Delta\phi = 2\pi \cdot d \tag{1.13}$$

A. State diagram

In the ideal case, the behavior of the PFD can be illustrated by the state diagram in Fig. 1.12. The three possible states of the PFD are NUL, UP and DN.

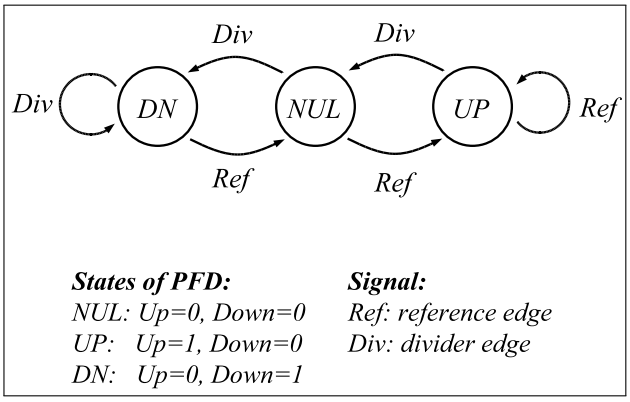


FIG. 1.12: *State diagram of the PFD: ideal case*

From Fig. 1.12 we can see that the Div edge always tries to drive the state towards DN, whatever the initial state is UP, NUL or DN. For example, DN → (Div) → DN, NUL → (Div) → DN, UP → (Div) → NUL → (Div) → DN. In contrast, the REF edge tends to drive the state towards UP. Ideally, the state CLR (Up=1 and Down=1) is neglected because it is transient and very short.

B. Characteristic of the PFD and dead zone

In the ideal case, the relationship between the phase difference and the net duty cycle of the PFD is shown in Fig. 1.13.

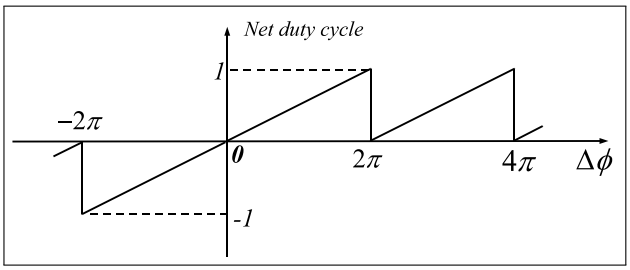


FIG. 1.13: *Characteristic of the PFD: ideal case*

The negative duty cycle means that the phase of the divider is ahead of that of the reference. Ideally, when the phase difference is null, the duty cycle of the PFD should be zero. Since the characteristic of the PFD is periodic outside the range $[-2\pi, 2\pi]$, the net duty cycle can be written as follows:

$$d = \Delta\phi \% 2\pi \tag{1.14}$$

where % represents modulo.

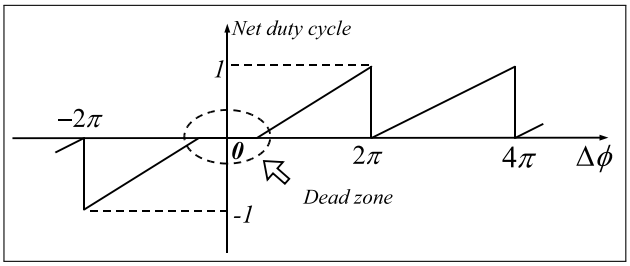


FIG. 1.14: *Characteristic of the PFD: dead zone*

However, non-idealities of PFD and CP circuit cause an important defect known as dead zone (shown in Fig. 1.14), which largely affects the performance of the PLL [2] [7]. The behavior of the dead zone is as follows: if the phase difference becomes very small, the output of PFD almost does not vary with the phase difference. Hence, the PLL can not adjust the frequency and phase of VCO. The PLL is thus becomes unstable and very hard to enter the locked state.

C. Delay element

To solve the dead zone problem, a delay element is inserted after the AND gate, as shown in Fig. 1.15. With this delay, the signals UP and DOWN do not return to zero immediately after the signals REF and DIV become to the high state simultaneously. On

the contrary, the signals UP and DOWN remain in high state long enough to avoid the glitches. [2] [3] [11]

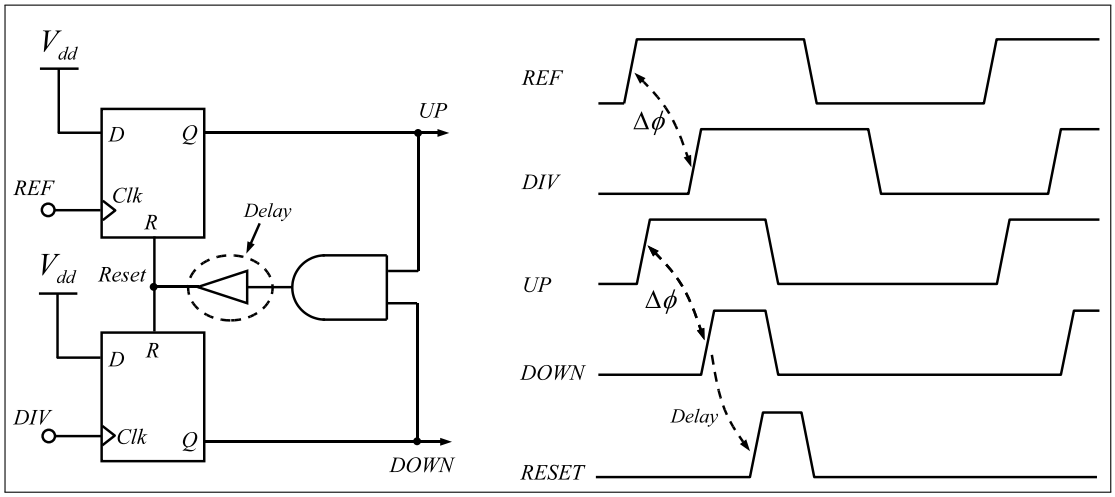


FIG. 1.15: *Modified topology of PFD and the behavior of PFD*

With the new structure of the PFD, the phase difference is no longer represented by the pulse width of UP or DOWN, but by the edge difference between the edges of UP and DOWN, as shown in Fig. 1.15. Moreover, the pulse width of UP and DOWN are always greater than the delay time. The delay time can be set large enough by the designer, thus the dead zone could be largely eliminated.

Note that the signals UP and DOWN being at high state simultaneously, means that the two current sources of CP circulate simultaneously and no current should flow out of the CP. Therefore, the two current sources should be designed perfectly symmetrical; otherwise the current flowing into/out of the CP will discharge/charge the filter and create the voltage ripple at the input of VCO.

Now we return to the state diagram of the PFD. With the delay element, the state CLR (Up=1, Down=1) can no longer be ignored. So there will be four states to describe the operation of the PFD, as shown in Fig. 1.16.

D. Hysteresis

Another important characteristic of the PFD is hysteresis [2] [20] [21], which is described briefly below. When the phase difference $\Delta\phi$ increases, the net duty cycle will follow the trajectory shown in Fig. 1.17 (a). On the contrary, when $\Delta\phi$ decreases, the net duty cycle will follow another trajectory shown in Fig. 1.17 (b). The reason is that the PFD interprets the phase relationship differently in these two cases [2]. The complete characteristic of the PFD is illustrated in Fig. 1.18.

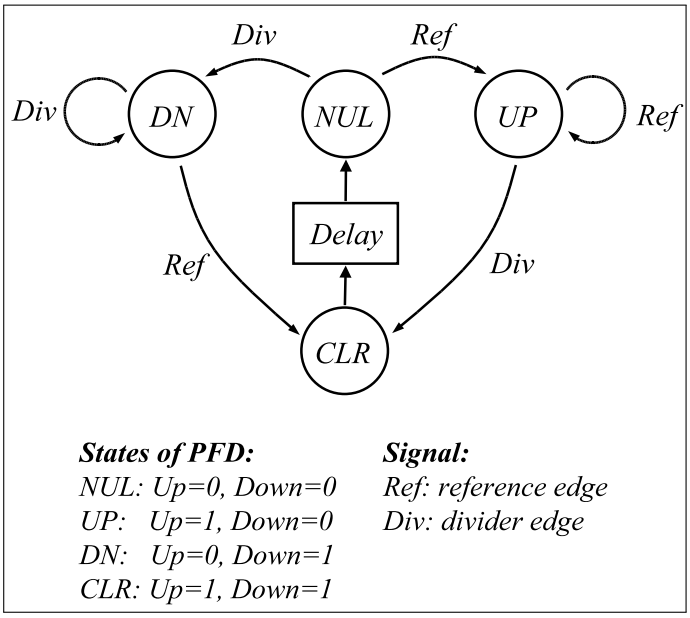


FIG. 1.16: *State diagram of the PFD : with delay element*

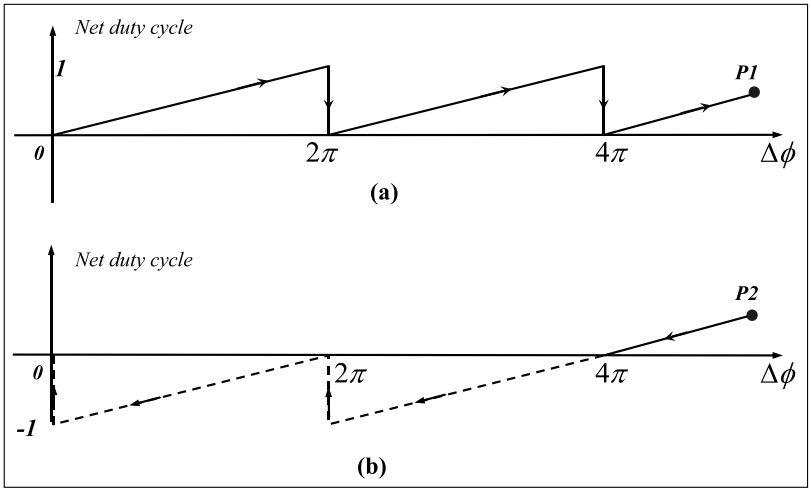


FIG. 1.17: *Illustration of the PFD hysteresis*

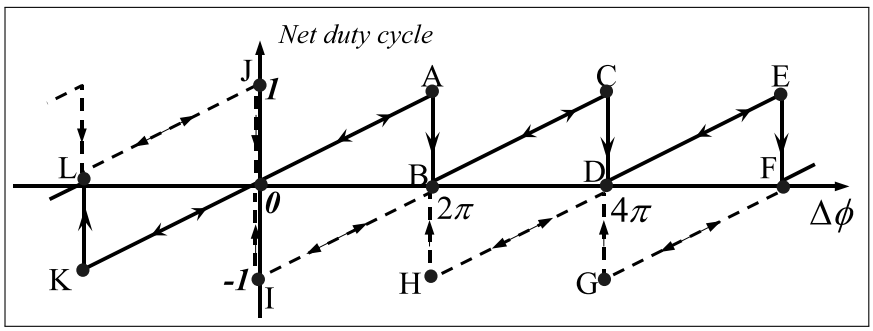


FIG. 1.18: *Hysteresis of the PFD*

E. Average duty cycle when the input frequencies are different

When the frequencies of REF and DIV are different, the pulse width of UP and DN will change their values in each period, resulting in variable duty cycles. What we care about is the average net duty cycle $\bar{d}$, because the high-frequency component is eliminated by the loop filter.

When the frequency of REF is slightly larger than that of DIV, $\Delta\phi$ is positive and the net duty cycle follows the sawtooth path located in the upper half plane of Fig. 1.18, the average duty cycle is 0.5. On the contrary, if the frequency of REF is slightly smaller than that of DIV, the average value of the duty cycle is -0.5. [21]

For the large frequency difference, $\bar{d}$ depends only on one parameter: frequency ratio between REF and DIV, as expressed by the following formula [22] :

$$\bar{d} = \begin{cases} 1 - f_{div}/2f_{ref} & \text{... if } f_{ref} > f_{div} \\ f_{ref}/2f_{div} - 1 & \text{... if } f_{ref} < f_{div} \end{cases} \tag{1.15}$$

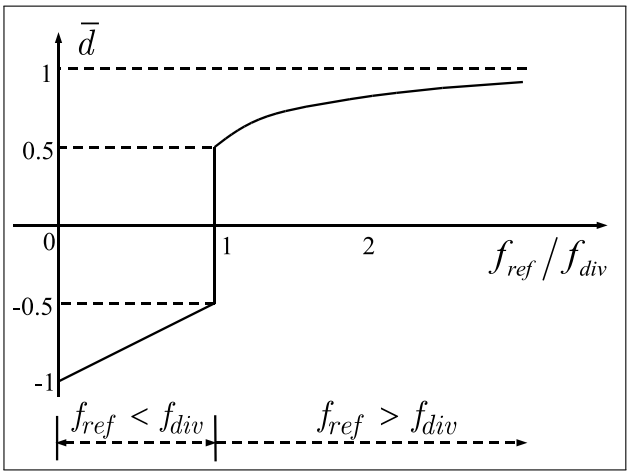


FIG. 1.19: *Average net duty cycle in the case of large frequency error*

Illustrated in Fig. 1.19, this formula shows that if $f_{ref} >> f_{div}$, the edges of REF arrive consecutively before a DIV edge arises, causing a net duty cycle close to 1, which allows charging the filter with an important average current, then raising the VCO frequency rapidly in order to catch up with the reference frequency f_{ref}. On the other hand, if $f_{ref} << f_{div}$, $\bar{d}$ will approach -1.

1.3.3.3 Charge Pump (CP)

The charge pump mainly consists of two current sources controlled by switches. CP is normally loaded with a filter to form an integrator with which the output current of the

CP is converted into a voltage signal controlling the VCO. When the current flows into or out of the filter, it charges or discharges the capacitors in the filter, and makes the VCO control voltage vary with time, as shown in Fig. 1.20. [19]

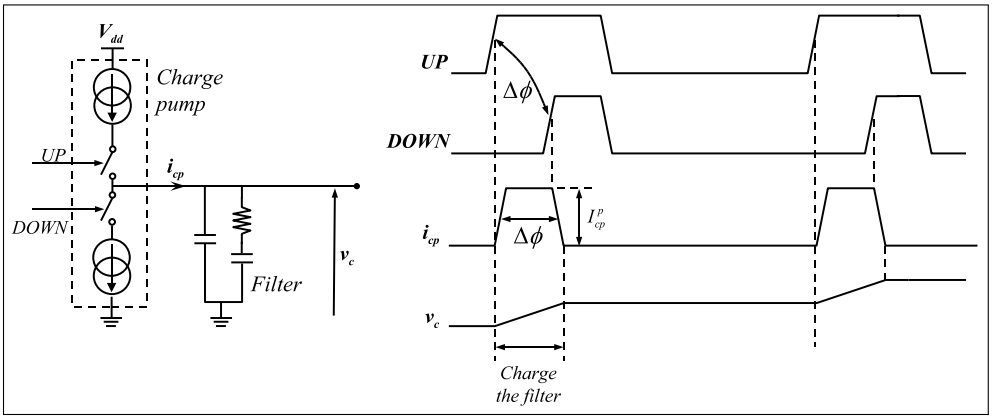


FIG. 1.20: *Charge pump and the time sequence of the signals*

The two current sources are expected to be symmetrical. Ideally, when UP and DOWN are in the high state simultaneously, there is no current flowing out of the CP. At the output of the CP it's the pulse width of current I_{cp} that represents the phase difference.

Concerning the CP output current, what we care about is actually the continuous component. So the average output current of the CP can be written as follows:

$$I_{cp}\,(DC) = \frac{\Delta\phi}{2\pi} I^p_{cp} \tag{1.16}$$

where I^p_{cp} is the peak value of the CP output current, as shown in Fig. 1.20.

Hence, ideally the gain of the block PFD+CP can be considered as:

$$K_p = \frac{I^p_{cp}}{2\pi} \tag{1.17}$$

However, it's usually very difficult to design two perfectly symmetrical current sources. The mismatch of the current sources will create a low-frequency voltage ripple at the input of VCO, which is very likely to modulate with the high-frequency oscillation of VCO and produce the deterministic noise which will be discussed in chapter 3.

1.3.3.4 Summary of phase detector circuits

As a summary of the above, we may say that the PFD+CP has superior advantages over PD. One may consult the following references for more details: [2] [12] [23].

- First, the PFD can detect the phase and the frequency at the same time, allowing the PLL to lock more rapidly when the frequency difference is large.
- Secondly, only the input signal edges can take effect on the PFD output, while the duty cycle of the input signal don't have the influence on the PFD output, which allows achieving a precise phase control.
- Thirdly, with the PFD+CP, while the PLL is in the locked state, the phase difference can be nearly zero, instead of π or $\pi/2$.
- Finally, the linear range of the PFD is $4\pi([-2\pi, 2\pi])$, larger than the any PD circuit.

In addition, the charge pump can be used to form a type II PLL, whose advantages will be introduced in the section 1.4.1. In view of these advantages of the PFD+CP, the rest of the thesis will consider mainly PFD+CP when the phase detector is involved.

1.3.4 Loop filter

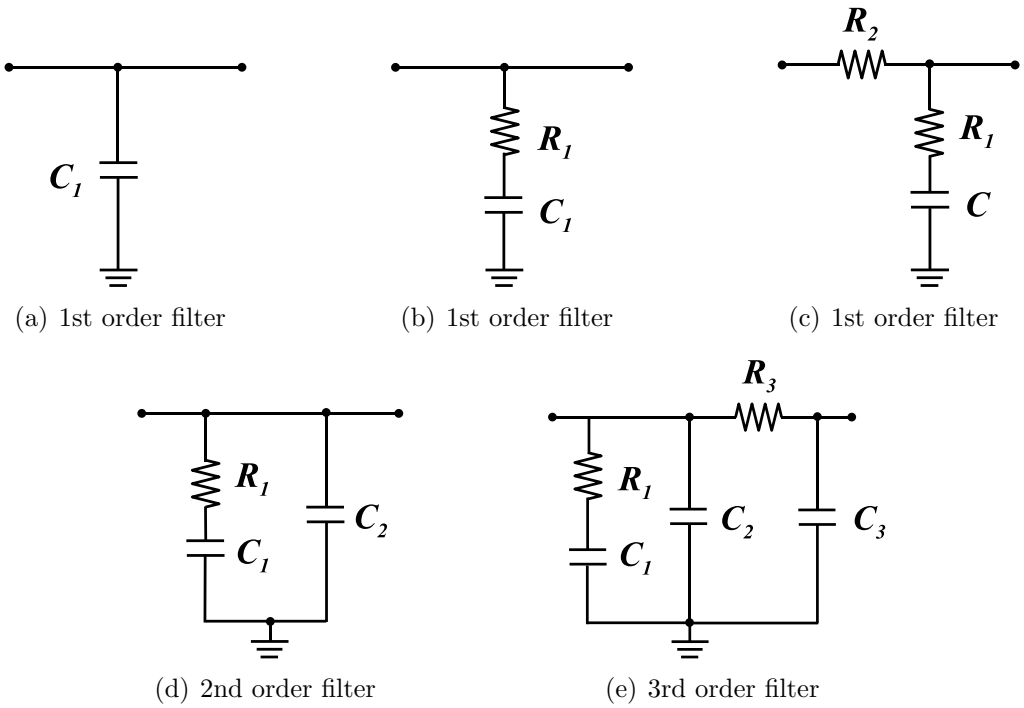
The loop filter is used to smoothen the signal by eliminating the high-frequency component and provide a slowly varying DC voltage to the input of the VCO. The input signal of the loop filter is a current for the PFD+CP case or a voltage for the PD case.

Two types of filters are often used: the passive filter and the active filter. The passive filter is simple, has low noise, but the output voltage range is fixed. The active filter can adjust its output voltage range, but it's more complex and produces more noise. The characteristics of the passive filter are shortly introduced next. [2] [3] [24] [25]

The simplest passive filter is the 1st order filter consisting of a simple capacitor (Fig. 1.21(a)). But the single pole in its transfer function located on the imaginary axis makes the PLL potentially unstable. If a resistance R_1 is added in series with the capacitor C_1 (Fig. 1.21(b)), one zero can be created on the negative axis, hence improving the stability of the PLL. In practice, a 2nd order filter is often used (a capacitor added in parallel with the RC branch, Fig. 1.21(d)) to attenuate the voltage ripple at the output of the filter.

The 3rd order filter (Fig. 1.21(e)) and the higher order filters are also used to get some specific performances, for example, zero phase error when the input frequency ramps, the attenuation of the noise from the reference, a sharper roll-off characteristic of the frequency response, etc. However, increasing the filter order deteriorates the stability of the PLL and makes the circuit more complex.

The mainly used passive filters are shown in Fig. 1.21, and the corresponding transfer functions are presented in the equations below, which can be used to develop the transfer function of the PLL in the section 1.4.1.



(a) 1st order filter (b) 1st order filter (c) 1st order filter

(d) 2nd order filter (e) 3rd order filter

FIG. 1.21: *Loop filter*

▶ Transfer function of the first order filter in Fig. 1.21(a):

$$F(s) = \frac{1}{\tau \cdot s} \tag{1.18}$$

where $\tau = C_1$. The pole is $\omega_p = 0$.

▶ Transfer function of the first order filter in Fig. 1.21(b):

$$F(s) = R_1 \frac{1 + \tau_{rc} \cdot s}{\tau_{rc} \cdot s} \tag{1.19}$$

where $\tau_{rc} = R_1 C_1$. The pole is $\omega_p = 0$, and the zero is $\omega_z = -1/\tau_{rc}$.

▶ Transfer function of the first order filter in Fig. 1.21(c):

$$F(s) = \frac{1 + \tau_2 \cdot s}{1 + \tau_1 \cdot s} \tag{1.20}$$

where $\tau_1 = (R_1 + R_2) C$, $\tau_2 = R_2 C$. The pole is $\omega_p = -1/\tau_1$, and the zero is $\omega_z = -1/\tau_2$.

▶ Transfer function of the second order filter in Fig. 1.21(d):

$$F(s) = \frac{1 + \tau_{rc} \cdot s}{\tau_{cc} \cdot s \left(1 + \tau_2 \cdot s\right)} \tag{1.21}$$

where $\tau_{rc} = R_1 C_1$, $\tau_{cc} = C_1 + C_2$, $\tau_2 = R_1 C_1 C_2 / \tau_{cc}$. The poles are $\omega_{p1} = 0$ and $\omega_{p2} =$

$-1/\tau_2 = -\left(1/C_1 + 1/C_2\right)/R_1$, and the zero is $\omega_z = -1/\tau_{rc}$.

▶ Transfer function of the third order filter in Fig. 1.21(e):

$$F(s) = \frac{1 + \tau_{rc} \cdot s}{s \cdot \tau_{ccc} \cdot (1 + s \cdot \tau_3)(1 + s \cdot \tau_{rc3})} \tag{1.22}$$

where $\tau_{rc} = R_1C_1$, $\tau_{ccc} = C_1 + C_2 + C_3$, $\tau_3 = R_1C_1C_2/\tau_{ccc}$, $\tau_{rc3} = R_3C_3$. The poles are $\omega_{p1} = 0$, $\omega_{p2} = -1/\tau_3$, $\omega_{p3} = -1/\tau_{rc3}$, and the zero is $\omega_z = -1/\tau_{rc}$.

1.4 Idealized description of the PLL: linear model

Now that we have briefly presented the main building blocks of the PLL, we will introduce some idealized characteristics of the PLL, such the frequency response characteristic, phase margin, etc. A more in-depth description of the PLL characteristics like the noise behavior and the dynamic performances will be carried in chapter 3 and chapter 4.

As known, the PLL is a nonlinear circuit with a very complex electrical dynamics, so a general analysis with a linear model is not feasible. Nevertheless, many characteristics of the PLL can be well approximated by linear models, especially in the vicinity of the locked state [2]. Linear models provide simple analysis tools that help to understand the basic characteristics of the PLL.

One of the most useful linear analysis tools is the transfer function, which is based on the Laplace transform of the input/output signals in the *s-domain*. As we will see below, most of the important concepts of the PLL, such as type and order of the PLL, phase margin, damping factor..., are derived from the definition of the transfer function. In addition, the transfer function of the PLL is also a very good tool to model the noise contribution of the different building blocks.

In this section, we give a concise introduction of the transfer function of the PLL, the noise transfer function, and the two typical PLL structures: type II 2nd order PLL and type II 3rd order PLL.

1.4.1 Transfer function of the PLL

A. Introduction

Generally, a transfer function describes the relationship between input and output signals of a device in *s-domain*. If we express the voltage or current signals as $x(t) = X(t)\,cos\left(2\pi f_x t + \phi_x(t)\right)$, for the PLL, the signal we are interested in is the phase rather

than the voltage or the current waveform. This is because the intimate function of the PLL is to process the phase of the signal rather than its waveform. Interestingly enough the phase components of the signal are slowly varying variables of time in contrast to the time constants of input and output voltages.

Considering the block diagram of the PLL shown in Fig. 1.22, the input/output signals of each building block are defined in *s-domain*. The phase difference $\Delta\Phi(s)$ is defined as $\Delta\Phi(s) = \Phi_{ref}(s) - \Phi_{div}(s)$. Note that the CP current and the VCO control voltage being the input and output signals to a low-pass filter, only the low frequency components of the signals are considered in the model.

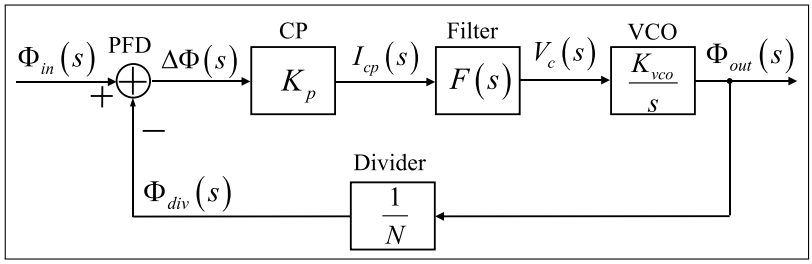


FIG. 1.22: *Block diagram of the PLL*

Next, the transfer functions for each building block are briefly described.

PFD+CP: If the PLL is in the locked state, the ideal characteristic of the PFD+CP is of the form:

$$I_{cp}(s) = K_p \cdot \Delta\Phi(s) \tag{1.23}$$

where K_p (A/rad) is the static gain of the PFD+CP.

Filter: The loop filter can be expressed by the following relation:

$$\frac{V_c(s)}{I_{cp}(s)} = F(s) \tag{1.24}$$

where $F(s)$ is the trans-impedance function described in the section 1.3.4.

VCO: The transfer function of the VCO is given by the equation (1.7) rewritten below:

$$\frac{\Phi_{out}(s)}{V_c(s)} = \frac{K_{vco}}{s} \tag{1.25}$$

where K_{vco} (rad/s/V) is the static frequency sensitivity regarding to the voltage change.

Divider: The relationship between the output phase and the input phase can be represented by the division ratio N :

$$\frac{\Phi_{div}(s)}{\Phi_{out}(s)} = \frac{1}{N} \tag{1.26}$$

where N is the division ratio.

Now that the transfer functions of the building blocks are obtained, the transfer functions for the open-loop and close-loop PLL can be derived as follows:

Forward transfer function of the PLL $G_{fwd}(s)$, defined as:

$$G_{fwd}(s) = \frac{\Phi_{out}(s)}{\Delta\Phi(s)} = \frac{K_p K_{vco} F(s)}{s} \tag{1.27}$$

Transfer function for the feedback chain $H(s)$:

$$H(s) = \frac{\Phi_{div}(s)}{\Phi_{out}(s)} = \frac{1}{N} \tag{1.28}$$

Thus the open-loop transfer function (named also open-loop gain) $G_{open}(s)$ can be obtained:

$$G_{open}(s) = \frac{\Phi_{div}(s)}{\Delta\Phi(s)} = G_{fwd}(s) \cdot H(s) = \frac{K_p K_{vco} F(s)}{N \cdot s} \tag{1.29}$$

And the close-loop transfer function (or close-loop gain) $G_{close}(s)$ is:

$$G_{close}(s) = \frac{\Phi_{out}(s)}{\Phi_{in}(s)} = \frac{G_{fwd}(s)}{1 + G_{fwd}(s) \cdot H(s)} = \frac{N K_p K_{vco} F(s)}{N \cdot s + K_p K_{vco} F(s)} \tag{1.30}$$

Another useful transfer function is the error transfer function $G_{err}(s)$ which describes the relationship between the phase difference and the input phase:

$$G_{err}(s) = \frac{\Delta\Phi(s)}{\Phi_{in}(s)} = \frac{1}{1 + G_{open}(s)} = 1 - \frac{G_{close}(s)}{N} = \frac{N \cdot s}{N \cdot s + K_p K_{vco} F(s)} \tag{1.31}$$

Note that these transfer functions ($G_{open}(s)$, $G_{close}(s)$, $G_{err}(s)$) of the PLL are very useful to describe the basic characteristics of the PLL, like the type and order of the PLL, the phase margin, etc, which will be introduced below.

B. Type and order of the PLL

The order of the PLL is defined by the number of poles in the close-loop transfer function. In fact, as can be seen from the equation (1.25) there is always a pole in the transfer function of the VCO. Therefore, a PLL with N^{th} order filter is a $(N+1)^{th}$ order PLL.

The type of PLL is defined as the number of poles at the origin in the open-loop transfer function, or the number of integrators in the loop [2]. Since the VCO naturally offers a pole at the origin, the PLL is at least type I.

The PLL with CP and the passive filter in Fig. 1.21(b) is type II, because the CP and passive filter create a pole at the origin. On the contrary, a PLL with PD + filter is type I, as the pole provided by the filter is nonzero. Nowadays most of the PLLs are type II, because they have many advantages over the type I PLLs, such as zero phase error in the case of frequency step of the reference signal, infinite DC gain, unlimited hold range and capture range. More details can be found in the bibliographies [12] [26].

Note that the type of the PLL does not always equal to the order of the PLL. For example, the PLL with the filter in Fig. 1.21(d) is 3rd order, but type II, because one of the two poles of the filter is not located at the origin.

This type II 3rd order PLL is one of the most popular PLLs. Besides the advantages of the type II PLLs mentioned above, the second pole located at the negative axis of s-plan allows filtering the high-frequency fluctuation and offers a good trade-off between the dynamic performances and the stability of the PLL behavior [17] . Some higher order PLLs also can be found to achieve some specific goals but their use is very limited due to the stability problem and the increase of the complexity. Hence in the following we focus on the type II 3rd order PLL. On the other hand, the analysis of the type II 3rd order PLL is usually carried as an extension to the type II 2nd PLL, because the type II 2nd PLL is more easily represented mathematically, so in the following we will also introduce the characteristics of the type II 2nd order PLL.

C. Phase margin of the PLL

Phase margin is a very critical parameter of the PLL which describes the degree of stability of the PLL. The phase margin is defined on the basis of the open-loop transfer function of the PLL, when the magnitude of the open-loop gain $G_{open}(s)$ equals to 1:

$$\begin{cases} |G_{open}(j\omega_c)| = 1 \\ \phi_m = \angle G_{open}(j\omega_c) + 180^\circ \end{cases} \tag{1.32}$$

where ω_c is named cross-over frequency, and ϕ_m is the phase margin of the PLL. Moreover, ω_c is often considered as the loop bandwidth of the PLL.

A PLL is stable if the phase margin is greater than 0, in another word, $\angle G_{open}(j\omega_c)$ is bigger than -180°:

$$\begin{cases} \phi_m > 0 \\ \angle G_{open}(j\omega_c) > -180^\circ \end{cases} \tag{1.33}$$

Since Bode plots can illustrate the magnitude and the phase of the open-loop gain in the same diagram, it is usually used to depict the phase margin of the PLL, as will be shown.

1.4.2 Noise transfer function

Fig. 1.23 shows the PLL diagram where the intrinsic noises of all blocks are indicated: $\tilde{\phi_{vco}}$ phase noise source from VCO, $\tilde{\phi_{div}}$ phase noise source from the divider, and $\tilde{i_n}$ current noise source from PFD+CP.

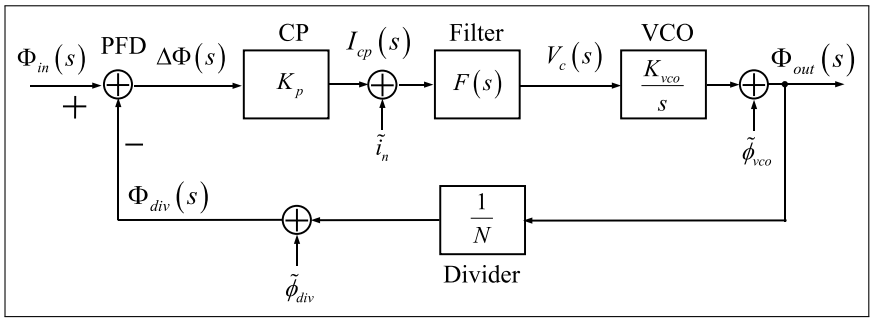


FIG. 1.23: *Diagram of the PLL and the noise sources*

The noise transfer function describes the relationship between the noise source from one block and the noise at the output of the PLL. It can be easily calculated, as shown in Fig. 1.24.

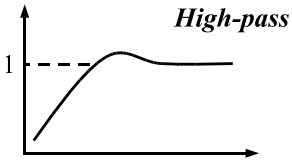
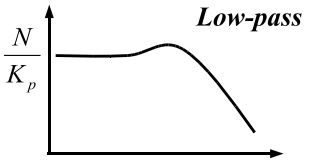
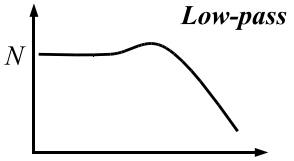
Noise source	***Noise transfer function***	***Figure***
VCO	$G_{err}(s) = \frac{N \cdot s}{N \cdot s + K_p K_{vco} F(s)}$	***High-pass*** 
PFD+CP	$\frac{G_{close}(s)}{K_p} = \frac{N K_{vco} F(s)}{N \cdot s + K_p K_{vco} F(s)}$	***Low-pass*** 
Divider	$G_{close}(s) = \frac{N K_p K_{vco} F(s)}{N \cdot s + K_p K_{vco} F(s)}$	***Low-pass*** 

FIG. 1.24: *Noise transfer function for the different blocks*

The total noise at the output of the PLL is the sum of all the noise contribution:

$$Noise\,(total) = \sum_{n=1}^{N} (S_n \cdot F_n) \tag{1.34}$$

where S_n is the noise power spectral density for one block and F_n is the corresponding noise transfer function for this block.

The noise contribution of a block depends not only on the amount of the noise source, but also the transfer function of the noise source. The latter, served as a filter, can increase or decrease the noise contribution of a block.

The noise transfer function for the VCO is characterized by error transfer function of the PLL $G_{err}(s)$ [24] [2], as shown in Fig. 1.24. It has a high-pass characteristic, i.e., the VCO noise source at low frequency is attenuated.

However, the noise transfer functions for the PFD/CP and the divider are characterized by the close-loop transfer function $G_{close}(s)$, which is low-pass (see Fig. 1.24), and the noise at high frequency could be filtered.

At low frequency the noise transfer functions for the PFD/CP and the divider are proportional to the division ratio N. As a result, a reduction of N by 2 allows lowering the noise contribution by 6dB.

1.4.3 Description of two typical PLL structures

As mentioned previously, the type II 2nd order PLL and the type II 3rd order PLL are the most often used PLL structures, so we give a brief presentation on their characteristics. First, the transfer functions for these two typical PLLs are developed, then the frequency response and phase margin characteristics are briefly presented, which serve as a basis for the PLL analysis in the following chapters.

A. Transfer functions for type II 2nd order PLL

The type II 2nd order PLL uses a loop filter shown in Fig. 1.21(b) whose block transfer function $F(s)$ is given by the equation (1.19). According to the definition of the open-loop transfer function in the equation (1.29), the open-loop transfer function of the type II 2nd order PLL is given by:

$$G_{open}(s) = \frac{K_p K_{vco} R_1}{N} \cdot \frac{1 + s\tau_{rc}}{s^2 \tau_{rc}} = K \cdot \frac{1 + s\tau_{rc}}{s^2 \tau_{rc}} \tag{1.35}$$

where K is defined as:

$$K = \frac{K_p K_{vco} R_1}{N} \tag{1.36}$$

With the definition of the closed-loop transfer function in the equation (1.30), the

closed-loop transfer function of the type II 2nd order PLL is obtained as follows:

$$G_{close}(s) = N\frac{Ks + K/\tau_{rc}}{s^2 + Ks + K/\tau_{rc}} = N\frac{2\zeta\omega_n s + \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1.37}$$

where the natural frequency ω_n and the damping factor ζ determine the stability and the dynamic performances of the PLL, and they are written by :

$$\omega_n = \sqrt{K/\tau_{rc}} \tag{1.38}$$

$$\zeta = \frac{1}{2}\sqrt{K \cdot \tau_{rc}} \tag{1.39}$$

Finally, the error transfer function can be written by:

$$G_{err}(s) = \frac{s^2}{s^2 + Ks + K/\tau_{rc}} = \frac{s^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1.40}$$

B. Transfer functions for type II 3rd order PLL

The type II 3rd order PLL contains a loop filter illustrated in Fig. 1.21(d) with its block transfer function $F(s)$ expressed by the equation (1.21). In the same way, the transfer functions can be obtained and written as follows:

Open-loop transfer function:

$$G_{open}(s) = \frac{K_P K_{vco}(\tau_{rc}s + 1)}{N\tau_{cc}s^2(\tau_2 s + 1)} \tag{1.41}$$

Close-loop transfer function:

$$G_{close}(s) = N\frac{K_P K_{vco}(\tau_{rc}s + 1)}{N\tau_{cc}s^2(\tau_2 s + 1) + K_P K_{vco}(\tau_{rc}s + 1)} \tag{1.42}$$

Error transfer function:

$$G_{err}(s) = \frac{N\tau_{cc}s^2(\tau_2 s + 1)}{N\tau_{cc}s^2(\tau_2 s + 1) + K_P K_{vco}(\tau_{rc}s + 1)} \tag{1.43}$$

C. Frequency response characteristics of the two typical PLLs

Frequency response characteristics of the PLL are very useful to study the dynamic performances and the noise characteristics of the PLL.

Based on the transfer functions presented in the previous section, the frequency response characteristics of the PLL can be easily obtained. Replacing the parameter s in

the transfer functions by $j\omega$, the frequency response of the PLL can be traced in the frequency domain.

For the close-loop transfer function of the type II 2nd order PLL, its characteristic is shown in Fig. 1.25, which has a low-pass characteristic. Consequently, the slow phase variation at the input of the PLL can be transmitted to the output without attenuation while the rapid phase variation is filtered. If one varies the damping factor ζ, a series of curve can be shown in Fig. 1.25. Note that there is always peaking on the curve regardless of the value of damping factor. The cause of the peaking is the presence of the zero $1/\tau_{rc}$ near the natural frequency ω_n [27].

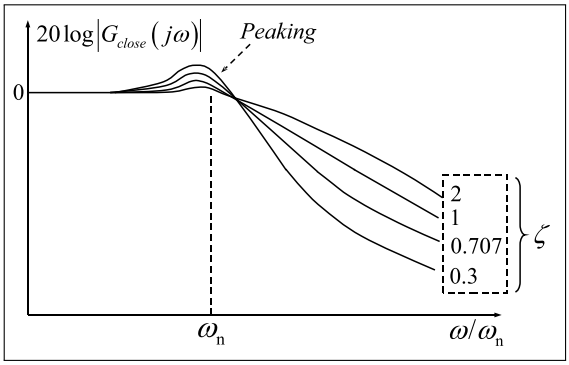


FIG. 1.25: *Frequency response of the close-loop gain for the type II 2nd order PLL*

Concerning the error transfer function $G_{err}(s)$, its characteristic is high-pass, as shown in Fig. 1.26. Therefore, if the input phase of the PLL varies slowly, the phase error will be small, thus the PLL will track closely the phase variation. Unlike the closed loop gain, only when the damping factor $\zeta < 0.707$, the peaking appears on the curve.

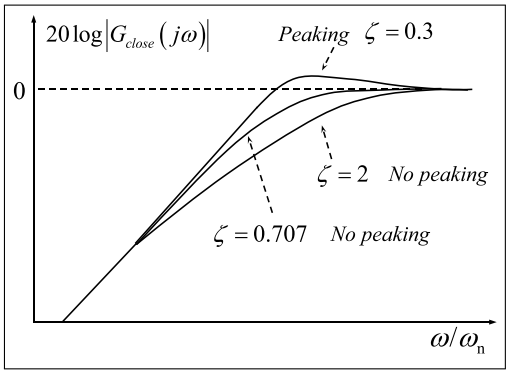


FIG. 1.26: *Frequency response of G_{err} for the type II 2nd order PLL*

The asymptote of the curves $G_{close}(s)$ or $G_{err}(s)$ has a constant slope, which depends on the degree of the denominator and the numerator. For the type II 2nd order PLL, the slope of the asymptote is -10dB/dec for the close-loop gain, and +20dB/dec for the gain $G_{err}(s)$.

As for the type II 3rd order PLL, it has the similar low-pass characteristic for the close-up transfer function $G_{close}(s)$ and the high-pass characteristic for the error transfer function $G_{err}(s)$.

D. Phase margin of the two typical PLLs

Phase margin of the PLL is defined in the equation (1.32) and can be described by Bode plots. Fig. 1.27 shows respectively Bode plots for the type II 2nd order PLL and type II 3rd order PLL. At low frequency, the two PLLs have the same characteristic: the slope is $-20dB/dec$, and the phase is about $-180°$ since both of them are type II PLLs. The presence of the third pole in the 3rd order PLL makes the curves different at high frequencies, where the slope of the curve returns to $-20dB/dec$ and the phase falls back to $-180°$. Phase margin of these two PLLs is marked on the graph. Phase margin also have an influence on the lock time of the PLL. The details are presented in [3] [28].

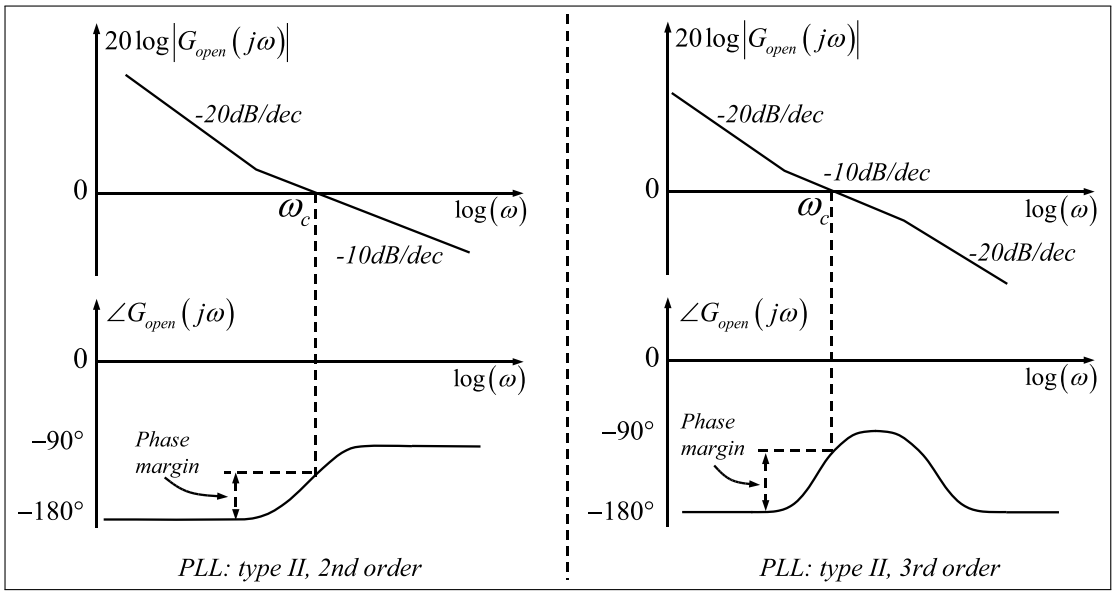


FIG. 1.27: *Bode plots for the type II 2nd order PLL and the type II 3rd order PLL*

1.5 Conclusion

Phase-locked loop is one of the most complex circuits in the RFIC domain. The verification of the PLL circuit becomes a great challenge due to its mixed (analog and digital) circuit property and its non-linear characteristics. This chapter was dedicated to present some important aspects related to the circuit structure and its non-linear characteristics, which serve as a prerequisite to the modeling and the simulation of the PLL.

To do so, the building blocks of the PLL are introduced along with their non-linear characteristics, such as the dead zone of the PFD, the non-linear voltage-frequency relationship in the VCO, etc. Then, the idealized characteristics of the PLL are presented based on the definition of the transfer function of the PLL, from which some important concepts of the PLL are derived, such as type and order of the PLL, phase margin.

We have seen that the PLLs can be classified into narrow-band PLL and wide-band PLL, and their key characteristics could be low phase noise or specific dynamic performance, or both. In addition, the large-signal steady-state response under the lock condition is also an important characteristics of the PLL. The simulation and modeling of the above three characteristics are of high interests to the designers, and constitute the main targets of this thesis, which will be respectively presented in the next three chapters.

1.6 Bibliography

[1] R. E. Best, *Phase-Locked Loops: Design, Simulation, and Applications*, 5th ed. McGraw-Hill Professional, 2003.

[2] F. M. Gardner, *Phaselock Techniques*, 3rd ed. Wiley-Interscience, Jul. 2005.

[3] K. Shu and E. Sánchez-Sinencio, *CMOS PLL Synthesizers: Analysis and Design*, 1st ed. Springer, Nov. 2004.

[4] T. Lee, K. Donnelly, J. Ho, J. Zerbe, M. Johnson, and T. Ishikawa, "A 2.5 V CMOS delay-locked loop for 18 Mbit, 500 megabyte/s DRAM," *Solid-State Circuits, IEEE Journal of*, vol. 29, no. 12, pp. 1491–1496, Dec 1994.

[5] J. Vankka and K. A. Halonen, *Direct Digital Synthesizers: Theory, Design and Applications*, 1st ed. Springer, May 2001.

[6] A. Layec, "Développement de modèles de CAO pour la simulation système des systèmes de communication. Application aux communications chaotiques," Ph.D. dissertation, Université de Limoges, 2006.

[7] M. Sie, "Synthétiseurs de fréquence monolithiques micro-ondes à 10 et 20 GHz en technologies BiCMOS SiGe 0,25 et 0,35 um," Ph.D. dissertation, Université Paul Sabatier de Toulouse, 2004.

[8] T. Riley, M. Copeland, and T. Kwasniewski, "Delta-sigma modulation in fractional-N frequency synthesis," *IEEE Journal of Solid-State Circuits*, vol. 28, pp. 553–559, 1993.

[9] W. Rhee, B. S. Song, and A. Ali, "A 1.1-Ghz CMOS fractional-N frequency synthesizer with a 3-bthird-order delta-sigma modulator," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 1453–1460, 2000.

[10] M. H. Perrott, T. L. Tewksbury, and C. G. Sodini, "A 27-mw CMOS fractional-N synthesizer using digital compensationfor 2.5-Mb/s GFSK modulation," *IEEE Journal of Solid-State Circuits*, vol. 32, pp. 2048–2060, 1997.

[11] T. H. Lee, *The Design of CMOS Radio-Frequency Integrated Circuits*, 2nd ed. Cambridge University Press, Dec. 2003.

[12] B. Razavi, *RF Microelectronics*. Prentice Hall, 1998.

[13] A. Aktas and M. Ismail, *CMOS PLLs and VCOs for 4G Wireless*, 1st ed. Springer, Jun. 2004.

[14] A. Hajimiri and T. H. Lee, *The Design of Low Noise Oscillators*, 1st ed. Springer, Feb. 1999.

[15] S. Trotta, H. Knapp, T. F. Meister, K. Aufinger, J. Bock, W. Simburger, and A. L. Scholtz, "110-Ghz static frequency divider in SiGe bipolar technology," in *IEEE Compound Semiconductor Integrated Circuit Symposium*, 2005, pp. 291–294.

[16] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*, 4th ed. Wiley, Feb. 2001.

[17] F. M. Gardner, "Charge-pump phase-lock loops," *IEEE Transactions on Communications*, vol. 28, pp. 1849–1858, 1980.

[18] A. Demir, "Analysis and simulation of noise in nonlinear electronic circuits and systems," Ph.D. dissertation, University of California, 1997.

[19] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 1st ed. McGraw-Hill Science/Engineering/Math, Aug. 2000.

[20] M. Tavares, "PLL Frequency synthesizers: Phase noise issues and wide-band loops," Ph.D. dissertation, INSA de Lyon, 1999.

[21] J. Chen, "An introduction to the PLL library," Cadence Design Systems, Tech. Rep. [Online]. Available: www.cadence.com/whitepapers/pllapp_note.pdf

[22] M. Soyuer and R. Meyer, "Frequency limitations of a conventional phase-frequency detector," *IEEE Journal of Solid-State Circuits*, vol. 25, pp. 1019–1022, 1990.

[23] D. Fischette, "Practical tips for phase-locked loops design," 2007. [Online]. Available: www.delroy.com

[24] D. Banerjee, *PLL Performance, Simulation and Design*, 4th ed. Dog Ear Publishing, LLC, Aug. 2006.

[25] Z. Madini, "Caractérisation et modélisation des dynamiques non linéaires dans les dispositifs pour systèmes de télécommunications : Cas de la PLL et de l'amplificateur de puissance," Ph.D. dissertation, Université de Limoges, 2008.

[26] S. Sancho, A. Suárez, and T. Fernández, "Nonlinear dynamics of microwave synthesizers - stability and noise," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, pp. 1792–1803, 2001.

[27] A. Berny, "Analysis and design of wideband LC VCOs," Ph.D. dissertation, University of California at Berkeley, 2006.

[28] C. S. Vaucher, "An adaptive PLL tuning system architecture combining high spectralpurity and fast settling time," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 490–502, 2000.

Chapter 2 :

Steady-state Analysis of the PLL

2.1 Introduction

As indicated in chapter 1, the PLL is a crucial circuit in the transceiver and it influences greatly the performance of the communications system. The main characteristics of the PLL include:

- Large signal steady-state response;
- Noise characteristics, such as phase noise, deterministic noise;
- Dynamic performances, e.g., lock time, capture range, hold range, etc.

As known, the steady-state solution is a prerequisite for small-signal noise analysis, which means that the noise analysis of the PLL should be performed in the presence of the large-signal steady-state solution [1]. Therefore in this chapter we discuss the large signal steady-state response of the PLL under locked condition, which serves as basis for the noise analysis of the PLL to be outlined in chapter 3.

Regarding the steady-state analysis, it is very important since the accuracy of the steady-state solution affects the calculation of the noise of the PLL. In addition, it is often very difficult to make the PLL get into the locked state due to the long transition time and the huge memory necessary to make the simulation converge [2]. That is why a quick and accurate simulation method is always in need.

In this chapter, first the steady-state of the PLL is introduced and a summary is made about the existing steady-state simulation methods. Then, a new method is proposed which considers the non-ideal effects of the PLL and uses an iterative algorithm to achieve good simulation accuracy. With the new method the simulation time is largely shortened thanks to a hierarchical approach. The whole procedure of the new method is presented, and is verified step by step on a circuit example. Finally, the simulation results are shown and compared with the conventional transistor level simulation method.

2.2 Steady-state of the PLL

In this section, we briefly review the steady-state of the PLL, and indicate the possible simulation methods.

In electronics, a steady-state occurs in a circuit when all the transients die away [3]. There are different types of steady-state in the electronic circuits. The first one is the *DC steady-state*, whose solution is a time-invariant equilibrium point. The second one is the *periodic steady-state*, which consists of a linear combination of a DC offset and a number

of harmonically related sinusoids. The periodic steady-state exists in both autonomous circuits (e.g. oscillator) and forced circuits with periodic excitation.

If a non-linear circuit is driven by periodic sources at non harmonically related frequencies, the circuit will have a *quasi-periodic* response. A quasi-periodic response consists of a linear combination of different fundamental frequencies and their harmonics. The periodic steady-state is a special case of the quasi-periodic steady-state.

As for the steady-state of the PLL circuit, its operating point may not be invariant over time, but periodic or sometimes chaotic [4]. Moreover, certain types of PLL don't have a steady-state solution, such as the $Sigma - Delta$ fractional-N PLL (which generates randomly a signal at the output of the divider), the CDR circuits (Clock and Data Recovery circuits), and the PLL with a dead zone. In this chapter, we'll address the PLLs which have a steady-state solution, primarily the integer-N PLL.

Generally, most of the analog circuits can reach the periodic steady-state or quasi-periodic steady-state from an initial condition, assuming that the excitation signals are periodic or quasi-periodic. So it is possible to calculate the steady-state response of the circuit using transient simulation, like SPICE approach or some other equivalent methods. It merely needs to solve the differential equations of the circuit from an arbitrary initial condition until the transient behavior disappears.

However, SPICE method becomes impracticable if the ratio between the highest frequency and the lowest frequency in the response is too large, because in that case the time step will be so small that the computation would be very costly.

The SPICE approach is based on direct time domain integration, there are many other methods for calculating the steady-state solution more effectively, either in the time domain (Time Shooting method) or in the frequency domain (Harmonic balance), which will be introduced in the next section. A careful association of these simulation methods is necessary to analyze the steady-state of the PLL as well as the building blocks of the PLL.

2.3 General steady-state analysis method

To analyze the steady-state of the PLL and the behavior of its building blocs, some efficient methods are indispensable, which will be briefly introduced in this section.

Time domain integration, time Shooting method and harmonic balance are widely used in most commercial simulators, and they are very powerful to analyze the periodic or quasi-periodic signal in the microwave circuits.

2.3.1 Time domain integration

Time domain integration (TDI) method [3] [5] [6] is based on the discretization of the device constitutive equation in the time domain, as shown in Fig. 2.1. After discretization of the ordinary differential equation (ODE), the circuit becomes resistive at each moment t_n. The complexity of equations depends on the discretization method (Forward, Backward, Euler, Gear...) [7]. A variable time step is usually necessary to achieve an acceptable accuracy and to speed up the simulation as well.

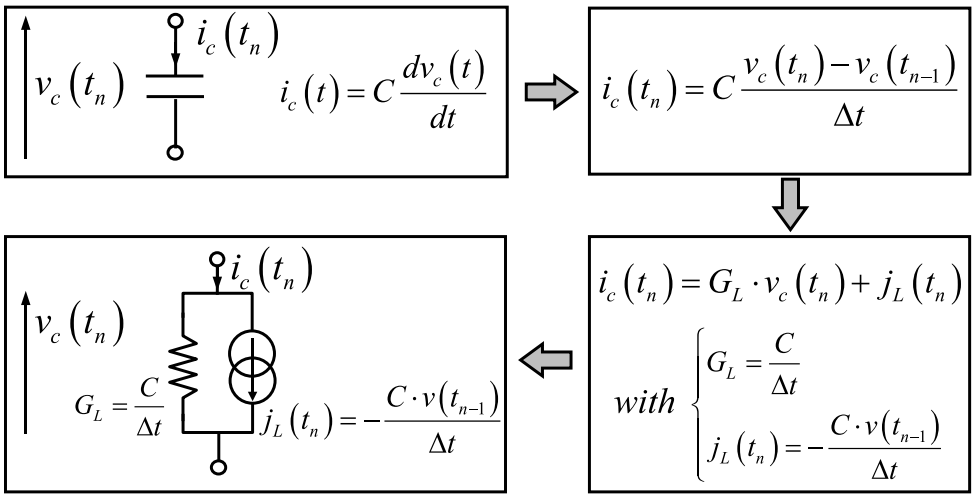


FIG. 2.1: *Discretization of the basic component: the capacitor*

The resolution of the equations starts at the initial condition where $t = t_0$. Once a solution is found at a sampling point, it proceeds to the next point after controlling a local truncation error. When the transient dies out, the steady-state of the circuit can finally be obtained.

Waiting for the transient to die out can require prohibitive memory resources and excessive simulation time. To accelerate the onset of the steady state, time domain Shooting method was developed [3], as introduced below.

2.3.2 Time domain Shooting method

Traditional time domain integration method solves initial-value problems, while time Shooting method solves boundary-value problems. Time Shooting method allows computing the periodic steady state response more efficiently by reducing the transient phase. Considering the steady state solution over one period $[t_0, t_M]$, its state variables can be written as $x(t_0)$, $x(t_1)$, ... , $x(t_M)$. So the periodic steady state solution satisfies the two-point boundary constraint:

$$x(t_0) = x(t_M) \tag{2.1}$$

The variable $x(t_M)$ can be expressed as an explicit function of $x(t_0)$, named state transition function:

$$x(t_M) = \theta(x(t_0), t_M - t_0) \tag{2.2}$$

Combining equation (2.1) with equation (2.2), we get the following equation which can be then resolved by the Newton-Raphson method.

$$x(t_M) = \theta(x(t_0), t_M - t_0) - x(t_0) = 0 \tag{2.3}$$

The time domain Shooting method is mostly suitable to the digital and low-frequency application. As the frequency range increases, the simulation of distributed elements poses modeling and accuracy problems, hence the frequency domain method is developed, and the most effective one is Harmonic balance.

2.3.3 Harmonic balance

Harmonic balance (HB) method [8] [9] is now implemented in most of RF simulators. It is a numerical method for calculating the steady-state of the non-linear circuit. HB can rapidly find the steady-state and represent the results precisely for the periodic or quasi-periodic signals.

The principle of the HB is described as follows (Fig. 2.2): the circuit to be analyzed is divided into a linear network and a non-linear network, which connect each other by the forward and reverse Fourier transform (FT and FT^{-1}). The linear network is analyzed in the frequency domain, while the non-linear network is easier to be calculated in the time domain. As soon as the current and voltage are identical on both sides of the two network interfaces, a solution is found.

The resolution process of the HB is iterative: first it gives some estimation values to the interface currents on the side of the linear network. After the associated voltages at the interface of the linear network are calculated in the frequency domain, they are transformed in the time domain. These voltages values are then used to calculate the current of non-linear network in the time domain. Then, the currents obtained are transformed in the frequency domain, and compared with estimate current values in the linear network. If they are identical at all the frequency harmonics, the resolution process is over. Otherwise, the previous estimation is optimized and the resolution process restarts.

The use of efficient Krylov projection algorithm has made HB an efficient simulation method even for very large circuit including tens of thousands of transistors. The accuracy is in general better than time domain Shooting method.

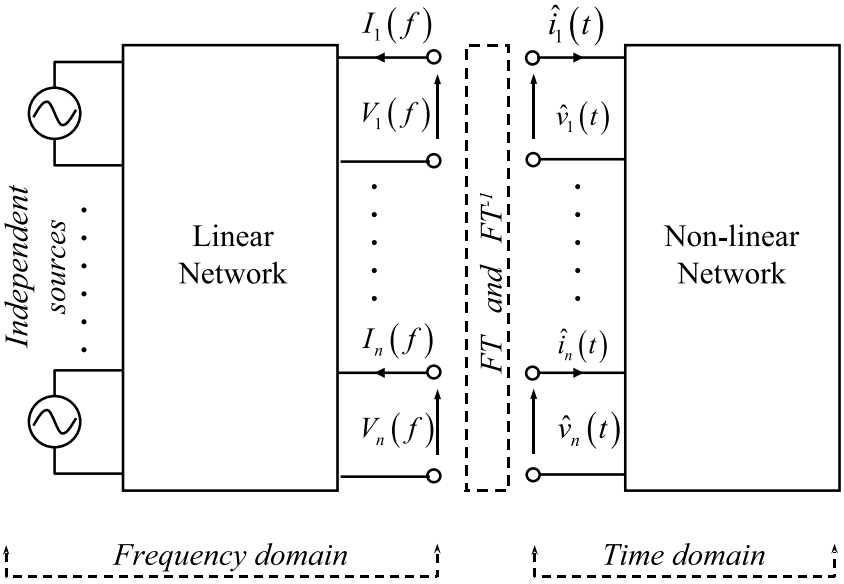


FIG. 2.2: *Harmonic balance method*

However, critical convergence problems can arise when simulating highly non-linear and digital circuits like frequency divider, PFD and charge pump. It is therefore necessary to assist harmonic balance with a prior transient analysis over a number of signal periods.

2.3.4 RF circuit Simulators

Here we briefly list some commercial RF circuit simulators in which harmonic balance or time domain Shooting method is employed (Tab. 2.1). Almost all RF circuit simulators use Harmonic balance, and some simulators implement two simulation methods (HB and time domain Shooting) to better adapt to the simulation requirements.

Simulator	Company	Environment	1st tech.	2nd tech.
SpectreRF	Cadence	ADE	PSS(Shooting)	HB
GoldenGate	Agilent	ADE	Carrier(HB)	Shooting
ADS	Agilent	ADS	HB	–
RFDE	Agilent	ADE	HB	–
Eldo RF	Mentor Graphic	DA-IC	HB	–

TAB. 2.1: *RF circuit simulators*

2.4 Existing PLL steady-state analysis method

As indicated previously, if the PLL has a steady-state, it can be analyzed with harmonic balance or Shooting method presented above. In this section, some existing methods

about the PLL simulation based on harmonic balance and Shooting method are concisely introduced. These methods may include:

- Brute force transistor-level simulation of the PLL
- Controlled brute force transistor-level simulation of the PLL
- Hierarchical simulation of the PLL

2.4.1 Brute force transistor-level simulation of the PLL

The simplest approach to calculate the steady-state of the PLL is the direct simulation at the transistor-level, which is referred to as brute force transistor-level simulation. This simulation can be done with Shooting method (PSS in the simulator SpectreRF [10]) or with the harmonic balance analysis "Harmonic balance" in the simulator ADS [11], or "Carrier analysis" of the GoldenGate simulator [12]). A transient simulation prior to HB or Shooting simulation is generally necessary in order to put a good initial condition for steady state analysis.

The idea behind the brute force transistor-level simulation is to leave the PLL to approach the steady-state on its own, and then use Shooting or HB method to refine the solution. However, in most cases it is difficult to do this for the following reasons: First, the PLL consists of two different circuits: analogue circuits (VCO, filter) and digital circuits (frequency divider, PFD), which makes the simulation of the PLL very complex. Second, the time constants in the PLL vary widely: the high frequency of the VCO's carrier is much larger than the loop bandwidth of the PLL, especially for a PLL with large division ratio. The simulation of such circuit requires a costly computation with much smaller time step over a wide time span called stabilization stage. Finally, to track the reference phase as accurately as possible, the PLL being very sensitive to the phase changes, it requires a very small time step. In fact, conversely for HB or Shooting analysis, it requires a large number of harmonics. The more number of harmonics is need, the smaller is the convergence valley. Consequently, the more memory would be spent with either Shooting or HB method.

In summary, the complexity of the brute force simulation of the PLL is proportional to the number of nodes of the circuit and the number of harmonics. Table 2.2 shows an example of the brute force transistor-level simulation for some simple PLL circuits. It's obvious that the simulation CPU time increases along with the division ratio (N), the number of nodes, the number of harmonics (NH) and the circuit stabilization time.

As said, the stabilization time is always needed to make the PLL approach the locked state. As described in the section 2.3.3, the resolution of the Harmonic balance begins at

Ckt	N	Num. of nodes	NH	Circuit stabilization time	Stabilization simulation time	HB/Shooting simulation time
I	2	120	128	2 us	11.6 min	0.9 min
II	4	132	128	3 us	19.2 min	3.1 min
III	8	217	128	4 us	45.8 min	3.5 min
IV	16	217	256	5 us	63.5 min	21.5 min
V	32	229	256	13 us	137.4 min	26.6 min
VI	56	229	512	20 us	213.6 min	59.1 min

TAB. 2.2: *Brute force transistor-level simulation. N: Division ratio; NH: Number of harmonic*

the estimation of the currents flowing into/out of the linear network. In fact, the more accurate the estimation is, the less number of iterations will be needed to converge. To make a good estimation, it's necessary to run the transient simulation sufficiently long until the transient significantly dies away, as shown in Fig. 2.3. The same is also true for Shooting method.

FIG. 2.3: *Evolution of the VCO control voltage*

In summary, the brute force transistor-level simulation is too slow, and is not convenient to verify the complex PLL with a large transistors count and division ratio.

2.4.2 Controlled brute force transistor-level simulation of the PLL

As seen from above, the long stabilization time is an obstacle to the brute force simulation of the PLL. To shorten this stabilization time, a number of techniques have been derived to constrain the transient analysis so that it gets quicker to the PLL locked state. Especially, these techniques have been systematized recently by G. Estep *et al.* [13], and an efficient transient control process has been proposed, as described below.

Recall that when the PLL is in the locked state, its output frequency and the phase shift with respect to the reference are constant. In general, the nominal output frequency of the PLL is known *a priori*. So the question becomes how to make the output frequency of the PLL move from an initial value to the nominal frequency as fast as possible. Once this target is achieved, the output frequency of the PLL is almost constant and the PLL gets into the lock process, in which the PLL can get locked quickly by varying primarily the phase shift. Since the PLL in the lock process is very close to the locked state, this may constitute a good estimation to HB or Shooting analysis, and bring about an easier convergence of the simulation.

In order to make the PLL output frequency move quickly from an initial value to the nominal frequency, the process depicted in Fig. 2.4 is proposed.

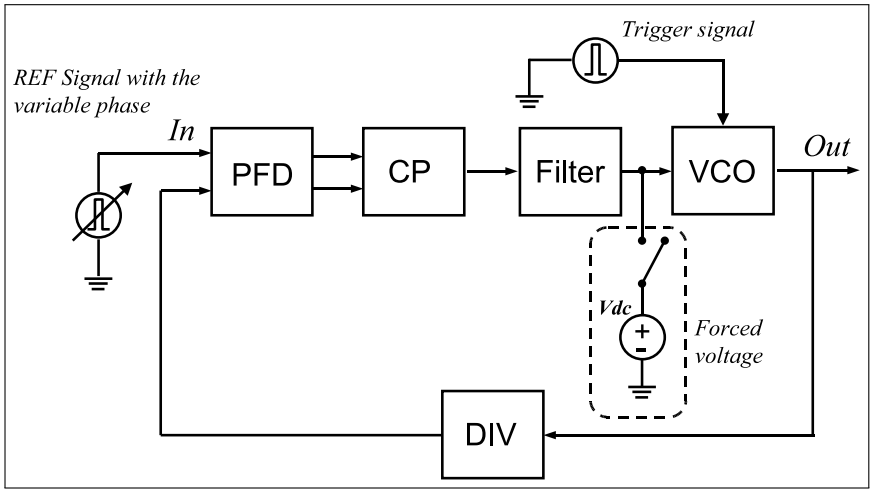


FIG. 2.4: *Controlled brute force transistor-level simulation of the PLL*

A forced DC voltage V_{dc} is connected to the input of the VCO through a switch, which is used to control the VCO output frequency. When the VCO input voltage is forced to a fixed value, the output frequency will move instantaneously to the specified frequency on the static voltage-frequency characteristic of the VCO.

At the input of the PFD, the reference signal can be delayed to modify the phase shift between VCO and the reference signal. A trigger signal is applied to the VCO in order to start the oscillation at an adequate time instant. However, being an autonomous circuit, the VCO can start the oscillation on its own due to the numerical noise. Therefore, it's impossible to predict when the VCO begins to oscillate. To make the VCO start the oscillation at a specified time, the trigger implements a "kick-start" principle (Fig. 2.5) [14]. For a LC-VCO, the trigger can be simply a current source (1mA, 0.1ns duration) in parallel with the inductor. For the ring oscillator, it is preferable to force one or two internal nodes of the circuit to high/low state at the beginning of the simulation.

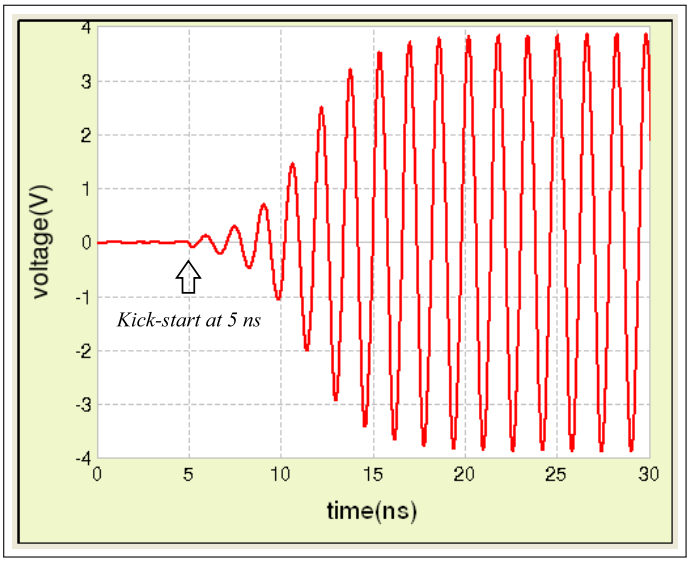


FIG. 2.5: *Start of the oscillation of the VCO with an external stimulation*

The details of the method are described as follows:

1. Provided that the nominal output frequency of the PLL is known, according to the static voltage-frequency characteristic of the VCO, the nominal control voltage of the VCO can be easily obtained.
2. Adjust the delay time of the reference signal, and the VCO trigger time to give an appropriate phase shift between the reference and the VCO.
3. Force the VCO control voltage with the nominal control voltage obtained in step 1 (i.e., turn on the switch), and do a Transient simulation for several periods so that the VCO output signal propagates to the divider output. Save the simulation results as initial conditions for the next simulation.
4. Remove the forced voltage source (i.e., turn off the switch), and do another Transient simulation based on the above initial condition, and let the PLL approach the locked state on its own. Save the results as an estimation of the next step.
5. Simulate the PLL with HB or Shooting using the previous estimation. If the analysis is converged, the steady-state of the PLL is found.

The simulation with controlled brute force method is fast, and the result obtained from HB analysis is precise. However, the final HB simulation time carried over the full PLL circuit is directly proportional to the division ratio. Hence the method is practically limited to PLL with division ration less than 100. Beyond that, the simulation time and the memory become prohibitive. In addition, the method requires a good approximation of the lock phase difference between the reference and the feedback to ensure a quick convergence. And this parameter is usually unknown to the designer.

2.4.3 Hierarchical simulation of the PLL

An alternative to the brute force transistor-level approach presented above is the hierarchical simulation approach. With this approach the circuit is broken into functional pieces for which the simulations are carried separately; then a top level simulation is carried to check the consistency of the block interfaces conditions.

An extensive number of works have been presented following this principle, which are usually termed as behavioral modeling/simulation techniques [1] [15] [16] [17]. By breaking the PLL into smaller pieces, the size of the simulated problems is reduced and each block can be simulated with an appropriate time rate and an appropriate algorithm or simulation tool. Doing so, the explosion of simulation time with PLL large division rank can be theoretically overcome. A very comprehensive description of this approach is given by K. Kundert in reference [1]. The difficulty with this approach however is that, to insure a consistent result accuracy, the top level simulation must consistently check the block interfaces loading conditions. And this is unfortunately not a trivial problem, especially as the PLL blocks have widely spaced time constants and deliver signals with power spectrum ranging from DC to the high frequency harmonics of the VCO. Hence most authors adopt the assumption of idealized block terminations, i.e., ideal square wave input and infinite load impedance (Fig. 2.6). This assumption results in hazardous simulation accuracy. Such a hierarchical simulation is therefore good for specification and top-down design phase, but not valuable for bottom-up verification, especially as the designs get more and more complex.

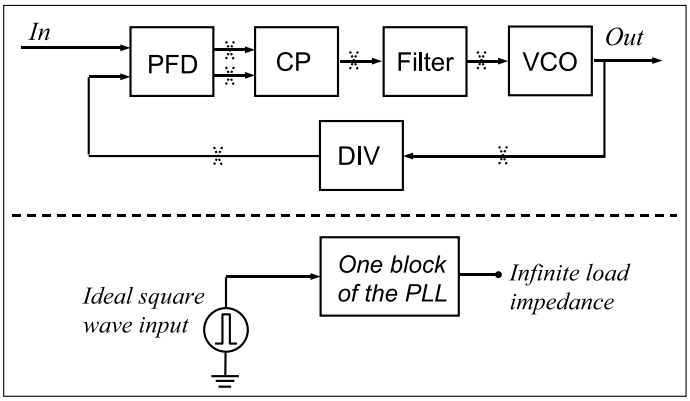


FIG. 2.6: *Hierarchical simulation with the idealized block loading conditions*

In this work we will reconsider a hierarchical simulation approach and propose a simulation and modeling strategy that accurately and effectively account for the interface conditions of the blocks, in order to provide a PLL simulation accuracy that is identical to brute force transistor level simulation.

2.5 Proposed method

As indicated in the previous section, the classical hierarchical method does not take into account the effects of the load impedance imperfections and the waveform distortions of the input signals, so the precision of the simulation is hazardous and does not fit well the transistor-level simulation. In this section, a new method is proposed considering the above effects, and an iterative algorithm is utilized to achieve a good precision and a quick convergence of simulation.

First, some principles of the new methods are introduced, then a general algorithm is presented, which is then described in detail. After that, the proposed method is verified by some examples of the PLL circuits and the simulation results are compared with the brute force transistor-level simulation.

2.5.1 Fundamentals of the proposed method

The diagram of the PLL is illustrated in Fig. 2.7. As seen from the diagram, the PLL forms a loop of 4 interconnected two-port blocks: VCO (B(1)), frequency divider or FD (B(2)), PFD/CP(B(3)), and the low-pass filter or LPF (B(4)). For the convenience of the technique to be described, the PFD and CP are grouped into a single block, and the reference source is considered as an internal signal to the block PFD/CP (B(3)).

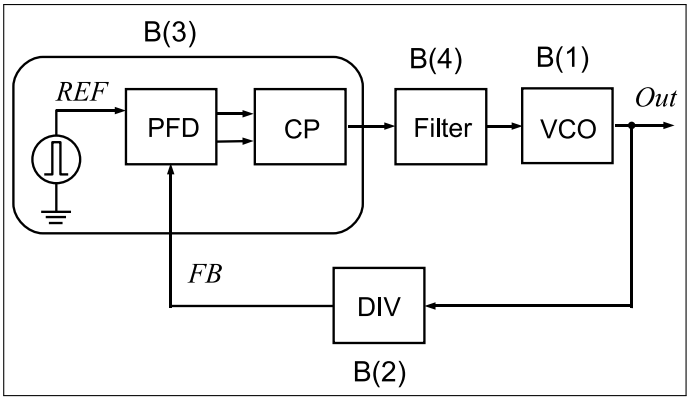


FIG. 2.7: *Diagram of the PLL*

In fact, in the close loop, each bloc $B(p)$ $(p = 1, 2, 3, 4)$ is loaded by its succeeding block $B(p+1)$ and serves as load to its preceding block $B(p-1)$, as illustrated in Fig. 2.8. The input voltage / current of the block B(p) is denoted $v_{in}(p,t)$ / $i_{in}(p,t)$, and its output voltage is denoted $v_{out}(p,t)$. The load impedance of the block B(p) is denoted $Z_{out}(p,k)$, where k represents the harmonic order of the frequency $f = k \cdot f_{ref}$. The input

impedance of the block B(p+1) $Z_{in}(p+1,k)$ is defined as below :

$$Z_{in}(p+1,k) = \frac{V_{in}(p+1,k)}{I_{in}(p+1,k)} \tag{2.4}$$

where $V_{in}(p+1,k)$ and $I_{in}(p+1,k)$ are Fourier transforms of $v_{in}(p,t)$ and $i_{in}(p,t)$.

Therefore, under lock condition, the output signal of the block $B(p)$: $v_{out}(p,t)$ equals the input signal of its succeeding block $B(p+1)$: $v_{in}(p+1,t)$. Also, the load impedance of the block $B(p)$: $Z_{out}(p,k)$ equals to the input impedance of its succeeding block $B(p+1)$: $Z_{in}(p+1,k)$.

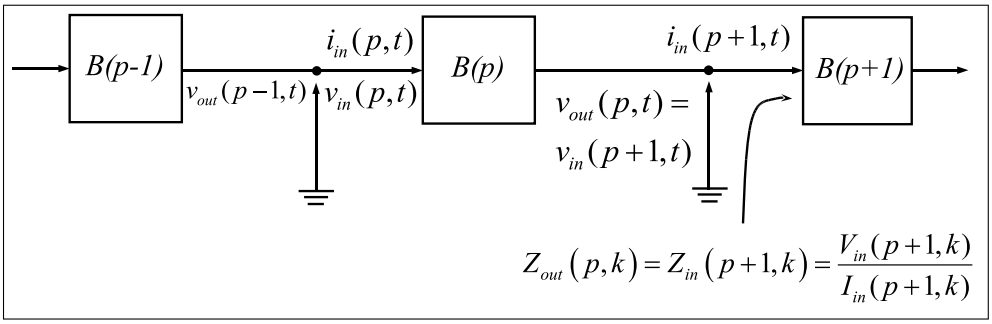


FIG. 2.8: *Interconnection of the blocks and the definition of the signals*

From the viewpoint of a given block $B(p)$, the internal structures of $B(p-1)$ or $B(p+1)$ are not of importance and can be regarded as black boxes. Given the input signal $v_{in}(p,t)$ or $i_{in}(p,t)$, and the load impedance $Z_{out}(p,k)$, the response of block $B(p)$ can be calculated without any information loss in the conditions shown in Fig. 2.9. $v_{in}(p,t)$ and $Z_{out}(p,k)$ will be termed the loading conditions of the block $B(p)$.

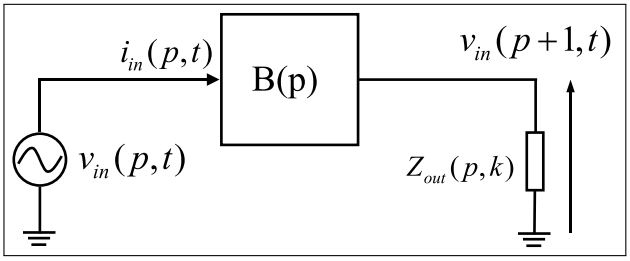


FIG. 2.9: *Simulation of a block with the input signal and the load impedance*

Thus, provided that the loading conditions of each block are known, the simulation of a close-loop PLL can be conducted as the simulation of the 4 individual blocks.

Therefore, the loading conditions $v_{in}(p,t)$ and $Z_{out}(p,k)$ become the unknown of our system, and will be computed iteratively until the convergence criterion is fulfilled. The corresponding algorithm will be presented in the next section.

To set the input signal of the block $B(p)$, a voltage/current source substitution technique should be used. This technique, introduced below, allows saving the output voltage of the preceding block $B(p-1)$ and then reusing it as a voltage source at the input of the block $B(p)$.

Introduction of voltage/current source substitution technique

Before going to the general algorithm of the proposed method, we briefly introduce the voltage/current source substitution technique, as illustrated in Fig. 2.10 below.

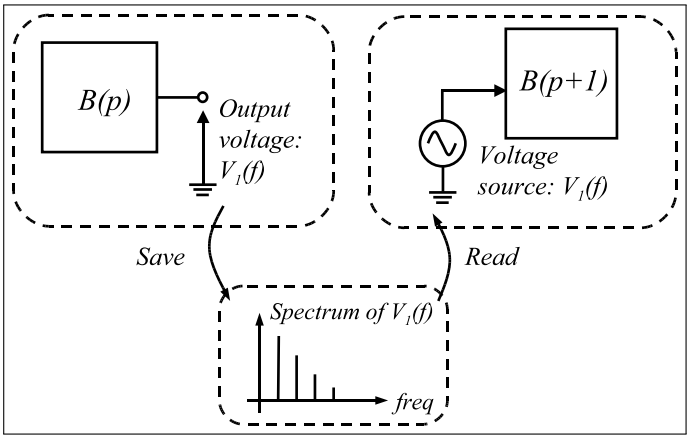


FIG. 2.10: *Mechanism of the voltage substitution*

Considering the assumption of a periodic steady-state operating point of the PLL, any node voltage or pin current can be represented by the Fourier series:

$$x(t) = \sum_{k=0}^{K} \hat{X}_k \cdot exp\left(j(2\pi k f t + P_k)\right) \tag{2.5}$$

where $\hat{X}_k$ is the magnitude of the signal, f the fundamental frequency, P_k the phase, and K number of harmonics.

Fourier series decomposition is very convenient for the method to be outlined, as we can save the spectrum of the voltage/current of a signal and then reuse it in the form of a voltage/current source for another circuit.

This voltage/current source substitution technique is available in most commercial simulators, like GoldenGate (COMBVS / COMBCS source [12]), and ADS / RFDE (V_spectrumDataset / I_SpectrumDataset source: frequency spectrum defined in dataset [11]). This technique will be used in the proposed algorithm in the following section.

2.5.2 Algorithm presentation

2.5.2.1 Introduction

Based on the principles described in the previous section, a general algorithm is derived, as presented below.

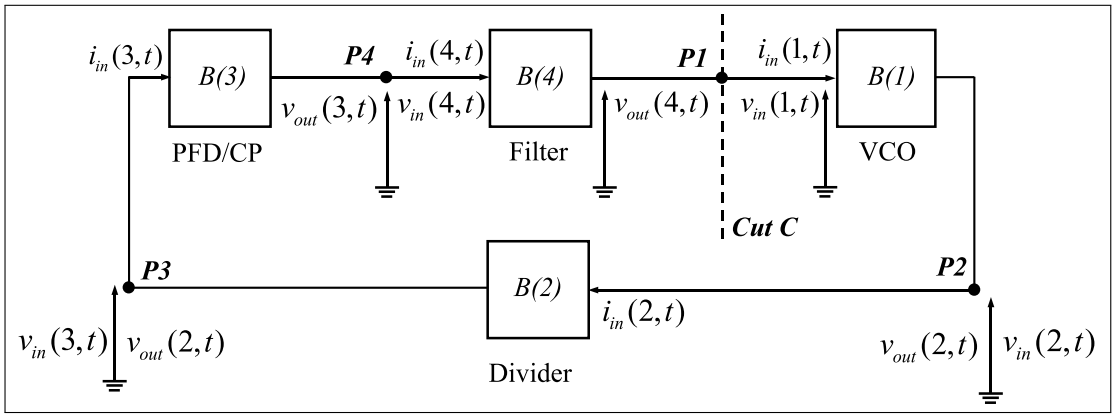


FIG. 2.11: *Block diagram of the PLL for the proposed algorithm*

Considering the PLL loop depicted in Fig. 2.11, where the input and output signals of each block are indicated, the general algorithm of the new method follows the process below:

- A cut C is made on the PLL loop at any of the points P_1 ,P_2, P_3, P_4.
- Let $v_{in}(R,t)$ be the voltage applied to the right of the cut C and $v_{out}(L,t)$ the voltage measured at the left of the cut C, as illustrated in Fig. 2.12.
- Next, starting from the right of the cut point with the excitation $v_{in}(R,t)$, each block is simulated at a time, with its output serving as the input for the succeeding block. An entire iteration is completed when we have computed the output at the left of the cut point $v_{out}(L,t)$.
- We now observe that the loop equilibrium is reached when $v_{out}(L,t) = v_{in}(R,t)$.

We therefore have defined a non-linear algebraic equation, which we will call the PLL lock equation, from which the steady-state condition can be computed:

$$\begin{cases} F(v_{in}(R,t)) = v_{out}(L,t) - v_{in}(R,t) = 0 \\ 0 \leq t < \infty \end{cases} \tag{2.6}$$

Now the question is, where to place the cut point in the loop?

In fact, equation (2.6) will be resolved by an iterative technique, as such it is important to select a point where a good estimation of $v_{in}(R,t)$ can be easily obtained.

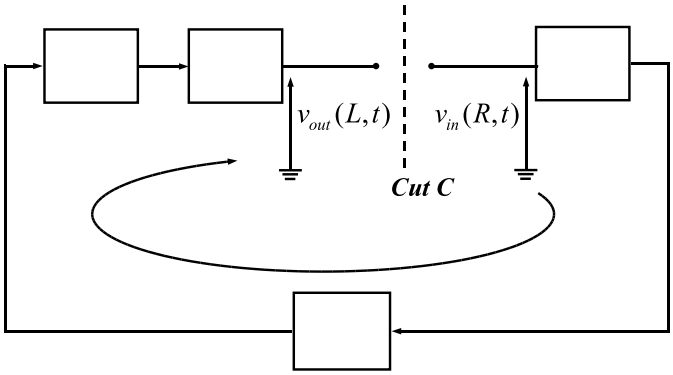


FIG. 2.12: *Block diagram of the PLL with arbitrary cut*

Then looking at the PLL structure, it's obvious that the cut should be placed at the input of the VCO. The reasons are as follows:

- Under locked condition, the VCO input voltage is merely a DC voltage.
- The nominal value of the VCO input voltage is easily obtained from the static frequency-voltage characteristic.

Doing so, the PLL lock equation writes:

$$\begin{cases} F\left(v_{in}(1,t)\right) = v_{out}\left(4,t\right) - v_{in}\left(1,t\right) = 0 \\ 0 \leq t < \infty \end{cases} \tag{2.7}$$

where $v_{out}\left(4,t\right)$ is the filter output voltage, and $v_{in}\left(1,t\right)$ is the VCO input voltage.

Now for the resolution of the equation (2.7), it is important to notice that the lock conditions in the PLL are achieved by the degree of freedom available in the VCO signal phase. Indeed, because the VCO is an autonomous circuit, for a given input signal $v_{in}\left(1,t\right)$, it accepts an infinite number of solutions that have the same waveform but differ on the phase at the time origin. Thus the lock conditions are achieved by adjusting both $v_{in}\left(1,t\right)$ and this free parameter.

Let us state this more quantitatively now. Under locked conditions or near locked conditions, the VCO input signal is a periodic signal whose fundamental frequency is the reference frequency f_{ref}, i.e.,

$$v_{in}\left(1,t\right) = \sum_{k} V_{in}\left(1,k\right) exp\left(j2\pi k f_{ref} t\right) \tag{2.8}$$

When this signal is applied to the input of the VCO, the reference frequency mixes with the self oscillation frequency of the VCO f_{vco}, so the VCO output $v_{out}\left(1,t\right)$ is a quasi-periodic signal with two fundamental frequencies f_{vco} and f_{ref}.

As a result, the output voltage of the VCO $v_{out}(1,t)$ can be expressed in the form of a modulated signal:

$$\begin{cases} v_{out}(1,t) = \sum_{l} \hat{v}_{out}(1,l,t) \exp(j2\pi l f_{vco} t) \\ \hat{v}_{out}(1,l,t) = \sum_{k} \hat{V}_{out}(1,l,k) \exp(j2\pi k f_{ref} t) \\ f_{vco} = N \cdot f_{ref} \end{cases} \tag{2.9}$$

In fact, it is possible to obtain this kind of signal directly by simulating the VCO as a self-oscillating mixer using harmonic balance.

However, since we are only looking at the PLL in the locked state, f_{vco} equals the reference frequency multiplied by the division ratio ($f_{vco} = f_{ref} \cdot N$), so a single tone f_{ref} will be enough for the HB analysis. As a result, the output voltage of VCO becomes periodic with one fundamental frequency f_{ref}:

$$v_{out}(1,t) = \sum_{k} V_{out}(1,k) \exp(j2\pi k f_{ref} t) \tag{2.10}$$

Actually, since the VCO is an autonomous circuit, any phase shifted version of the equation (2.9) along the frequency axis f_{vco} is also an admissible solution. So let us define $\Delta\phi_L$ the arbitrary VCO phase shift, and ΔT_L the corresponding time shift, so we have:

$$\Delta\phi_L = 2\pi f_{vco} \Delta T_L = 2\pi N f_{ref} \Delta T_L \tag{2.11}$$

From the above, the general form of the VCO output takes the following form:

$$\begin{aligned} v_{out}(1,t,\Delta\phi_L) &= \sum_{l} \hat{v}_{out}(1,l,t) \exp(jl(2\pi f_{vco} t - \Delta\phi_L)) \\ &= \sum_{l} \hat{v}_{out}(1,l,t) \exp(2\pi j l f_{vco}(t - \Delta T_L)) \end{aligned} \tag{2.12}$$

The divider output voltage can be obtained as follows:

$$\begin{aligned} v_{out}\left(2,t,\frac{\Delta\phi_L}{N}\right) &= \sum_{k} \hat{v}_{out}(2,k,t) \exp\left(jk(2\pi f_{ref} t - \frac{\Delta\phi_L}{N})\right) \\ &= \sum_{k} \hat{v}_{out}(2,k,t) \exp(2\pi j k f_{ref}(t - \Delta T_L)) \end{aligned} \tag{2.13}$$

Note that the output phase of the divider follows the output phase of the VCO, and this will in turn affects the PFD/CP output current and then the LPF output voltage $v_{out}(4,t)$. Because $\Delta\phi_L$ is arbitrary and thus independent of the VCO input voltage

$v_{in}(1,t)$, we now see that the equation (2.7) actually depends on two variables: the VCO input voltage $v_{in}(1,t)$, and phase shift $\Delta\phi_L$. Equation (2.7) can then be rewritten as follows:

$$\begin{cases} F_L\left(v_{in}(1,t), \Delta\phi_L\right) = v_{out}(4,t) - v_{in}(1,t) = 0 \\ 0 \leq t < \infty \end{cases} \tag{2.14}$$

Equation (2.14) defines the lock condition of the PLL, from which the steady-state solution will be determined.

To solve the equation (2.14), we propose a piecewise simulation strategy, starting from VCO (B (1)), sequentially, to LPF (B (4)). The simulation of a block B(p) assumes an arbitrary load impedance profile $Z_{out}(p,k)$, $p = 1,2,3,4$, $k = 0,1,2,...K$, and the input signal generated by the preceding block.

Doing so, we see that the lock conditions of the PLL are then obtained when the following system of equations is fulfilled.

$$\begin{cases} F_{outer}\left(Z_{out}(p,k), v_{in}(1,t)\right) = Z_{in}(p+1,k) - Z_{out}(p,k) = 0 \\ F_{inner}\left(\Delta\phi_L\right) = v_{out}(4,t) - v_{in}(1,t) = 0 \\ Z_{in}(p,k) = \dfrac{V_{in}(p,k)}{I_{in}(p,k)} \\ p = 1,2,3,4 \\ k = 0,1,2,...K \end{cases} \tag{2.15}$$

The overall resolution process of this equation set is then a two-tier equation solution loop. The inner loop equation $F_{inner}\left(\Delta\phi_L\right)$ solves for $\Delta\phi_L$, and the outer loop equation $F_{outer}\left(Z_{out}(p,k), v_{in}(1,t)\right)$ solves for $v_{in}(1,t)$ and the impedances $Z_{out}(p,k)$. The resolution of the above equations follows an iterative procedure which is depicted in Fig. 2.13.

First we give an overview of the general algorithm. Then in the next subsection the algorithm is simplified. After that, the detailed description is presented.

The algorithm begins with the estimation of the lock conditions described in the equation (2.15):

- Estimation of VCO input voltage $v_{in}(1,t)$
- Estimation of VCO phase shift $\Delta\phi_L$
- Estimation of load impedance of each block $Z_{out}(p,k)$ with $p = 1,2,3,4$ and $k = 0,1,2,...K$

With these estimations, the building blocks are sequentially simulated one by one. First, the VCO is simulated with the estimated input voltage $v_{in}(1,t)$ and the estimated

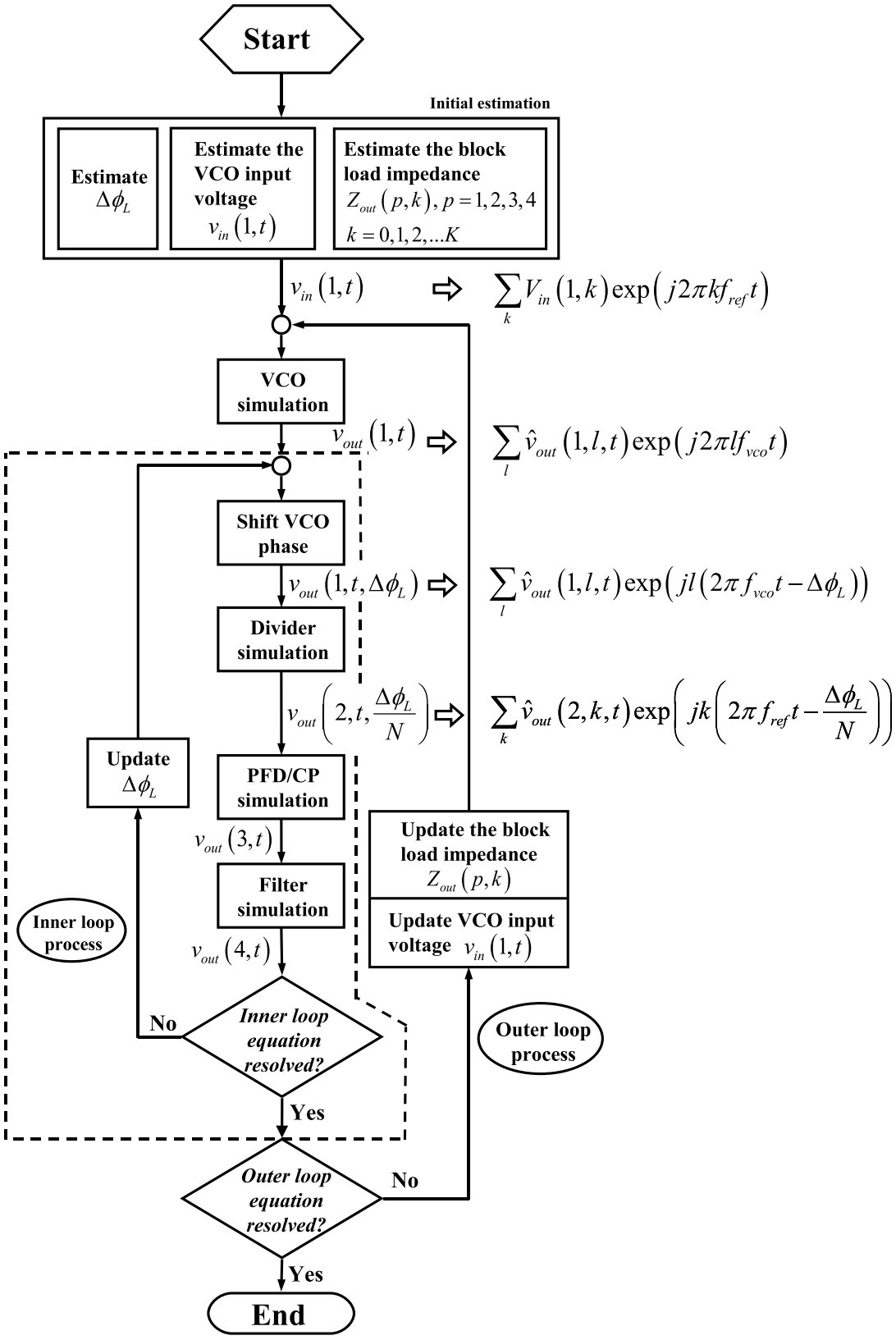


FIG. 2.13: *Procedure of the proposed algorithm*

load impedance $Z_{out}(1,k)$, then its output voltage $v_{out}(1,t)$ is obtained. The output phase of the VCO is shifted according to the estimated phase shift $\Delta\phi_L$, and the VCO voltage becomes $v_{out}(1,t,\Delta\phi_L)$, whose spectrum is saved and used as the input voltage source of the divider. Then the divider is simulated using the voltage source substitution technique and the estimated load impedance $Z_{out}(2,k)$. Its output voltage $v_{out}(2,t,\Delta\phi_L/N)$ is obtained and used as the input voltage of the PFD/CP. Next, the simulation of PFD/CP is driven by the divider output voltage and loaded by the estimated $Z_{out}(3,k)$. The simulation of PFD/CP is more complex than the simulation of the other blocks, and it will be described in detail in the subsection 2.5.3.4. The output voltage of PFD/CP $v_{out}(3,t)$ is then saved and sent to the input of the filter. After the filter simulation, one verifies if the inner loop equation is satisfied. If not, the phase difference value $\Delta\phi_L$ is updated and the output phase of the VCO shifted accordingly and then sent to the input of the divider. After that, divider, PFD/CP and filter are simulated again until the inner loop equation is satisfied.

Once the inner loop equation is fulfilled, the outer loop equation is checked. If the output loop equation is not solved, the load impedances of all blocks are updated as well as the VCO input voltage $v_{in}(1,t)$. When these updates are done, a new iteration begins from the VCO simulation until the satisfaction of the two loop equations.

Diagram 2.13 actually describes the resolution of a typical non-linear system of equation, and the resolution of which could theoretically be carried with Newton iteration. This however would require the computation of substantially large and dense Jacobian matrix for the outer loop equation, whose size is 4 times the number of harmonics in the circuit. Fortunately enough, we have observed that the outer loop equation can be made to converge very quickly with a simple fixed point iteration, provided that a good estimate is given. This avoids the computation and factorization of the Jacobian. The update of the VCO input voltage and load impedances at iteration i is obtained as:

$$\begin{cases} v_{in}(1,t)^{(i+1)} = v_{out}(4,t)^{i} \\ Z_{out}(p,k)^{(i+1)} = Z_{in}(p+1,k)^{(i)} = \dfrac{V_{in}(p+1,k)^{(i)}}{I_{in}(p+1,k)^{(i)}} \end{cases} \tag{2.16}$$

where the superscript i and $i+1$ represent the i^{th} and $(i+1)^{th}$ iteration.

The inner loop equation, being a single variable equation in $\Delta\phi_L$, it can in turn be efficiently resolved by either a Newton iteration or dichotomy.

Note that the inner loop includes the simulation of 3 blocks (FD, PFD/CP and LPF) while the outer loop involves the four blocks. The inspection of the inner loop however shows that under some assumptions easily verified, it can be simplified and reduced only

to PFD/CP block simulation, as outlined below.

2.5.2.2 Simplified algorithm

According to the equation (2.13), the divider output voltage can be written as:

$$v_{out}\left(2,t,\Delta T_L\right)=\sum_k \hat{v}_{out}\left(2,k,t\right)\exp\left(j2\pi k f_{ref}\left(t-\Delta T_L\right)\right) \tag{2.17}$$

Note that for convenience and without loss of generality, we express the reference voltage with zero phase at time origin, i.e.,

$$v_{ref}\left(t\right)=\sum_k \hat{V}_{ref}\left(k\right)\exp\left(j2\pi k f_{ref}t\right) \tag{2.18}$$

Hence, the phase shift between the reference and the feedback signals is simply:

$$\Delta\phi_{FB}=2\pi f_{ref}\Delta T_L \tag{2.19}$$

Comparing the equation (2.19) and (2.11), we have:

$$\Delta\phi_{FB}=\Delta\phi_L/N \tag{2.20}$$

According to the general algorithm presented in Fig. 2.13, if the inner loop equation is not fulfilled, the phase shift $\Delta\phi_L$ is updated and the output phase of the VCO is shifted according to the new value of $\Delta\phi_L$. Then the divider is simulated with the phase-shifted VCO output voltage. In fact, since the input signal of the divider is simply phase shifted, the output of the divider will also be a simple phase-shifted version of previous divider solution. Hence a phase shift $\Delta\phi_L$ at the VCO output is equivalent to a phase shift $\Delta\phi_L/N$ at the output of the divider. Therefore, we can eliminate the divider simulation from the inner loop process and adjust directly the output phase of the divider with the equivalent phase shift $\Delta\phi_L/N$, as illustrated in Fig. 2.14.

Furthermore, the simulation of the filter can also be removed from the inner loop process if we make the reasonable assumption that LPF attenuates significantly the reference frequency and all its harmonics. With that assumption we then observe that under the desired PLL lock conditions, both the LPF output voltage $v_{out}\left(4,t\right)$ and VCO input voltage $v_{in}\left(1,t\right)$ are practically constant. So let us approximate them to DC signals for a

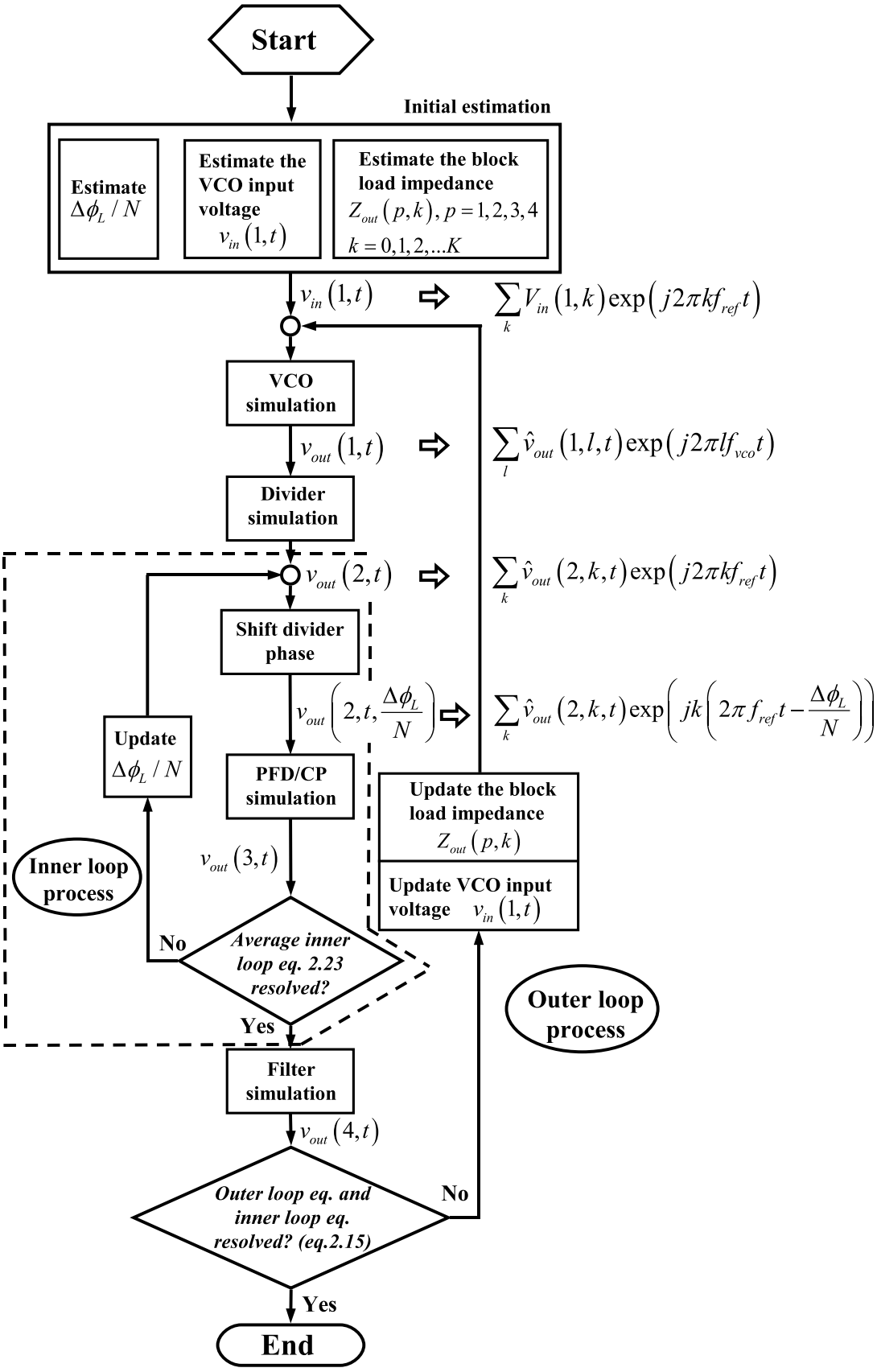


FIG. 2.14: *Simplified procedure of the proposed algorithm*

while:

$$\begin{cases} v_{in}(1,t) \simeq v_{in}^{DC}(1) = \overline{v_{in}(1,t)} \\ v_{out}(4,t) \simeq v_{out}^{DC}(4) = \overline{v_{out}(4,t)} \end{cases} \tag{2.21}$$

On the other hand, the usual architecture of LPF circuits (see chapter 1), are such that under DC conditions, the LPF output voltage is identical to the PFD/CP output voltage, namely $v_{out}(3,t)$, thus:

$$\overline{v_{out}(3,t)} \approx \overline{v_{out}(4,t)} = v_{out}^{DC}(4) \tag{2.22}$$

Hence, it is not necessary to simulate the LPF but consider only the PFD/CP. Under these assumptions, the inner loop problem simplifies to

$$F_{inner}(\Delta\phi_L) = \overline{v_{out}(3,t)} - \overline{v_{in}(1,t)} = 0 \tag{2.23}$$

The phase shift $\Delta\phi_{FB} = \Delta\phi_L/N$ of the feedback signal is to be adjusted until the above condition is satisfied. The resolution of the equation involves therefore only the simulation of the PFD/CP block, and the lock condition is verified only on the average value (DC component) of the output voltage (eq. (2.23)).

The simplified algorithm is then summarized in the diagram 2.14. Note that the assumption for DC conditions at LPF input is made only within the inner loop process; once the output from the PFD/CP is obtained, this assumption is relaxed, the real filter conditions are simulated and the exact lock condition is verified in the outer loop process, in order to guaranty an exact transistor level accuracy.

2.5.3 Algorithm description

In the previous section, the general algorithm is presented, in which several simulation steps are involved. Basically, these simulation steps can be classified into the following groups and will be described in detail in this section.

- Initial estimates (VCO input voltage, phase shift, load impedances)
- VCO / Divider simulation
- Inner loop process (including PFD+CP simulation)
- Outer loop process (including LPF simulation, complete inner loop and outer loop equations check, VCO input voltage and output load impedances updates)

2.5.3.1 Initial estimates

The following procedure describes the acquisition of initial conditions for VCO input voltage, VCO phase shift and load impedances that guaranty a quick convergence of the algorithms.

A. VCO input voltage

To find out the nominal value of VCO input voltage $v_{in}(1,t)$, the VCO is driven by a DC voltage, and an HB analysis of the VCO is performed, sweeping the input DC voltage. Then the voltage-frequency curve of the VCO is traced, as shown in Fig. 2.15. Since the nominal frequency of the VCO is known *a priori*, the nominal value of v_0 can be easily found using a Newton iteration. The initial estimate of $v_{in}(t)$ is set then to $v_{in}(t) = v_0$, $\forall t$.

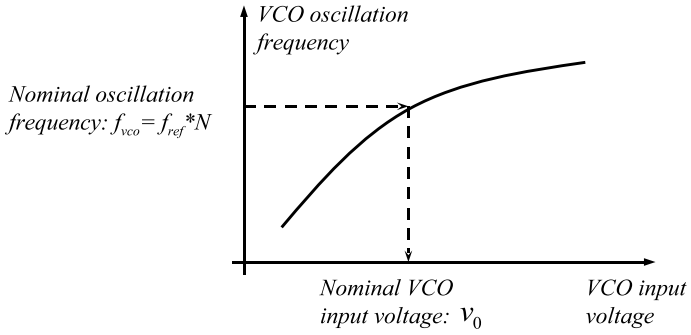


FIG. 2.15: *VCO: search for the nominal control voltage from the voltage-frequency curve*

B. Initial VCO phase shift

In the proposed algorithm, a zero phase is assumed to the reference signal and an arbitrary phase shift $\Delta\phi_L$ is considered for the VCO output, which results in a phase shift $\Delta\phi_L/N$ at the divider output. The estimation of the phase shift will depend on the designer's knowledge about the PLL circuit. Generally, for a charge-pump PLL, the nominal phase difference is nearly zero [18], so $\Delta\phi_L = 0$ can constitute a good initial value. In any case this initial value is not a critical parameter since it will be easily adjusted in the inner loop resolution, as will be seen next.

C. Load impedances

A brief description of the various PLL blocks has been given in chapter 1. To estimate the load impedance value of each block, the characteristics of each block are considered below. Recall that the load impedances of four blocks are expressed by $Z_{out}(p,k)$ where $p = 1,2,3,4$, and k is the harmonic order of frequency $f = kf_{ref}$ with $k = 0,1,2,...K$.

1. **VCO block:** As indicated in chapter 1, a buffer circuit is usually added at the output of the VCO, so the VCO is generally terminated by a high impedance. Therefore a

good estimate for its load impedance is:

$$Z_{out}(1,k) \approx \infty \text{ ... k} = 0,1,2,...\text{K} \tag{2.24}$$

2. **Divider block:** Similarly to VCO, due to the buffer circuit at the output of the divider, its load impedance is also usually very large, so a good estimate is:

$$Z_{out}(2,k) \approx \infty \text{ ... k} = 0,1,2,...\text{K} \tag{2.25}$$

3. **PFD+CP block:** The PFD+CP is terminated with a LPF with a cut-off frequency well below the reference frequency. Hence a good estimate of its load impedance is:

$$Z_{out}(3,k) \approx \begin{cases} \infty \text{ ... k} = 0 \\ 0 \text{ ... k} \geq 1 \end{cases} \tag{2.26}$$

4. **LPF block:** The load of low-pass filter is the VCO input. The input impedance of the VCO at DC is close to the infinity, and at high frequencies it's much larger than the filter output impedance. Thus, the load impedance of the filter can be estimated to:

$$Z_{out}(4,k) \approx \infty \text{ ... k} = 0,1,2,...\text{K} \tag{2.27}$$

2.5.3.2 VCO simulation

The VCO simulation bench is illustrated in Fig. 2.16. For the initial iteration, the input voltage $v_{in}(1,t)$ is DC voltage as given by the estimate described in the above section, and the load impedance is infinite, as above (eq. (2.24)). Therefore the HB analysis is usually very fast.

In the subsequent iterations, $v_{in}(t)$ and $Z_{out}(1,k)$ are updated in the outer loop process (section 2.5.3.5 B and section 2.5.3.5 C). The convergence is quickly obtained as we start the simulation from the previous solution.

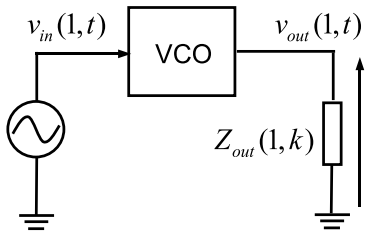


FIG. 2.16: *VCO simulation bench*

2.5.3.3 Divider simulation

The divider simulation bench is illustrated in Fig. 2.17. The input voltage $v_{in}(2,t)$ is given by the VCO output voltage $v_{out}(1,t)$. With regard to the load impedance, it's is infinite for the initial iteration, as indicated in eq. (2.25). For subsequent iterations, it is updated in the outer loop process (section 2.5.3.5 C).

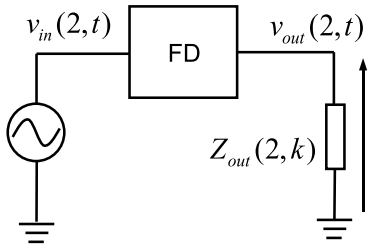


FIG. 2.17: *Divider simulation bench*

2.5.3.4 Inner loop process

In this section we will describe the inner loop process. As depicted in Fig. 2.18, the inner loop process involves the divider phase shift, the PFD/CP simulation, the average inner loop equation check and the VCO phase shift update, . The PFD/CP simulation is the critical part of the algorithm, the other parts are trivial. We will distinguish between the initial iteration and the subsequent iterations.

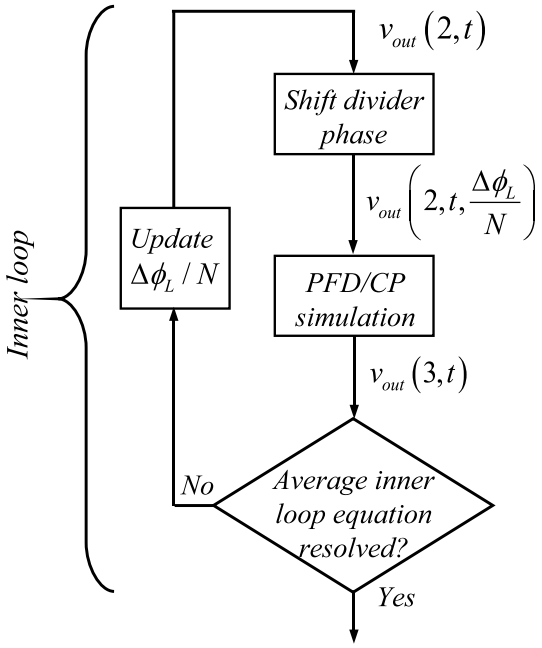


FIG. 2.18: *Resolution of the inner loop equation*

A. Initial iteration

In the initial inner loop iteration, the PFD/CP simulation is carried out with a load impedance estimation profile indicated in equation (2.26), i.e., an open circuit at DC and a short circuit for the reference frequency and its harmonics. Note however that because the PFD/CP has a digital circuit behavior, it is necessary to assist Harmonic balance simulation with a Transient analysis to achieve a fast convergence. Unfortunately, the discrete impedance profile specified by equation (2.26) is not very practical for a Transient analysis. The solution consists therefore in synthesizing this impedance profile by means of an ideal DC voltage source that is constrained to have a zero average current flow (see Fig. 2.19). Doing so results in a very effective Transient and HB simulation.

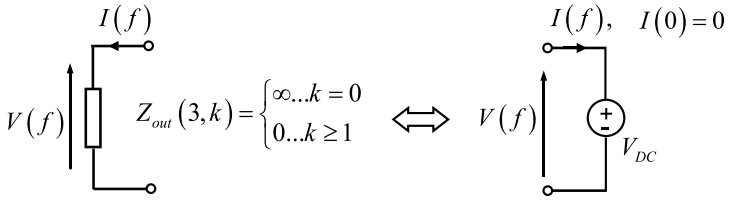


FIG. 2.19: *Synthesis of PFD/CP load impedance profile by means of an ideal DC voltage source*

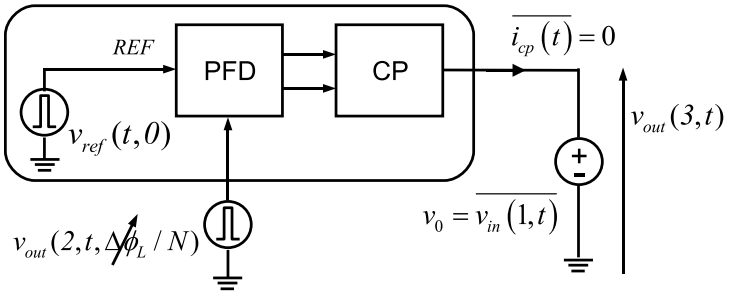


FIG. 2.20: *PFD+CP simulation bench in the initial iteration*

Fig. 2.20 thus represents the circuit testbench for the PFD/CP simulation. The output voltage of the PFD/CP is forced to the average of the estimated VCO input voltage $v_0 = \overline{v_{in}(1,t)}$. An estimated phase shift $\Delta\phi_L/N$ is affected to the FD output signal $v_{out}(2,t,\Delta\phi_L/N)$. The circuit is simulated with Harmonic balance, a current $i_{cp}(t)$ is measured at the CP output and the output impedance emulation is satisfied if the average current $\overline{i_{cp}(t)}$ is zero. Observe that, in these conditions, the average inner loop equation (2.23) is always satisfied, so that we have turned the inner loop constraint from equation (2.23) to the following new equation:

$$F_{inner}(\Delta\phi_L) = \overline{i_{cp}(t)} = 0 \tag{2.28}$$

Equation (2.28) is a single variable equation in $\Delta\phi_L$, which can be resolved iteratively with Newton-Raphson algorithm. Once the equation (2.28) is resolved, we can obtain the

nominal VCO phase shift $\Delta\phi_L^*$ as sketched in Fig. 2.21. Note that $\overline{i_{cp}(t)}$ is a periodic function in $\Delta\phi_L$ with a period $N \times 2\pi$, as shown in Fig. 2.22.

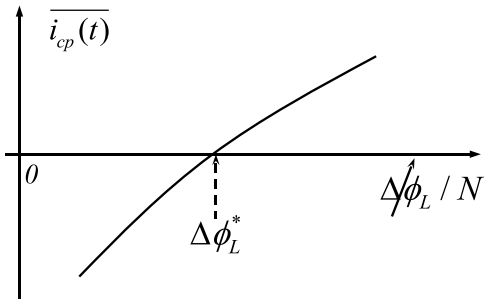


FIG. 2.21: *Relationship between the DC output current of CP and $\Delta\phi_L/N$*

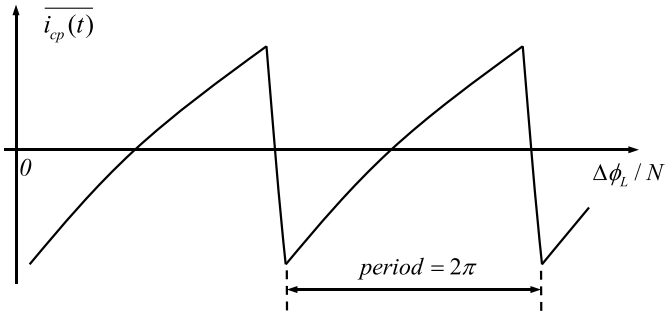


FIG. 2.22: *Relationship between DC output current of CP and $\Delta\phi_L/N$ - 2 periods*

B. Subsequent iterations

In the subsequent iterations, one does no more need a Transient assistance to help Harmonic balance analysis, since the results from the previous iteration will constitute a good estimate to achieve a fast HB convergence. So we may directly terminate the PFD/CP with the discrete load impedance profile $Z_{out}(3,k)$, as shown in Fig. 2.23. An estimated phase shift $\Delta\phi_L/N$ is affected to the FD output signal $v_{out}(2,t,\Delta\phi_L/N)$. Upon the convergence of Harmonic balance, one obtains the CP output voltage $v_{out}(3,t)$. The original average inner loop equation (2.23), recalled below for convenience, is checked for loop convergence.

$$F_{inner}(\Delta\phi_L) = \overline{v_{out}(3,t)} - \overline{v_{in}(1,t)} = 0 \tag{2.29}$$

Note that the average CP output voltage $\overline{v_{out}(3,t)}$ is also a periodic function of the phase shift $\Delta\phi_L$ of period $N \times 2\pi$. Fig. 2.24 shows a typical plot of that relationship. Interestingly one notices that this curve alternates between flat regions and narrow / steep regions. For a good and stable PLL, the locking point $\Delta\phi_L^*$ happens in the steep region, so that any small phase deviation breaks the lock conditions and the loop restores back. The frequency synthesis accuracy is therefore proportional to the slope of this plot.

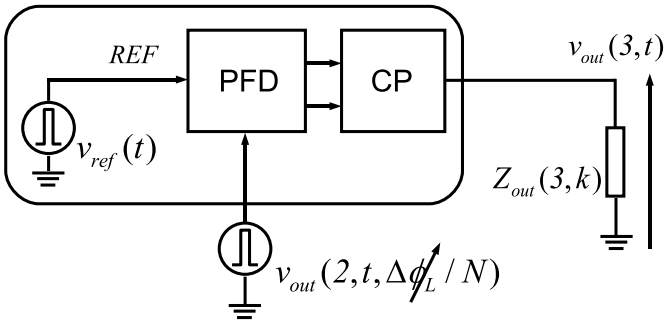


FIG. 2.23: *PFD+CP simulation bench in the second and the following iterations*

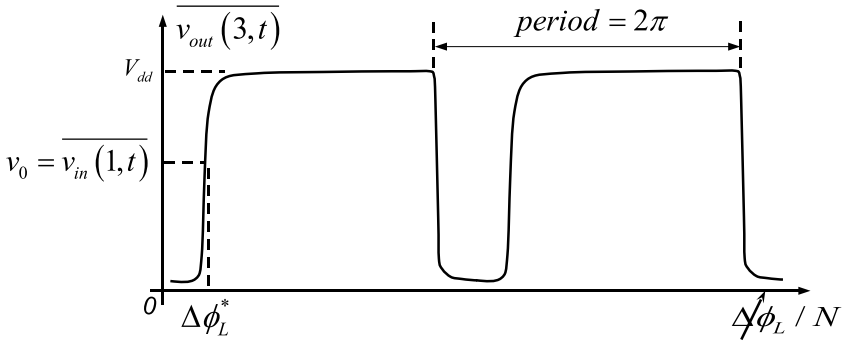


FIG. 2.24: *Relationship between the DC output voltage of CP and $\Delta\phi_L/N$ - 2 periods*

A Newton-Raphson algorithm can also be used to solve equation (2.29) for $\Delta\phi_L^*$. This is usually quickly performed because the previous loop iterations will provide a good initial estimation, despite the Newton iteration convergence valley can be narrow here.

2.5.3.5 Outer loop process

The outer loop process involve many operations: LPF simulation, complete inter-loop and outer-loop equation checks, VCO input and the four blocks load impedances updates, as described below.

A. LPF simulation

The filter simulation is simple and does not pose any problem. The schematic is illustrated in Fig. 2.25. The CP output voltage $v_{out}(3,t)$ drives the filter terminated with estimated load $Z_{out}(4,k)$. The HB simulation is virtually instantaneous and provides the voltage $v_{out}(4,t)$.

B. Complete inner loop and outer loop equations check

After the LPF simulation we are in a position to check the two loop equilibrium equations (eq. (2.15)). When these are satisfied we obtain the PLL solution with transistor-level accuracy; otherwise we need to update the VCO input voltage $v_{in}(1,t)$ and the load

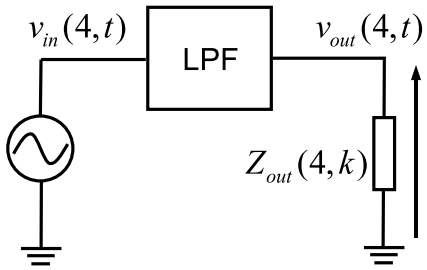


FIG. 2.25: *Loop filter simulation*

impedances of all the four blocks $Z_{out}(p,k)$, $p = 1,2,3,4$, $k = 0,1,...,K$ as below, and restart another iteration.

C. VCO input voltage and output load impedances updates

Recall that, given the phase shift $\Delta\phi_L^*$ from the inner loop resolution, equation (2.15) is a non-linear system of equations in $4 \times K + 1$ unknowns. As stated, we have chosen to resolve that with a fixed point iteration technique, which has shown good convergence in this particular problem. Thus the updates at iteration $i+1$ express simply as:

$$\begin{cases} v_{in}(1,t)^{(i+1)} = v_{out}(4,t)^{i} \\ Z_{out}(p,k)^{(i+1)} = Z_{in}(p+1,k)^{(i)} = \dfrac{V_{in}(p+1,k)^{(i)}}{I_{in}(p+1,k)^{(i)}} \\ p = 1,2,3,4 \\ k = 0,1,...,K \\ Z_{in}(5,k) = Z_{in}(1,k) \end{cases} \tag{2.30}$$

where the superscript i and $i+1$ represent the i^{th} and $(i+1)^{th}$ iteration.

2.5.3.6 Discussion on the algorithm termination conditions

Because of the fixed point iteration updates (eq. (2.30)), at iteration $i+1$, the check on the lock conditions (2.15) expresses as:

$$\begin{cases} F_{outer}(\ ) = Z_{in}(p+1,k)^{(i+1)} - Z_{in}(p+1,k)^{(i)} = 0 \\ F_{inner}(\ ) = v_{out}(4,t)^{(i+1)} - v_{out}(4,t)^{(i)} = 0 \end{cases} \tag{2.31}$$

Hence we may define an effective relative tolerance criterion ε_{tol} for stopping the algorithm, i.e.,

$$\begin{cases} \left| \dfrac{Z_{in}(p+1,k)^{(i+1)} - Z_{in}(p+1,k)^{(i)}}{Z_{in}(p+1,k)^{(i)}} \right| < \varepsilon_{tol} \\ \left| \dfrac{V_{out}(4,k)^{(i+1)} - V_{out}(4,k)^{(i)}}{V_{out}(4,k)^{(i)}} \right| < \varepsilon_{tol} \end{cases} \tag{2.32}$$

2.5.4 Numerical application – Step by step simulation of a PLL

In this section we present a numerical application of the proposed method. We will apply step by step the algorithm onto a realistic PLL circuit to verify the effectiveness of the proposed solution. The simulation process is recapitulated in Fig. 2.26, which consist in simulating the PLL blocks one by one and checking for the interface consistency conditions at the end of each algorithm cycle.

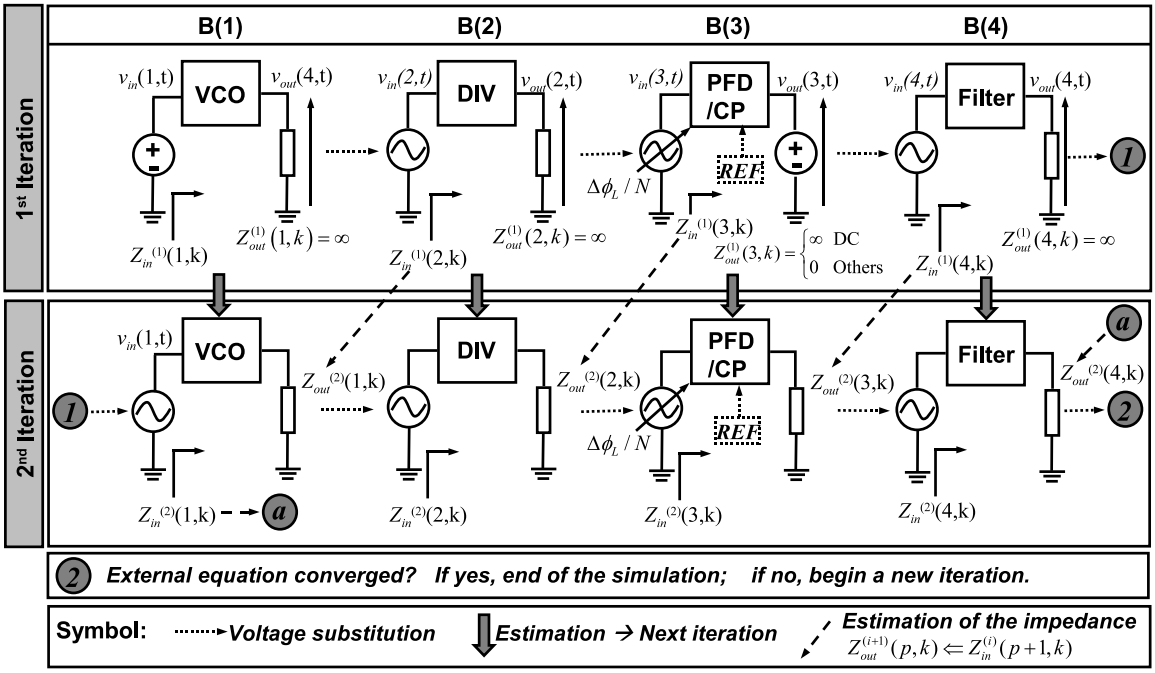


FIG. 2.26: *Verification of the new method*

The PLL circuit example is realized on CMOS technology with a division rank of 16 (Reference frequency = 38.88 MHz, output frequency = 622 MHz). It comprises 463 MOS devices, and uses a PFD+CP structure and a passive 2nd order LPF.

The first thing to start with is the estimation of the nominal VCO input voltage. Given a reference frequency of 38.88 MHz and a division rank of 16, the nominal output frequency is to be 622 MHz. Then the static input-output characteristic of the VCO is traced with HB to obtain the nominal VCO input voltage value. The VCO input-output characteristic is shown in Fig. 2.27 and the nominal control voltage is found to be $v_0 = 1.71V$. For that nominal control voltage value, the VCO output voltage waveform is shown in Fig. 2.28. Considering that waveform at the input of the FD, we obtain the nominal divider output waveform shown in Fig. 2.29, which is nearly a square waveform.

The next stage is then to simulate the PFD+CP. For the initial iteration, it needs to load a DC voltage of $v_0 = 1.71V$ at the output of PFD/CP and to drive its feedback input with the nominal FD output waveform. Then we vary the feedback input phase

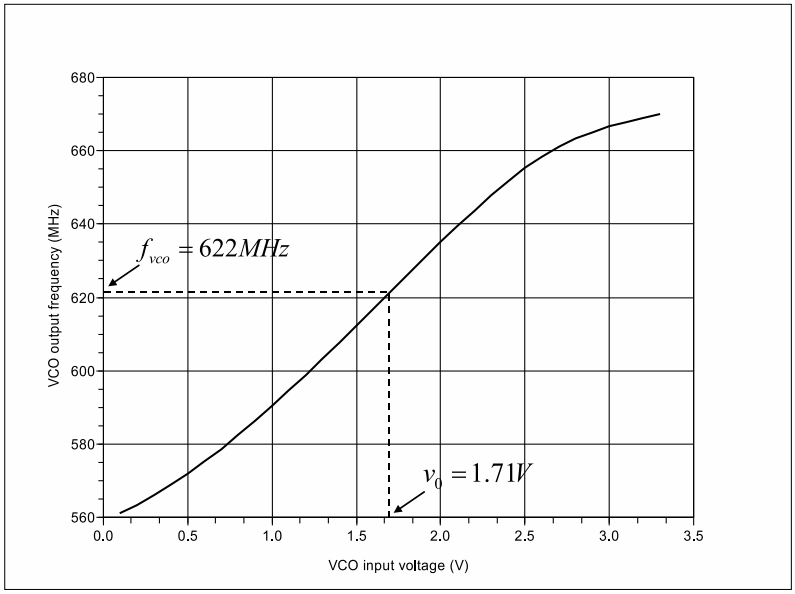


FIG. 2.27: *VCO input-output static characteristic*

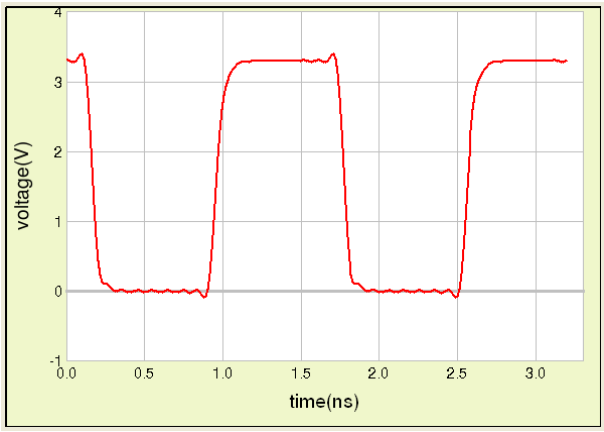


FIG. 2.28: *VCO output voltage* $v_{out}(1,t)$

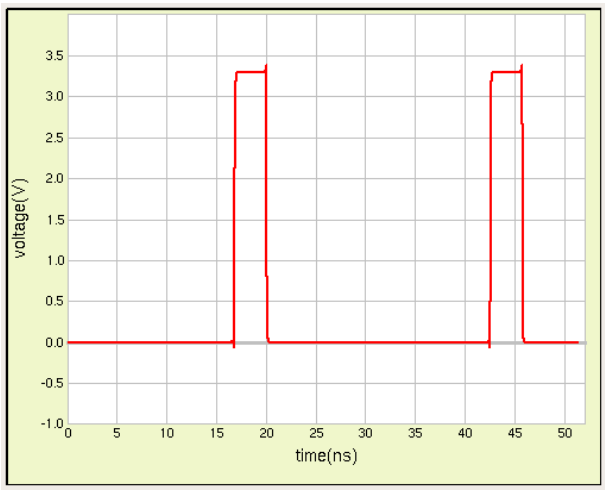


FIG. 2.29: *Divider output voltage* $v_{out}(2,t)$

$\Delta\phi_L/N$ and monitor the average output current $i_{cp}(t)$. The characteristic $\overline{i_{cp}(t)}$ vs. $\Delta\phi_L/N$ is shown in Fig. 2.30, the nominal phase shift $\Delta\phi_L^*$ is found to be $\Delta\phi_L^* = 1.33°$, when the characteristic crossed the zero axis. For the subsequent iterations, the DC voltage source is removed and replaced by a discrete load impedance profile $Z_{out}(3,k)$. By varying the feedback input phase $\Delta\phi_L/N$, we monitor the average output voltage $\overline{v_{out}(3,t)}$. The characteristic $\overline{v_{out}(3,t)}$ vs. $\Delta\phi_L/N$ is shown in Fig. 2.31, the nominal phase shift $\Delta\phi_L^* = 1.33°$ is found when $\overline{v_{out}(3,t)} = v_0$, and this is unchanged as to the first iteration.

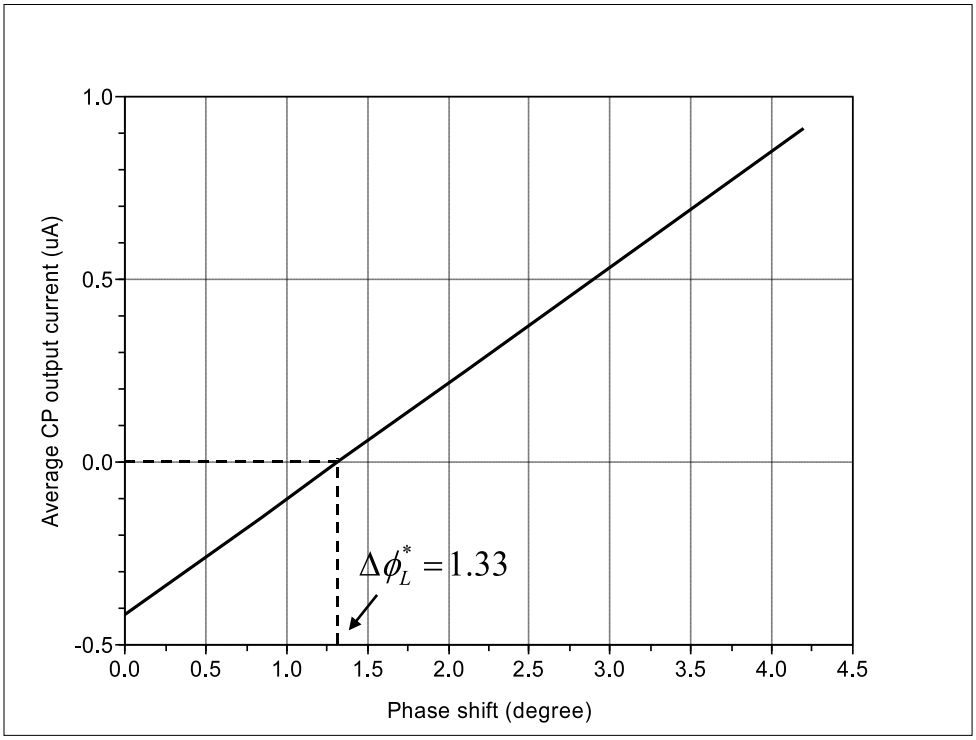


FIG. 2.30: *Relation* $\overline{i_{cp}(t)}$ *vs. phase shift* $\Delta\phi_L/N$

The first iteration of the algorithm is then terminated. For entering a new iteration, we update the VCO input voltage and the load impedances of all blocs, as stated in equation (2.30). At the end of each new iteration, we monitor the variations of the block input impedances and the LPF output voltage, according to equation (2.32). Fig. 2.32 to Fig. 2.35 show in a logarithmic scale the evolution of the input impedances of the various blocks with the iteration number. We then see that the algorithm converges very quickly within 4 iterations. The variation of the LPF output voltage spectrum at each loop iteration end is shown in Fig. 2.36, and it converges within 4 iterations too.

Fig. 2.37 shows the waveforms of the reference and the feedback signals of the PFD under locked condition obtained with the proposed method. We see that they are well synchronized. The zoom of the rising signal front (Fig. 2.38) shows that the two input signals are time shifted by $\Delta T_L = 95ps$, corresponding to a phase

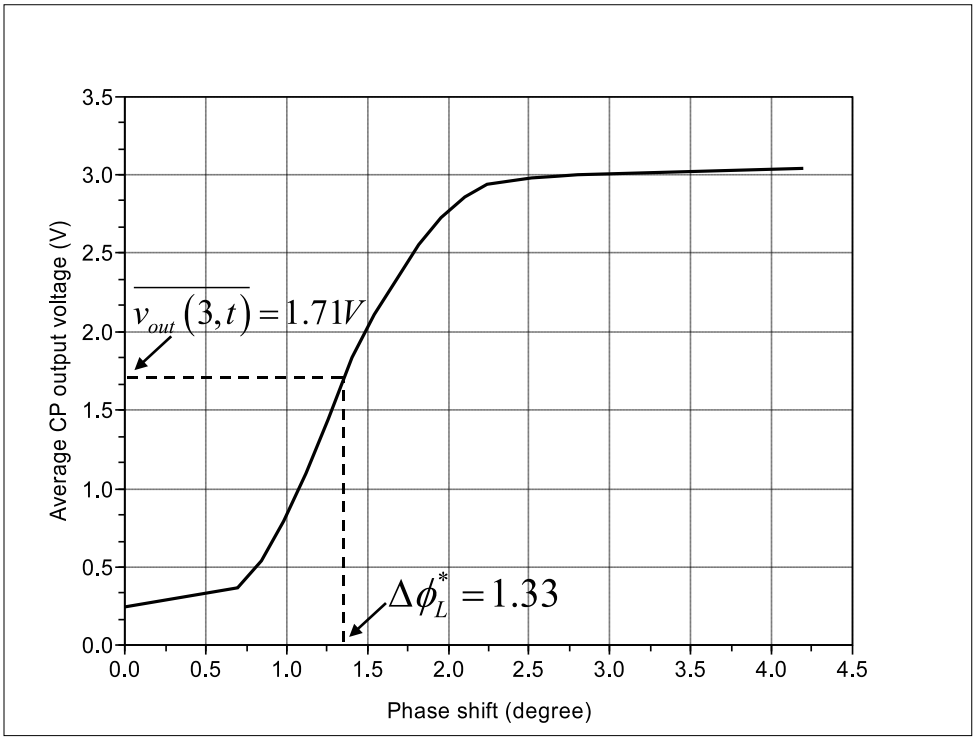


FIG. 2.31: *Relation* $\overline{v_{out}(3,t)}$ *vs. phase shift* $\Delta\phi_L/N$

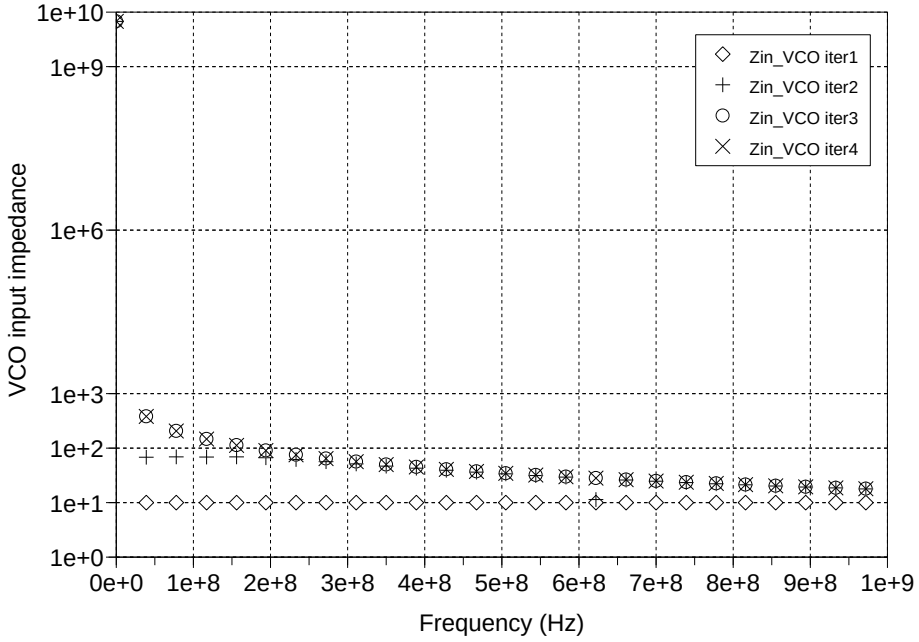


FIG. 2.32: *Evolution of the input impedance of the VCO:* $Z_{in}(1,k)$

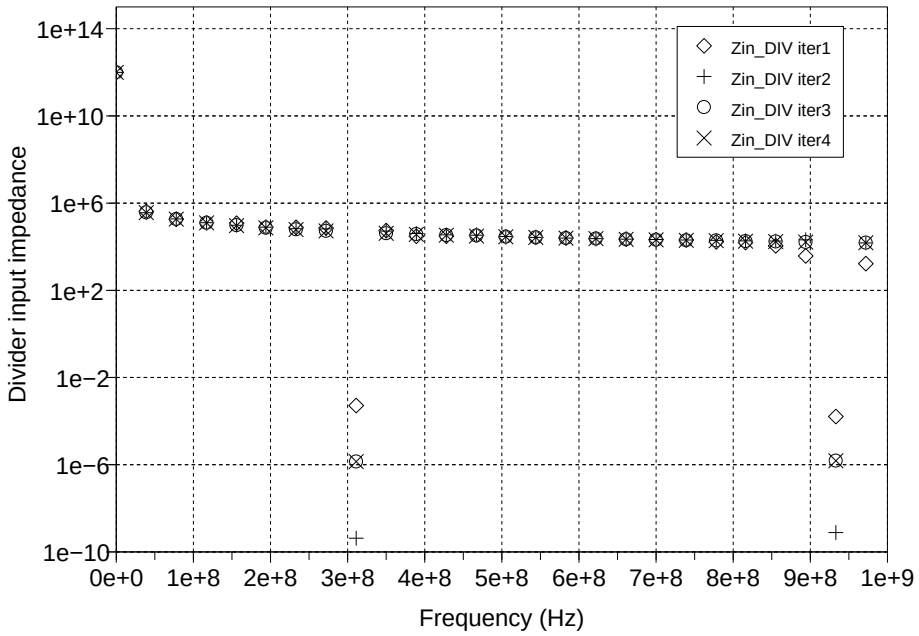


FIG. 2.33: *Evolution of the input impedance of the divider:* $Z_{in}(2,k)$

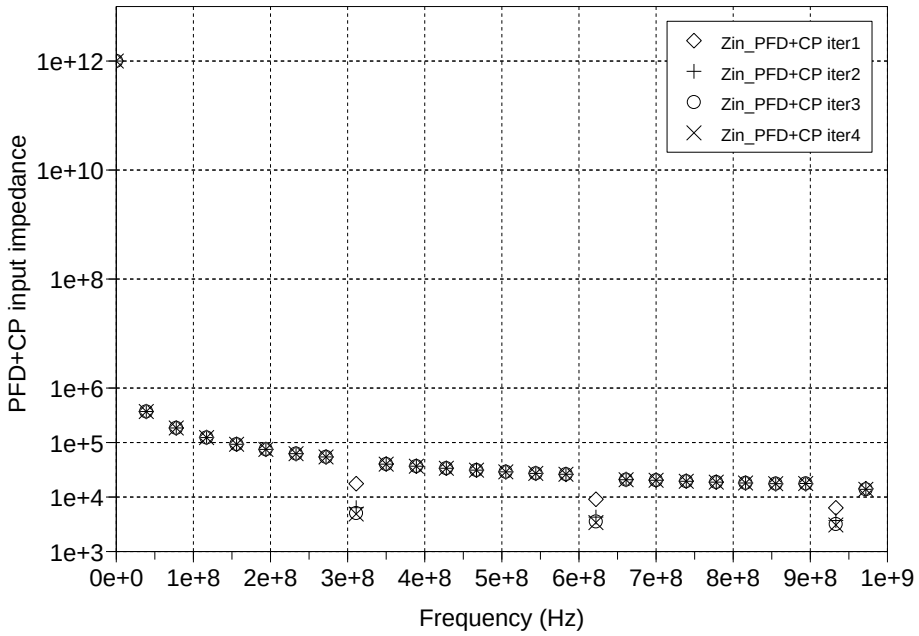


FIG. 2.34: *Evolution of the input impedance of the PFD/CP:* $Z_{in}(3,k)$

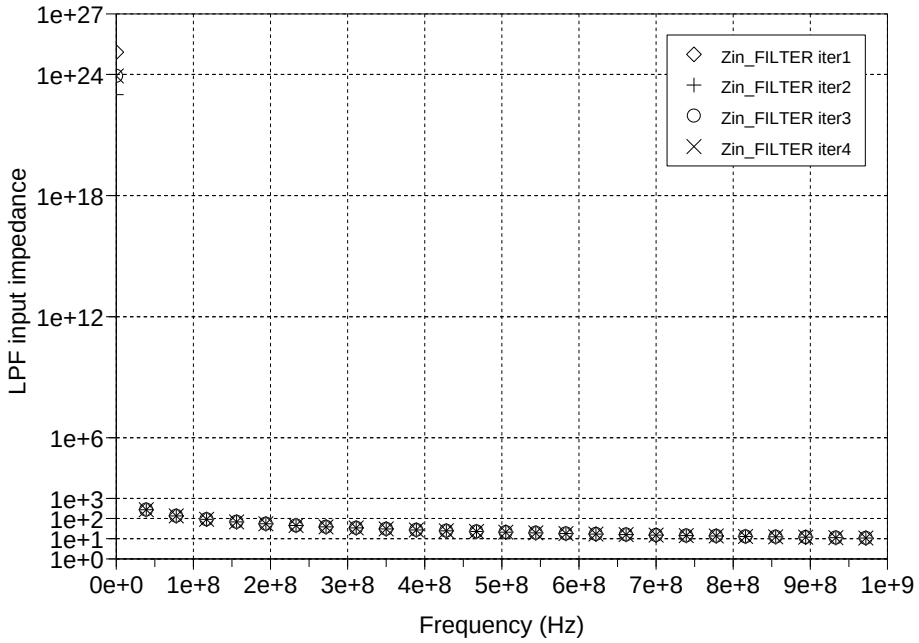


FIG. 2.35: *Evolution of the input impedance of the filter:* $Z_{in}(4,k)$

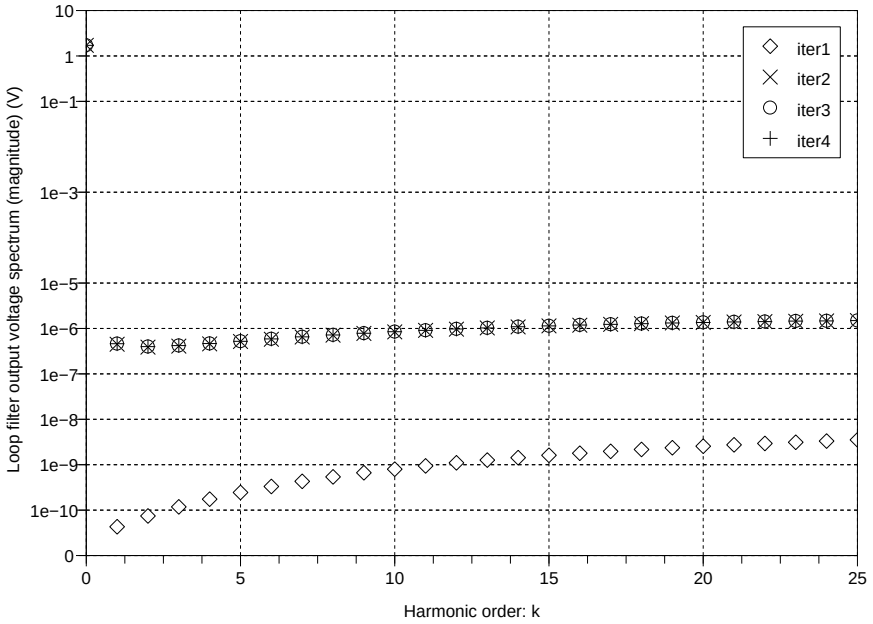


FIG. 2.36: *Evolution of the loop filter output voltage spectrum:* $V_{out}(4,k)$

shift $\Delta\phi/N = 1.33°$. Fig. 2.39 shows the reference signal and the feedback signal obtained with the brute force transistor-level approach. This plot is indistinguishable with the plot Fig. 2.37 from the proposed method. A zoom near the rising signal front (Fig. 2.40) shows a time shift of $\Delta T_L = 103ps$ (or $\Delta\phi/N = 1.44°$). In fact the new method tends to be however the most accurate, because each HB simulation involves a single block at a time, thus a smaller number of devices and less numerical error.

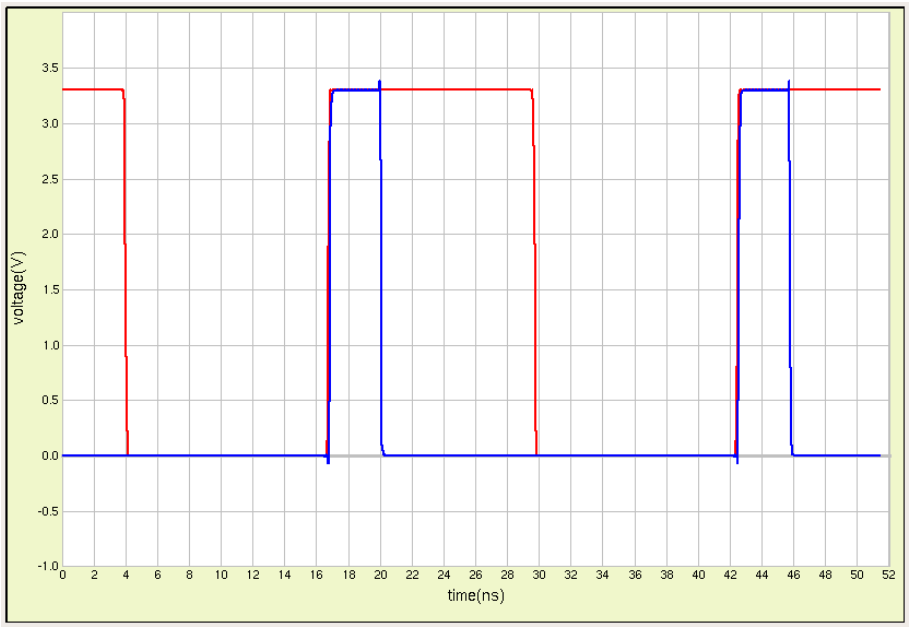


FIG. 2.37: *Nominal phase difference: new method*

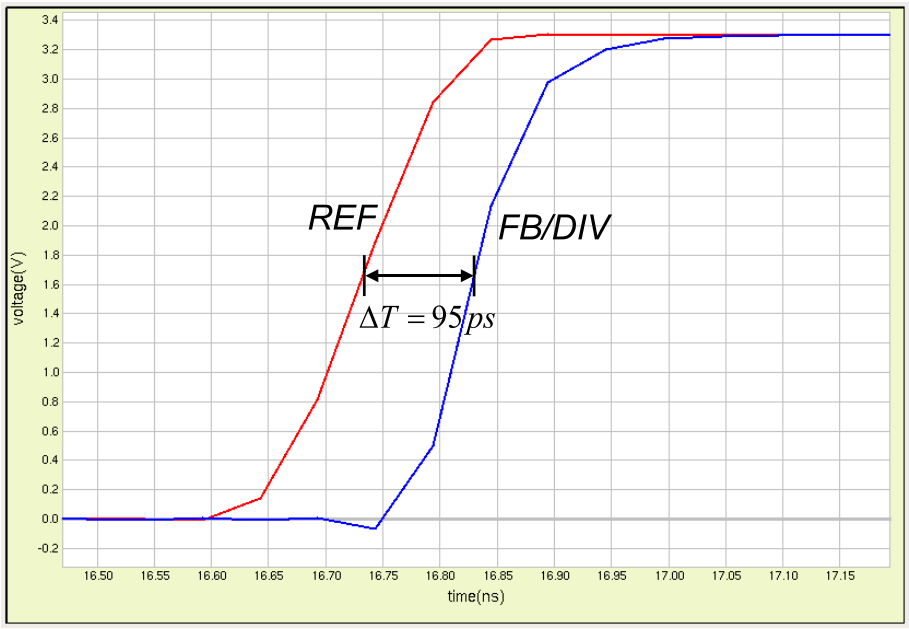


FIG. 2.38: *Zoom of Fig. 2.37*

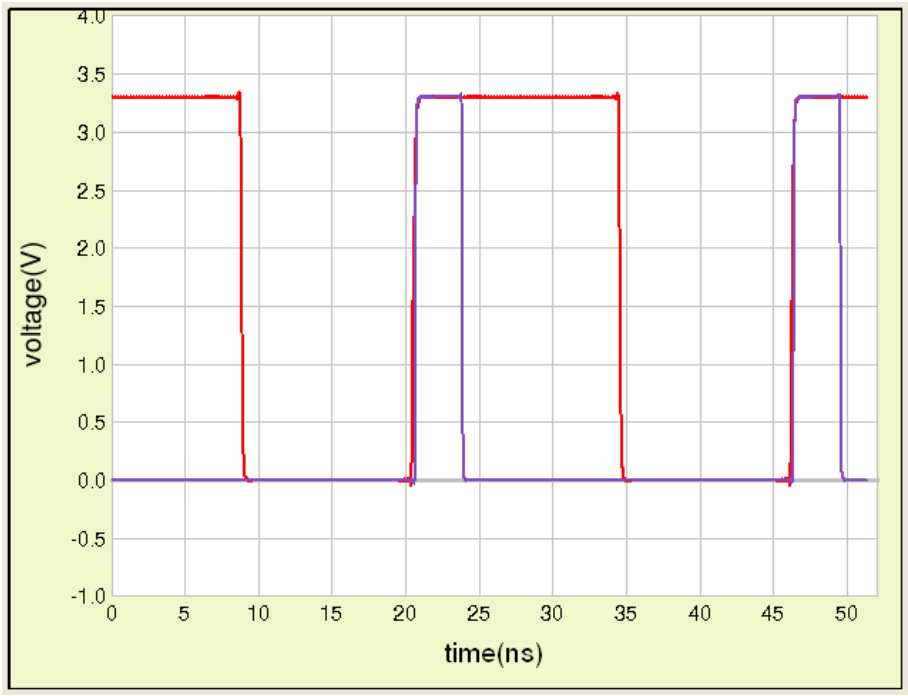


FIG. 2.39: *Nominal phase difference: brute force simulation*

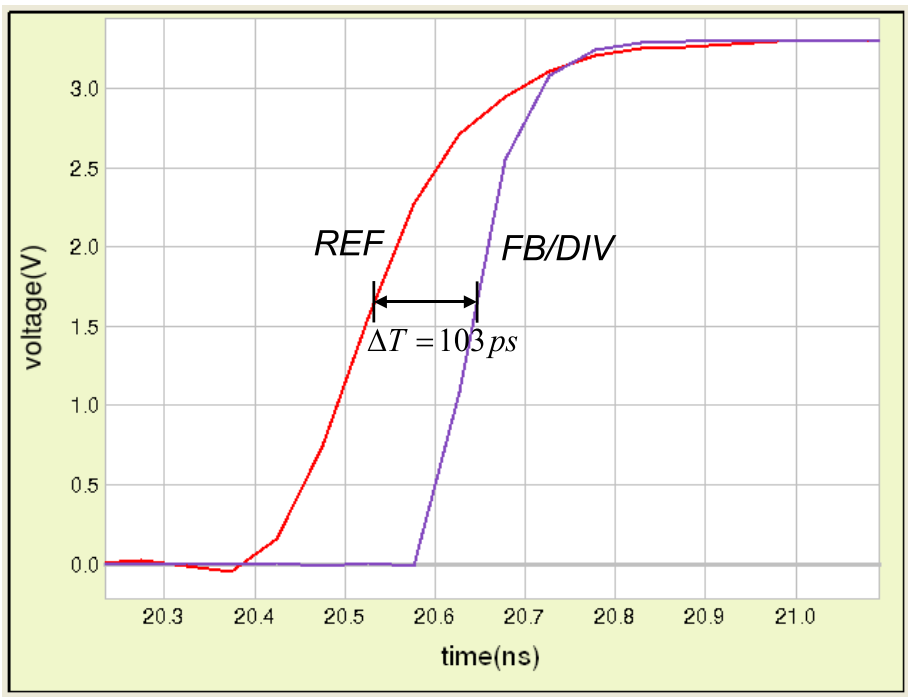


FIG. 2.40: *Zoom of Fig. 2.39*

Note that we have not so far completely automated the presented algorithm in the form of a thorough program, so most of the steps described have been carried manually. Though, we have carefully recorded the net simulation time necessitated in each step, and these are listed in the table below. It is obvious that these simulation times should be substantially lower on a completely automated process.

Simulation step	Simulation time
Initial iteration	7.5 min
VCO input estimate	0.7 min
VCO simulation	0.98 min
FD simulation	2.15 min
Phase shift estimate (PFD+CP simul.)	3.5 min
LPF simulation	0.2 min
2nd iteration	5.8 min
3rd iteration	5.0 min
4th iteration	4.4 min
Total	22.7 min

TAB. 2.3: *List of simulation time of each step*

Most of simulation time is consumed in the FD simulation and the phase shift estimate. During the phase shift estimate, the phase shift value has been updated 5 times before the resolution of the average inner loop equation, which means simulating PFD+CP for 5 times. The subsequent iterations (2nd, 3rd, and 4th iteration) spend less simulation time than the initial iteration because the Transient simulation is skipped.

▶ Comparison of the simulation time for some PLL circuits examples

We have carried the simulation of a number of PLL and compare the simulation time with the brute force approach, as shown in Tab. 2.4. It shows that the proposed method is much faster, especially as the division rank increases.

Ckt	CMOS count	Division ratio	Simulation time (new method)	Simulation time (brute force approach)
I	463	16	22.7 min	85 min
II	491	32	24.8 min	163 min
III	320	40	20.5 min	430 min

TAB. 2.4: *Comparison of the simulation time*

▶ Discussion on the number of harmonics (NH)

The volume of the memory and the simulation time are proportional to the number of harmonics. Since the number of harmonics is specified by the user, one always desires to use a NH as small as possible. With the new method, it's possible to verify whether the

specified number of harmonic is sufficient or not. Of the four PLL blocks, PFD/CP is the most sensitive to the number of harmonics. If the number of harmonics is not enough, the simulation may will likely diverge or converge with an incorrect solution. Fig. 2.41 shows the relation between the CP output voltage and VCO phase shift $\Delta\phi_L$, and how the number of harmonics can affect the simulation results. Therefore, the number of harmonics should be increased whenever the $\overline{v_{out}(3,t)}$ vs. $\Delta\phi_L$ plot shows oscillatory conditions.

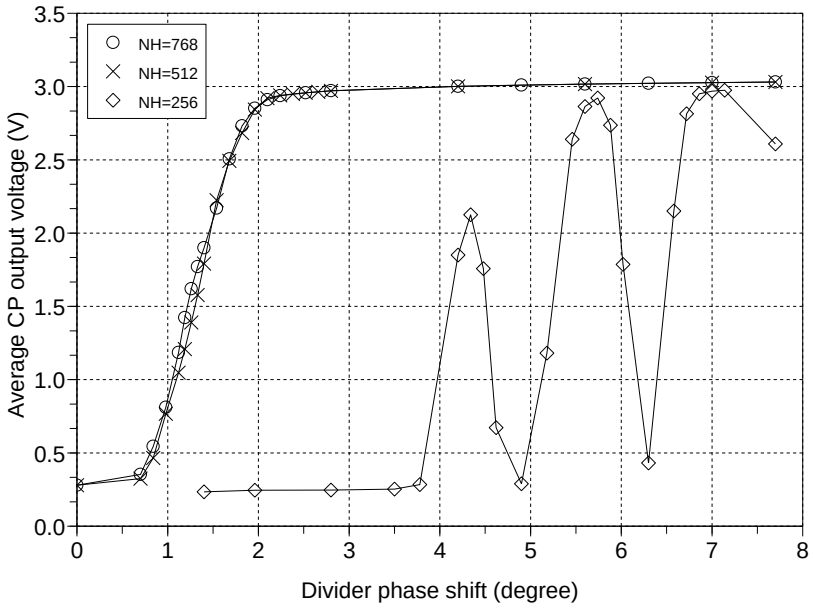


FIG. 2.41: *Influence of the number of harmonics (NH) on the relation* $\overline{v_{out}(3,t)}$ *vs.* $\Delta\phi_L/N$ *: NH is sufficient when* $NH = 768$ *and* $NH = 512$*, and NH is NOT sufficient when* $NH = 256$*.*

2.6 Conclusion

The steady-state analysis of the PLL is very crucial and it's the basis of the other analysis of the PLL (phase noise, deterministic noise, dynamic performance). However, the steady-state analysis of the PLL could be very difficult: on one hand, it's not easy to bring the PLL into the locked state since the transient stage can be very long, especially in the case of a large division ratio; on the other hand, even if the PLL reaches the locked state, the existing analysis methods can encounter critical convergence problems due to the large number of harmonics and the size of the circuit.

In this chapter, we proposed a new simulation method at the transistor-level which breaks the PLL in small pieces. The simulation with the individual block allows reducing

the simulation size and accelerating the analysis. The accuracy of the simulation is insured by a careful consideration of the block interface conditions to fulfill a transistor-level accuracy. A speedup factor of many order in CPU time and memory volume is obtained as compared with the classical brute force approach.

Now that we have developed a realistic solution for the steady state analysis of the PLL, we are in a position to investigate the noise analysis solutions, which will be the subject of the next chapter.

2.7 Bibliography

[1] K. Kundert. (2006) Predicting the phase noise and jitter of PLL-based frequency synthesizers. [Online]. Available: www.designers-guide.org/Analysis/PLLnoise.pdf

[2] ——, "Introduction to RF simulation and its application," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 9, pp. 1298–1319, 1999.

[3] K. S. Kundert, J. K. White, and A. L. Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*, 1st ed. Springer, Mar. 1990.

[4] S. Sancho, A. Suárez, and T. Fernández, "Nonlinear dynamics of microwave synthesizers - stability and noise," *IEEE Transactions on Microwave Theory and Techniques*, vol. 49, pp. 1792–1803, 2001.

[5] A. Soury, "Modélisation des phénomènes de mémoire à long terme des amplificateurs de puissance pour la simulation des systèmes de télécommunications," Ph.D. dissertation, Université de Limoges, 2002.

[6] A. Bennadji, "Implémentation de modèles comportementaux d'amplificateurs de puissance dans des environnements de simulation système et co-simulation circuit système," Ph.D. dissertation, Université de Limoges, 2006.

[7] K. Singhal and J. Vlach, *Computer Methods for Circuit Analysis and Design*, 2nd ed. Springer, Aug. 1993.

[8] M. Nakhla and J. Vlach, "A piecewise harmonic balance technique for determination of periodic response of nonlinear systems," *IEEE Transactions on Circuits and Systems*, vol. 23, pp. 85–91, 1976.

[9] E. Ngoya, "Contribution à la création d'outils de CAO des circuits non linéaires microondes," Ph.D. dissertation, Université de Limoges, 1988.

[10] *SpectreRF simulation option user guide*, Cadence design system, July 2005.

[11] *Advanced Design System 2008 documentation*, Agilent Technologies, 2008. [Online]. Available: http://edocs.soco.agilent.com/display/ads2008/Sources

[12] *GoldenGate Device/Model Reference Manual*, Agilent Technologies, 2008.

[13] G. Estep, E. Ngoya, and D. Webber, "Solving the periodic steady-state operating condition of a phase-locked loop," US Patent 2007/0 255 547 A1, 2006.

[14] *Simulating complex RF/Mixed-signal designs using Virtuoso ultrasim full-chip simulator*, Cadence design system, 2005, cadence white paper. [Online]. Available: https://www.cadence.com/rl/Resources/white_papers/SimRFMS_wp.pdf

[15] A. Demir, "Analysis and simulation of noise in nonlinear electronic circuits and systems," Ph.D. dissertation, University of California, 1997.

[16] M. Hinz, I. Konenkamp, and E.-H. Horneber, "Behavioral modeling and simulation of phase-locked loops for RF front ends," *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems*, vol. 1, pp. 194–197 vol.1, 2000.

[17] J. Zou, D. Mueller, H. Graeb, U. Schlichtmann, E. Hennig, and R. Sommer, "Fast automatic sizing of a charge pump phase-locked loop based on behavioral models," *Proceedings of the IEEE International Behavioral Modeling and Simulation Workshop*, pp. 100–105, 2005.

[18] F. M. Gardner, *Phaselock Techniques*, 3rd ed. Wiley-Interscience, Jul. 2005.

Chapter 3 :

Noise Analysis of the PLL

3.1 Introduction

The PLL is supposed to provide a stable and precise frequency/phase. A perturbation in phase can dramatically degrade the performance of the PLL. Generally, the noise of the PLL comes from either the random source or the deterministic source [1]. The random noise source can cause the phase noise in the PLL, and the deterministic source brings about the deterministic noise.

Known as one of the most critical characteristics of the PLL, the phase noise is probably the most difficult to predict among all the characteristics of the PLL. On one hand, the traditional analysis as SPICE can not calculate the noise in the PLL which is modulated with the large periodic signals, and on the other hand, the other simulation methods at the transistor-level also have difficulties to calculate the noise of the PLL due to the complexity of the circuit [2]. Therefore the behavioral modeling at the block level is usually necessary. To analyze the phase noise in an efficient way, the noise characteristics of each building block of the PLL should be studied, and a noise analysis method adapted to the specificities of the PLL is necessary.

The structure of this chapter is as follows:

First, the characteristics of the noise in electronic circuits are summarized. Then the noise analysis of the PLL at the transistor-level is briefly presented, and the most used methods are reviewed. Then we address the noise modeling at the block level, which is often necessary due to the difficulties encountered in the transistor-level noise analysis. We present the broad lines of the block-level modeling principles, and for each building block, we propose a simple modeling approach that takes benefit of the steady-state method of the previous chapter and accounts for the non ideal loading conditions of various blocks. Both random noise and deterministic noise are taken into consideration. Finally, the simulation results are presented and compared with the conventional methods.

3.2 Noise in the electronic circuits

This section aims to introduce the basic aspects of the noise in electronic circuits. First the basics of the stochastic process are reviewed, and the random noise representation method is introduced. Then the cyclo-stationary noise and AM / PM noise which are critical to the understanding of the phase noise mechanism is briefly introduced.

3.2.1 Random noise and deterministic noise

Two types of noise sources are in consideration: the random noise source produced inside components, and the deterministic noise caused by the perturbation outside the device. The first type of noise relates to the random agitation of electrical charges, leading to the fluctuations in voltage or current. It may include thermal noise, shot noise and flicker noise (1/f noise). The second type of noise, however, concerns the deterministic signals generated by operations in different parts of the circuit, modulated or coupled with each other. It may come from the power supply, substrate or the leakage current, etc.

In the context of the PLL, the random perturbation results in phase noise spectrum distributed over a frequency range, while the deterministic perturbation produces the noise at discrete frequencies points, basically the reference frequency and its harmonics, as shown in Fig. 3.1.

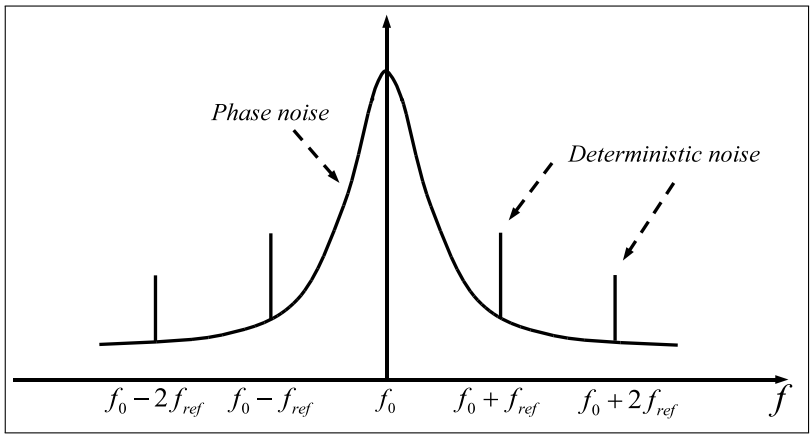


FIG. 3.1: *Phase noise and the deterministic noise of the PLL*

3.2.2 Basic definitions of the random noise

In this section, some basic definitions of the random noise are introduced. The random noise can be regarded as a stochastic process, which implies that several identical trials can lead to slightly different results. Generally the stochastic process is described with the statistical properties of all trials. In this section, first we review briefly the theory of stochastic process, which is helpful to understand the mechanism of random noise, described later in this section.

A *stochastic process* is defined by a set of time functions [1] [3]. Each of them corresponds to an individual trial, and constitutes a sample of stochastic process. The statistical description of this time function is called the *probability density*. The random noise can be characterized by the average of all the trials, or *expectation*, which is the

limit of the overall average value when the number of trials increases and tends towards the infinity.

In practice, the stochastic process is often described by the first moment (mean) and the second moment (autocorrelation).

The *mean* of a process is represented by its expectation:

$$m_X = E\left[X\left(t\right)\right] \tag{3.1}$$

where m_X: mean, $E\left[\cdot\right]$: expectation operator, $X\left(t\right)$: stochastic process.

The *autocorrelation* is a measure of similarity of the process at different time points, which is defined as follows:

$$R_X\left(t_1, t_2\right) = E\left[X\left(t_1\right) X\left(t_2\right)\right] \tag{3.2}$$

To describe the noise in the non-linear circuits, the stochastic process presented below is often used:

- *Stationary process*: the probability density is independent of the observation time or it is invariable with the time shift.
- *Wide-sense stationary process (WSS)*: only the mean and the autocorrelation are invariable with the time shift, as defined below:

$$m_X\left(t_1\right) = m_X\left(t_2\right) \tag{3.3}$$

$$R_X\left(t_1, t_2\right) = R_X\left(t_1 + h, t_2 + h\right) \tag{3.4}$$

- *Wide-sense cyclo-stationary process*: the mean and the autocorrelation are periodic with the period T:

$$m_X\left(t + T\right) = m_X\left(t\right) \tag{3.5}$$

$$R_X\left(t_1 + T, t_2 + T\right) = R_X\left(t_1, t_2\right) \tag{3.6}$$

For a wide-sense stationary process, the autocorrelation depends only on the time shift $\tau = t_1 - t_2$. Hence, the equation (3.2) becomes the expression with one variable τ:

$$R_x\left(\tau\right) = E\left[X\left(t\right) X\left(t - \tau\right)\right] \tag{3.7}$$

3.2.3 Power spectral density of the noise

After introducing the notion of stochastic process, we address now the random noise $n(t)$ [1] [3], whose power can be measured by its variance:

$$var(n(t)) = E\left[n(t)^2\right] \tag{3.8}$$

Actually, the autocorrelation of $n(t)$ is more often used to represent the power of noise:

$$R_n(t,\tau) = E[n(t)\, n(t-\tau)] \tag{3.9}$$

The formula (3.9) describes the noise in the time domain. In practice, the noise distribution in the frequency domain is more often used, called PSD (Power Spectral Density). When $n(t)$ is a stationary process, then $R_n(t,\tau)$ is only a function of τ. And with the help of Fourier Transform, the equation (3.9) can be expressed in the frequency domain:

$$S_n(f) = \int_{-\infty}^{+\infty} R_n(\tau)\, e^{-j2\pi f\tau} d\tau \tag{3.10}$$

If the input X is a WSS process, and the system is stable, linear and time invariable (LTI), the output Y is also a WSS process, whose PSD can be written as:

$$S_Y(f) = |H(f)|^2 \cdot S_X(f) \tag{3.11}$$

where $H(f)$ is the transfer function of the LTI system, and $S_X(f)$ is the input PSD. The equation (3.11) is essential to analyze the influences of a WSS noise on the LTI system.

3.2.4 White noise and colored noise

The white noise is a term commonly used by engineers. If the power spectral density is constant for all frequencies, it is considered *white*.

The white noise is a WSS process with the mean equal to zero and PSD constant:

$$\begin{cases} E[n(t)] = 0 \\ S_n(f) = \text{constant} \end{cases} \tag{3.12}$$

In the time domain, the ideal white noise is a pulse $\tau = 0$ and equal to zero for $\tau \neq 0$. Actually, the ideal white noise does not exist since it has an infinite power. Nevertheless, the white noise is a practical way to represent a noise whose spectral density is constant over a sufficiently large frequency range. Thermal noise and shot noise can

be approximated by Gaussian white noise because their frequency limit is up to 10^{12} Hz, which is well above the operating frequency of the common electric components.

However, if the noise spectral density is shaped by the filter, i.e., the PSD is a function of the frequency, it is said that noise is colored [4], as illustrated in Fig. 3.2.

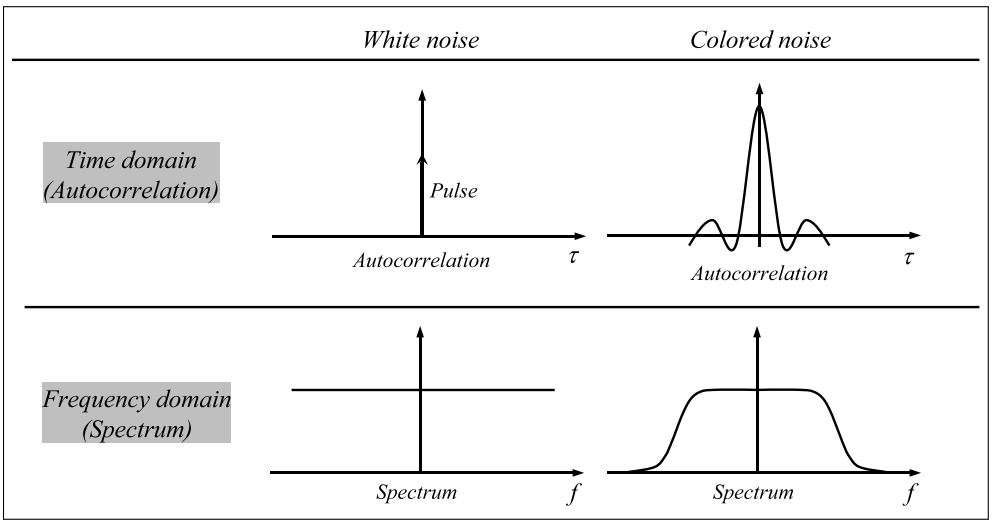


FIG. 3.2: *Comparison between the white noise and the colored noise*

3.2.5 Cyclo-stationary noise

The cyclo-stationary noise exists in the RF circuit, and plays a very important role in understanding the phase noise in the PLL circuit.

The noise is cyclo-stationary if its autocorrelation $R_n(t, \tau)$ is periodic. If a noise source is modulated by a periodic signal, it can generate the cyclo-stationary noise [4]. For example, shot noise mechanism in a transistor excited by a periodic signal produces a cyclo-stationary noise. The cyclo-stationary noise occurs therefore when a periodic large signal is applied to a non-linear circuit.

As stated above, the cyclo-stationary noise is the result of the modulation of the periodic signal with the noise source. If $n(t)$ is the noise and $m(t)$ the periodic modulation signal, then the modulated noise $y(t)$ could be expressed as:

$$y(t) = m(t)\, n(t) \tag{3.13}$$

The multiplication of the two terms corresponds to a convolution in the frequency

domain, so the spectrum of $y(t)$ is:

$$Y(f) = \sum_{k=-K}^{+K} M(kf_0)\, N(f - kf_0) \tag{3.14}$$

where k is an integer, f_0 is the fundamental frequency of the periodic signal, $M(f)$ and $N(f)$ are the Fourier transform of the $m(t)$ and $n(t)$.

Looking at the spectrum of $Y(f)$ (Fig. 3.3), one can see that low-frequency noise $N(f)$ is transformed to $N(f - kf_0)$, $k = [-K, K]$, which locates around each harmonic of the fundamental frequency. The noise around harmonics will correlate each other because they come from the same source but shifted by kf_0. These noises create the replicated sidebands around each harmonic, as shown in Fig. 3.3. Therefore, the noise at a certain frequency contains the contribution of all noise sources at frequency $f \pm kf_0$.

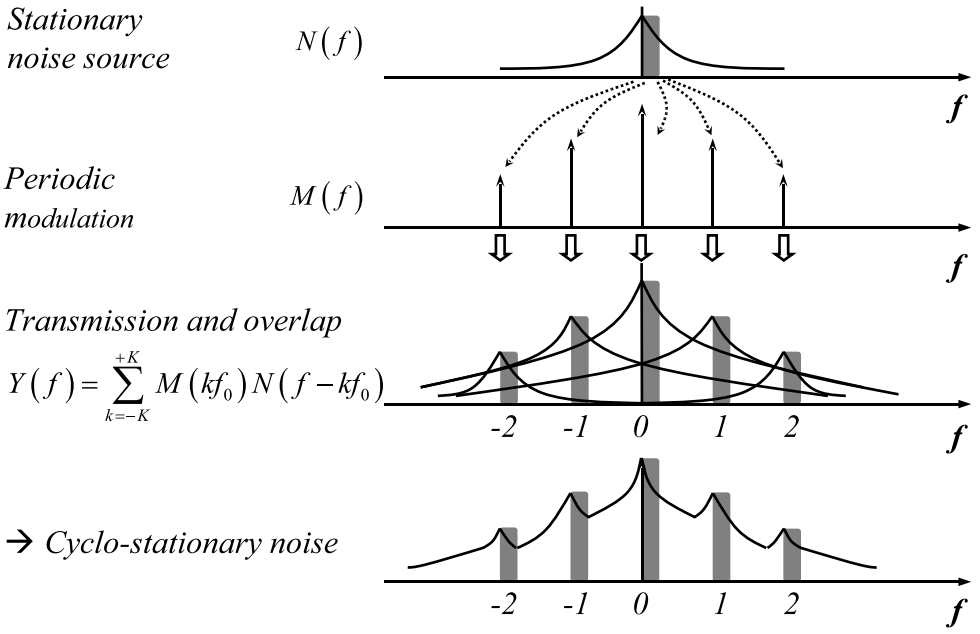


FIG. 3.3: *Correlation in cyclo-stationary noise*

3.2.6 AM noise and PM noise

The real signal is often described with the single sideband, so it needs to put the conjugation of the noise at negative frequencies to the positive frequencies. Thus the upper sideband and the lower sideband correlate, as shown in Fig. 3.4 and Fig. 3.5.

If the sidebands and the carrier are treated as phasor, the noise can be separated into AM noise and PM noise. Note that the decomposition of AM and PM noise is associated to the carrier, therefore AM and PM noise are also named as carrier referred AM noise

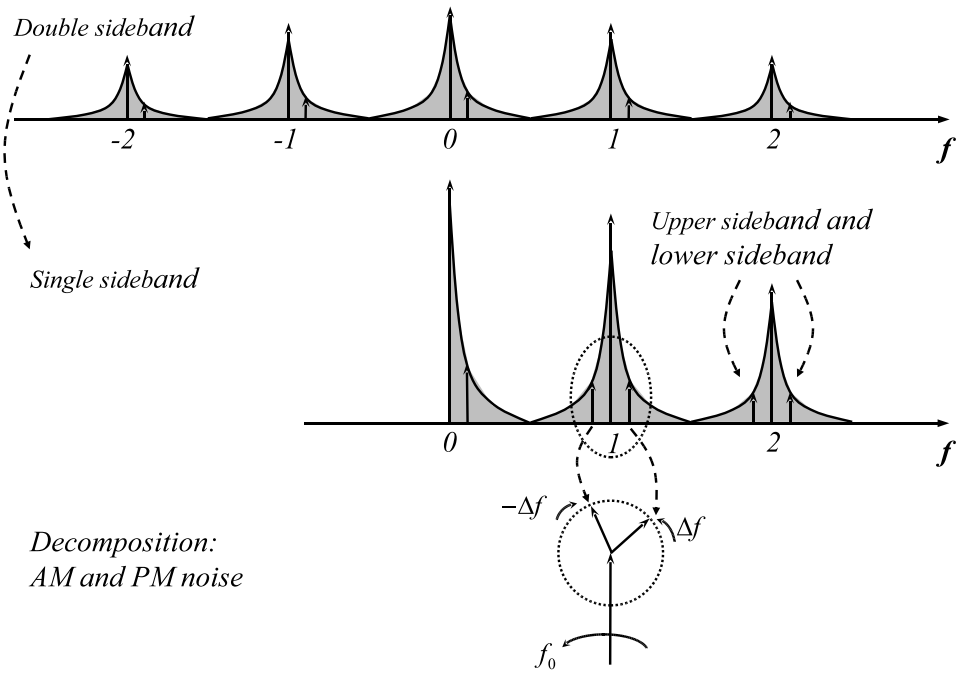


FIG. 3.4: *Double sideband and single sideband*

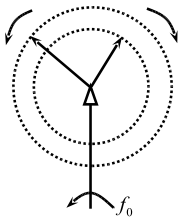
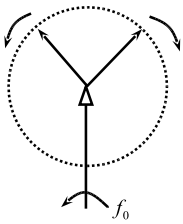
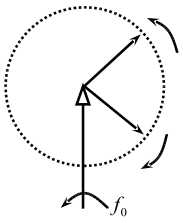
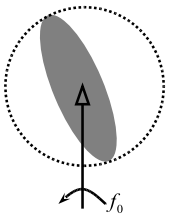
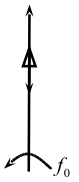
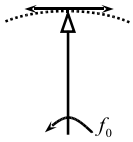
	Non-correlated sideband	AM correlated sideband	PM correlated sideband
Upper sideband and lower sideband (separated)			
Upper sideband and lower sideband (grouped)			

FIG. 3.5: *Decomposition of AM noise and PM noise*

and carrier referred PM noise. Thus, the phasor of sideband noise may be added on the phasor of carrier. The carrier phasor rotates at high frequency f_0, while the sideband noise phasor rotates at the frequency Δf above and below the phasor carrier frequency [1] [4]. The negative frequency $-\Delta f$ means that the phasor rotates in the opposite direction to the frequency Δf.

For an oscillator circuit, PM noise is much more important than AM noise, as we will discuss in the following section.

In this section, the main characteristics of noise in the circuit were briefly introduced. Next, we will describe how to analyze noise in the context of the PLL circuit.

3.3 Noise analysis of the PLL at the transistor level

Once the steady-state response of the PLL is determined, the noise analysis can be treated as the perturbation of the steady-state solution through the linearization of the equilibrium point [2]. The principle of the noise analysis is:

1. Turn off all noise sources and calculate the large-signal steady-state solution.
2. Linearize the circuit around steady-state solution, apply the perturbation, and calculate the output noise.

In this section, we review briefly the main existing noise analysis methods which are performed at the transistor-level: conversion matrix, envelope transient and Monte Carlo.

3.3.1 Conversion matrix

The conversion matrix method is implemented in most commercial simulators.

As mentioned above, the sideband noises around different harmonics are correlated. Therefore, the noise at a certain frequency depends on the signals and noise of all the harmonics. The conversion matrix method can account for these correlations between different harmonics. [5] [6]

To summarize this method, consider a two-port device, as shown in Fig. 3.6.

Assuming the steady-state of the two-port device has been determined, it can be described as follows:

$$\begin{aligned} &i_2(t) = f(v_1(t), v_2(t)) \quad \overset{FFT}{\Leftrightarrow} I_2(f_k), \ k = 0, 1, 2..., K \\ &I_2(f_k) = F(V_1(f_0), ...V_1(f_k) ..., V_1(f_K), V_2(f_0), ...V_2(f_k) ..., V_2(f_K)) \end{aligned} \tag{3.15}$$

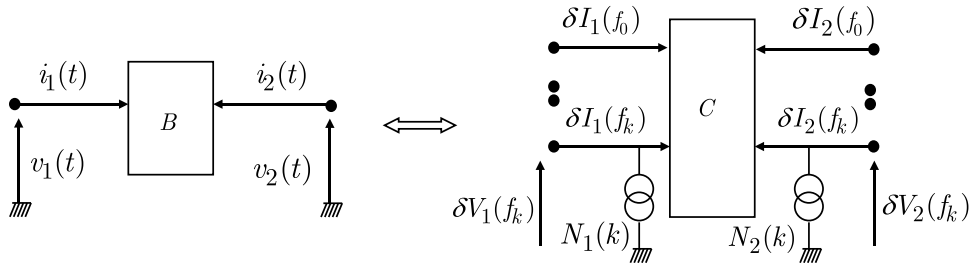


FIG. 3.6: *Model for the noise analysis*

Note that $I_2(f_k)$ at a frequency f_k depends on $V_1(f_p)$ and $V_2(f_p)$ for all harmonics ($p = 0, 1, 2..., K$).

Then the noise will produce a small displacement of the steady-state solution as:

$$I_2(f_k) + \delta I_2(f_k) = F\left(\begin{array}{l} V_1(f_0) + \delta V_1(f_0), ...V_1(f_k) + \delta V_1(f_k) ..., V_1(f_K) + \delta V_1(f_K), \\ V_2(f_0) + \delta V_2(f_0), ...V_2(f_k) + \delta V_2(f_k) ..., V_2(f_K) + \delta V_2(f_K) \end{array}\right) \quad k = 0, 1, 2..., K \tag{3.16}$$

Developing the Taylor series to the first order yields the following:

$$\delta I_2(f_k) = \sum_{p=0}^{K} \frac{\partial I_2(f_k)}{\partial V_1(f_p)} \delta V_1(f_p) + \sum_{p=0}^{K} \frac{\partial I_2(f_k)}{\partial V_2(f_p)} \delta V_2(f_p) \ , k = 0, 1, 2..., K \tag{3.17}$$

Consequently, the current variation $\delta I_2(f_k)$ for all the harmonics is represented in a matrix form as:

$$\left[\begin{array}{c} \delta I_2(f_0) \\ \delta I_2(f_1) \\ ... \\ \delta I_2(f_K) \end{array}\right] = \left[\begin{array}{cccc} \frac{\partial I_2(f_0)}{\partial V_1(f_0)} & \frac{\partial I_2(f_0)}{\partial V_2(f_0)} & ... & \frac{\partial I_2(f_0)}{\partial V_1(f_K)} \quad \frac{\partial I_2(f_0)}{\partial V_2(f_K)} \\ \frac{\partial I_2(f_1)}{\partial V_1(f_0)} & \frac{\partial I_2(f_1)}{\partial V_2(f_0)} & ... & \frac{\partial I_2(f_1)}{\partial V_1(f_K)} \quad \frac{\partial I_2(f_1)}{\partial V_2(f_K)} \\ ..... & & & \\ \frac{\partial I_2(f_K)}{\partial V_1(f_0)} & \frac{\partial I_2(f_K)}{\partial V_2(f_0)} & ... & \frac{\partial I_2(f_K)}{\partial V_1(f_K)} \quad \frac{\partial I_2(f_K)}{\partial V_2(f_K)} \end{array}\right] \left[\begin{array}{c} \delta V_1(f_0) \\ \delta V_2(f_0) \\ ... \\ \delta V_1(f_K) \\ \delta V_2(f_K) \end{array}\right]$$
$$k = 0, 1, 2..., K \tag{3.18}$$

where $C = \left[\frac{\partial I_i(f_k)}{\partial V_j(f_p)}\right]$, $i, j = 1, 2$, $k, p = 0, 1, ..., K$ is the conversion matrix. The matrix C represents the current sensitivity on the voltage variation. By interconnecting the conversion matrix of all the devices, the noise characteristics of the circuit can be obtained by a linear analysis.

3.3.2 Noise analysis based on envelope transient technique

Envelope transient method [7] [8] are very suitable to analyze modulated signal, which is a combination of a high-frequency carrier and a low-frequency envelope. With envelope transient, the carrier and the envelope are treated separately, which allows using fewer sampling points and accelerating the simulation. The interesting thing about the noise analysis using envelope transient technique lies on the fact that the noise source can be regarded as modulation signal thus its amplitude/phase fluctuation can be easily computed by the envelope transient analysis [9]. In this section, envelope transient technique is briefly presented first, then noise analysis with envelope transient is introduced.

3.3.2.1 Introduction of envelope transient technique

Envelope transient method combines a transient analysis and a modified harmonic balance analysis, so it is also referred to as mixed time-frequency analysis.

Envelope transient method can be considered as an enhanced version of harmonic balance method. Generally, harmonic balance method allows simulating a periodic signal with one tone or a quasi-periodic signal with two/several tones, in which cases the Fourier coefficients are constant or periodic. However, when a carrier signal is modulated with an arbitrary signal, the Fourier coefficients are arbitrarily time-varying. As a result, harmonic balance method is no longer applicable, and envelope transient method should be used.

In fact, any signal in the circuit can be expressed in the form of a time varying complex envelope:

$$\begin{cases} z\left(t\right) = \sum\limits_{k} \hat{Z}_k\left(t\right) \exp\left(jk\omega_0 t\right) \\ \hat{Z}_k\left(t\right) = \dfrac{1}{2\pi} \displaystyle\int_{-BW/2}^{BW/2} \hat{Z}_k\left(\Omega\right) \exp\left(j\Omega t\right) d\Omega \end{cases} \tag{3.19}$$

where ω_0 is carrier frequency, BW is the largest bandwidth of modulation envelope.

Assuming the bandwidth of the modulation envelope is much lower than the carrier frequency, the envelope varies slowly compared with the high-frequency carrier. Therefore, a low sampling frequency is allowable. At each sampling point (or time step), a harmonic balance analysis is performed. Then the low-frequency envelope waveform is treated by time domain integration.

Envelope transient is a general purpose method than can apply to forced or autonomous circuit with or without the presence of the noise.

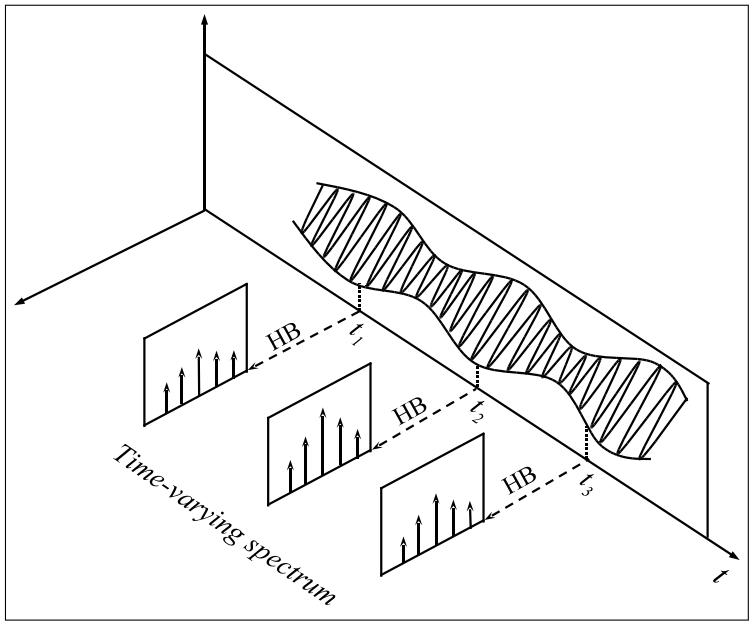


FIG. 3.7: *Principle of envelope transient*

3.3.2.2 Noise analysis using envelope transient technique

As known, the noise sources can be described by a summation of distinct sinusoids in the following form: [9] [10] [11]

$$\begin{cases} n_s(t) = \sum_{k=-H}^{H} \hat{N}_{sk}(t) \exp(jk\Omega_0 t) \\ \hat{N}_{sk}(t) = N_k^+ \exp(j\Omega_s t) + N_k^- \exp(-j\Omega_s t) \end{cases} \tag{3.20}$$

where Ω_s is noise sideband frequency, Ω_0 is the fundamental frequency. Note that all the harmonic frequencies are taken into account simultaneously, so the exchange of energy among the harmonic frequencies are included in the analysis.

To carry out noise analysis, the circuit is perturbed around the noiseless steady-state solution. Then a perturbation equation is obtained [9]. Since envelope transient treats an arbitrary signal in the form of time varying phasor (amplitude, phase and frequency), the noise perturbation is considered as one of the modulation signals.

Solving the perturbation equation, one gets the carrier referred amplitude perturbation and the total phase perturbation in the time domain. Then the frequency spectrum of above perturbations are obtained through Fourier transform, and the corresponding AM noise and PM noise can be easily computed. With envelope transient method, one has a direct access to the amplitude and phase modulation, so the AM and PM noise can be explicitly separated. In addition, the absolute voltage noise spectrum can also be calculated by solving the above perturbation equation.

An extension of the envelope transient method to analyze the PLL using three time scales (VCO frequency, reference frequency, and loop bandwidth) has also been proposed in the reference [12], though it has been applied at behavioral level. The variables such as phase difference, reference signal, VCO input voltage and CP output signal are expressed in the form of time-varying phasor. The CP output signal depends on divider output phase and reference signal. The derivative of phase difference (or frequency difference) is described by a non-linear function of VCO input voltage. The noise perturbations are described as a summation of pseudo-sinusoids, and a noise-perturbation equation in the frequency domain is obtained to calculate the noise spectral density at the output of the PLL.

3.3.3 Monte Carlo method

Monte Carlo method is the only method that does not treat the large-signal and the noise separately [3]. The non-linear circuit equations contain the excitation signal and noise sources, as shown in the equation below:

$$I\left(X,t\right)+\frac{d}{dt}Q\left(X\right)+B\left(X\right)U\left(t\right)=0 \tag{3.21}$$

where X is a stochastic process vector, which represents the nodal voltage, and U is a vector representing the noise source in the circuit.

The method is based on a numerical integration of the non-linear equations to create some sample path for the vector X. A random number generator is necessary to realize the noise sources. Then the correlation and the spectral density can be calculated.

The advantage of this method is that a steady-state analysis is not a prerequisite, and it does not assume the perturbation is sufficiently small compared with the large signals.

The disadvantage of the Monte Carlo method is the prohibitive memory and simulation time, and also it uses a pseudo-random number generator, which may not produce a sufficiently large sequence of independent number. If a circuit has many noise sources, this can cause significant calculation errors.

3.4 Noise modeling and analysis at the block level

3.4.1 Introduction

In the previous section we briefly introduced the noise analysis method at the transistor-level. Indeed, the difficulties of the PLL transistor-level noise analysis are well known: large memory resources, prohibitive simulation time, convergence problems with iterative methods. Therefore, PLL modeling at block level, also termed behavioral modeling, is usually necessary to overcome the above difficulties. The PLL modeling at block level consists in simulating the building blocks of the PLL, creating the behavioral models and then computing the overall PLL noise. Note that the PLL noise accuracy relies on the behavioral model accuracy, so creating accurate block models is very critical for the PLL noise analysis.

3.4.2 Behavioral modeling of the PLL blocks

To create the block models for the noise analysis, it needs first to define some important notions about the behavioral modeling process, as follows:

- Access: it indicates the port property of the model: dipole, two-port...
- Access variable: it defines the physical variable that can be accessed at the port of the model: voltage, current, frequency, phase, etc.
- Large signal operating point: it defines the steady-state operating point.
- Small signal transfer (or sensitivity) matrix of the block: it describes the relationship between small fluctuations at the block ports.
- Internal noise to the block: the noise generated within one block, it could be voltage/current noise or phase noise.

Now consider a N-port block which is described by a set of two access variables $x_i(t)$ and $y_i(t)$, as illustrated in Fig. 3.8. The noise sources internal to the block are denoted $\hat{n}_j(t)$.

If the large signal steady state response of the block is obtained, and a small perturbation is applied to the block, then this N-port block can be described by:

$$\begin{cases} x_i(t) = \hat{x}_i + \delta x_i \\ y_i(t) = \hat{y}_i + \delta y_i \\ i = 1, ..., N \end{cases} \tag{3.22}$$

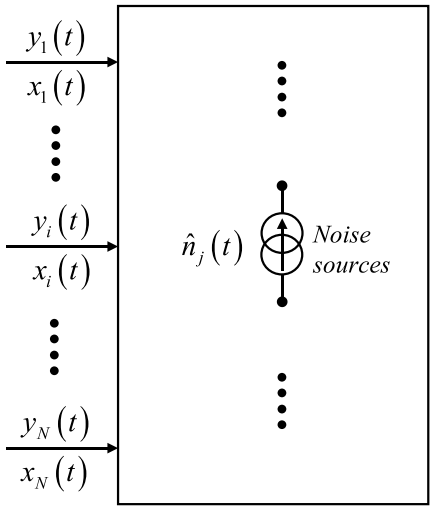


FIG. 3.8: *Model of a N-port block*

where $\hat{x}_i$ and $\hat{y}_i$ represent the noiseless large signal operating point, δx_i and δy_i are the small perturbations due to noise.

If we consider a voltage/current formalism (Fig. 3.9), equation (3.22) writes:

$$\begin{cases} v_i(t) = \hat{v}_i(t) + \delta v_i(t) \\ i_i(t) = \hat{i}_i(t) + \delta i_i(t) \\ i = 1, ..., P \end{cases} \tag{3.23}$$

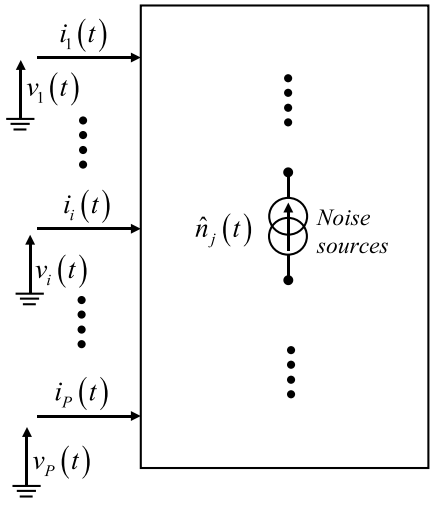


FIG. 3.9: *Model of a N-port block using a voltage/current formalism*

Accounting the internal noise contributions, the port current $i_i(t)$ may be expressed as:

$$i_i(t) = f_i(v_1(t), ..., v_P(t), t) + n_i(t) \quad i = 1, ..., P \tag{3.24}$$

Since from eq. (3.23), $\delta v_i(t)$ is small, $i_i(t)$ can be expressed by a 1st order Taylor series expansion:

$$i_i(t) = f_i(\hat{v}_1(t), ..., \hat{v}_P(t), t) + \sum_{j=1}^{P} \frac{\partial f_i}{\partial v_j} \cdot \delta v_j(t) + n_i(t) \tag{3.25}$$

Noting that $f_i(\hat{v}_1(t), ..., \hat{v}_P(t), t)$ corresponds to the large signal operating point $\hat{i}_i(t)$, we deduce the small signal perturbation mechanism as:

$$\begin{pmatrix} \delta i_1(t) \\ \vdots \\ \delta i_P(t) \end{pmatrix} = \begin{bmatrix} \frac{\partial f_1(t)}{\partial v_1(t)} & \cdots & \frac{\partial f_1(t)}{\partial v_P(t)} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_P(t)}{\partial v_1(t)} & \cdots & \frac{\partial f_P(t)}{\partial v_P(t)} \end{bmatrix} \begin{pmatrix} \delta v_1(t) \\ \vdots \\ \delta v_P(t) \end{pmatrix} + \begin{pmatrix} n_1(t) \\ \vdots \\ n_P(t) \end{pmatrix} \tag{3.26}$$

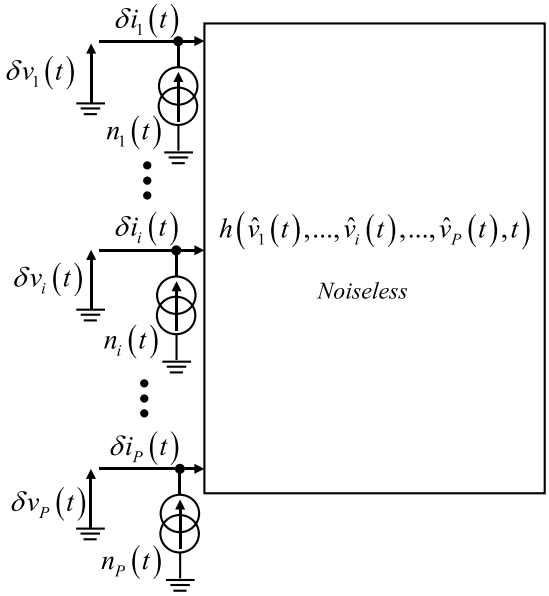


FIG. 3.10: *Norton equivalent circuit with noiseless circuit, noise sources and small signal perturbations at the block access*

Equation (3.26) results in the Norton equivalent circuit depicted in Fig. 3.10, where

$$h(\hat{v}_1, ..., \hat{v}_P, t) = \begin{bmatrix} \frac{\partial f_1(..., t)}{\partial v_1(t)} & \cdots & \frac{\partial f_1(..., t)}{\partial v_P(t)} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_P(..., t)}{\partial v_1(t)} & \cdots & \frac{\partial f_P(..., t)}{\partial v_P(t)} \end{bmatrix}$$

is the time varying impulse response of the circuit block, and $(n_1(t), ..., n_P(t))^T$ is the equivalent Norton noise source. Note that $(n_1(t), ..., n_P(t))^T$ is the noise current flowing at block terminals when these are shorted, i.e., $\delta v_i(t) = 0$.

For each block of the PLL, we must determine the impulse response $(h(\hat{v}_1(t), ..., \hat{v}_P(t), t))$ and the Norton noise sources $(n_1(t), ..., n_P(t))$ seen at the block access. Note that on one hand, a precise steady state analysis is needed to calculate the impulse response $h(\hat{v}_1(t), ..., \hat{v}_P(t), t)$; on the other hand, a precise transistor-level noise analysis is required in order to calculate the Norton noise sources at the block ports.

Indeed, the above process is mathematically accurate, but the impulse response is usually very complex to draw. To make this process practical, it is therefore necessary to consider some practical assumptions for each of the PLL blocks. Next, we will address the noise modeling of the various PLL building blocks.

3.4.2.1 VCO

VCO is a two-port network as defined in Fig. 3.11, where $v_c(t)$, $v_o(t)$, $i_c(t)$ and $i_o(t)$ are respectively the input and output voltages and currents.

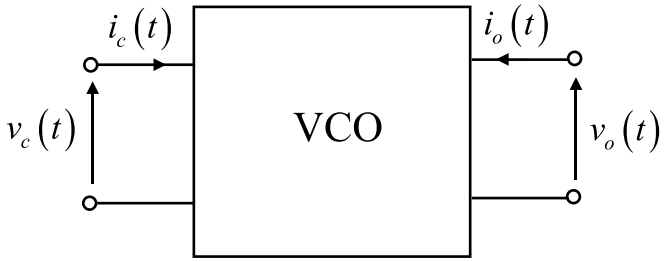


FIG. 3.11: *VCO block: two port network*

As indicated above, we must identify the equivalent impulse response $h(\hat{v}_c(t), \hat{v}_o(t), t)$ and the equivalent Norton noise sources as depicted in Fig. 3.12.

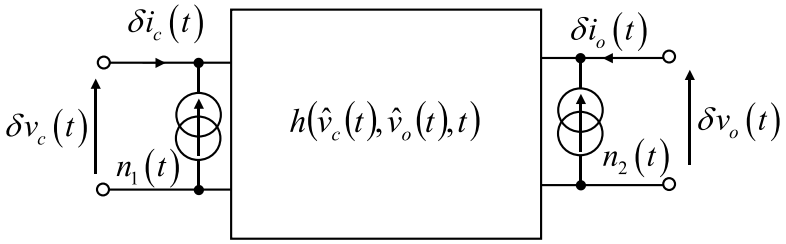


FIG. 3.12: *VCO block: Norton equivalent circuit*

As indicated above, the full extraction of $h(\hat{v}_c(t), \hat{v}_o(t), t)$ and $n_1(t)$, $n_2(t)$ is indeed very complex. Fortunately a full extraction is not necessary to capture the essential characteristics of the VCO noise behavior in the PLL, as will be seen.

Consider the noisy VCO output voltage:

$$v_o(t) = \sum_k \left(\hat{V}_k^o(t) + \delta V_k^o(t) \right) \exp\left(jk \left(\hat{\phi}_o(t) + \delta\phi_o(t) \right) \right)$$

It is known that for an oscillator, the amplitude perturbation $\delta V_k^o(t)$ is negligible, especially for slowly varying perturbations. Therefore, once the steady-state parameters $\hat{V}_k^o(t)$ and $\hat{\phi}_o(t)$ are determined (see chapter 2), the knowledge of the phase perturbation $\delta\phi_o(t)$ is sufficient to describe the output signal. Taking this into consideration, we can change the VCO output access formalism, from voltage/current formalism to the phase domain formalism. Hence we obtain the mixed voltage-phase domain equivalent small signal network depicted in Fig. 3.13.

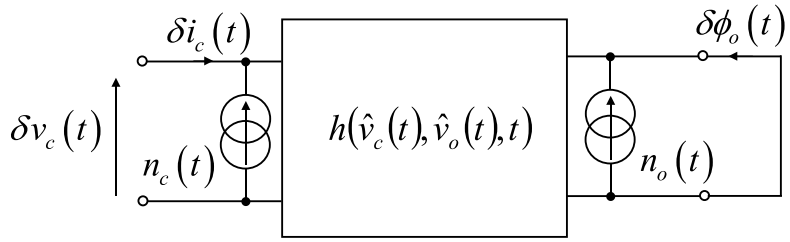


FIG. 3.13: *VCO model: I/V at the input and phase at the output*

This kind of representation is interesting for the following reasons:

- Both $\delta v_c(t)$ and $\delta\phi_o$ are slowly varying signals, so we may efficiently use either time or frequency domain simulation techniques;
- The relation between $\delta v_c(t)$ and $\delta\phi_o$ is practically quasi-static.

A huge amount of work has been devoted to the characterization of the above equivalent network. A brief review of the main ideas and the principal works is given in the Appendix I. In the following we present our proposed approach.

The previous works (Appendix I) on VCO modeling consider ideal block impedance termination. The model we propose is in the continuity of the hierarchical steady-state analysis technique presented in chapter 2, where the device is characterized under the specific loading conditions obtained from the PLL steady-state regime.

Thus returning to Fig. 3.13, we first observe that the output port being characterized in the phase domain, and the equivalent voltage drop is identically zero. We need therefore to determine the input and output Norton sources $n_c(t)$, $n_o(t)$ and the noiseless impulse response below, which is only a single column.

$$h\left(\hat{v}_c\left(t\right),\hat{v}_o\left(t\right),t\right)=\begin{bmatrix}\dfrac{\partial i_c}{\partial v_c} & 0\\ \dfrac{\partial \phi_o}{\partial v_c} & 0\end{bmatrix} \tag{3.27}$$

The first item $\dfrac{\partial i_c}{\partial v_c}$ in this response is actually the time domain corollary of the conversion matrix defined previously. However, as indicated, the VCO being terminated at its input port by a low-pass filter, $\delta v_c\left(t\right)$ is a low frequency signal. Therefore $\dfrac{\partial i_c}{\partial v_c}$ is in fact the low frequency input admittance of the VCO, which is usually a simple parallel RC network, i.e., $Y_{in}\left(\omega\right)=G_{in}+j\omega C_{in}$. This can be obtained readily from Harmonic balance sensitivity upon the steady-state regime.

The second term $\dfrac{\partial \phi_o}{\partial v_c}$ is better expressed in the form:

$$\frac{\partial \phi_o}{\partial v_c}=\int_0^t \frac{\partial \omega_o}{\partial v_c}d\tau \tag{3.28}$$

Since as said, $\delta v_c\left(t\right)$ is a low frequency signal, we will postulate that the oscillator frequency $\omega_o\left(t\right)$ is a static function of the control voltage, especially for small amplitude variations. This assumption is practically true in bandwidth from the zero to 100MHz where the noise analysis is usually desired. This has been verified for many oscillator circuits, with Envelope transient analysis. Coefficient $\dfrac{\partial \omega_o}{\partial v_c}$ is also obtained readily from Harmonic balance sensitivity upon the steady-state regime.

The two Norton noise sources $n_c\left(t\right)$, $n_o\left(t\right)$ are correlated. However, since the VCO output is in the phase domain formalism (i.e., shorted output port), only $n_o\left(t\right)$ will contribute to the output phase $\delta\phi_o\left(t\right)$. We can therefore ignore $n_c\left(t\right)$ and the correlation. The source $n_o\left(t\right)$ or more precisely the power spectral density $S_{n_o}\left(f\right)$ of $n_o\left(t\right)$ can be determined from a transistor-level noise analysis carried on the VCO, when it is terminated on the loading conditions obtained from chapter 2. The equivalent network of the VCO is then as depicted in Fig. 3.14. This equivalent network can be then easily implemented in Verilog-A.

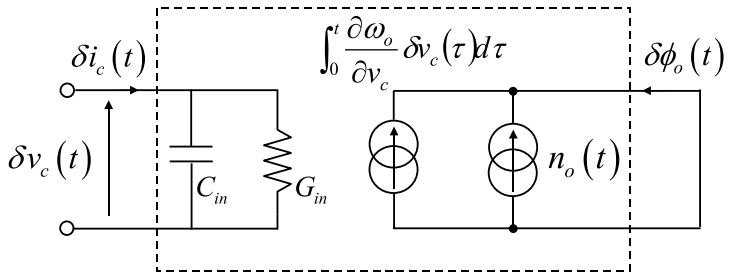


FIG. 3.14: *VCO equivalent network*

3.4.2.2 PFD+CP

PFD+CP is a three-port block, as illustrated in Fig. 3.15: $v_{ref}(t)$ and $i_{ref}(t)$ are the reference signals, $v_{fb}(t)$ and $i_{fb}(t)$ the feedback signals, $v_{cp}(t)$ and $i_{cp}(t)$ the CP output signals of PFD+CP.

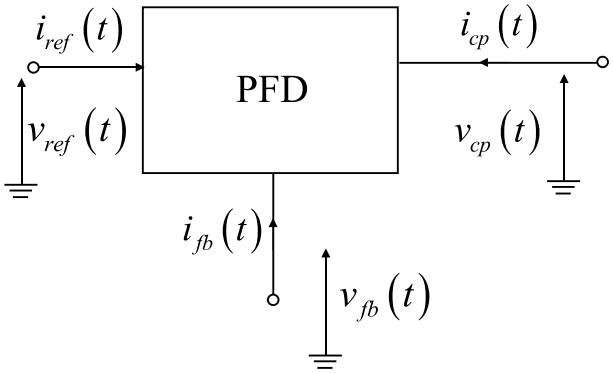


FIG. 3.15: *PFD+CP block*

Similarly to the VCO, we need to determine the voltage-domain small-signal equivalent network depicted in Fig. 3.16.

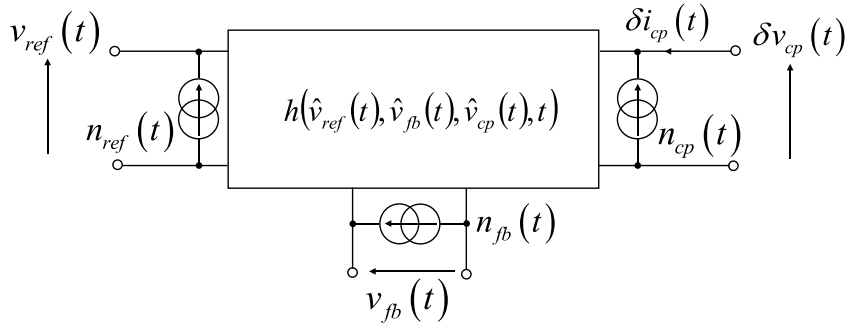


FIG. 3.16: *PFD+CP block: voltage domain small-signal equivalent network*

Similarly to the VCO, the full determination of the impulse response $h(\hat{v}_{ref}(t), \hat{v}_{fb}(t), \hat{v}_{cp}(t), t)$ is complex and fortunately not necessary for an accurate noise characterization. Indeed, if we consider the signals at the reference and feedback ports:

$$\begin{aligned} v_{ref}(t) &= \sum_k \left(\hat{V}_k^r(t) + \delta V_k^r(t)\right) \exp\left(jk\left(\hat{\phi}_{ref}(t) + \delta\phi_{ref}(t)\right)\right) \\ v_{fb}(t) &= \sum_k \left(\hat{V}_k^{fb}(t) + \delta V_k^{fb}(t)\right) \exp\left(jk\left(\hat{\phi}_{fb}(t) + \delta\phi_{fb}(t)\right)\right) \end{aligned} \tag{3.29}$$

then we observe from the PLL operation that the amplitude perturbations $\delta V_k^r(t)$ and $\delta V_k^{fb}(t)$ are negligible. Hence, similarly to VCO, once the steady-state coefficients $\hat{V}_k^r(t)$ and $\hat{V}_k^{fb}(t)$ are determined, the knowledge of the phase perturbations $\delta\phi_{ref}(t)$ and $\delta\phi_{fb}(t)$

is sufficient to describe the reference and feedback ports. The resulting mixed voltage-phase domain equivalent network is then as depicted in Fig. 3.17.

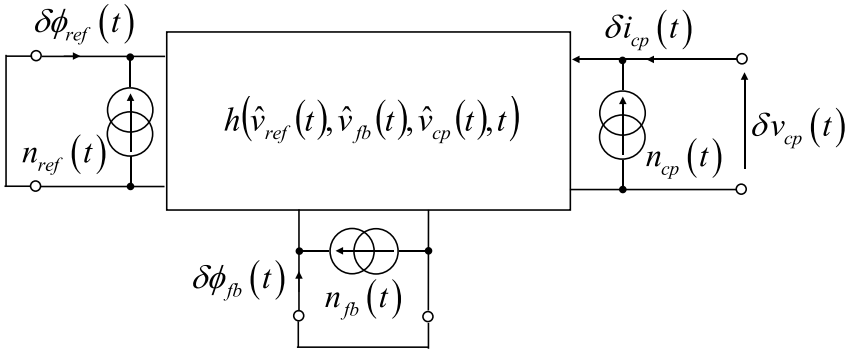


FIG. 3.17: *PFD+CP model: phases at the input and I/V at the output*

For the PFD+CP block, it is of interest to consider $\delta\phi_{ref}(t)$, $\delta\phi_{fb}(t)$ and $\delta v_{cp}(t)$ as the independent variables. Hence since the reference and feedback ports are shorted (phase-domain formalism), this leaves us with a single dependent variable $\delta i_{cp}(t)$. Also for the same reason, we may drop the Norton noise sources at the reference and feedback ports, and retain only the noise source at CP output.

From the above, the block modeling reduces to the determination of the CP output noise source $n_{cp}(t)$ and the impulse response below, which is a single line matrix.

$$h\left(\hat{v}_{ref}(t),\hat{v}_{fb}(t),\hat{v}_{cp}(t),t\right)=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \dfrac{\partial i_{cp}}{\partial\phi_{ref}} & \dfrac{\partial i_{cp}}{\partial\phi_{fb}} & \dfrac{\partial i_{cp}}{\partial v_{cp}} \end{bmatrix} \tag{3.30}$$

Equation (3.30) is the basis for most small signal PFD+CP models, though this is usually not stated explicitly. Also the CP output conductance $\dfrac{\partial i_{cp}}{\partial v_{cp}}$ is usually ignored, which may be a source of substantial error in noise analysis. Another usual limitation is that the equation (3.30) is determined under idealized port loading conditions (i.e.: open circuit at CP output port and ideal square wave signals at reference and feedback) [2] [13] [14]. In our work, thanks to the hierarchical steady-state method of chapter 2, we may effectively account for the real loading conditions and consider all the 3 terms of the impulse response.

In this purpose, we note that, within the PLL loop, all the 3 variables $\delta\phi_{ref}(t)$, $\delta\phi_{fb}(t)$ and $\delta v_{cp}(t)$ are slowly time varying signals due to the loop filter. Hence we may postulate

that, in this context, $\frac{\partial i_{cp}}{\partial \phi_{ref}}$, $\frac{\partial i_{cp}}{\partial \phi_{fb}}$ and $\frac{\partial i_{cp}}{\partial v_{cp}}$ are static characteristics, i.e.,

$$\begin{aligned} \frac{\partial i_{cp}(t)}{\partial \phi_{ref}(t)} &= \alpha_{ref} \\ \frac{\partial i_{cp}(t)}{\partial \phi_{fb}(t)} &= \alpha_{fb} \\ \frac{\partial i_{cp}(t)}{\partial v_{cp}(t)} &= G_{cp} \end{aligned} \tag{3.31}$$

The resulting equivalent circuit of the PFD+CP is depicted in Fig. 3.18, which can be easily implemented in Verilog-A. The coefficients α_{ref}, α_{fb} and G_{cp} are straightforward sensitivity analysis results from Harmonic balance analysis. Note that the PFD+CP is non-autonomous circuit, the two coefficients α_{ref} and α_{fb} have identical amplitudes and opposite signs, i.e.:

$$\alpha_{ref} = -\alpha_{fb} = \alpha \tag{3.32}$$

where α is known as the PFD+CP gain.

Finally, the power spectral density $S_{n_{cp}}(f)$ of $n_{cp}(t)$ is also determined from the transistor-level noise analysis carried on the PFD+CP, with the loading conditions indicated above.

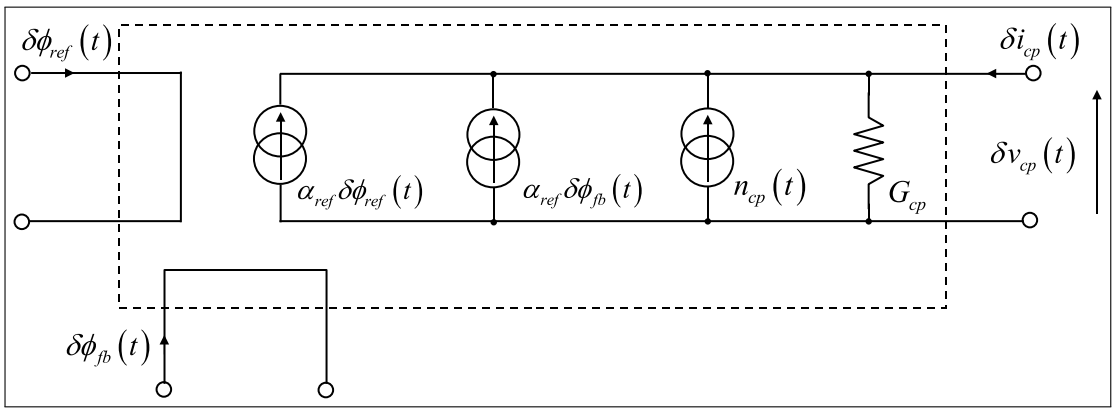


FIG. 3.18: *PFD+CP equivalent network*

3.4.2.3 Divider

Illustrated in Fig. 3.19, the divider is a two-port block with input and output variables: $v_o(t)$, $i_o(t)$, $v_d(t)$ and $i_d(t)$.

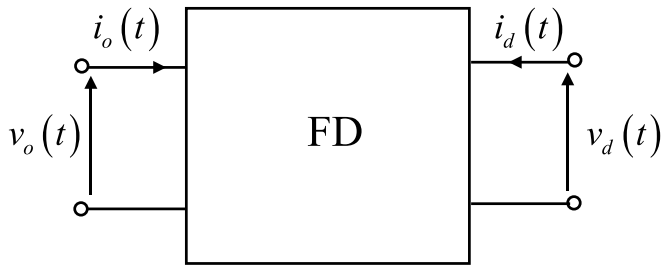


FIG. 3.19: *Divider block: two-port network*

Considering the input and output voltages expression:

$$\begin{aligned} v_o(t) &= \sum_k \left(\hat{V}_k^o(t) + \delta V_k^o(t)\right) \exp\left(jk\left(\hat{\phi}_o(t) + \delta\phi_o(t)\right)\right) \\ v_d(t) &= \sum_k \left(\hat{V}_k^d(t) + \delta V_k^d(t)\right) \exp\left(jk\left(\hat{\phi}_d(t) + \delta\phi_d(t)\right)\right) \end{aligned} \tag{3.33}$$

Similarly to the previous blocks, once the steady-state response $\hat{V}_k^o(t)$, $\hat{V}_k^d(t)$ $(k = 0, 1, ..., K)$ is obtained, we postulate that the amplitude perturbation $\delta V_k^o(t)$, $\delta V_k^d(t)$ is negligible, and the divider response to noise is fully characterized by the input and output signal phase perturbations $\delta\phi_o(t)$ and $\delta\phi_d(t)$.

Following the same reasoning as previously, and observing that

$$\frac{\partial\phi_d(t)}{\partial\phi_o(t)} = \frac{1}{N} \tag{3.34}$$

where N is the division rank, we readily end up with the equivalent phase domain network depicted in Fig. 3.18.

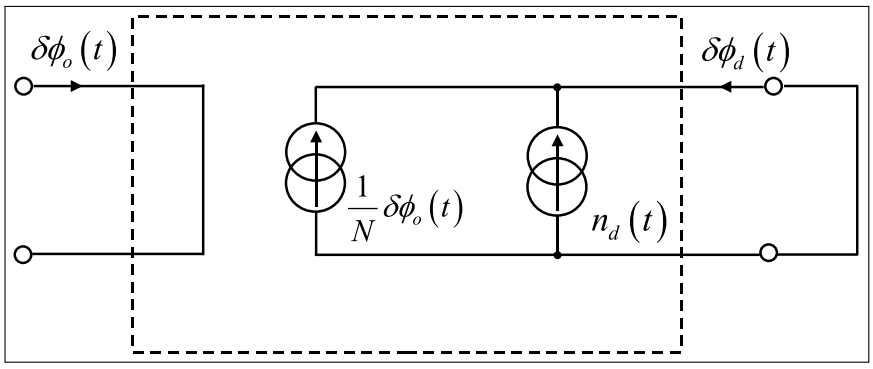


FIG. 3.20: *Divider equivalent network*

Finally the only operation required here is to compute the PSD $S_{n_d}(f)$ of $n_d(t)$ using a transistor-level noise analysis of the FD, with the loading conditions from the PLL steady-state response.

3.4.2.4 Low-pass filter

As indicated in chapter 1, the LPF circuit is usually simple passive structure that can be efficiently simulated at the transistor-level, especially because this is traversed by a low frequency signal. Using a transistor-level model maintains the accuracy of the LPF simulation at the highest level.

3.4.3 Deterministic noise modeling

As already indicated, the deterministic noise may come from the PLL internal signals, or from the external sources like the power supply and the substrate. Here we will only consider the internal noise. This has many causes such as CP current mismatch, CP and LPF leakage current, etc [15] [16]. All these causes lead to the perturbation of the VCO control voltage and then have an impact on the VCO output by modulating the VCO oscillation frequency. As illustrated in Fig. 3.21, the current mismatch of the CP causes periodic ripple on the control line of the VCO. As a result, the ripple modulates the VCO frequency and generates deterministic noise at $F_0 + k \cdot F_{ref}$.

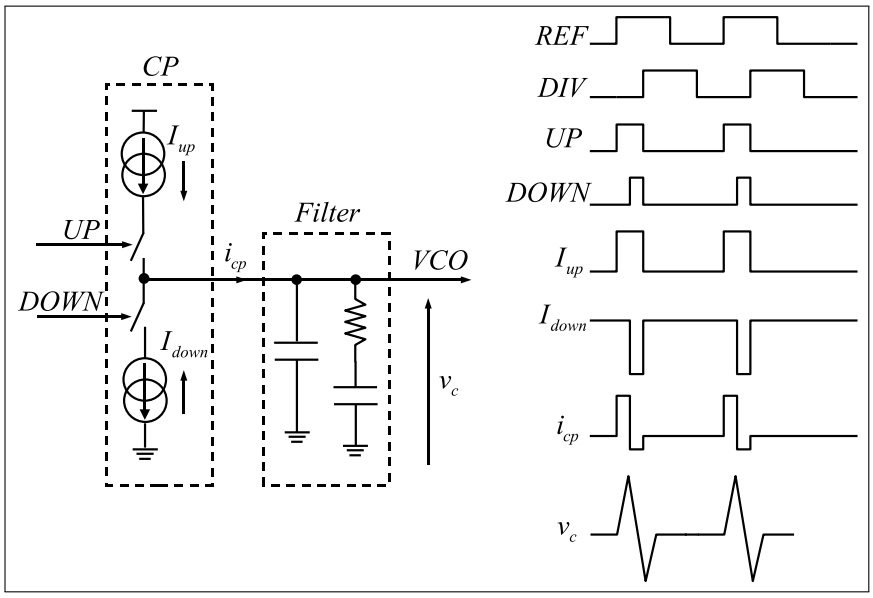


FIG. 3.21: *Cause of the deterministic noise: current mismatch*

As mentioned earlier, the deterministic noise is directly available from the steady-state analysis of the PLL, using the techniques presented in chapter 2. However, because the ripple in the VCO control voltage is usually small, the resulting accuracy on the noise is highly subjected to the aliasing error in the Harmonic balance or truncation error in Time shooting processes used to compute the steady-state response. A better accuracy can be obtained by considering the VCO control ripple contribution to phase

noise from a linear perturbation analysis. By doing so, we avoid the direct addition of the small amplitude ripple to the large signal state, thus avoiding truncation and aliasing errors. Another benefit of considering deterministic noise analysis as a small perturbation analysis is the uniformity of noise analysis. Phase noise contributions from both random and deterministic noise can be calculated using the same simulation models, as will be shown.

Since the deterministic noise as illustrated above results from the high frequency components present in the CP current, this can be effectively modeled as the deviation of the CP output current from its average value.

Consider $\hat{i}_{cp}(t) = \sum_k \hat{I}_k^{cp} \exp(j2\pi k f_{ref} t)$ the current delivered by the CP, upon the steady-state analysis of chapter 2. Because $\hat{i}_{cp}(t)$ is a large signal with large amplitude harmonic contents, it does not suffer aliasing and truncation errors. Hence it is calculated with high accuracy. We thus express the deterministic current noise source as:

$$d_{cp}(t) = \hat{i}_{cp}(t) - \overline{\hat{i}_{cp}(t)} \tag{3.35}$$

The power spectral density of the deterministic noise source is therefore:

$$S_{d_{cp}}(f) = \sum_{k \neq 0} \hat{I}_k^{cp} \cdot \delta(f - k \cdot f_{ref}) \tag{3.36}$$

Gathering the above with the small signal equivalent network of the PFD+CP block obtained in section 3.4.2.2, we obtain the final equivalent network depicted in Fig. 3.22.

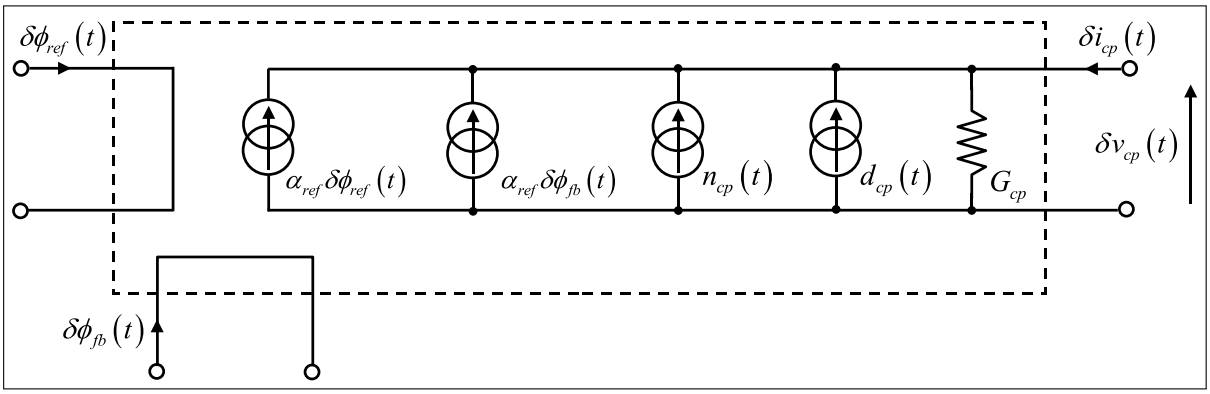


FIG. 3.22: *PFD+CP equivalent network with deterministic source*

3.4.4 Overall block-level noise simulation

Using the small signal equivalent network derived in the preceding section, we are in a position to simulate the PLL behavior as a response to a small perturbation of the large signal steady-state equilibrium. We can therefore calculate the PLL noise response. Because the block models have been drawn on the assumption of negligible amplitude perturbation, we may compute the PM noise characteristics. The other small perturbation characteristics of the steady-state regime (loop bandwidth and phase margin, presented in chapter 1 / section 1.4.1), can also be computed with this model. The overall block-level diagram for noise and small perturbation analysis of the PLL is shown in Fig. 3.23, where

- $\alpha_{ref} = -\alpha_{fb} = \alpha$: the PFD+CP gain
- $n_{cp}(t)$: Norton equivalent current source from random noise internal to PFD+CP
- $d_{cp}(t)$: Norton equivalent current source from deterministic noise internal to PFD+CP
- G_{cp}: CP output conductance
- $\dfrac{\partial \omega_o}{\partial v_c}$: VCO static frequency sensitivity
- G_{in}, C_{in}: Low frequency VCO input conductance and capacitance
- $n_o(t)$: Norton equivalent phase source from random noise internal to VCO
- N: Division rank of FD
- $n_d(t)$: Norton equivalent phase source from random noise internal to FD

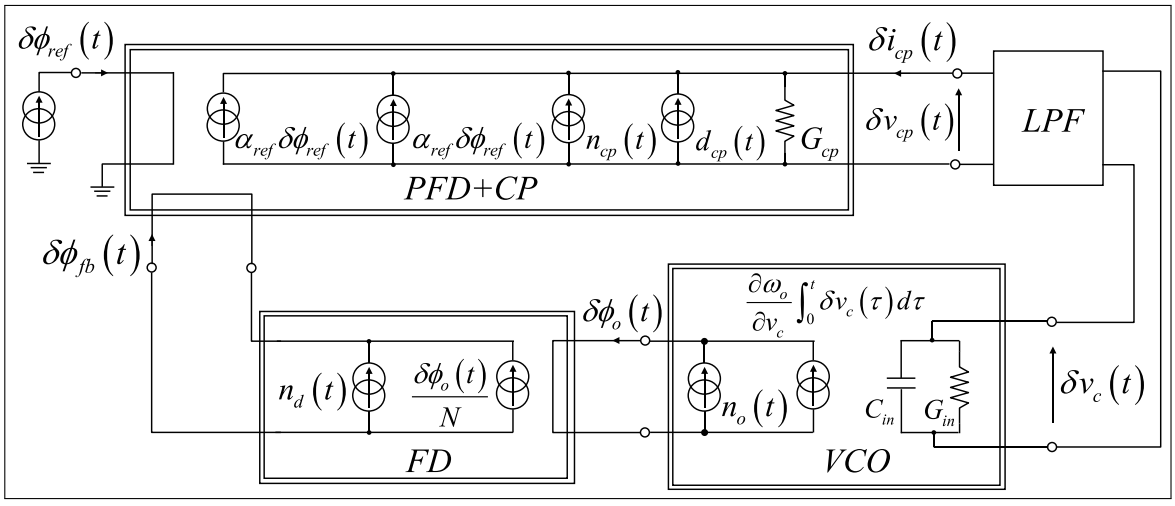


FIG. 3.23: *Block-level PLL model for noise and small perturbation analysis*

3.4.5 Simulation results

The model in the above section has been implemented in Verilog-AMS [17] [18]. We have carried the characterization of a number of PLL circuits and performed the noise analysis. The result of our block-level approach is indistinguishable from the brute-force transistor-level simulation, while the simulation time and memory usage are many orders lower. As an illustration example, we have considered a PLL comprising 320 CMOS devices, division rank = 40, the reference frequency $f_{ref} = 264MHz$. The noise characterization of the various blocks and the overall noise results of the PLL are shown below.

3.4.5.1 Noise characterization of the various blocks

Fig. 3.24 to Fig. 3.26 show respectively the internal VCO phase noise power spectral density (PSD), the internal divider phase noise PSD and the internal PFD+CP current noise PSD.

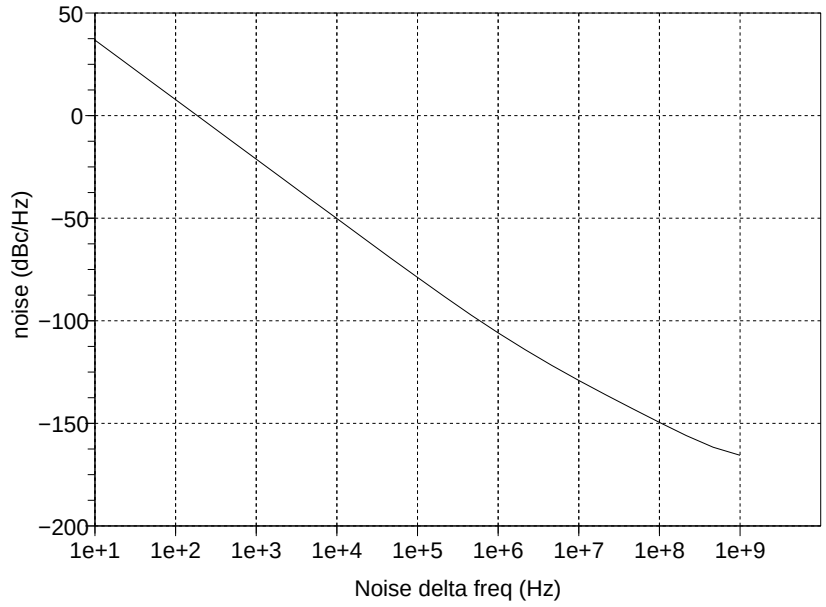


FIG. 3.24: *Internal VCO phase noise PSD:* $S_{n_o}(f)$

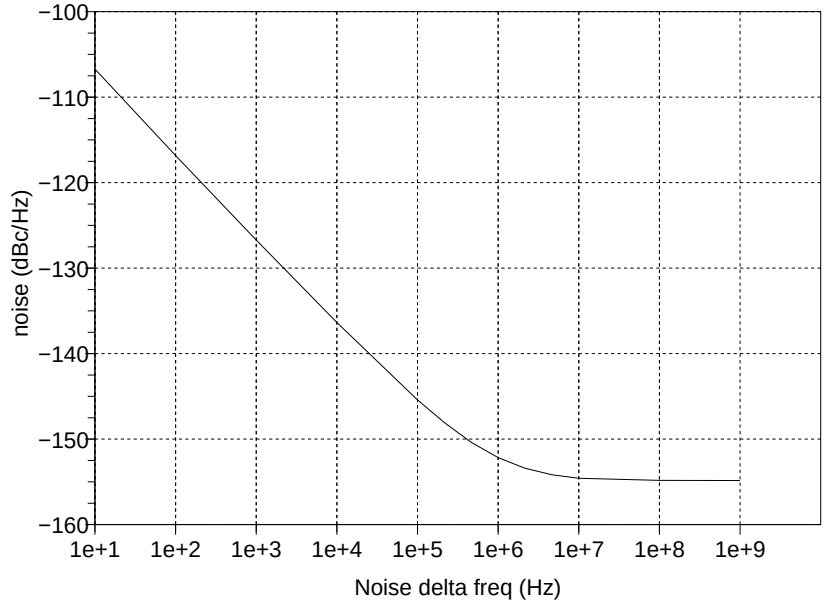


FIG. 3.25: *Internal divider phase noise PSD:* $S_{n_d}(f)$

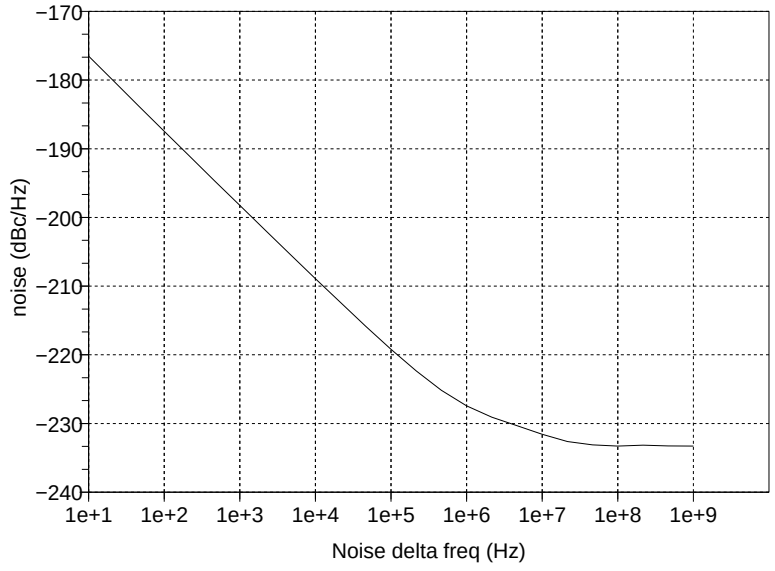


FIG. 3.26: *Internal PFD/CP current noise PSD:* $S_{n_{cp}}(f)$

3.4.5.2 Simulation results of PLL phase noise

First we have ignored the deterministic noise and considered the reference port free of noise, we thus obtain at the VCO output, the typical PLL output PSD shown in Fig. 3.27, which is compared with the result from the brute-force transistor-level simulation. The simulation time for block-level is less than 2 hours. This time is essentially spent on the block modeling process. The transistor-level simulation in turn requires about 2 days.

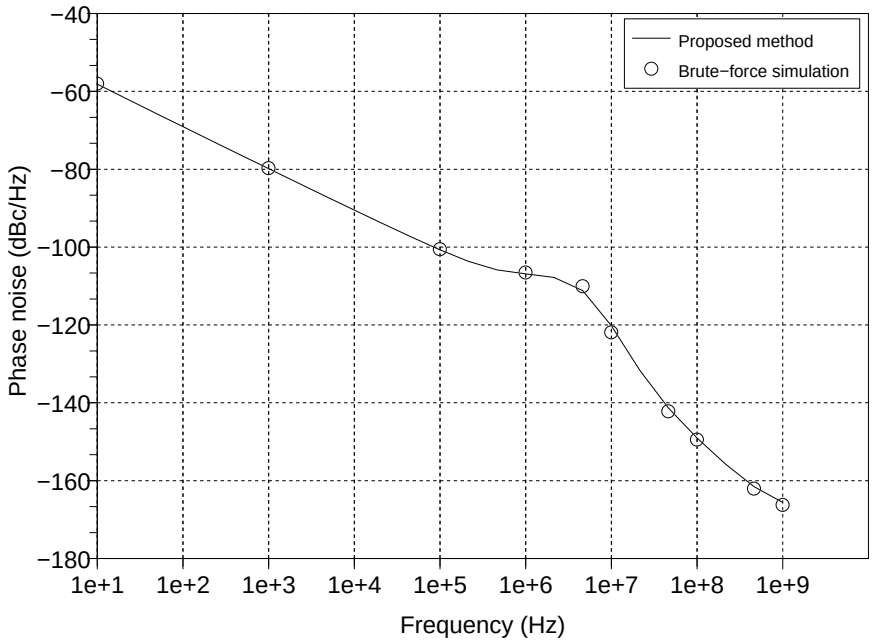


FIG. 3.27: *Phase noise of the PLL: Proposed method and brute force transistor-level simulation*

Another important aspect of block-level simulation is that we may more easily isolate the contribution of each block to the total phase noise, as shown in Fig. 3.28. We then see that the PFD+CP is the largest noise contributor within the loop bandwidth, while the VCO phase noise dominates outside the bandwidth. The divider and LPF noise contribution is small.

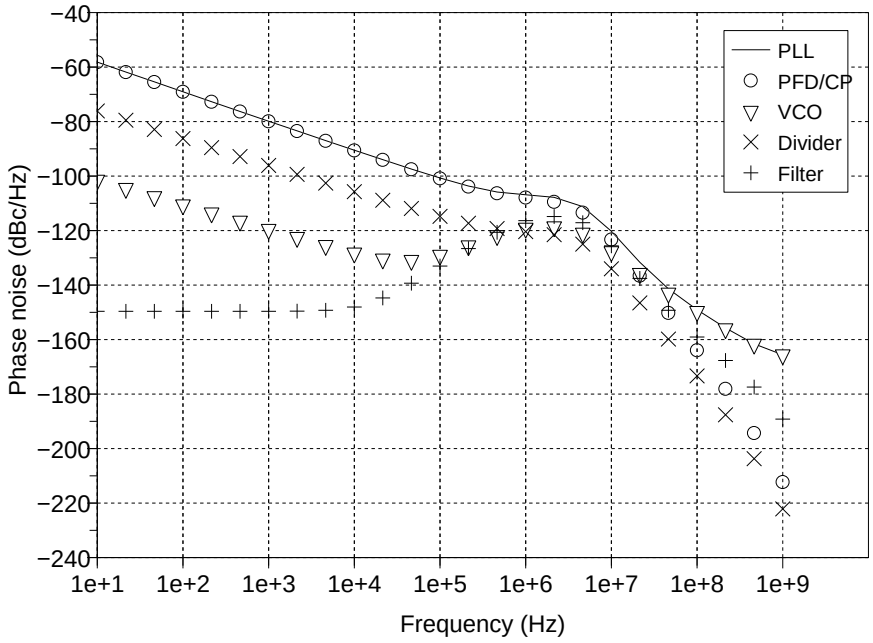


FIG. 3.28: *Noise contribution of each block: proposed method*

3.4.5.3 Simulation results of PLL deterministic noise

Finally we have incorporated the deterministic noise in the PLL model. The discrete PSD of the CP current is shown in Fig. 3.29, and the resulting phase noise PSD at PLL output including random and deterministic noise is shown in Fig. 3.30. The 3 visible spikes can be compared with the power spectrum obtained from the transistor-level steady-sate analysis in Fig. 3.31. We thus see that they match well, as shown in Tab. 3.1.

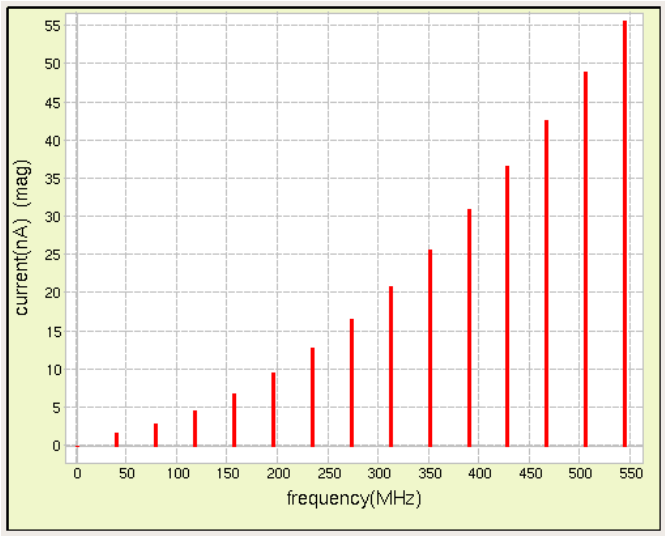


FIG. 3.29: *Discrete PSD of the CP current:* $S_{d_{cp}}$

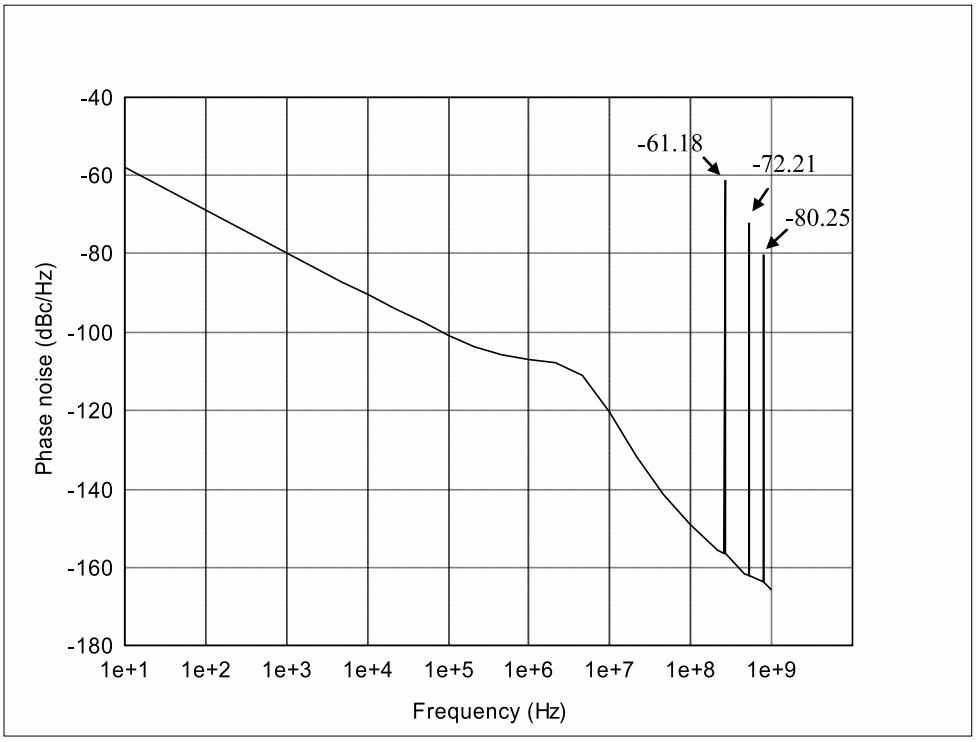


FIG. 3.30: *Deterministic noise: proposed model*

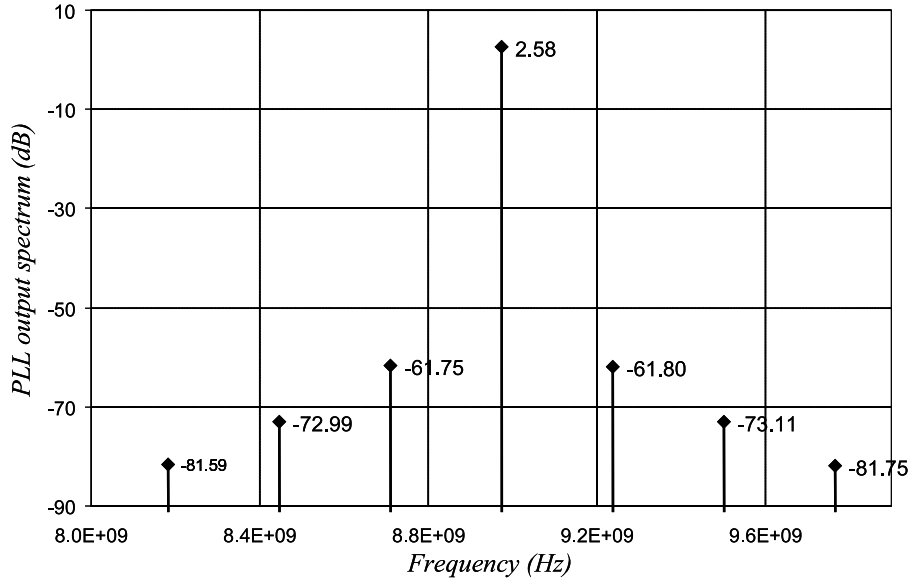


FIG. 3.31: *Deterministic noise: brute-force transistor-level simulation*

Frequency	Proposed method	Brute force simulation
f_{ref}	-61.18 dB	-61.36 dB
$2f_{ref}$	-72.21 dB	-72.63 dB
$3f_{ref}$	-80.43 dB	-81.25 dB

TAB. 3.1: *Deterministic noise: comparison between the proposed method and the brute force simulation*

3.4.5.4 Simulation time comparison

With the proposed method, the overall simulation time is listed in Tab. 3.2 for a few PLL circuits, and compared with the brute-force transistor-level simulation.

Circuit	MOS device count	Division ratio	Proposed method	Transistor-level
I	463	16	0.7 hours	3.8 hours
II	491	32	1.1 hours	5.4 hours
III	320	40	1.7 hours	48 hours

TAB. 3.2: *Noise analysis time of the proposed method and the transistor-level simulation*

3.5 Conclusion

In this chapter a noise analysis method concerning the random and deterministic phase noise of the PLL is presented, which is built upon the hierarchical steady-state solution obtained in chapter 2.

The block models are proposed in which the non-idealities of the PLL are taken into account, such as output impedance of the charge pump, input impedance of the VCO, and the non-ideal square wave signals at PFD ports.

The simulation results have shown that the proposed method is accurate and fast.

3.6 Bibliography

[1] M. Tavares, "PLL Frequency synthesizers: Phase noise issues and wide-band loops," Ph.D. dissertation, INSA de Lyon, 1999.

[2] K. Kundert. (2006) Predicting the phase noise and jitter of PLL-based frequency synthesizers. [Online]. Available: www.designers-guide.org/Analysis/PLLnoise.pdf

[3] A. Demir, "Analysis and simulation of noise in nonlinear electronic circuits and systems," Ph.D. dissertation, University of California, 1997.

[4] K. Kundert. (2000) Noise in mixers, oscillators, samplers, and logic: an introduction to cyclostationary noise. [Online]. Available: http://www.designers-guide.org/Theory/cyclo-paper.pdf

[5] M. Prigent, "Contribution à l'étude de la conversion de fréquence dans les circuits non-linéaire : Application à la CAO d'oscillateurs à bruit de phase minimum," Ph.D. dissertation, Université de Limoges, 1987.

[6] J. Nallatamby, "Détermination des caractéristiques en bruit des circuits non-linéaires à l'aide des formalismes de conversion de fréquence et de matrice de corrélation des sources de bruit," Ph.D. dissertation, Université de Limoges, 1992.

[7] E. Ngoya and R. Larcheveque, "Envelop transient analysis: a new method for the transient and steady state analysis of microwave communication circuits and systems," *IEEE MTT-S International Microwave Symposium Digest*, vol. 3, pp. 1365–1368, 1996.

[8] P. Feldmann and J. Roychowdhury, "Computation of circuit waveform envelopes using an efficient, matrix-decomposed harmonic balance algorithm," in *IEEE International Conference on CAD Digest, pp. 295-300, November*, 1996.

[9] E. Ngoya, J. Rousset, and D. Argollo, "Rigorous RF and microwave oscillator phase noise calculation byenvelope transient technique," *IEEE MTT-S International Microwave Symposium Digest*, vol. 1, 2000.

[10] V. Rizzoli, F. Mastri, and D. Masotti, "General noise analysis of nonlinear microwave circuits by thepiecewise harmonic-balance technique," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 5, pp. 807–819, 1994.

[11] A. Suárez, S. Sancho, S. V. Hoeye, and J. Portilla, "Analytical comparison between time-and frequency-domain techniques for phase-noise analysis," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 10, pp. 2353–2361, 2002.

[12] S. Sancho, A. Suárez, and J. Chuan, "General envelope-transient formulation of phase-locked loops using three time scales," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, pp. 1310–1320, 2004.

[13] J. Zou, D. Mueller, H. Graeb, U. Schlichtmann, E. Hennig, and R. Sommer, "Fast automatic sizing of a charge pump phase-locked loop based on behavioral models," *Proceedings of the IEEE International Behavioral Modeling and Simulation Workshop*, pp. 100–105, 2005.

[14] M. Hinz, I. Konenkamp, and E.-H. Horneber, "Behavioral modeling and simulation of phase-locked loops for RF front ends," *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems*, vol. 1, pp. 194–197 vol.1, 2000.

[15] K. Shu and E. Sánchez-Sinencio, *CMOS PLL Synthesizers: Analysis and Design*, 1st ed. Springer, Nov. 2004.

[16] B. Razavi, *RF Microelectronics*. Prentice Hall, 1998.

[17] *Verilog-AMS Language Reference Manuals. version 2.3*, Accellera, August 2008. [Online]. Available: http://www.eda.org/verilog-ams/htmlpages/lit.html

[18] K. S. Kundert, *The Designer's Guide to Verilog-AMS*, 1st ed. Kluwer Academic Publishers, Boston, May 2004.

Chapter 4 :

Dynamic performances analysis of the PLL

4.1 Introduction

In the previous chapter, we have developed new models to analyze the noise of the PLL. These models are valid only for small perturbations that maintain the PLL in the locked state. For large perturbations that make the PLL wander back and forth between locked and unlocked states, other models are necessary.

Take an example of a frequency synthesizer, and assume that it is initially in the locked state. When the division ratio N changes, the PLL becomes momentarily unlocked and it adjusts the VCO frequency until the divider output frequency catch up again with the reference frequency. In this process, what we are interested in is how long it takes the PLL to return to the locked state. When the PLL switches between the locked and non-locked states, its behavior is characterized by what we will call the dynamic performances.

In this chapter we first introduce some basic concepts on the dynamic performances of the PLL, such as the tracking and acquisition process. Then we review the closed-form formulas used to calculate the lock time of the PLL. Although these simple methods are useful in the PLL specification stage, they are not precise enough for the PLL design verification. As a result, a simulation of the PLL is often necessary to precisely verify its dynamic performances.

Then, we review some existing simulation methods based on the block-level modeling, and propose a new modeling approach.

4.2 Principles

In this section, we present the basic concepts of the PLL dynamic performances, such as tracking process and acquisition process, and the characteristics associated with these processes, like hold range, capture range, lock time, etc.

4.2.1 Tracking process / Acquisition process

The tracking process describes how the PLL follows the variation of the reference signal. First we denote f_{ref} the reference frequency , and f_{div} the feedback frequency. Assume that a PLL is initially in the locked state. If one varies f_{ref} smoothly and slowly, the PLL can track this variation while remaining in the locked state. This is called *tracking process*, which can be characterized by the *hold range* $\Delta\omega_H$, within which the PLL remains in lock. However, if one changes f_{ref} in an abrupt manner, in other words, if a frequency step is applied to f_{ref}, the PLL is likely to remain in the locked state only if the frequency step

does not exceed a limit, which is defined as the *pull-out range* $\Delta\omega_{PO}$. One may easily imagine that the pull-out range is smaller than the hold range.

The acquisition process describes how the PLL moves from the unlocked state towards the locked state. While the feedback frequency f_{div} approaches the reference frequency, and arrives at a boundary, the PLL would become locked eventually. Within this boundary the PLL is in the *capture range* $\Delta\omega_C$, and the corresponding process is called *pull-in process*, which is relatively slow. If the two frequencies continue to approach, then the PLL would enter the *lock range* $\Delta\omega_L$, within which the absolute value of the phase difference would be less than 2π, the two frequencies are almost identical, and the PLL adjusts primarily the phase for entering the locked state. This process is named *lock-in process* in which the PLL can reach the locked state very quickly.

Generally, the capture range is smaller than the hold range. However, for the charge-pump PLL, the capture range and the hold range are only limited by the VCO tuning range, because the charge pump and the filter constitute an integrator whose DC gain is almost infinite, hence generating a voltage which can cover the VCO tuning range, as mentioned in Chapter 1. Therefore, the PLL can get locked from any frequency within the VCO tuning range. In summary, we can get the following relationship between the four frequency ranges, as shown in Fig. 4.1: [1]

$$\Delta\omega_L < \Delta\omega_{PO} < \Delta\omega_{PI} < \Delta\omega_H \tag{4.1}$$

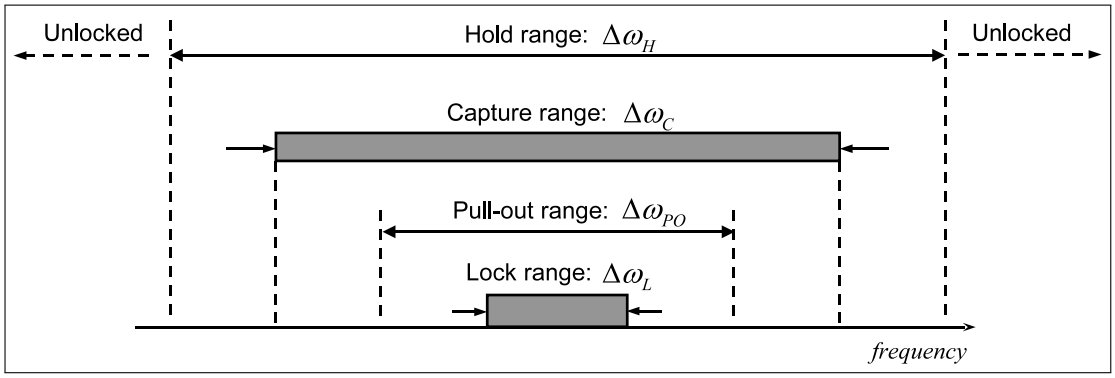


FIG. 4.1: *Relationship between the different frequency ranges*

The estimation of these frequency ranges is often very difficult and not precise [1] [2]. In practice, a complete PLL simulation is needed to obtain these parameters accurately.

4.2.2 Lock time

The lock time (named also settling time) is the time needed for a PLL output frequency f_{vco} to jump from one specified value f_1 to another specified value f_2 within a given frequency tolerance f_{tol} [2], as shown in Fig. 4.2.

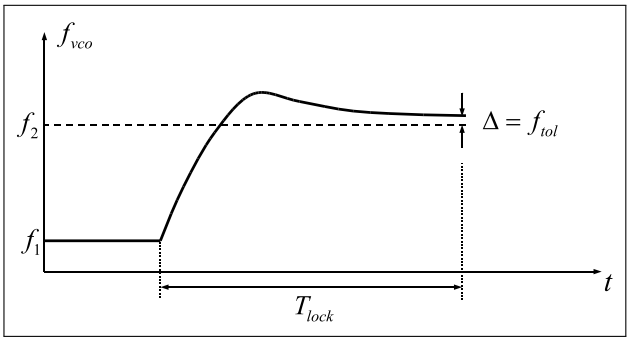


FIG. 4.2: *Lock time definition*

For a 2nd order PLL, the lock time can be approximated as follows: [2]

$$T_{lock} = \frac{1}{\zeta\omega_n} \ln\left(\frac{1}{\varepsilon\sqrt{1-\zeta^2}}\right) \tag{4.2}$$

where ζ is the damping factor of the 2nd order PLL (defined in the equation (1.39) in the chapter 1), f_n is the natural frequency (defined in the equation (1.38) in the chapter 1), ε represents a normalized frequency tolerance:

$$\varepsilon = \frac{f_{tol}}{|f_2 - f_1|} \tag{4.3}$$

In practice, a rule of thumb is often used to estimate the lock time of the PLL:

$$T_{lock} = \frac{4}{f_c} \tag{4.4}$$

where f_c is the loop bandwidth of the PLL.

For the 3rd order PLL, the calculation of the lock time is more difficult since there is no damping factor for the 3rd order PLL. Recently, a method [3] is proposed to define an effective coefficient ζ_{eff}, and use it to express the lock time:

$$T_{lock} = \frac{\ln(1/\varepsilon)}{f_c\zeta_{eff}} \tag{4.5}$$

Equation (4.2) to (4.5) are idealized figures useful for specification purpose. An accurate simulation of the PLL is necessary to precisely calculate the dynamic performances

during the design phase.

4.3 Existing simulation methods

An accurate calculation of the dynamic characteristics of the PLL can be obtained from a brute-force transistor-level transient simulation of the PLL. But this is almost impractical, as it is extremely time-consuming (from hours to days and weeks). Therefore the only practical solution is to consider a block-level simulation. The accuracy of the block-level simulation relies heavily on the accuracy of the block models.

Among all the blocks of the PLL, the most critical ones for modeling accuracy are PFD+CP and VCO. We will briefly indicate the previous work in this area [4] [5] [6] [7], before introducing our methodology.

4.3.1 PFD modeling

A number of approaches can be found for PFD or PFD+CP modeling. The simplest approach is the mixer model, where the output is considered as a direct product of the two input signals [1]. In this simple model the non-idealities at transistor level are ignored, and the model does not apply to digital circuit.

The second most popular model is the one based on an idealized truth table [8] [9]. The model applies to the digital type PFD, but ignores the analog characteristics of the signals. The accuracy of these types of models is limited and may not efficiently apply to design verification.

The other models of PFD are derived from the analysis or the characterization of the PFD at transistor-level, and they have considerations about the analog nature of the signals. We will briefly describe two of them: one is based on duty cycle averaging and the other is based on uniform time step sampling.

4.3.1.1 Average duty cycle model

As indicated in chapter 1, it is observed that when the frequency difference between the two inputs of the PFD is large, the PLL starts by adjusting its frequency to approach the locked state, so we call it the *frequency mode*. In this mode, the output current of the CP is determined by the *average* duty cycle of the PFD output voltage which can be calculated by the equation (1.15) in the chapter 1. When the two input frequencies become close, the PLL starts adjusting its phase, and we call it the *phase mode*, in which the average

duty cycle is about 0.5 or -0.5. Finally when the PLL is in the locked state, its nominal duty cycle approaches zero. Unfortunately, the equation (1.15) can not explain how the duty cycle changes from ± 0.5 to zero when the PLL gets into lock. The difficulty of the modeling lies in how to combine these two distinct modes and deal with the transition between them.

An interesting solution has been proposed by Jess Chen [4], which uses the reference frequency and the divider frequency as its input values. A weighting factor k is introduced to realize a smooth transition between the frequency mode and the phase mode. The model is expressed by the equation (4.6).

$$duty\_cycle = k \cdot H + (1-k)^2 \int_0^t (freq\_diff/2\pi)d\tau \tag{4.6}$$

where H represents the average duty cycle in the frequency mode, and the integration of frequency difference $freq\_diff$ represents the phase difference in the phase mode.

Fig. 4.3 illustrates the model as it transits between the two modes.

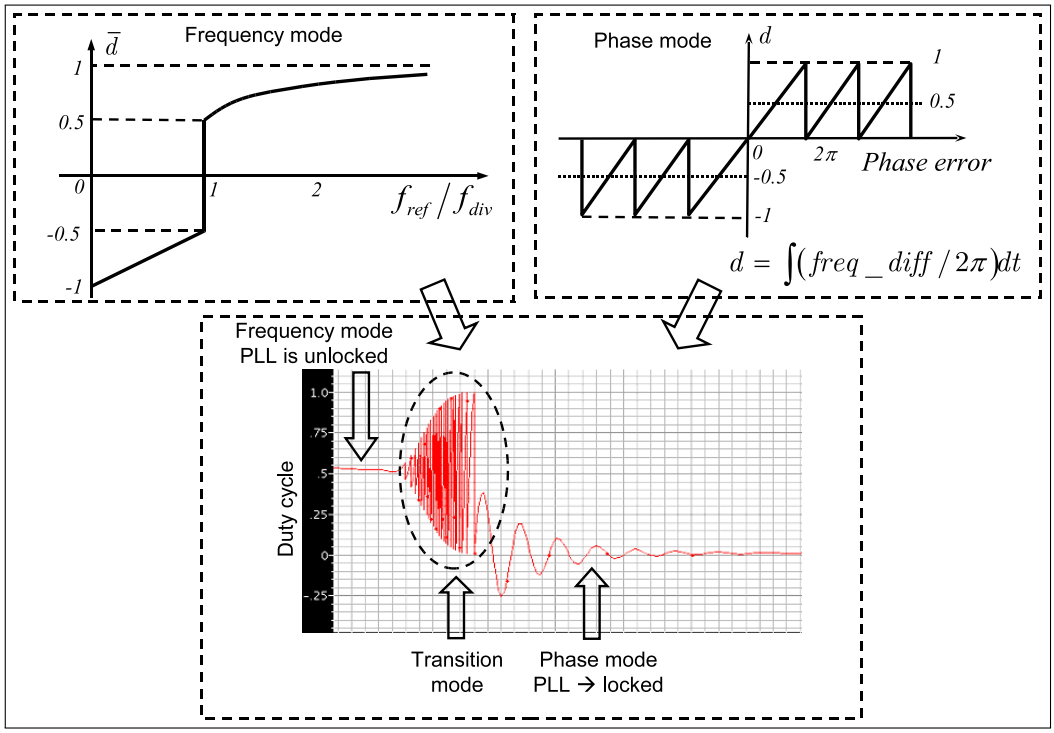


FIG. 4.3: *Model of Jess Chen*

Nevertheless, the value of the weighting factor k is empirical, so it may result in large error if k is not well chosen. Moreover, the model uses an averaged value of duty cycle, so the high frequency content of the output signal is ignored.

4.3.1.2 Uniform time step sampling model

Perrot *et al* [5] proposes a PFD modeling method in which the continuous-time (CT) output of the PFD is approximated by the discrete-time (DT) sequence using an area conservation principle. The input signals of the PFD are sampled with a uniform time step. For the rectangular pulses at edge boundaries, the pulse width varies between 0 and the sampling period. In the aim of reducing the quantization noise around the signal edge, the DT value are not fixed to 0 and 1 but can vary between 0 and 1 depending on the location of the transition edge. In order to precisely characterize the PFD, two assumptions should be met: the sampling frequency is at least 100 times greater than the bandwidth of the PLL; the sampling frequency is set to be an integer multiple of the reference frequency.

However, the model for CP is a simplified one, and the dependence of the CP output current on its output voltage is not taken into account.

4.3.2 VCO and VCO+divider modeling

As indicated in chapter 1, one of the reasons for the long simulation time is that the VCO frequency is much higher than the reference frequency and the loop bandwidth. Consequently, the transient simulation of the PLL is performed over a very long time span with a very small time step.

One way to bypass the small time constant present at the VCO output is to consider a single model for the VCO and the divider. Hence we have to deal with a block cascade where the input signal (VCO control signal) and the output signal (divider output) are all about the same time constant. The high frequency components from the VCO are thus eliminated and the simulation speed is accelerated.

A number of works have been carried in this direction [5] [6] [7]. However, the proposed solutions usually focus only on the output signal phase component and ignore the signal waveform. Hence the effects due to the waveform analog distortion can not be accounted for in the PLL simulation.

In the purpose of providing a more accurate simulation, in the next section we propose a modeling methodology for VCO+divider and PFD+CP that carefully consider the analog aspects of the signals and their non-ideal loading conditions of the blocks.

4.4 Proposed method

4.4.1 VCO+divider modeling

In this section, we will consider the VCO and the divider as a combined block and extract a VCO+divider model, which is then verified in the PLL simulation.

4.4.1.1 Model derivation

As shown in Fig. 4.4, the VCO and the divider are combined into one block, whose input and output signals are $v_c(t)$, $i_c(t)$, $v_d(t)$ and $i_d(t)$.

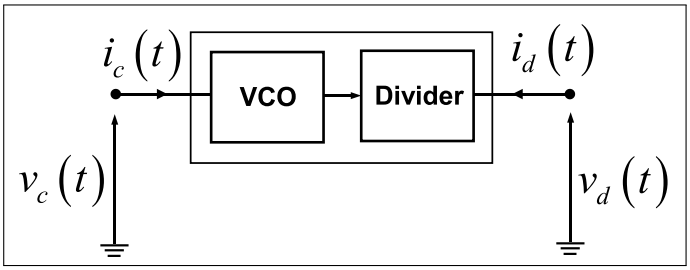


FIG. 4.4: *Definition of the input and output signals of VCO+divider block*

It's necessary to find the relationship between $v_c(t)$, $v_d(t)$, $i_c(t)$, $i_d(t)$, i.e:

$$\psi\left(v_c(t), v_d(t), i_c(t), i_d(t), t\right) = 0 \tag{4.7}$$

1/ We assume that the divider load (i.e., the PFD feedback input) is a high impedance (this has been shown in chapter 2) so that the divider output current can be ignored. Hence only the output voltage $v_d(t)$ is of concern for this block modeling.

2/ We assume that because of the low cut-off frequency of the LPF, the VCO control signal $v_c(t)$ contains only low frequency components; so it's a slowly varying function of time. Under this condition we may expect that the VCO output voltage expresses in the following form:

$$v_o(t) = \sum_k V_k^o\left(v_c(t)\right) exp\left(j2\pi k \int_0^t f_o\left(v_c(\tau)\right) d\tau\right) \tag{4.8}$$

where $V_k^o\left(v_c(t)\right)$ is the time varying Fourier coefficients, and $f_o\left(v_c(t)\right)$ the time varying oscillation frequency. Both $V_k^o\left(v_c(t)\right)$ and $f_o\left(v_c(t)\right)$ are slowly varying time functions, and are static characteristics in terms of $v_c(t)$, i.e., implicit functions of time through $v_c(t)$.

Therefore the output signal of the divider can be expressed as:

$$v_d(t) = \sum_k V_k^d(v_c(t)) exp\left(j2\pi k \int_0^t f_o(v_c(\tau))/N d\tau\right) \tag{4.9}$$

where N is the division rank.

3/ We assume that the input impedance of the VCO is high at low frequency. Taking account of that, the input current to the VCO $i_c(t)$ is negligible. This also is verified from chapter 2. The above assumptions simplify the modeling problems, so that we have only to find the relationship between $v_c(t)$ and $v_d(t)$.

Because $V_k^d(v_c(t))$ and $f_o(v_c(t))$ are static characteristics, they can be easily obtained by considering a DC control voltage and sweeping its amplitude from zero to the maximal admissible value. Fig. 4.5 illustrates the characteristic of $f_o(v_c)$ for a typical VCO+divider circuit; Fig. 4.6 and 4.7 show the characteristics of $V_k^d(v_c)$ for the same VCO+divider circuit. We may see that they are both smooth functions of v_c, and this is usually the case for most of applications. This interesting feature will allow us to use a low order polynomial in model implementation.

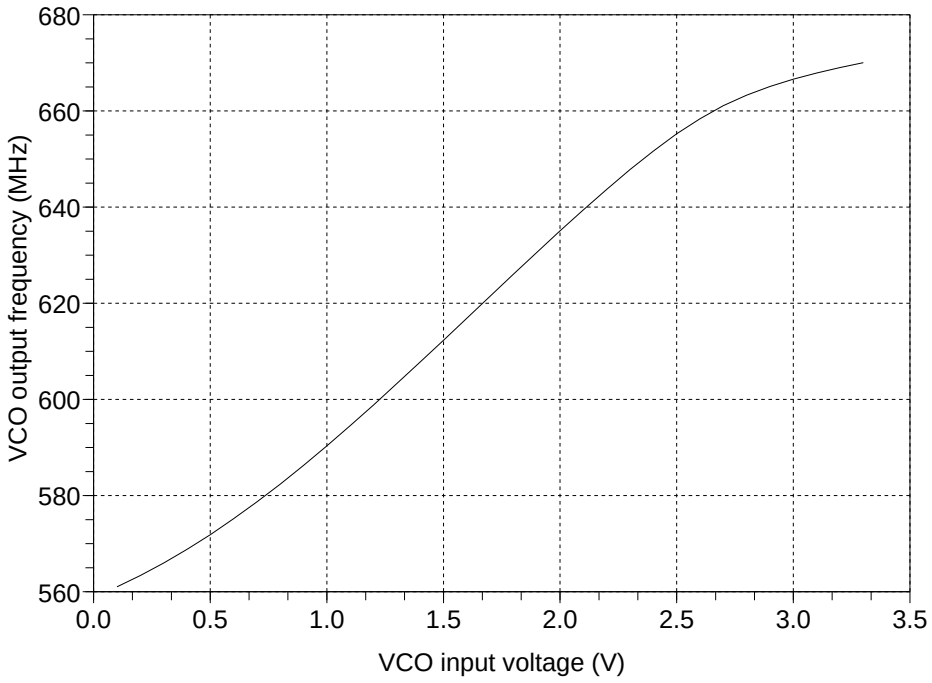


FIG. 4.5: *v_c vs. VCO output frequency $f_o(v_c)$*

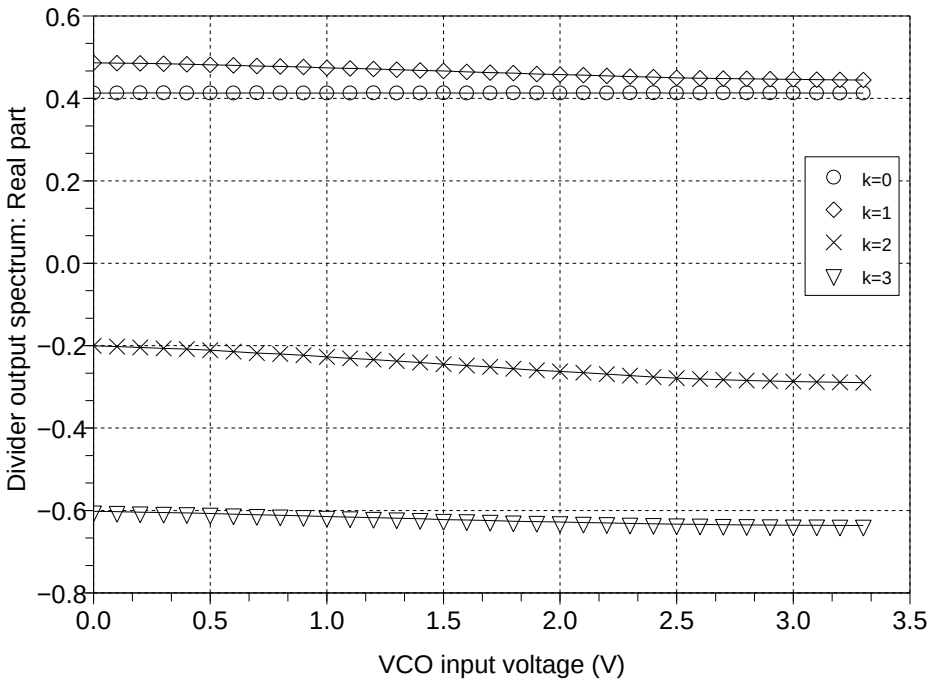


FIG. 4.6: v_c *vs. divider output spectrum coefficients - real part* $Re\left(V_k^d\left(v_c\right)\right)$

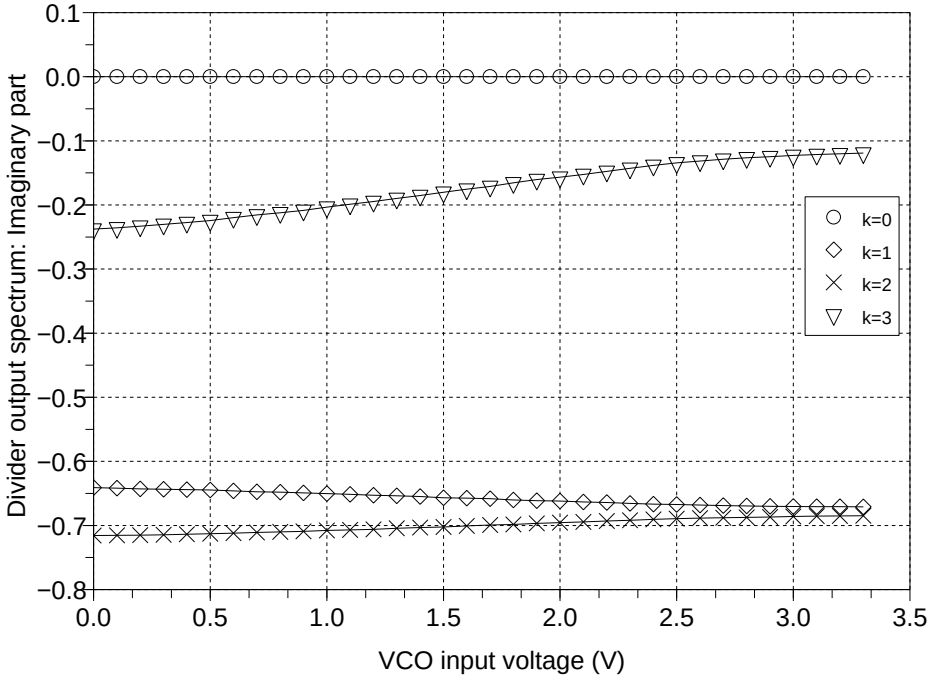


FIG. 4.7: v_c *vs. divider output spectrum coefficients - imaginary part* $Im\left(V_k^d\left(v_c\right)\right)$

4.4.1.2 Model implementation

Once the static characteristics $V_k^d(v_c)$ and $f_0(v_c)$ are obtained by Harmonic balance, the VCO+divider model can be easily implemented, e.g., in Verilog-A.

$$\begin{cases} v_d(t) = \sum_k \left[a_k(t)\cos(k\phi(t)) - b_k(t)\sin(k\phi(t))\right] \\ a_k(t) = \mathrm{Re}\left[V_k^d(v_c(t))\right] \\ b_k(t) = \mathrm{Im}\left[V_k^d(v_c(t))\right] \\ \phi(t) = \int_0^t 2\pi f_0(v_c(\tau))/N d\tau \end{cases} \tag{4.10}$$

4.4.1.3 Model verification

To verify the accuracy and the efficiency of the proposed model, in this section we will simulate the main dynamic performances of the PLL (lock time, hold range and capture range) using the proposed VCO+divider model. As illustrated in Fig. 4.8, the VCO+divider model is inserted into the PLL circuit to replace the transistor-level VCO+divider circuit. The PFD+CP and the filter remain at the transistor-level, thus their non-ideal characteristics like the CP output current mismatch and dead zone can be included in the simulation. The PLL considered here is a CMOS PLL comprising 320 transistors, with a reference frequency of 264MHz and a division rank of 40.

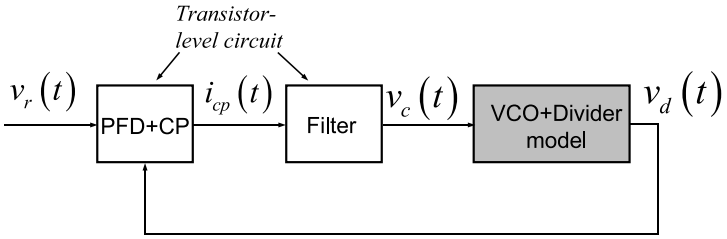


FIG. 4.8: *PLL simulation with the proposed VCO+divider model*

Fig. 4.9 shows the lock time simulation. First the reference frequency f_{ref} is set to the nominal value $264MHz$, and the PLL is brought to lock. This takes about $0.4us$. At time $t = 1us$ the reference frequency is made to suddenly jump from 264 to $271.75MHz$, and the PLL is again brought to lock. After that, at time $t = 3us$, the reference frequency is set back to the nominal value and the PLL locks again. We may observe the lock time for the up and down frequency jumps, which is about $0.6us$ in the two cases.

For measuring the capture and hold ranges, a practical way is to carry the experiment illustrated in Fig. 4.10, where the reference frequency $f_{ref}(t)$ is made to follow slowly the indicated time trajectories. To determine the lower boundaries of the capture and hold ranges, $f_{ref}(t)$ is moved upward from outside into the capture range and then

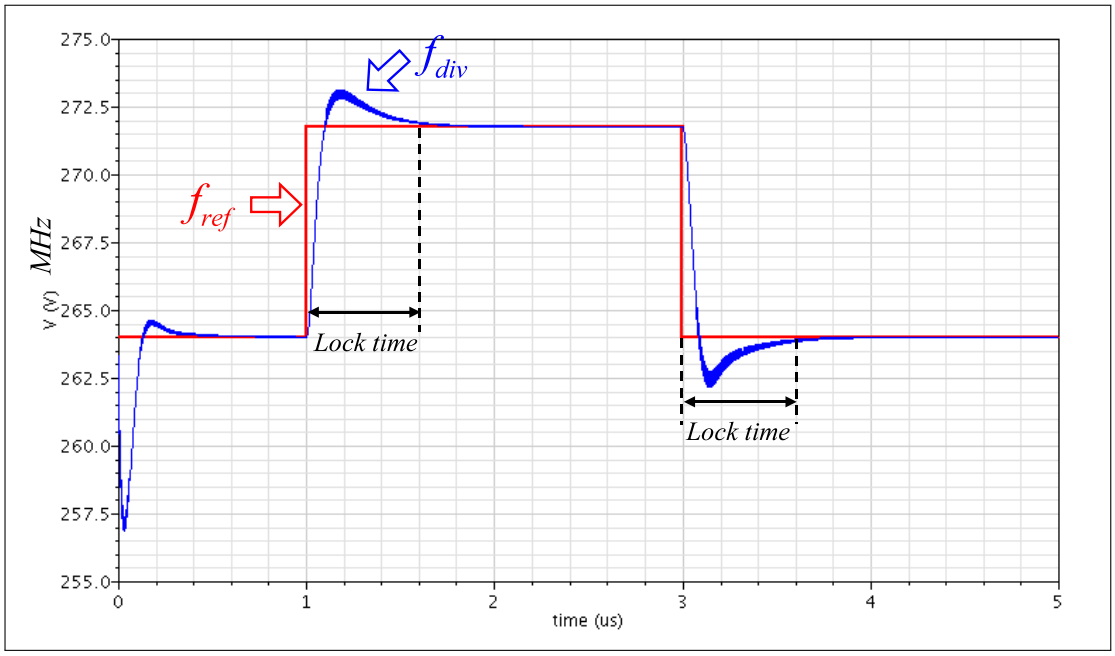


FIG. 4.9: *Lock time simulation: waveform of f_{ref} and f_{div} (First jump of f_{ref}: 264MHz → 271.75MHz, second jump of f_{ref}: 271.75MHz → 264MHz)*

downward to outside the hold range. Similarly, for the upper boundaries, $f_{ref}(t)$ is moved downward from outside into the capture range and then upward to outside the hold range. The boundaries are thus identified as the points from which $f_{div}(t)$ coincides to or depart from $f_{ref}(t)$ within a given tolerance f_{tol}. Fig. 4.11 and Fig. 4.12 show respectively the experiment for the lower and upper boundaries measurements for the PLL considered. It shows a capture range $[C1, C2] = [248MHz, 278.6MHz]$ and a hold range $[H1, H2] = [247.2MHz, 279.5MHz]$ for a tolerance 0.04%.

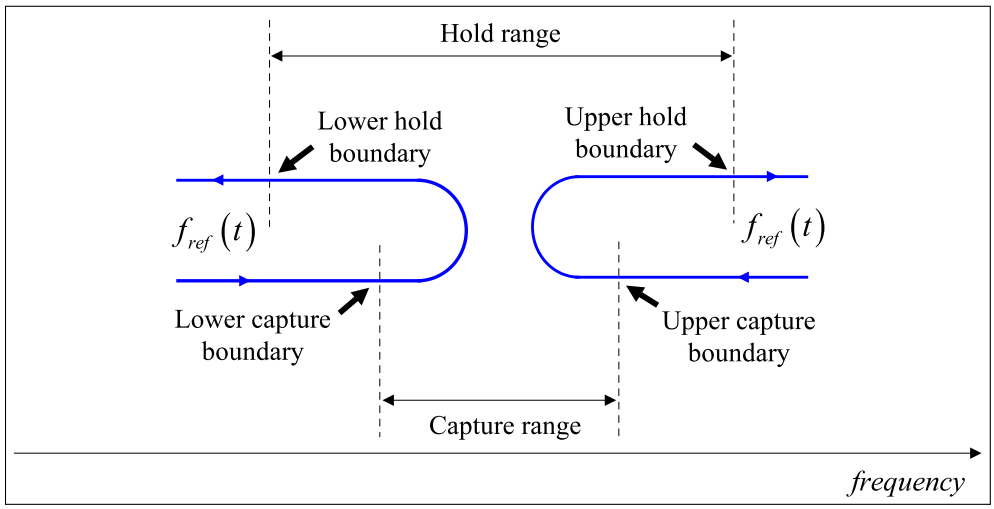


FIG. 4.10: *Sweep f_{ref} to measure capture range and hold range*

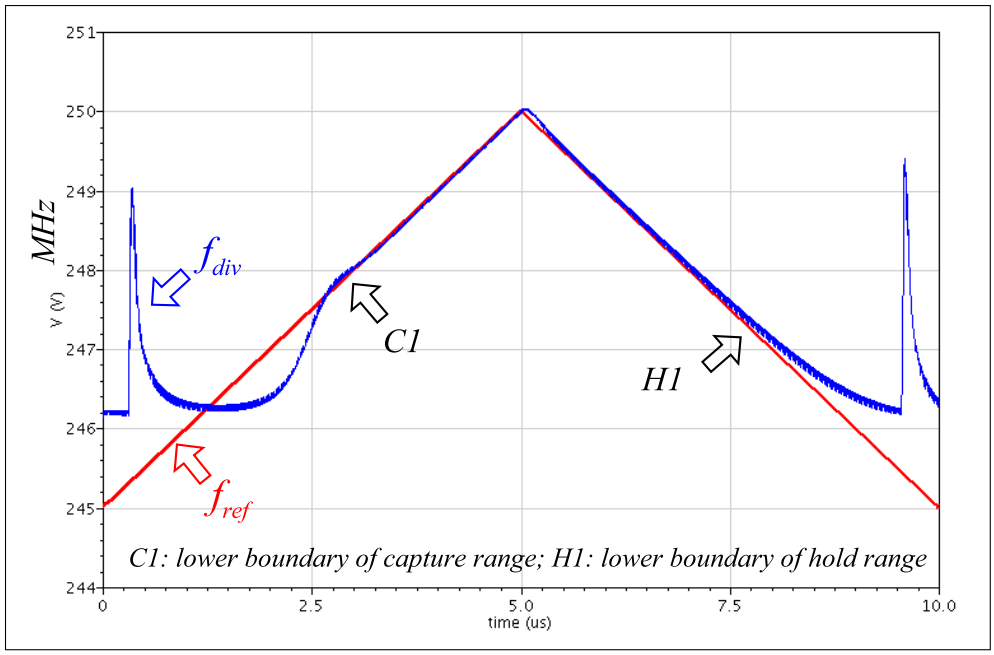


FIG. 4.11: *Lower boundaries of hold and capture range: (f_{ref} upward: 245MHz → 250MHz, f_{ref} downward: 250MHz → 245MHz) - C1: lower capture boundary, H1: lower hold boundary*

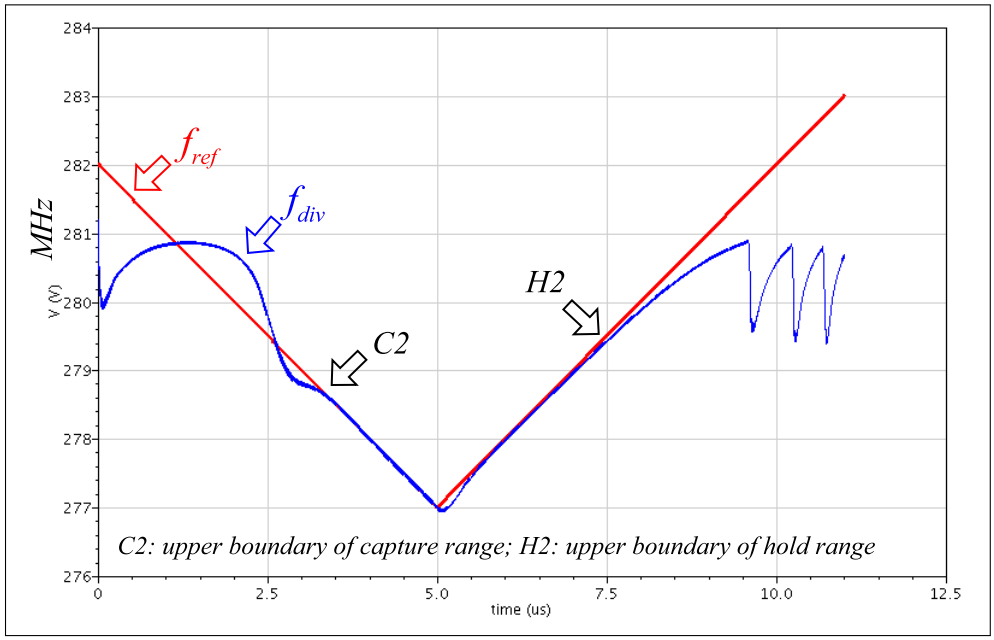


FIG. 4.12: *Upper boundaries of hold and capture range: (f_{ref} downward: f_{ref} 282MHz → 277MHz, f_{ref} upward: 277MHz → 283MHz) - C2: upper capture boundary, H2: upper hold boundary*

The HB simulation time necessary for the extraction of the VCO+divider model has been 18 min. The PLL simulation time with the proposed VCO+divider model is listed in Tab. 4.1. The table shows also the full PLL simulation at transistor-level under the same conditions. In comparison, the simulation with the block level model is about 15-20 times faster than a full transistor-level simulation. The simulation results match is very good, as will be shown in next section.

Simulation	Full transistor-level CPU time	Mixed transistor-level and VCO+divider model CPU time
Lock time	20 hours	1.4 hours
Hold/capt. range(lower)	40 hours	2.1 hours
Hold/capt. range(upper)	52 hours	2.5 hours

TAB. 4.1: *Simulation time comparison*

4.4.2 PFD+CP modeling

4.4.2.1 Model derivation

As shown in Fig. 4.13, the PFD and the CP are combined in a single block. The three ports signals are indicated in the figure.

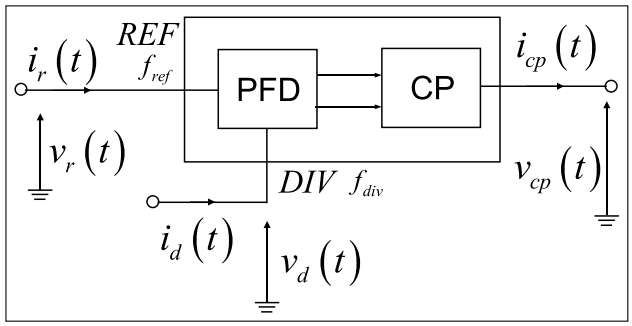


FIG. 4.13: *PFD+CP input and output signals*

To model the block we need to find a concise relationship between the ports currents and voltages, i.e.:

$$\psi\left(i_r\left(t\right), i_d\left(t\right), i_{cp}\left(t\right), v_r\left(t\right), v_d\left(t\right), v_{cp}\left(t\right), t\right) = 0 \tag{4.11}$$

In that purpose we will take consideration of the following specifics of the PFD operation in a PLL circuit.

The signals at the reference (REF) and divider (DIV) ports of the PFD are formed by a series of pulses, whose shape is practically invariable with time, and only the pulse

period and phase shift vary slowly with time. More precisely the signals may be expressed as follows:

$$\begin{aligned} v_r(t) &= \sum_k V_k^r exp\left(jk\int_0^t 2\pi f_{ref}(\tau)\,d\tau\right) = \sum_k V_k^r(k)exp\left(jk\phi_{ref}(t)\right) \\ i_r(t) &= \sum_k I_k^r exp\left(jk\int_0^t 2\pi f_{ref}(\tau)\,d\tau\right) = \sum_k I_k^r(k)exp\left(jk\phi_{ref}(t)\right) \\ v_d(t) &= \sum_k V_k^d exp\left(jk\int_0^t 2\pi f_{div}(\tau)\,d\tau\right) = \sum_k V_k^d(k)exp\left(jk\phi_{div}(t)\right) \\ i_d(t) &= \sum_k I_k^d exp\left(jk\int_0^t 2\pi f_{div}(\tau)\,d\tau\right) = \sum_k I_k^d(k)exp\left(jk\phi_{div}(t)\right) \end{aligned} \tag{4.12}$$

where the set of the pulses generating Fourier coefficients $\left\{V_k^r, I_k^r, V_k^d, I_k^d, k = 0, \pm1, ..., \pm K\right\}$ are obtained from the steady-state PLL analysis techniques of chapter 2.

From this observation, it is then apparent that as the dynamic analysis is concerned, the knowledge of the two instantaneous phase $\phi_{ref}(t)$ and $\phi_{div}(t)$ is sufficient to characterize the REF and DIV ports. Therefore, we have reduced the number of variable in equation (4.11) from 6 to 4. We need now to identify the relationship between $\phi_{ref}(t)$, $\phi_{div}(t)$, $v_{cp}(t)$ and $i_{cp}(t)$, i.e.,

$$\theta\left(\phi_{ref}(t), \phi_{div}(t), i_{cp}(t), v_{cp}(t), t\right) = 0 \tag{4.13}$$

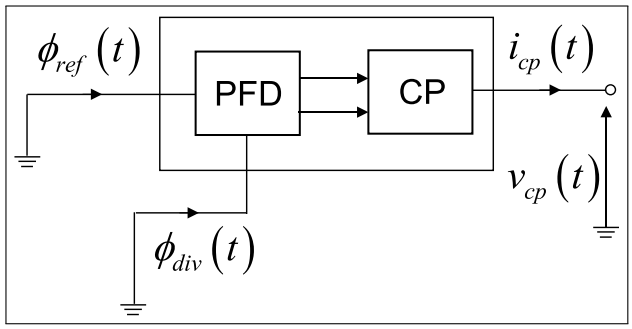


FIG. 4.14: *PFD+CP model*

The equivalent network representing equation (4.13) can then be sketched as in Fig. 4.14. Note that, similarly to the noise model approach in chapter 3, the CP port remains under the voltage/current formalism, while the REF and DIV ports are in the phase-domain formalism. Also note that in the phase-domain formalism the port is always shorted (no voltage drop). Equation (4.13) is better expressed in the form of the hybrid transfer characteristic below:

$$\begin{cases} i_{cp}(t) = \Gamma\left(\phi_{ref}(t), \phi_{div}(t), v_{cp}(t), t\right) \\ f_{ref}(t) = \dfrac{1}{2\pi}\dfrac{d\phi_{ref}(t)}{dt} \\ f_{div}(t) = \dfrac{1}{2\pi}\dfrac{d\phi_{div}(t)}{dt} \end{cases} \tag{4.14}$$

Equation (4.14) is a quasi-static characteristic that we will show the identification process in the next section.

4.4.2.2 Model identification

In the purpose of identifying the above model characteristic, let us observe some characteristics of the PFD+CP operation. We consider the following three experiments where the three state variables $f_{ref}(t)$, $f_{div}(t)$ and $v_{cp}(t)$ are taken constant:

$$\begin{cases} f_{ref}(t) = F_{ref}, \quad f_{div}(t) = F_{div}, \quad v_{cp}(t) = V_{cp} \\ F_{diff} = |F_{div} - F_{ref}| \end{cases} \tag{4.15}$$

1. Experiment 1: $F_{ref} < F_{div}$, with $F_{ref} = Q \cdot F_{diff},\ Q \in N$;
2. Experiment 2: $F_{ref} < F_{div}$, with $F_{ref} = \rho \cdot F_{diff},\ \rho \in \Re$;
3. Experiment 3: $F_{ref} > F_{div}$.

Experiment 1: $F_{ref} < F_{div}$, with $F_{ref} = Q \cdot F_{diff},\ Q \in N$

First, when $F_{ref} < F_{div}$ with the reference frequency being a multiple of the frequency difference, i.e., $F_{ref} = Q \cdot F_{diff},\ Q \in N$, the waveforms of the PFD input signals and the CP output current i_{cp} are as drawn in Fig. 4.15. It shows that in each time span of $1/F_{ref}$ there is one and only one pulse of i_{cp}, because the rising edge of i_{cp} aligns always with the rising edge of the signal REF. Now define T_1 as the time length of a basic segment (containing a logic 0 and a logic 1) of i_{cp}, thus $T_1 = 1/F_{ref}$. The pulse width of i_{cp} varies gradually from one time span of T_1 to the other T_1. Note that the waveform of i_{cp} repeats itself every time span T_2. If one regards the signal i_{cp} within a time span T_2 as an envelope, then it's easy to find $T_2 = 1/F_{diff}$. Hence for the case $F_{ref} = Q \cdot F_{diff},\ Q \in N$, i_{cp} is periodic with its period equal to T_2.

Actually, Fig. 4.15 is an idealized figure; the waveform obtained from a real transistor-level simulation of a PFD+CP circuit with $F_{ref} < F_{div}$ ($F_{ref} = 38MHz$, $F_{div} = 40MHz$, $F_{diff} = 2MHz$ and $F_{ref} = 19 \cdot F_{diff}$) is shown in Fig. 4.16. The waveform of the first envelope is identical with that of the second envelope.

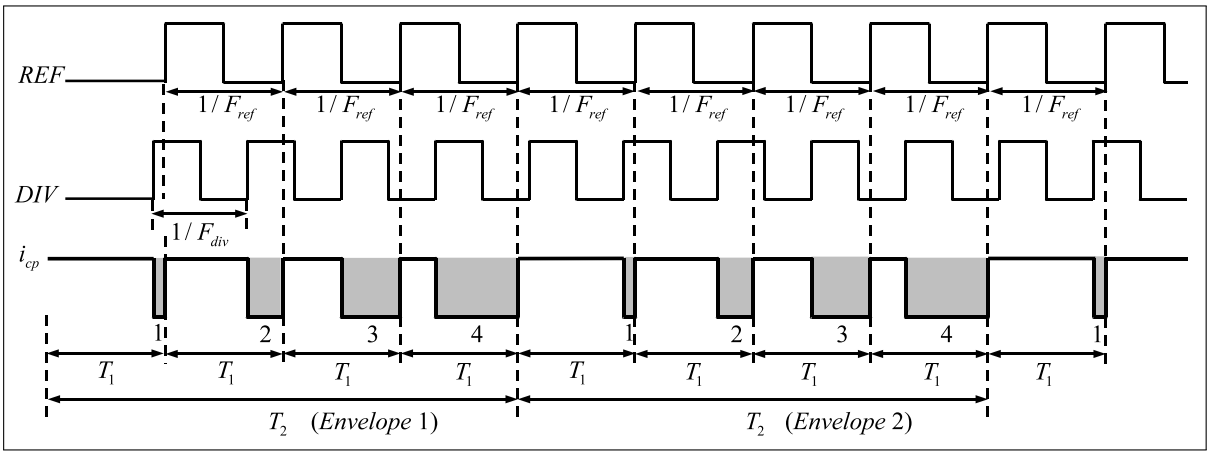


FIG. 4.15: *Input and output signals of PFD+CP:* $F_{ref} < F_{div}$, $F_{ref} = Q \cdot F_{diff}$ *with* $Q = 4$

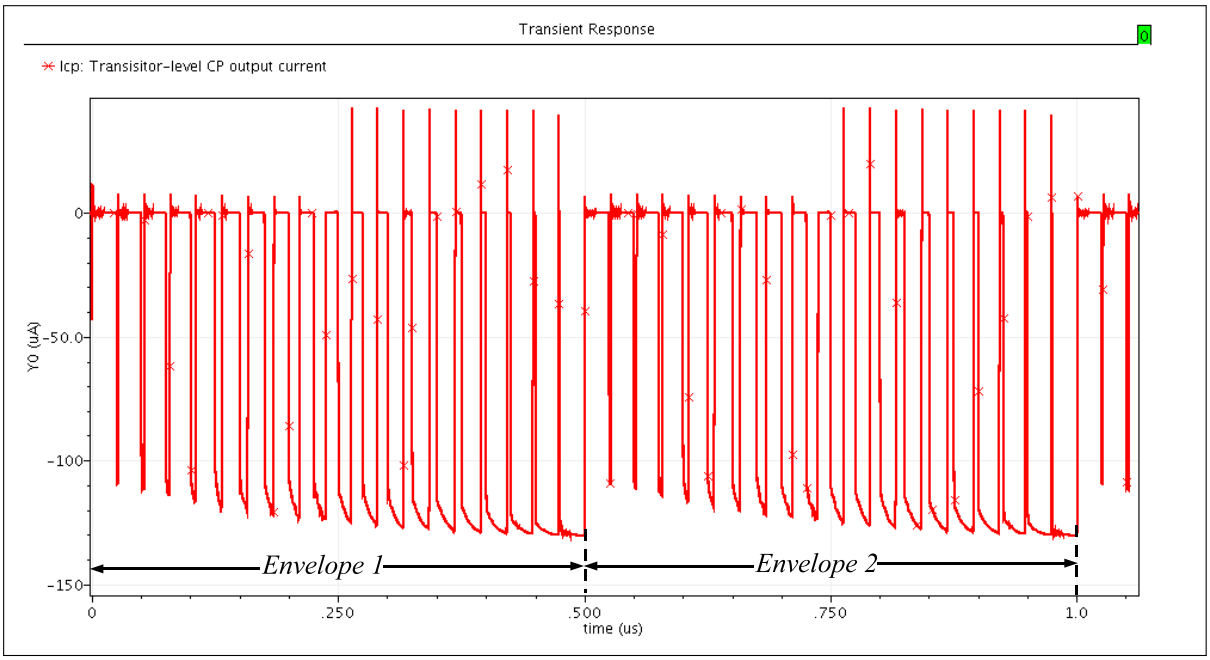


FIG. 4.16: *Waveform of* i_{cp}*:* $F_{ref} < F_{div}$, $F_{diff} = 2MHz$ *and* $F_{ref} = 19 \cdot F_{diff}$

Experiment 2: $F_{ref} < F_{div}$**, with** $F_{ref} = \rho \cdot F_{diff},\ \rho \in \Re$

If the reference frequency is not a multiple of the frequency difference, i.e., $F_{ref} = \rho \cdot F_{diff},\ \rho \in \Re$, the waveforms of the PFD input signals and the CP output current i_{cp} are drawn in Fig. 4.17. It shows also as previously that in each time span $1/F_{ref}$ there is one and only one pulse of i_{cp}, so the length of the basic segment of waveform is also $T_1 = 1/F_{ref}$. The waveform of i_{cp} does not repeat itself exactly in each time span of $1/F_{diff}$. However, i_{cp} still shows an envelope of $1/F_{diff}$, so we define the time length of the envelope as $T_2 = 1/F_{diff}$.

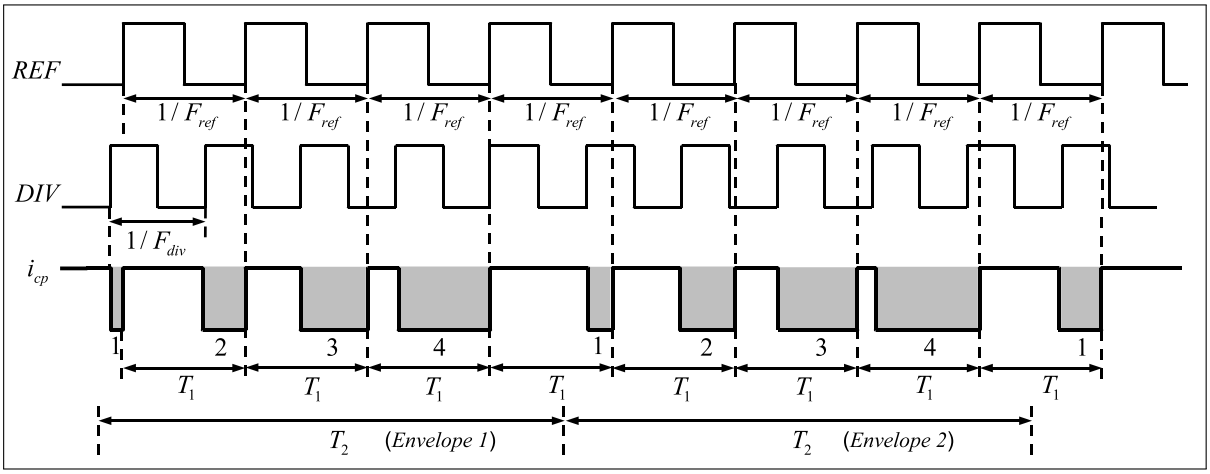


FIG. 4.17: *Input and output signals of PFD+CP:* $F_{ref} < F_{div}$, $F_{ref} = \rho \cdot F_{diff}$ *with* $\rho = 3.8$

In summary, from the above observation, an essential characteristic of the PFD+CP can be drawn as follows:

> *The PFD+CP performs like a mixer: whatever the PFD input frequencies are, the output current of CP* i_{cp} *takes the form of a modulated signal which consists of two time constants:* T_1 *(basic segment length of* i_{cp}*) and* T_2 *(envelope of* i_{cp}*).*

Accordingly, in the case where both the input frequencies and the output voltage v_{cp} are time-invariant, i_{cp} can be expressed like a modulated signal as below:

$$\begin{cases} i_{cp}(t) = \sum\limits_{k=0}^{K} \hat{i}_k^+(t)\, exp(2\pi jkF_{ref}t) \\ \hat{i}_k^+(t) = \sum\limits_{q=1}^{Q} I_{kq}^+(V_{cp})\, exp(2\pi jqF_{diff}t) \end{cases} \tag{4.16}$$

where $\hat{i}_k^+(t)$ represents the complex envelope containing the low-frequency component F_{diff}, and $I_{kq}^+(V_{cp})$ are the model kernels to be identified. These can be obtained from a two-tone HB simulation or shooting simulation.

To investigate the modulation characteristic of the PFD+CP, let's observe Fig. 4.18 which describes the acquisition process of the PLL. During the process, F_{diff} decreases with time, which makes the envelope of i_{cp} become longer and longer. It can be seen that each i_{cp} envelope segment coincides with a period of v_{cp}. Finally when the PLL is in the locked state ($F_{diff} \rightarrow 0$), the envelope segments of i_{cp} become infinitely long, and v_{cp} becomes constant.

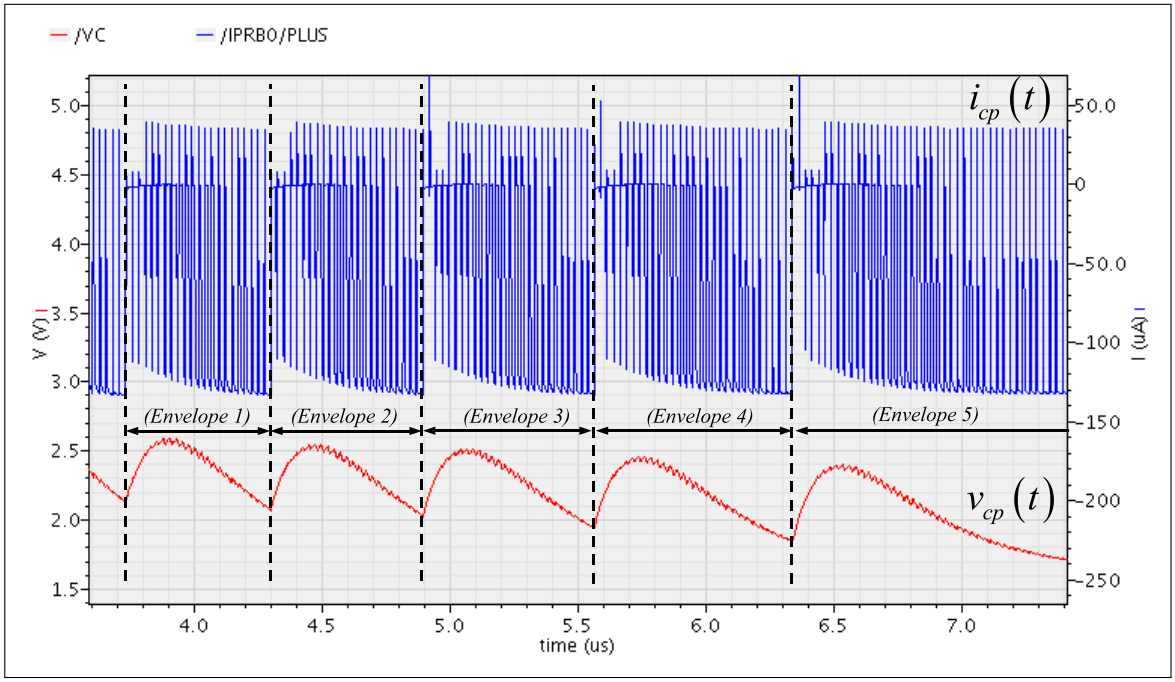


FIG. 4.18: *Evolution of the envelope of i_{cp} when the PLL approaches the locked state*

Based on the above observation, we thus can see that in the configuration of arbitrary time varying variables $f_{ref}(t)$, $f_{div}(t)$, $v_{cp}(t)$, the CP output current expression can be obtained as an extension of the equation (4.16), i.e:

$$\begin{cases} i_{cp}(t) = \sum_k \hat{i}_k^+(t)\, exp\left(jk\phi_{ref}(t)\right) \\ \hat{i}_k^+(t) = \sum_q I_{kq}^+\left(v_{cp}(t)\right) exp\left(jq\left(\phi_{div}(t) - \phi_{ref}(t)\right)\right) \\ \phi_{ref}(t) = \int_0^t 2\pi f_{ref}(\tau)\, d\tau \\ \phi_{div}(t) = \int_0^t 2\pi f_{div}(\tau)\, d\tau \end{cases} \tag{4.17}$$

Note that the kernel $I_{kq}^+(V_{cp})$ is independent of the frequencies F_{ref} and F_{div} where it has been characterized because of the assumption of slowly time varying phase shift (as compared to the PFD+CP time response). This assumption is well verified in practice.

Experiment 3: $F_{ref} > F_{div}$

In the above discussion, we have assumed $F_{ref} < F_{div}$. However, if $F_{ref} > F_{div}$, we can observe a different characteristic of the PFD+CP, which is shown in Fig. 4.19.

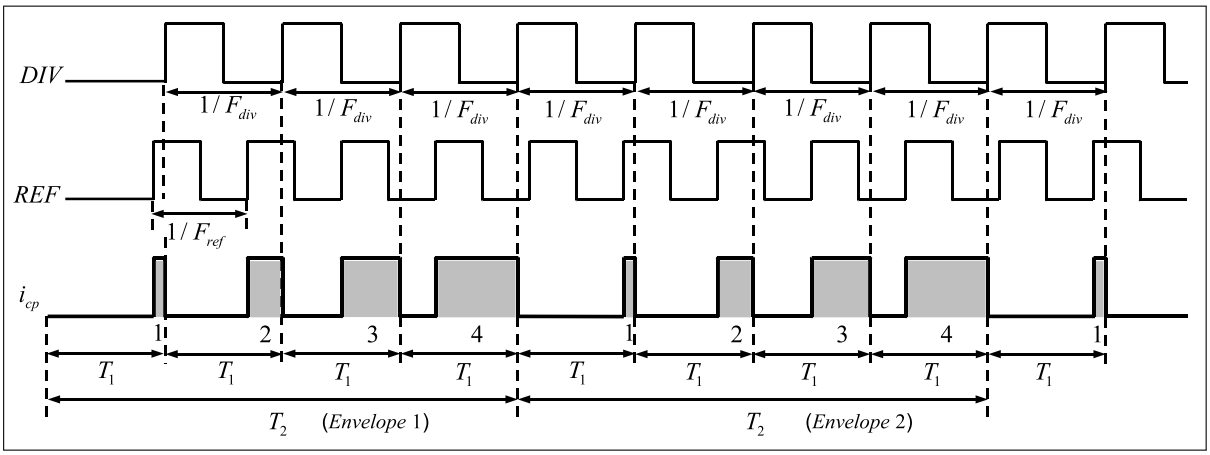


FIG. 4.19: *Input and output signals of PFD+CP:* $F_{ref} > F_{div}$, $F_{div} = Q \cdot F_{diff}$ *with* $Q = 4$

We can see that the falling edge of i_{cp} aligns with the rising edge of the signal DIV, hence in each time span of $1/F_{div}$ there is one and only one pulse of i_{cp}. Since T_1 is defined as the basic segment of i_{cp}, we have $T_1 = 1/F_{div}$. The time length of the envelope is still T_2, with $T_2 = 1/F_{diff}$. The above definitions are applicable to either $F_{div} = Q \cdot F_{diff},\ Q \in N$ or $F_{div} = \rho \cdot F_{diff},\ \rho \in \Re$. Therefore, by extension of equation (4.17), for the case $F_{ref} > F_{div}$, the characteristic of the PFD+CP writes:

$$\begin{cases} i_{cp}(t) = \sum\limits_{k} \hat{i}_k^-(t)\, exp\left(jk\phi_{div}(t)\right) \\ \hat{i}_k^-(t) = \sum\limits_{q} I_{kq}^-\left(v_{cp}(t)\right) exp\left(jq\left(\phi_{ref}(t) - \phi_{div}(t)\right)\right) \\ \phi_{ref}(t) = \int_0^t 2\pi f_{ref}(\tau)\, d\tau \\ \phi_{div}(t) = \int_0^t 2\pi f_{div}(\tau)\, d\tau \end{cases} \tag{4.18}$$

where $\hat{i}_k^-(t)$ is the complex envelope, and $\hat{I}_{kq}^-$ its Fourier coefficients which can be obtained by a two-tone simulation.

Finally, taking account of equations (4.17) and (4.18), the general expression of the proposed PFD+CP model is:

$$\begin{cases} i_{cp}(t) = \sum_k \hat{i}_k(t)\, exp\left(jk\phi_{ref}(t)\right) \\ \hat{i}_k(t) = \sum_q I_{kq}\left(|\Delta\phi(t)|, v_{cp}(t)\right) exp\left(jq\,|\Delta\phi(t)|\right) \\ I_{kq}\left(\Delta\phi(t), v_{cp}(t)\right) = \frac{1}{2}(1 + sign(\frac{d\Delta\phi(t)}{dt}))I_{kq}^{+}(v_{cp}(t)) \\ \qquad\qquad + \frac{1}{2}(1 - sign(\frac{d\Delta\phi(t)}{dt}))I_{kq}^{-}(v_{cp}(t)) \\ \Delta\phi(t) = \phi_{div}(t) - \phi_{ref}(t) \\ f_{ref}(t) = \frac{1}{2\pi}\frac{d\phi_{ref}(t)}{dt} \\ f_{div}(t) = \frac{1}{2\pi}\frac{d\phi_{div}(t)}{dt} \end{cases} \tag{4.19}$$

The model identification bench is as sketched in Fig. 4.20 and Fig. 4.21:

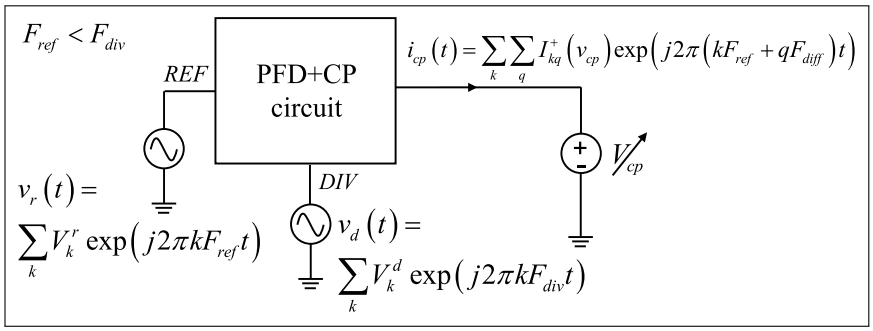


FIG. 4.20: *Model identification bench:* $F_{ref} < F_{div}$

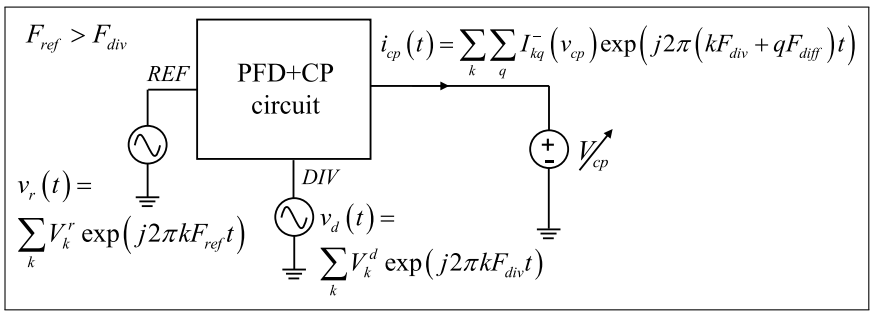


FIG. 4.21: *Model identification bench:* $F_{ref} > F_{div}$

4.4.2.3 Model implementation and verification

Once the characteristics $I_{kq}^{+}(v_{cp})$ and $I_{kq}^{-}(v_{cp})$ are acquired from a two-tone steady-state simulation, the model expression can be implemented either in C code or in Verilog-A using polynomial or spline interpolation.

The schematic of 4.22 has been then used to verify the model. The frequency values of F_{ref} and F_{div} used in this verification are different from the ones used to extract the model. The CP output is set to a DC voltage.

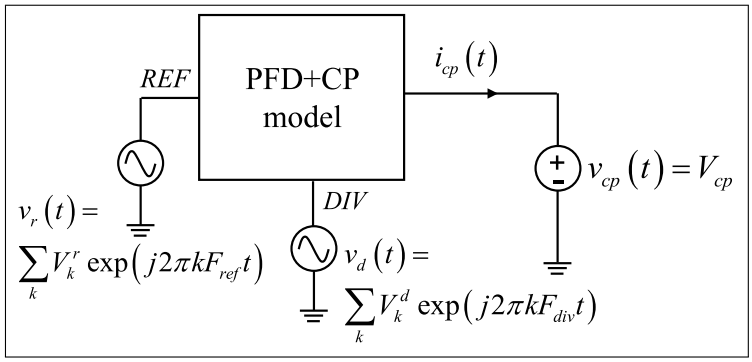


FIG. 4.22: *Schematic to verify the PFD+CP model*

In a first experiment, F_{ref} and F_{div} are set to commensurate values $F_{ref} = 38MHz$ and $F_{div} = 40MHz$ ($F_{ref} = 19F_{diff}$). Fig. 4.23 compares the model output with the transistor-level simulation result. The zoom figure Fig. 4.24 shows that the two results fit very well. The ringing on the PFD+CP model output current is due to the limited number of harmonics used to create the model. The ringing can be reduced by increasing the number of harmonics but at the price of slowing down the simulation speed. Note however that the ringing is not critical to the PLL simulation accuracy as they are filtered out by the LPF.

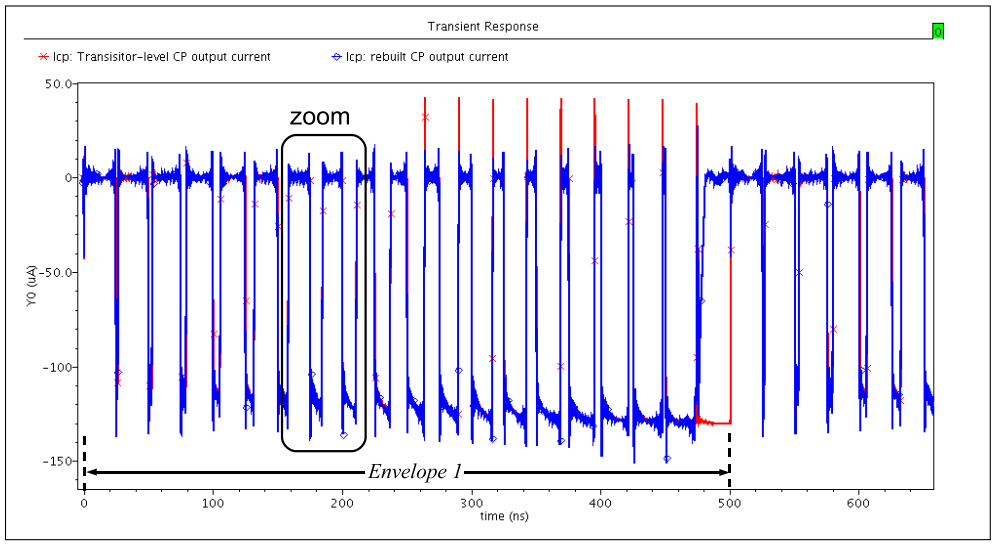


FIG. 4.23: *i_{cp} : Comparison between PFD+CP model simulation and PFD+CP circuit simulation, $f_{ref} = 38MHz$, $f_{div} = 40MHz$*

In a second experiment, we have considered two identical frequencies $F_{ref} = F_{div} = 38MHz$. The output current is periodic with period 26.3 ns. There is also a very good match with the transistor-level simulation shown in Fig. 4.25.

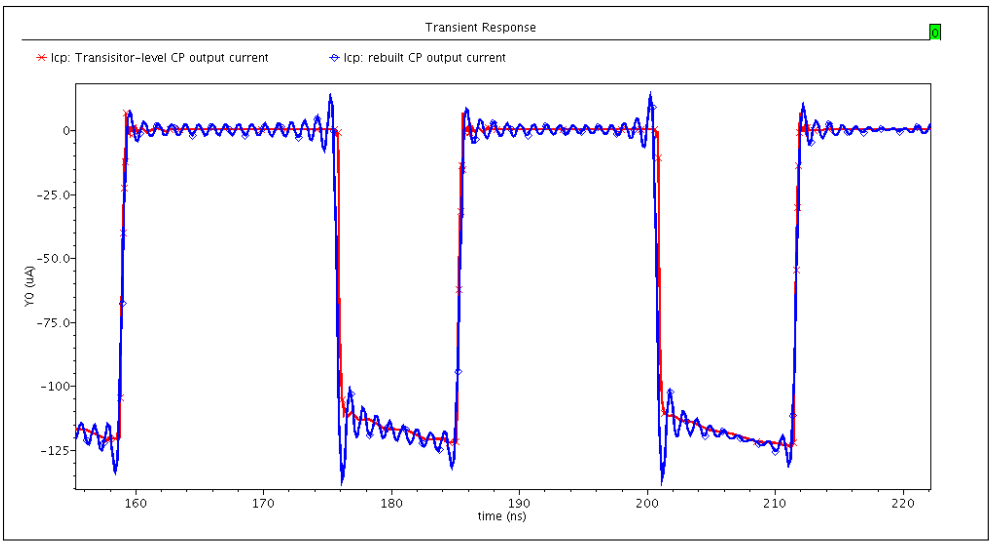


FIG. 4.24: *Zoom of Fig. 4.23*

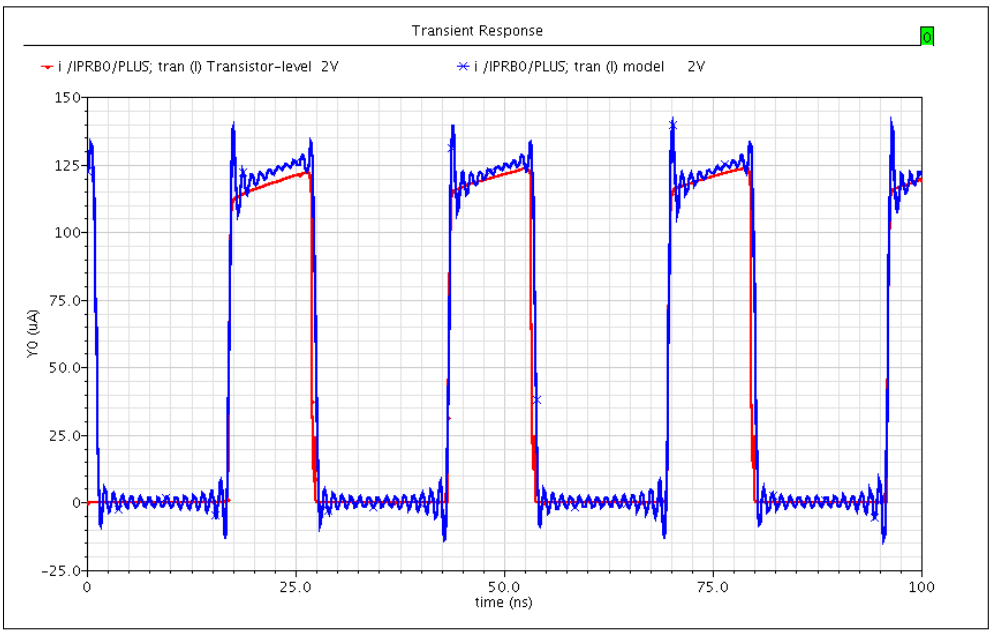


FIG. 4.25: i_{cp}*: Comparison between PFD+CP model simulation and PFD+CP circuit simulation,* $f_{ref} = f_{div} = 38MHz$

4.4.2.4 Numerical application of proposed modeling approach

In this section we will carry a full PLL simulation in three configurations:

- CF1: full transistor-level simulation
- CF2: Mixed transistor-level (PFD+CP+LPF) and block level (VCO+divider) (Fig. 4.8)
- CF3: Full block level (Fig. 4.26)

The simulations have been carried with the same PLL as in the preceding section.

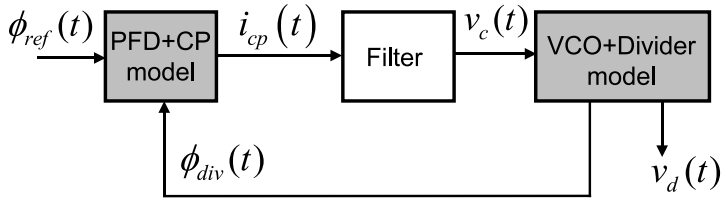


FIG. 4.26: *PLL simulation with the configuration CF3*

Fig. 4.27 shows the result for the lock time simulation, we observe a good match between the three configurations.

Fig. 4.29 and Fig. 4.28 show the results for the hold and capture ranges. There is also a good match between the three methods. The range boundaries obtained with the three methods are identical. The results however diverge slightly outside the hold range; the explanation could be the differences in time domain integration error control in the simulation engine when the distance between f_{ref} and f_{div} increases.

Comparison of the simulation time

Tab. 4.2 lists the simulation time for the three configurations. It shows a speed-up of many orders using the configurations CF2 and CF3.

Simulation	CF1: Full transistor level	CF2: Mixed transistor / block-level	CF3: Full block level
Lock time	20 hours	1.4 hours	1.9 min
Hold/capt. range(lower)	40 hours	2.1 hours	2.7 min
Hold/capt. range(upper)	52 hours	2.5 hours	3.4 min

TAB. 4.2: *Simulation time comparison between three methods*

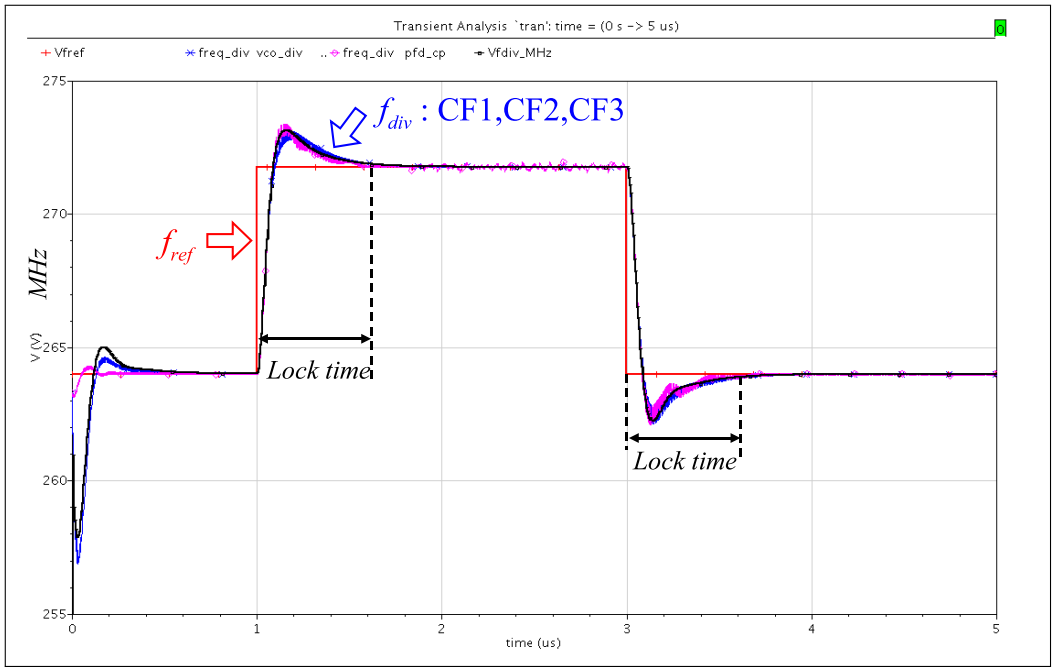


FIG. 4.27: *Lock time simulation - CF1: full transistor-level; CF2: Mixed transistor / block-level; CF3: Full block level*

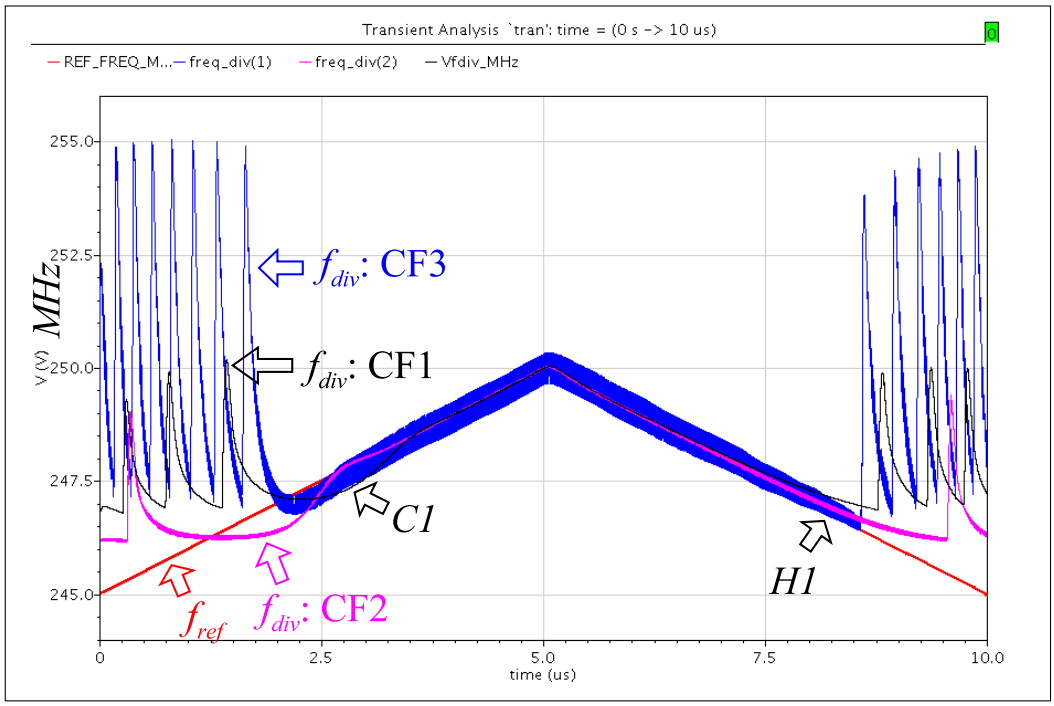


FIG. 4.28: *Lower boundaries of hold and capture range - CF1: full transistor-level; CF2: Mixed transistor / block-level; CF3: Full block level; C1: lower capture boundary, H1: lower hold boundary*

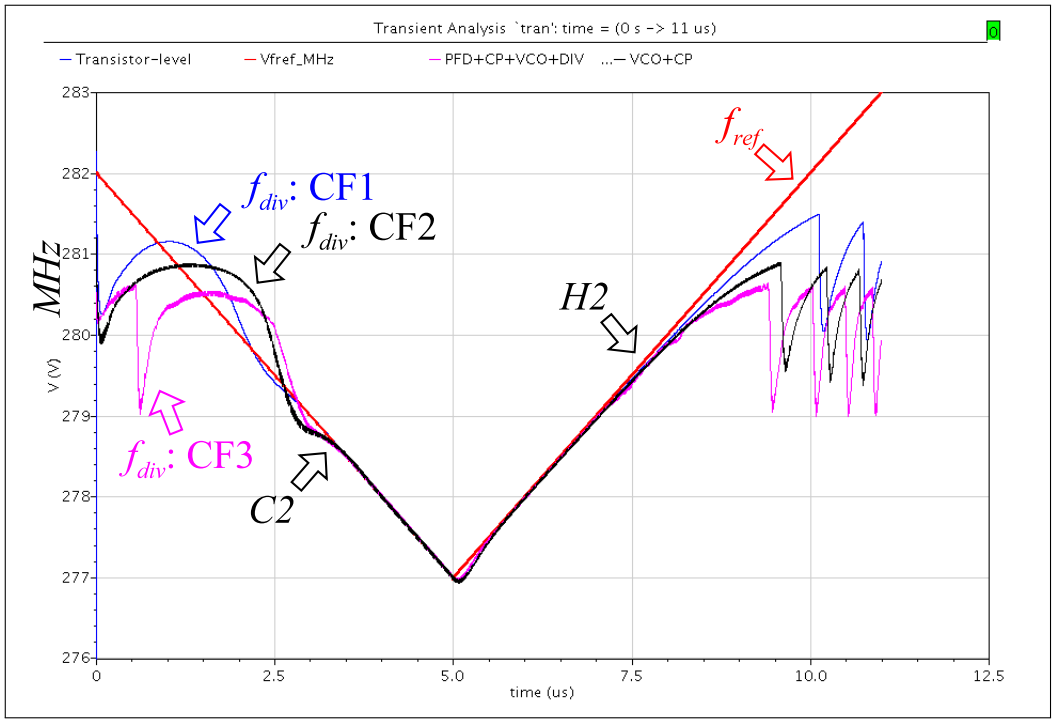


FIG. 4.29: *Upper boundaries of hold and capture range - CF1: full transistor-level; CF2: Mixed transistor / block-level; CF3: Full block level; C2: upper capture boundary, H2: upper hold boundary*

4.5 Conclusion

In this chapter, we have dealt with the simulation problems concerning the dynamic performances of the PLL. First, some concepts of the dynamic performances of the PLL are reviewed. It shows that a simulation of the PLL is usually necessary to predict and verify the dynamic performances of the PLL. To avoid the difficulties in the transistor-level simulation, the block-level simulation is needed to reduce the simulation time.

After reviewing existing simulation approaches, we have proposed block level models for the VCO+divider and the PFD+CP. These models rely on the availability of an accurate PLL steady state analysis tool. The proposed models for either VCO+divider or PFD+CP provide in an explicit way both voltage domain (i.e., voltage or current waveform) and phase domain (i.e., time varying phase and frequency) information. They therefore allow the block-level simulation to contain enough details on the dynamics of the PLL. It is shown that the accuracy of the results obtained is very close to the transistor-level simulation, and the simulation time is reduced by one or more orders of magnitude. The proposed modeling approach has made it possible to simulate thoroughly and accurately the dynamics characteristics (lock time, capture range and hold range) of a number of the PLL circuits within minutes of simulation time where this would need hours or days with the transistor-level simulation.

4.6 Bibliography

[1] R. E. Best, *Phase-Locked Loops: Design, Simulation, and Applications*, 5th ed. McGraw-Hill Professional, 2003.

[2] K. Shu and E. Sánchez-Sinencio, *CMOS PLL Synthesizers: Analysis and Design*, 1st ed. Springer, Nov. 2004.

[3] C. S. Vaucher, "An adaptive PLL tuning system architecture combining high spectralpurity and fast settling time," *IEEE Journal of Solid-State Circuits*, vol. 35, pp. 490–502, 2000.

[4] J. Chen, "An introduction to the PLL library," Cadence Design Systems, Tech. Rep. [Online]. Available: www.cadence.com/whitepapers/pllapp_note.pdf

[5] M. H. Perrott, "Fast and accurate behavioral simulation of fractional-N frequency synthesizers and other c/DLL circuits," *Proceedings of the 39th conference on Design automation*, pp. 498–503, 2002.

[6] M. Hinz, I. Konenkamp, and E.-H. Horneber, "Behavioral modeling and simulation of phase-locked loops for RF front ends," *Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems*, vol. 1, pp. 194–197 vol.1, 2000.

[7] S. Huang, H. Ma, and Z. Wang, "Modeling and simulation to the design of Sigma-Delta fractional-N frequency synthesizer," *Proceedings of the conference on Design, Automation and Test in Europe*, pp. 291–296, 2007.

[8] A. Demir, "Analysis and simulation of noise in nonlinear electronic circuits and systems," Ph.D. dissertation, University of California, 1997.

[9] A. Layec, "Développement de modèles de CAO pour la simulation système des systèmes de communication. Application aux communications chaotiques," Ph.D. dissertation, Université de Limoges, 2006.

General conclusion

In recent years, the design of the PLL circuits has experienced a rapid growth: new circuit topologies are proposed and new implementation technologies are adopted. However, there is always a lack of an efficient simulation tool which allows the PLL designers to analyze the critical characteristics of the PLL accurately and quickly. For this purpose, the CAD engineers have taken great effort trying to provide efficient solutions to the analysis of these PLL characteristics. Nevertheless, compared with the advances in PLL design techniques, there exists a lag in PLL simulation and verification techniques which is principally caused by the complexity of the PLL characteristics and lack of efficient modeling method.

This thesis has been dedicated to the study of efficient solutions for the PLL simulation and verification, and proposes new algorithms and modeling methods to calculate the steady state response and predict the noise as well as dynamic performances of the PLL. The main contributions of this thesis are summarized as follows:

- We have brought forward an efficient algorithm to compute the steady-state response of the integer-N PLL. The proposed algorithm, totally at the transistor-level, allows calculating the close-loop steady-state of the PLL very efficiently. The simulation converges within very few iterations with a very short simulation time. The simulation results are compared with the brute-force transistor-level simulation, and a good match is found. The proposed algorithm offers an attractive choice for PLL designers to verify efficiently the steady state response of the PLL.

- We have proposed models of the PLL building blocks to calculate the phase noise and the deterministic noise of the PLL. Non-idealities of the PLL are taken into account, such as the output impedance of the charge pump, input impedance of the VCO, etc. The simulation time lies principally on the block modeling and the PLL noise simulation is almost instantaneous. Hence as many frequency points as necessary can be considered in the phase noise plot. Small perturbation stability analysis can also be carried with these models.

- We have brought new VCO+divider and PFD+CP models to analyze the dynamic performances of the PLL. These two models provide explicitly the PLL dynamics details both in voltage domain and in phase domain. The simulation with the new proposed models can be accomplished within few minutes compared with hours and days for the conventional transistor-level simulation. The proposed VCO+divider and PFD+CP models offer the designers more flexibilities for the design verification with mixed transistor-block-level or full block-level simulation. The models can achieve a very fast simulation, which is very suitable for the sensitivity analysis about the temperature or the supply voltage.

Perspectives:

Although the simulation results obtained from the proposed methods have been very encouraging, the simulation processes, however, are carried manually for the moment. It is necessary to identify the PLL blocks, the stimulus and the probes at the interfaces, netlist them separately, set the simulation control parameters, then run the block simulations one by one, record the various results, gather them together, and finally generate Verilog-A models and run noise and dynamic performances analysis. All these manipulations are tedious and very time-consuming. Therefore, the perspective of this thesis is to automate the proposed algorithm and model extraction procedures so that the simulation process can be more easily carried. The users only need to provide some essential information of the PLL (such as input/output ports of the blocks, nominal PLL output frequency and division rank), and the rest of simulation work will be done automatically by a carefully defined simulation process. This automation process is expected to be integrated into a commercial RF simulator where a full verification of the PLL design can be carried efficiently.

For limited time reason, we have not been able to consider the problem of factional-N PLLs; the extension of the methods proposed in this work to factional-N PLLs is also one of the perspectives of this thesis.

Appendix $: Previous work on VCO noise modeling

Appendix $
Previous work on VCO noise modeling

A.1 Phase noise of oscillator

As known, the noise at the output of the oscillator is mainly the phase noise, which can not be removed by a filter or amplitude limiter [1]. Hence it has an important influence on the communication system.

In general, the oscillator output voltage can be expressed by:

$$V_{out}(t) = A(t) \cdot f[\omega_0 t + \phi(t)] \tag{A.1}$$

where $A(t)$ and $\phi(t)$ are functions of time. Due to the amplitude and phase perturbation, the spectrum of output voltage exhibits a sideband around the oscillation carrier ω_0, as illustrated in Fig. A.1.

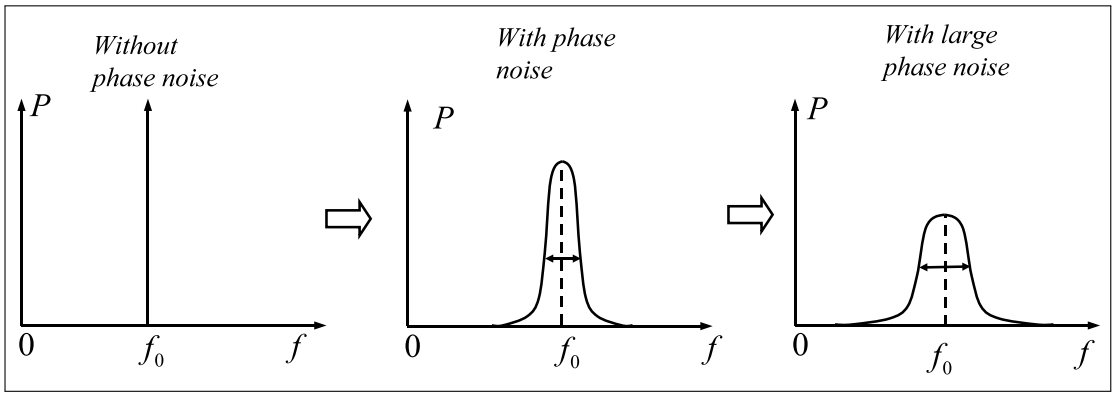


FIG. A.1: *Influence of the phase noise on the output spectrum of the oscillator*

In fact, the phase noise does not change the total signal power. What changes is only the distribution of signal power [2]. If the phase noise becomes larger, the side band spreads wider with a lower peak.

The phase noise can greatly degrade the performance of the receiver, especially when there is a large interference near the useful signal, as shown in Fig. A.2. With a noiseless

LO (local oscillator), the interfere near the useful signal can be easily removed by a low-pass filter. However, with the existence of the phase noise in the LO, it's hard to eliminate it with a filter, and the performance of the system is degraded.

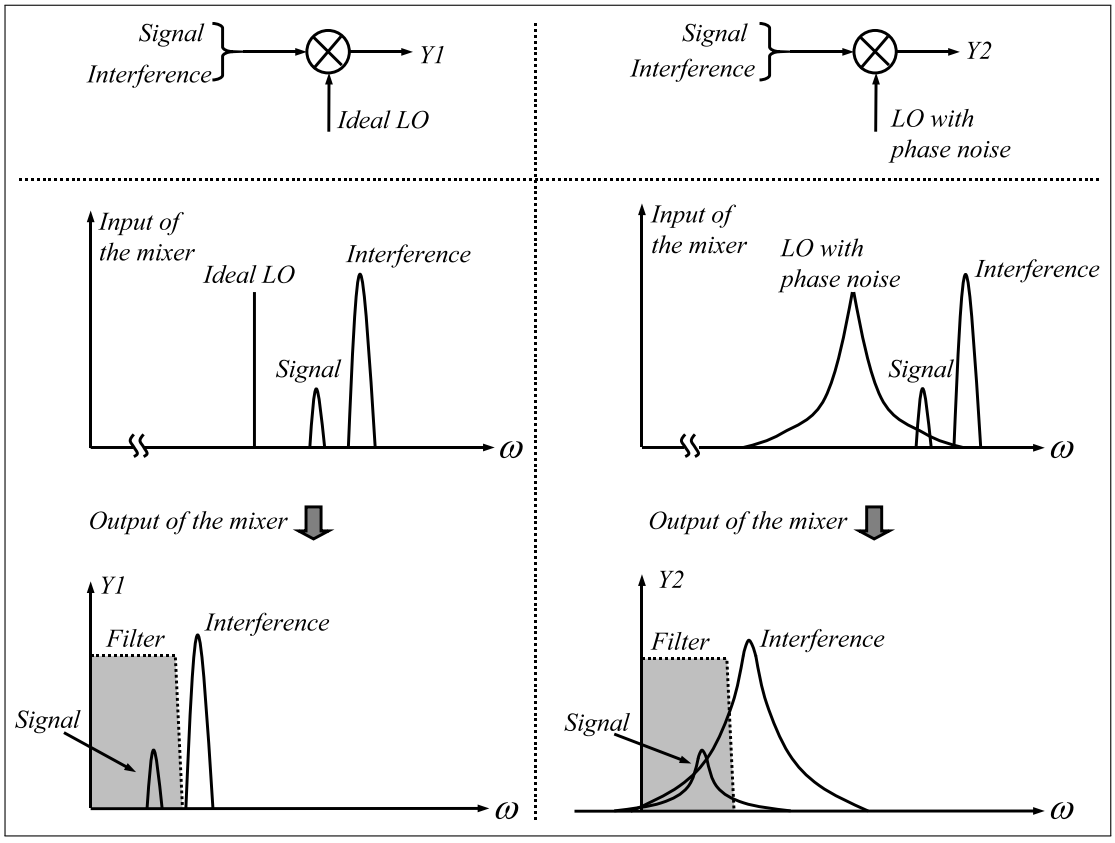


FIG. A.2: *Influence of the phase noise on the performance of the receiver*

Next, we briefly review the impulse response of the oscillator and the noise representation of the oscillator.

A.1.1 Impulse response to perturbation

Being autonomous circuit, the oscillator has some special characteristics, such as its impulse response to the perturbation.

1. *If the oscillator is disturbed in phase, the resulting phase deviation persists indefinitely* [1]. For an autonomous circuit, any time-shifted version of solution is the also a solution. In others words, if the phase of the oscillator is shifted, the oscillator continues to work as if nothing was disturbed except the phase shift. Therefore, there is no means to correct the phase deviation, and the phase error may accumulate continuously. This phase accumulation can be modeled by a step function, as shown in Fig. A.3, where $i(t)$ is the perturbation, and $\phi(t)$ is the resulting phase deviation.

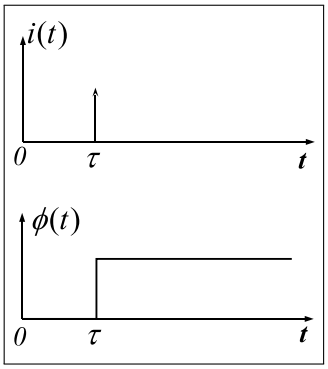


FIG. A.3: *Impulse response of the phase*

2. *If an oscillator is disturbed in amplitude, the amplitude deviation will disappear along with time* [1]. Actually, there is a limitation mechanism in the oscillator which tends to react against the amplitude variation. If we trace two state variables of the oscillator, we get a state space of oscillator represented by a closed-loop path, named the limit cycle, as illustrated in Fig. A.4. All the fluctuations tending to perturb the amplitude of the oscillator will be attenuated by this mechanism, which is a natural consequence of non-linearity of the oscillator.

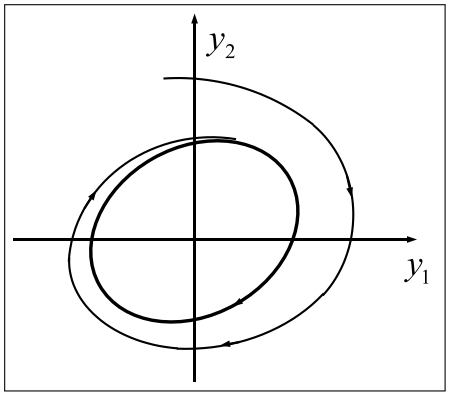


FIG. A.4: *Limit cycle of the oscillator*

As mentioned above, the phase accumulation may be modeled by a step function $s(\tau)$. Thus, the phase shift $\phi(t)$ due to the random perturbation u can be written as: [3]

$$\phi(t) \sim \int_{-\infty}^{+\infty} s(t-\tau)\, u(\tau)\, d\tau = \int_{-\infty}^{t} u(\tau)\, d\tau \tag{A.2}$$

and the power spectral density of the phase is given by:

$$S_\phi(\Delta f) \sim \frac{S_u(\Delta f)}{(2\pi\Delta f)^2} \tag{A.3}$$

Equation (A.3) shows that near carrier frequency (i.e. $\Delta f \rightarrow 0$), the phase error tends to infinity. Assuming that the perturbation u is constituted by white noise and 1/f noise, thus its PSD is given by:

$$S_u(\Delta f) \sim 1 + \frac{f_0}{\Delta f} \tag{A.4}$$

where f_0 is the oscillation frequency. Accordingly, the equation (A.3) becomes:

$$S_\phi(\Delta f) \sim n\left(\frac{1}{\Delta f^2} + \frac{f_c}{\Delta f^3}\right) \tag{A.5}$$

where n is a constant. This expression can explain the conversion of the white noise and the 1/f noise: the white noise source is converted in the phase noise with the slope $1/f^2$, while the 1/f noise source is converted into the phase noise with the slope $1/f^3$.

A.1.2 Noise representation

In fact, there are several ways to describe the phase noise of the oscillator:

- $S_\phi(\Delta\omega)$: spectral density of phase (unit: rad^2/Hz)
- $S_v(\Delta\omega)$: spectral density of voltage (unit: V^2/Hz)
- $\mathcal{L}(\Delta\omega)$: normalized spectral density of voltage: ratio between $S_v(\Delta\omega)$ and the carrier power (unit: dBc/Hz)

For uniformity of the definition, $\mathcal{L}(\Delta\omega)$, $S_v(\Delta\omega)$ and $S_\phi(\Delta\omega)$ are defined as the PSD with single sideband.

$\mathcal{L}(\Delta\omega)$ is widely used since it is easy to measure with a spectrum analyzer. $\mathcal{L}(\Delta\omega)$ is defined by the ratio between the carrier power and the single sideband power at a offset frequency $\Delta\omega$ from the carrier frequency within a bandwidth of 1Hz, as shown in Fig. A.5.

$$\mathcal{L}(\Delta\omega) = 10 \cdot \log\left[\frac{P_{sideband}(\omega_0 + \Delta\omega, 1Hz)}{P_{carrier}}\right] \tag{A.6}$$

In fact, $\mathcal{L}(\Delta\omega)$ contains the amplitude and phase variation. But for an oscillator, the phase deviation is much larger than the amplitude deviation, so $\mathcal{L}(\Delta\omega)$ is often used to describe the phase noise. As $\mathcal{L}(\Delta\omega)$ represents a ratio between the two power, and it is measured in a bandwidth of 1 Hz, the unit of $\mathcal{L}(\Delta\omega)$ is dBc/Hz. (dBc: dB relative to carrier)

Concerning $P_{sideband}(\omega_0 + \Delta\omega, 1Hz)$, it is equal to the voltage density $S_v(\Delta\omega)$. Similarly, $S_v(\Delta\omega)$ contains the amplitude and phase variation. The relationship between

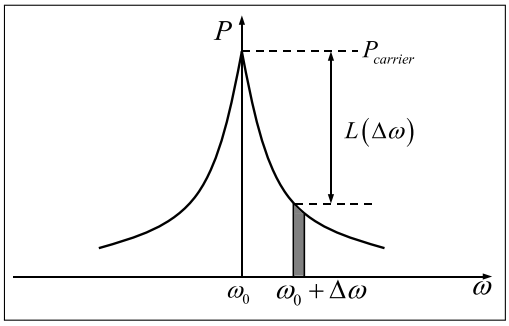


FIG. A.5: *Definition of* $\mathcal{L}(\Delta\omega)$

$S_v(\Delta\omega)$ and $\mathcal{L}(\Delta\omega)$ is:

$$\mathcal{L}(\Delta\omega) = \frac{S_v(\Delta\omega)}{P_{carrier}} \tag{A.7}$$

Unlike $\mathcal{L}(\Delta\omega)$ and $S_v(\Delta\omega)$, the phase spectral density $S_\phi(\Delta\omega)$ can not be observed with a spectrum analyzer, because it characterizes the phase variation instead of the voltage variation. As mentioned above, the phase variation is a random process and is described by the autocorrelation, which resembles a power. That is why the unit of $S_\phi(\Delta\omega)$ is rad^2/Hz.

The relation between $\mathcal{L}(\Delta\omega)$ and $S_\phi(\Delta\omega)$ has been proved as below: [4]

$$\mathcal{L}(\Delta\omega) = \frac{1}{2} S_\phi(\Delta\omega) \tag{A.8}$$

A.2 Existing models

To compute the noise sources in the VCO, one uses either the noise analysis methods presented in chapter 3 / section 3.3 or the empirical models / semi-empirical models described below. The parameters in these models could be estimated by a rule of thumb or experimented by a circuit simulation.

A.2.1 Leeson model

Leeson model [5] is a widely used empirical model based on a LTI (Linear Time Invariant) assumption for the oscillator. Leeson model shows that the phase noise distribution is composed of three parts: a part where the phase noise is a function of $1/f^3$, a part where the phase noise is a function of $1/f^2$, and a part where the phase noise is independent of frequency, as shown in Fig. A.6. Leeson model shows that the phase noise is inversely proportional with the average power dissipation P_S and the loaded quality factor Q_L. It

predicts the phase noise of the oscillator as follows:

$$\mathcal{L}\{\Delta\omega\} = 10 \cdot \log\left\{\frac{2FkT}{P_S} \cdot \left[1 + \left(\frac{\omega_0}{2Q_L\Delta\omega}\right)^2\right] \cdot \left(1 + \frac{\Delta\omega_{1/f3}}{\Delta\omega}\right)\right\} \tag{A.9}$$

- F is an empirical parameter.
- k is the Boltzmann constant.
- T is the absolute temperature.
- P_S is the average power dissipated in the resistive part of resonator.
- ω_0 is the oscillation frequency.
- Q_L is the effective quality factor of the resonator (also called loaded Q).
- $\Delta\omega$ is the offset frequency from the carrier.
- $\Delta\omega_{1/f^3}$ is the corner frequency between $1/f^3$ and $1/f^2$ region.

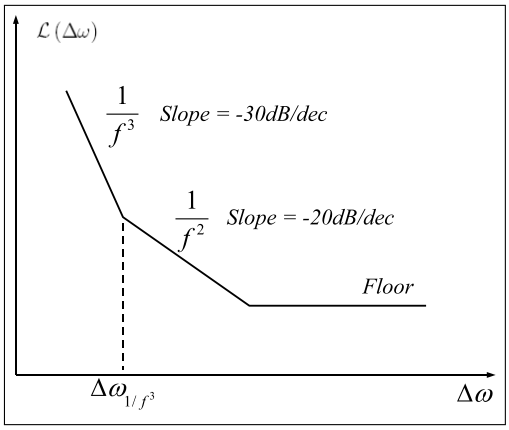


FIG. A.6: *Typical curve of the oscillator phase noise predicted by Leeson model*

However, It is not obvious to estimate the empirical parameters F and $\Delta\omega_{1/f^3}$ in the Leeson model. If these empirical parameters are not properly estimated, it may cause large errors.

Recently, J.C. Nallatamby *et al* revisit the Leeson model and offer a new phase noise analysis method for the feed-back oscillator, in which the empirical parameters are calculated explicitly [6]. For the Colpitts oscillator shown in Fig. A.7, its noise spectral density is written by the equation (A.10).

$$S_{\Delta\phi_{out}} = 2\frac{\left\langle |I_n|^2 \right\rangle}{|V_{01}|^2} \frac{1}{C_1^2\left(\frac{C_1}{C_2}+1\right)^2} \frac{1}{\Delta\omega^2} \tag{A.10}$$

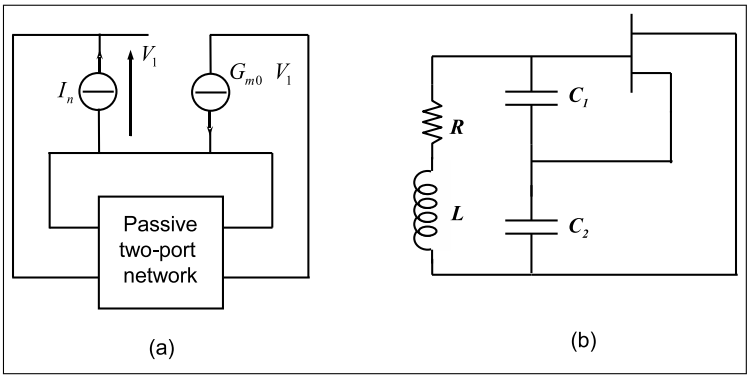


FIG. A.7: *(a) Schematic of an oscillator with noise, (b) Schematic of a Colpitts oscillator*

where I_n is the noise source and V_{01} is the peak value of the carrier signal at the oscillation frequency.

The Lesson model is also unable to explain the phenomena of noise conversion at low-frequency: how does the low-frequency noise convert to the sideband noise around the carrier? To try to answer this question, Hajimiri *et al* proposed a model based on the LPTV (Linear Phase Time-Varying) analysis with ISF (Impulse Sensitivity Function), which is presented afterwards.

A.2.2 Hajimiri model

The ideas of Hajimiri model [1] are based on the following hypothesis:

1. If an oscillator is disturbed in phase (Fig. A.3), the phase error persists indefinitely.
2. If an oscillator is disturbed in amplitude, the amplitude error will disappear along with time.
3. The change in the amplitude or phase is dependent on the time when the perturbation happens. On the one hand, if a perturbation occurs at the peak of voltage, it causes the change only on the amplitude and no change on the phase. On the other hand, if a perturbation occurs at zero crossing of the voltage, it causes change only in phase and no change in amplitude. Therefore, a phase linear time-varying analysis (LPTV) is proposed in the Hajimiri model.

Based on these ideas, Hajimiri *et al* treat the noise conversion in two sequential processes. The first process concerns the conversion from the noise source to the phase perturbation, while the second one is a phase modulation (PM), serving to transform the phase perturbation to the output voltage, as shown in Fig. A.8. [1]

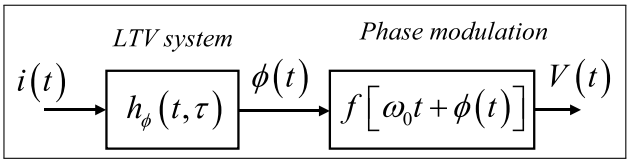


FIG. A.8: *Noise conversion: from noise source to the voltage output of the VCO*

First, Hajimiri *et al* [1] show the relationship between a current noise source and the resulting phase change. In fact, a linear relationship is found between the charge injected into a circuit node and the resulting phase shift. The amount of the charge injected represents the noise source. This linear relationship is critical to predict the phase noise by avoiding the empirical parameters. Hajimiri *et al* describe that relationship by using an ISF function (Impulse Sensitivity Function) that represents how much phase shift is caused by a unity impulse. Given the ISF $\Gamma(\omega_0\tau)$, the phase shift can be calculated as follows:

$$\phi(t) = \frac{1}{q_{\max}} \int_{-\infty}^{t} \Gamma(\omega_0\tau) \cdot i(\tau) d\tau \tag{A.11}$$

where $i(\tau)$ is the noise current, and $q_{\max}$ is the maximal injected charge into the node.

The ISF function is periodic and dependent on time. It can be developed in a Fourier series. Once the coefficients of the ISF $c_n\,(n = 0, 1, ...N)$ are determined, the conversion from the current noise to the phase noise can be obtained, as shown in the following equation:

$$\phi(t) = \frac{I_n c_n \sin(\Delta\omega t)}{2 q_{\max} \Delta\omega} \tag{A.12}$$

where I_n is the amplitude of the current noise and $\Delta\omega$ is the offset frequency, $q_{\max}$ is the charge injected into the node. Note that the phase shift is low-frequency because $\Delta\omega$ is the offset frequency. Accordingly, there will be two sidebands at $\pm\Delta\omega$ in the power spectral density of ϕ(t), denoted as $S_\phi(\omega)$.

The second process is the phase-voltage transformation, which can be represented by a transfer function with the phase shift as input. In fact, it is a phase modulation (PM), in which the low frequency phase shift is transferred around the fundamental frequency. As shown in Fig. A.9. The noise at the frequency $n\omega_0 \pm \Delta\omega$ contributes the phase noise by the coefficients c_n.

Finally, the two processes are grouped and the power spectral density of the output voltage is described by the coefficients of the ISF. Since the noise is mainly the phase noise, the voltage spectral density obtained is equal to phase noise.

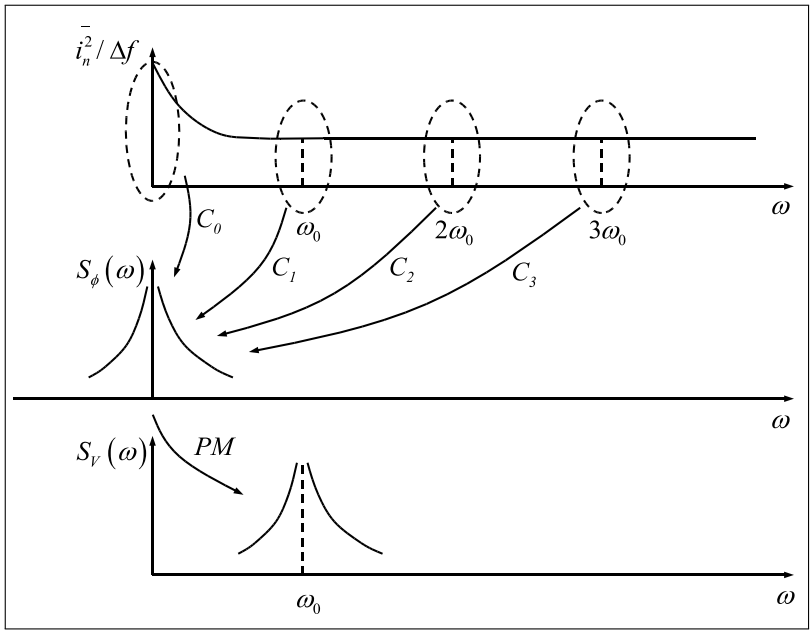


FIG. A.9: *Noise conversion in the Hajimiri model*

$$\mathcal{L}\{\Delta\omega\} = 10 \cdot \log\left(\frac{\bar{i_n^2} \cdot \sum_{n=0}^{\infty} c_n^2}{8q_{\max}^2 \Delta\omega^2}\right) \tag{A.13}$$

One can observe the noise conversion from the equation (A.13): If the noise source i_n contains a low-frequency noise in the form $1/f^n$, in the output phase noise spectrum it may appear as $1/f^{n+2}$ region. For example, the noise sources in 1/f is converted to the noise in the $1/f^3$ region. Also, the white noise source causes the region $1/f^2$ on the phase noise spectrum. The equation (A.13) however can not explain for noise floor characteristic of the oscillator (Fig. A.10) . It may be applied only at a few nodes of the circuit, as the computation burden grows rapidly with the number of sources considered. Hence its efficiency is limited for arbitrary oscillator topologies.

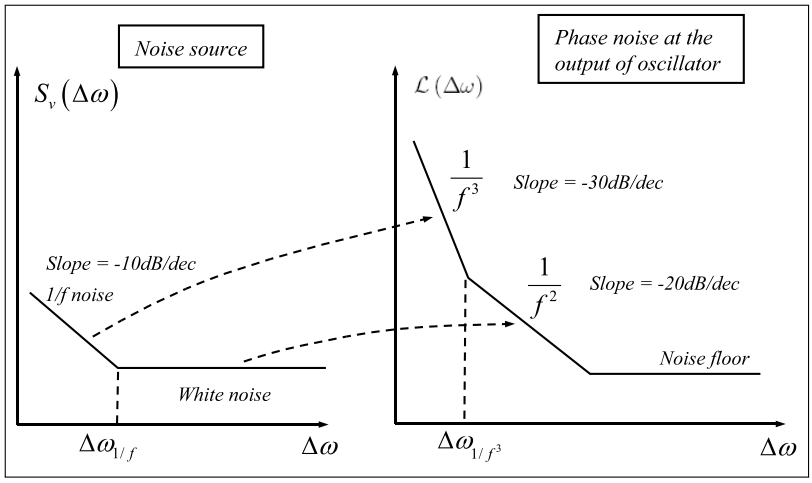


FIG. A.10: *Noise conversion*

A.2.3 Demir model

Demir *et al* [7] bring forward a rigorous non-linear method to analyze the noise of any kind of oscillator. A stochastic differential equation is derived for phase error $\alpha(t)$:

$$\frac{d\alpha}{dt} = v_1^T(t+\alpha(t))\, B(x_s(t+\alpha(t)))\, b(t) \tag{A.14}$$

where $B(x_s(t+\alpha(t)))\, b(t)$ represents the perturbation.

Demir *et al* claim that $\alpha(t)$ is a Gaussian random variable with its mean $\mu(t) = m$ and its variance varies linearly with time:

$$\sigma^2(t) = ct \tag{A.15}$$

According to the authors, one constant c is enough to characterize the spectrum of phase noise:

$$\mathcal{L}(\Delta f) = \frac{f_0^2 c}{\pi^2 f_0^4 c^2 + \Delta f^2} \quad (0 \leq \Delta f << f_0) \tag{A.16}$$

where Δf is the offset frequency, f_0 is the oscillation frequency.

In [8] an extension of this method for the colored noise is proposed, but uses an empirical parameter for the asymptotic behavior of flicker source that has to be properly estimated.

A.3 Bibliography

[1] A. Hajimiri and T. H. Lee, "A general theory of phase noise in electrical oscillators," *IEEE Journal of Solid-State Circuits*, vol. 33, pp. 179–194, 1998.

[2] F. M. Gardner, *Phaselock Techniques*, 3rd ed. Wiley-Interscience, Jul. 2005.

[3] K. Kundert. (2006) Predicting the phase noise and jitter of PLL-based frequency synthesizers. [Online]. Available: www.designers-guide.org/Analysis/PLLnoise.pdf

[4] W. F. Egan, *Frequency Synthesis by Phase Lock*, 2nd ed. Wiley-Interscience, 1999.

[5] D. Leeson, "A simple model of feedback oscillator noise spectrum," *Proceedings of the IEEE*, vol. 54, no. 2, pp. 329–330, Feb. 1966.

[6] J. Nallatamby, M. Prigent, M. Camiade, and J. Obregon, "Phase noise in oscillators: Leeson formula revisited," *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, pp. 1386–1394, 2003.

[7] A. Demir, A. Mehrotra, and J. Roychowdhury, "Phase noise in oscillators: A unifying theory and numerical methods for characterization," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 47, p. 655, 2000.

[8] A. Demir, "Phase noise and timing jitter in oscillators with colored-noise sources," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, pp. 1782–1791, 2002.

Glossary

CAD	Computer-Aided Design
CDR	Clock and Data Recovery
CP	Charge Pump
CR	Carrier analysis
DLL	Delay-Locked Loop
DSB	Double-Sideband
DDS	Direct Digital Synthesizer
FD	Frequency divider
GSM	Global System for Mobile communication
HB	Harmonic Balance
ISF	Impulse Sensitivity Function
LO	Local Oscillator
LPF	Low Pass Filter
$LPTV$	Linear Phase Time-Varying
LTI	Linear Time-Invariant
NH	Number of Harmonics
PD	Phase Detector
PFD	Phase Frequency Detector
PLL	Phase-Locked Loop
PM	Phase Modulation
PPV	Perturbation Projection Vector
PSD	Power Spectral Density
PSS	Periodic Steady State analysis
SNR	Signal-to-Noise Ratio
SSB	Single-Sideband
$TCXO$	Temperature Compensated Crystal Oscillators
TI	Time-domain Integration
TR	Transient analysis
WSS	Wide-Sense Stationary
VCO	Voltage-Controlled Oscillator
XOR	Exclusive OR logic

Publications and communications related to this work

[1] B. Wang, E. Ngoya "A piecewise transistor-level simulation technique for the steady state and phase noise analysis of Integer-N PLLs," *IEEE MTT-S International Microwave Symposium Digest*, page 1429-1432, 2008.

Techniques de modélisation et de simulation pour la vérification précise de PLLs à facteur de division entier

Résumé:

Cette thèse traite de la modélisation et simulation pour la vérification précise de PLLs à facteur de division entier. Les principaux problèmes dans la vérification de la PLL tels que la réponse d'état établi, le bruit aléatoire/déterministe, et les performances dynamiques sont concernés, et l'objectif de cette thèse est de fournir de nouveaux algorithmes et modèles permettant de prédire les principales caractéristiques de la PLL avec autant de précision que la simulation au niveau transistor en utilisant beaucoup moins de temps de simulation. D'abord, concernant le calcul de la réponse d'état établi de la PLL, nous avons soigneusement examiner les conditions de charge aux interfaces des blocs et proposé un algorithme itératif rigoureux qui permet d'obtenir une bonne précision et une convergence rapide. Ensuite, pour prévoir le bruit de phase et déterministe de la PLL, des modèles pour les blocs élémentaires sont proposés, basé sur la réponse d'état établi obtenu. Enfin, l'analyse des performances dynamiques est étudiée. Nous proposons une méthodologie de modélisation pour les différents blocs qui peut atteindre une accélération énorme avec une précision comparable avec la simulation au niveau transistor.

Mots clés: Verrouillage de phase, réponse d'état-etabli, bruit de phase, bruit déterministe, perforamance dynamique.

Modeling and simulation techniques for the accurate verification of Integer-N PLLs

Abstract:

This thesis deals with the modeling and simulation of the Integer-N PLL circuits for the purpose of an accurate and quick design verification. The main PLL verification issues such as steady-state response, random/deterministic noise, and dynamic performances are involved, and the objective is to provide new algorithms and models which allow the prediction of the main characteristics of the PLL with the same precision of the transistor-level simulation using much less simulation time. First, concerning the calculation of the PLL steady-state response, we carefully consider the block interfaces loading conditions and propose a rigorous iterative algorithm which allows achieving a good precision and a quick simulation convergence. Then, to predict the phase noise and deterministic noise of the PLL, block-level models for the PLL building blocks are brought forward, based on the obtained steady state response. Finally the analysis of the PLL dynamic performances is studied. We propose a modeling methodology for various blocks which can achieve an enormous simulation speed-up with a precision comparable with the transistor-level simulation.

Key words: Phase-locked loop, steady-state response, phase noise, deterministic noise, dynamic performance.

XLIM - UMR CNRS n^{o}6172
123, avenue Albert Thomas - 87060 LIMOGES CEDEX