NOISE-SENSITIVE LOOPS IDENTIFICATION FOR LINEAR TIME-VARYING ANALOG CIRCUITS

A Thesis

by

ANG LI

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Peng Li
Committee Members,	Jiang Hu
	Aydin I. Karsilayan
	Alexander Parlos
Head of Department,	Miroslav M. Begovic

August 2016

Major Subject: Computer Engineering

ABSTRACT

The continuing scaling of VLSI technology and the increase of design complexity have rendered the robustness of analog circuits a significant design concern. Analog circuits with strong parasitic effects can be modeled using a multi-loop structure, which is more sophisticated than the traditional single feedback loop structure and results in a more complex small signal stability analysis from the noise perspective. A Loop Finder algorithm has been proposed to allow designers to detect and identify noise-sensitive return loops, which are also called "unstable" loops in previous works,without the need to add breakpoints in the circuit. Besides, efficient pole discovery and impedance computation methods have been explored so that the Loop Finder algorithm can deal with very large scale analog circuits in a reasonable amount of time. However, this algorithm only works for circuits that can be described using a linear time-invariant (LTI) system model. Many practical circuits, such as switch capacitor filters, mixers and so on, have time-varying behaviors. To describe such circuits, a linear time-varying (LTV) system model needs to be employed.

In this research, we first examine the stability property of LTV systems in time domain, mostly based upon the Floquet Theory. We then take an in-depth look at the transfer function of an LTV system in the frequency domain and build the link between it and the Floquet theory. Finally, we propose an efficient algorithm for identifying noise-sensitive loops in linear time-varying circuits. This methodology provides a unifying solution for loop-based noise analysis for both LTI and LTV circuits.

ACKNOWLEDGEMENTS

I would like to thank my committee chair Dr. Peng Li for his instructive advice and proper guidances on my thesis. I am deeply grateful of his help in the completion of this work.

I would also like to thank my committee members Dr. Jiang Hu, Dr. Aydin I. Karsilayan and Dr. Alexander Parlos for their support.

Special thanks to Dr. Aakash Tyagi for attending my defense and giving lots of useful suggestions.

Last but not the least, I would like to thank my parents, my girl friend and all of my friends for their supporting and accompanying, which make the three-year life in TAMU colorful and pleasant.

This work is based upon work supported by the National Science Foundation under grant No. ECCS-1405774.

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

TABLE Page

1. INTRODUCTION

1.1 Motivation

Small signal stability analysis, which is used for the noise-sensitive circuit loop analysis in this work, is usually a necessary phase in the analog circuit design process. Traditional methods [1] [2] to do so require a decent understanding of the circuit to be analyzed and work not well for very large scale circuits with a large number of feedback loops. At the same time, as the feature sizes of transistors shrinks and the size of analog circuits grows rapidly, the effect of parasitics in the circuits becomes more and more dominant. It may influence the performance of analog circuits in several aspects, such as speed, accuracy, stability, power consumption and so on. In this work, the noise sensitivity, which is an aspect of the small signal stability, of analog circuits with strong parasitic effects is analyzed in detail.

Parasitics may be coupled and connected with transistors to form more feedback loops than we expect, which makes the circuits more complicated and need to be modeled as a multi-loop structure instead of the traditional single-loop structure. The noise-sensitive blocks are more likely to exist in certain parts of these circuits.

In this case, running noise-stability analysis for such high-performance analog circuits is a necessary. Since traditional methods don't meet the growing needs, an automatic noise-sensitivity checking tool is needed for not only detecting whether noise-sensitive property exists, but also point out which part of the circuit is potentially noise-sensitive for any large scale analog circuits with multi-loop structures.

One thing to be mentioned is that, in previous works [3] [4], "noise-sensitivity" is called "stability". To avoid confusions with some important definitions in the analog circuit design area, we rename some concepts in [3] [4], which will be introduced in

detail in the following several chapters.

1.2 Loop finder algorithm for linear time-invariant circuits

An automatic noise sensitivity checking algorithm called Loop Finder [3] has been come up with to deal with this problem. The algorithm can not only detect which part of the circuit is noise-sensitive, but also pick up the noise-sensitive loops without knowing the circuit well. What's more, an efficient implementation [4] using some advanced computation techniques, such as model order reduction, parallel computing and so on, is accomplished to make the algorithm speed up a lot and work well for very large scale circuits with potentially over thousands of feedback loops.

However, the Loop Finder algorithm is based on that the small signal model of most analog circuits can be described using a linear time-invariant (LTI) system model. LTI system transfer functions are used for the noise sensitivity analysis. However, not all of analog circuits can be modeled using LTI systems. The small signal models of some circuits can be non-linear, while some can be linear but time-varying. What if we want to perform noise-sensitivity analysis on such kinds of circuits? In this situation, the Loop Finder algorithm is not applicable and we need to think about some new methods.

1.3 Linear time-varying analog circuits

In this research, we mainly focus on the loop-based noise-sensitivity analysis of linear circuits with time-varying behaviors.

For some crucial analog and RF circuit blocks, such as mixers, switch-capacitor filters, their small signal models can't be expressed as LTI systems because of the time-varying performance. However, they can be modeled as linear time-varying (LTV) systems from a certain point of view. The following double balanced mixer [5] is a good example to illustrate this situation.

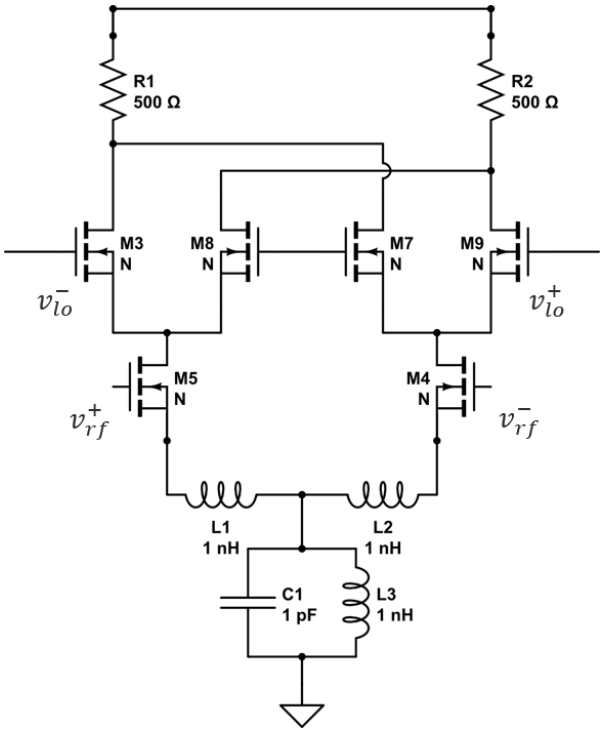

Figure 1.1: A double balanced mixer

In the mixer as shown in Figure 1.1, we can treat V_{LO} as the large signal input, and V_{RF} as the small signal input. To get its small signal model, we can linearize the mixer based on the time-varying large signal V_{LO} at each time point. Since V_{LO} is changing with time, the small signal model is also time-varying. Such a circuit is called a linear time-varying (LTV) circuit.

What's more, the most part of LTV circuits that designers concern also have periodic behaviors. For example, the clock signal in a switch capacitor circuit is always periodic, while the V_{LO} in a double balanced mixer is also periodic. To be more accurate, these kinds of circuits are called linear time-periodically-varying (LTPV) circuits. In our work, LTV circuits also refer to LTPV circuits.

1.4 Research on linear time-varying circuits

LTV circuits and LTV systems are not widely researched due to their relatively narrow application areas compared to other topics. However, there are still lots of critical progresses have been made for the modeling and stability analysis of them.

In 1950, Zadeh proposed an approach to the analysis of linear time-varying networks, which is essentially an extension of the frequency analysis techniques commonly used in linear time-invariant networks [6]. In this work, the transfer function $H(j\omega, t)$ of LTV systems is defined, which makes LTV systems can be analyzed easily in frequency domain. $H(j\omega, t)$ can be treated as an extension to the classic LTI transfer function $H(j\omega)$. It possesses many of the fundamental properties of $H(j\omega)$. What's more important is that it can be used in a similar manner to capture the input and output relationship of an LTV network just as LTI cases.

In [7] and [8], the LTV transfer function $H(j\omega, t)$ is used to solve the problem of how to perform model order reduction on RF circuits. Most RF circuits are actually LTPV circuits, such as the mixers described before. The periodic behaviors of them result in a Fourier expansion form of $H(j\omega, t)$, which is derived in [8]. In this form, $H(j\omega, t)$ can be expressed as the summation of LTI systems followed by memory-less multiplications with $e^{ji\omega_0 t}$, in which ω_0 is the fundamental frequency of the corresponding circuit. By using this expression, the connection between LTI and LTPV systems becomes tighter.

When it comes to the stability analysis of the LTPV system, the most important related work is Floquet Theory [9]. The original Floquet Theory talks about the stability property of the LTPV systems that can be described using Ordinary Differential Equation (ODE) as $\dot{x}(t) = A(t)x(t)$ in time domain. In [10] and [11], works have been done to demonstrate that the Floquet Theory is also helpful to analyze

the stability of those LTPV systems in frequency domain.

However, LTPV circuits are usually modeled as Differential Algebraic Equation (DAE), which is a generalized form of ODE. Their core parts can be written as $B(t)\dot{x}(t) = A(t)x(t)$, in which $B(t)$ may not be full rank. In this case, the original Floquet theory cannot be applied. In Lamour's work [12], The Floquet Theory for Index-1 DAE is derived. Some concepts in the original Floquet Theory are redefined so that they can be used for DAE cases. This theory is actually the ground truth of our work.

In this research, we shall combine the LTPV transfer function $H(j\omega, t)$ and the Floquet theory in the DAE case to perform the loop-based noise-sensitivity analysis for LTPV circuits.

1.5 Basic concepts in the stability analysis of linear systems

In this section, we shall introduce some concepts in the stability analysis of linear systems since they are basic and important in our work.

1.5.1 Stability of linear circuits

A linear circuit can always be modeled as a linear system and draw as a block diagram. Let's consider a linear circuit with single-input and single-output. It can be described using a block diagram shown in Figure 1.2. If the linear circuit is stable,

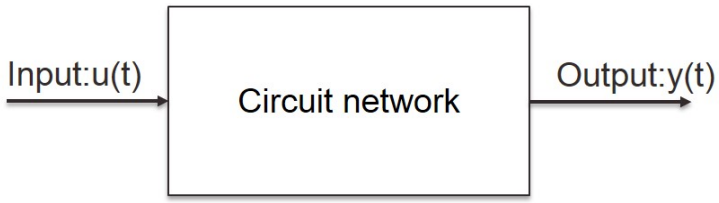

Figure 1.2: Diagram of a linear circuit

for any given bounded inputs $u(t)$ (voltages or currents), its outputs $y(t)$ will also be bounded and convergent to certain values y as time goes to infinity. y is called the steady state of the circuit. For linear circuits, such kind of stability is called "expotential stability". There is another definition of stability called "marginally stability". Marginally stability describes the convergence speed of an exponential stable circuit, it can be measured by some margin concepts such as phase margin and gain margin. In our work, the noise-sensitivity is somehow referred to the marginally stability of linear circuits.

1.5.2 Transfer function and node impedance of linear circuits

A transfer function describes the input/output relationship of a linear circuit as shown in Figure 1.2 in frequency domain. The transfer function of LTI circuits can be defined as (1.1), in which $y(s)$ and $u(s)$ are the Laplace transform of output/input of the corresponding circuit separately.

$$H(s) = \frac{y(s)}{u(s)} \tag{1.1}$$

If the input of an LTI circuit is a current injected to a certain node A while the output is the voltage of the same node, $H(s)$ is representing the impedance of node A. The node impedance transfer function of an LTI circuit can also be noted as $Z(s)$.

We shall show that the transfer function and node impedance of LTV circuits can also be defined in a similar manner.

1.5.3 Poles of linear time-invariant circuits

For linear invariant-circuits, the transfer function $H(s)$ can always been represented using a lumped high order fractional form, which is shown in (1.2).

$$H(s) = \frac{a_n s^n + a_{n-1} s^{n-1} + ... + a_1 s + a_0}{b_n s^n + b_{n-1} s^{n-1} + ... + b_1 s + b_0} \tag{1.2}$$

In this form, the zeros of the denominator are called "poles" of the circuit. An LTI circuit is exponentially stable if and only if it has poles with strictly negative real parts. For marginally stability, the "zeros" of the circuit, which are the zeros of the numerator will also be considered. Pole-zero analysis is a classic method for the stability analysis of LTI circuits. We will prove that it also works for LTPV circuits in this work.

In chapter II, we shall first introduce the Loop Finder algorithm for LTI circuits, which works as the foundation of our proposed algorithm for LTPV circuits. In chapter III, we shall describe the theory background and several phases of our algorithm in detail. Some concepts are redefined and developed based on the previous LTI work. In chapter IV, we shall present the experiment results produced by running our algorithm on several practical circuits. Finally, the conclusion is discussed in chapter V.

2. LOOP FINDER ALGORITHM FOR LINEAR TIME-INVARIANT CIRCUITS

In this chapter, we shall describe the previous work done by Albert and Parijatte [3] [4], which is the Loop Finder algorithm for LTI circuits. This work is a foundation to our algorithm since we extend and rename several concepts based on their framework. We shall start off by introducing single loop theory, which is the traditional stability analysis method for LTI circuits. We shall then discuss the system transfer function of LTI systems and its second order approximation. In the last, we shall provide a deeper loop-based analysis of the noise sensitivity properties of any LTI systems. Several concepts, such as "noise-sensitive poles" and "noise-sensitive loops", will be discussed.

2.1 Single loop theory

Traditionally, analog circuits are modeled as single loop structures by designers for stability analysis. A typical negative feedback single loop system is shown in Figure 2.1. The feedback loop contains two parts. One is the main path A, which

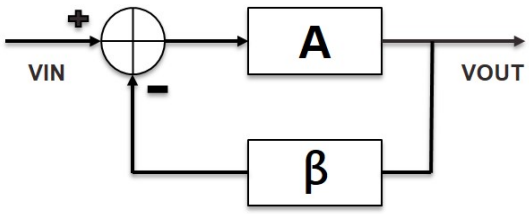

Figure 2.1: A typical negetive feedback loop in LTI circuits

refers to the open loop gain of the system. The other is the return path β, which is also called the feedback factor of the system. The feedback equation of the system

is shown in (2.1). Using this equation, the transfer function G of this single loop system can be written as (2.2).

$$(VIN - \beta VOUT)A = VOUT \tag{2.1}$$

$$G = \frac{VOUT}{VIN} = \frac{A}{1 + \beta A} \tag{2.2}$$

G describes the relationship of the input and output of this feedback system, which is actually an LTI system. The zeros of $1 + \beta A$ are the poles and contain the stability information of the system. Efficient ways to measure the stability of such a circuit are using some "margin" concepts, such as phase margin and gain margin.

However, when circuits become larger, more and more parasitics and transistors are connected and coupled with each other, which results in a large amount of feedback loops formed. A multi-loop structure system model is more accurate than the single-loop structure system to represent such kinds of circuits. In this circumstance, single loop theory is no longer applicable.

Though single loop theory cannot be used in a multi-loop structure circuit, using transfer function to analyze the stability of circuits is illuminating for us since it's a good way to avoid "spot checking [2]". "Spot checking" is a time-consuming way for the stability analysis of multi-loop structure circuits. Test inputs are given and a huge number of transient simulations are needed to be run for each suspected loop in the circuits. This method is impractical for circuits with potentially over thousands of feedback loops.

To sum up, if we want to analyze the stability of multi-loop structure circuits efficiently, a more accurate transfer function needs to be come up with to model them.

2.2 System transfer function for linear time-invariant circuits

2.2.1 The LTI transfer function

In the Loop Finder algorithm [3], the system transfer function of LTI circuits is used. From the perspective of control theory, a linear time-invariant circuit can be modeled as a constant feedback network, which also refers to an LTI system and may contain a large number of loops. The network can be described using differential algebraic equations(DAE) as following:

$$C\dot{x}(t) = -Gx(t) + Bu(t) \tag{2.3}$$

$$y(t) = L^T x(t) \tag{2.4}$$

In the DAE above, $G_{n\times n}$ contains the memory-less elements such as resistors. $C_{n\times n}$ contains memory elements such as capacitors and inductors. $x(t)$ contains state variables(node voltages, branch currents). $u(t)$ and $y(t)$ are the input and output of the circuit separately. $B_{n\times 1}$ and $L_{n\times}$ are vectors for single-input and single-output case. They map the relationships between the input/output and the state variable $x(t)$.

Laplace transform can be performed on $x(t)$, $y(t)$ and $u(t)$ so that they become $x(s)$, $y(s)$ and $u(s)$ separately. The DAE is also transformed into frequency domain as (2.5) and (2.6). The system transfer function $H(s)$ can be computed and has the form as shown in (2.7).

$$sCx(s) = -Gx(s) + Bu(s) \tag{2.5}$$

$$y(s) = L^T x(s) \tag{2.6}$$

$$H(s) = \frac{y(s)}{u(s)} = L^T(G + sC)^{-1}B \tag{2.7}$$

Since we are considering single-input and single-output case, if B ad L are chosen corresponding to a node in the circuit, the node impedance transfer function can be written as (2.8).

$$Z(s) = \frac{res(s)}{(s-p_1)(s-p_2)(s-p_3)...(s-p_n)} \tag{2.8}$$

The denominator part of $Z(s)$ stands for the pole information of the circuit. To compute p_i for $i = 1, 2, 3, ..., n$, a generalized eigenvalue problem described in (2.9) needs to be solved.

$$GX = \lambda CX \tag{2.9}$$

In (2.9), λ is the eigenvalue of the system. There are at most n non-zero eigenvalues $\lambda_i, i = 1, 2, 3..., n$, which also refer to the poles p_i of the system.

$(G+sC)^{-1}$ remains the same for all node impedance transfer functions of a circuit, which means that all nodes are sharing the same poles. At the same time, $res(s)$, which is the residue information of the node impedance transfer function, may vary among different nodes.

By using $H(s)$, a multi-loop structure circuit is modeled and treated as a whole. No information will be lost in this process. What's more, C and G are easy to be extracted from standard analog circuit simulators, since they are how the circuit is represented in the simulator naturally.

2.2.2 Second order approximation for LTI systems

Second order systems are commonly used in the feedback network stability analysis, since they deal with two-pole circuits which are commonly used by designers. Even for the complicated circuits with further more than 2 poles, the second order approximations can still be used to capture the stability performance accurately [4].

The transfer function $H(s)$ for a single-input and single-output second order

system is shown in (2.10).

$$H(s) = \frac{res(s)}{s^2 + 2\zeta\omega_0 s + {\omega_0}^2} \tag{2.10}$$

$$p_{1,2} = p_r \pm ip_i \tag{2.11}$$

$$\omega_0 = |p_{1,2}|, \zeta = -\frac{p_r}{|p_{1,2}|} \tag{2.12}$$

The denominator of $H(s)$ has two zeros p_1 and p_2, which are the poles of the system. They are conjugate to each other and can be written as (2.11).

There are two important parameters, ω_0 and ζ, in $H(s)$. ω_0 is defined as the magnitude of $p_{1,2}$, which is called natural frequency. ζ is called damping factor and defined as the ratio of the negative real part of $p_{1,2}$ and ω_0.

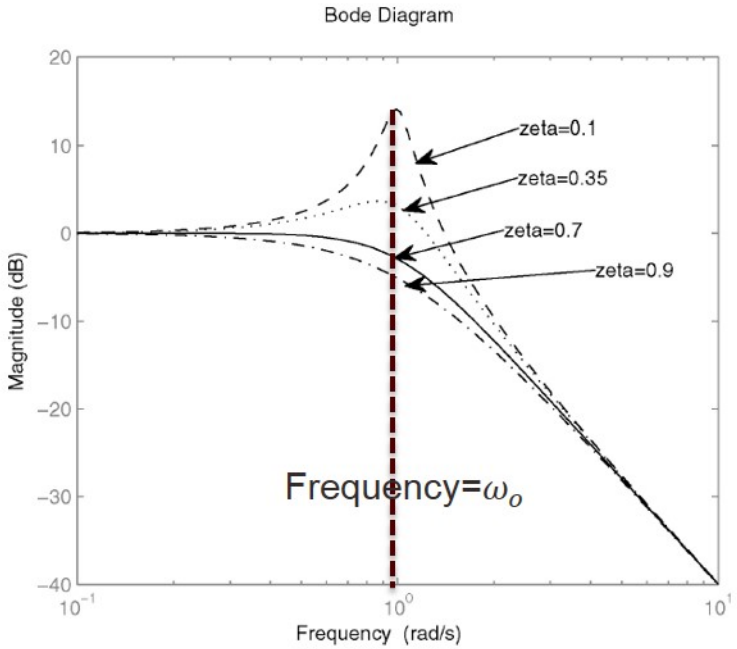

Figure 2.2: A second order system's bode plot

In Figure 2.2, a classic second order system bode diagram is shown. It can be found that a peaking behavior shows at frequency $\omega = \omega_0$ when damping factor ζ is small. A threshold value $\zeta_{th} = 0.7$ is often used by designers to measure the peaking. When ζ is less than ζ_{th}, the peaking behavior is considered to be large. If the second order system is the model of a circuit, a large peaking in frequency domain is corresponded to a ringing or overshooting behavior in the output waves of certain circuit nodes in time domain. It means that when a small noise signal with frequency near ω_0 is injected, the output of the circuit may be influenced and have some undesired behaviors. Thus, when such kind of peaking shows in the bode diagram of a second order system, we say that the system is potentially noise-sensitive and its pole is called a potentially noise-sensitive pole(a complex pole pair can be represented by any one of them since they always show up in pair).

2.3 Loop-based noise sensitivity analysis of linear time-invariant circuits

In the previous section, the LTI transfer function $H(s)$ and its second order case are described. The noise sensitivity property of the second order system is also introduced in detail. Now we shall start to analyze the noise sensitivity of any LTI systems.

2.3.1 Second order approximation of any linear time-invariant systems

Just like (2.8), for a given LTI circuit, each node impedance transfer function $Z(s)$ of it can be written in a lumped high-order linear fraction form. To make it easier to be investigated, a further factorization can be performed on $Z(s)$ and it can be written in the following form:

$$Z(s) = \sum_{i=1}^{N_R} \frac{k_i}{s - p_i} + \sum_{j=1}^{N_C} \frac{res_j(s)}{s^2 + 2\zeta_j \omega_{0j} s + \omega_{0j}^2} \tag{2.13}$$

$$\omega_{0j} = |p_j|, \zeta_j = -\frac{p_{j,r}}{|p_j|} \tag{2.14}$$

It means that $Z(s)$ can be expressed as the linear combination of several first-order systems and second-order systems. Each second-order system has its own natural frequency ω_{0j} and damping factor ζ_j. N_R corresponds to the number of first order sub-systems while N_C refers to the number of second order sub-systems in the whole system.

The second order system parts can potentially contribute to the noise-sensitive behavior of the whole node impedance since they may have peaking behaviors in frequency domain. Just as described before, a second order system is considered to be potentially noise-sensitive for $0 < \zeta < 0.7$. The natural frequency ω_0 tells us where the noise-sensitive behavior is and whether it will degrade the circuits' performance within our frequency range of interest $[0, \omega_{max}]$ or not. In general, if any second order systems in the node impedance $Z(s)$ satisfy the following 2 conditions (2.15), their corresponding poles are said to be potentially noise-sensitive poles for this node impedance transfer function [4].

$$\zeta < 0.7, \omega_0 < \omega_{max} \tag{2.15}$$

To compute the potentially noise-sensitive poles of a node impedance transfer function, all poles need to be computed first. Then the poles satisfying the above conditions are potentially noise-sensitive. The pole information will be further used for the noise-sensitive loop detection in the circuit.

2.3.2 Noise-sensitive pole in linear time-invariant systems

Since several second order systems are combined to form the node impedance transfer function, not all potentially noise-sensitive poles will contribute to the bad

noise performance at each node. The following figure can help to explain this situation.

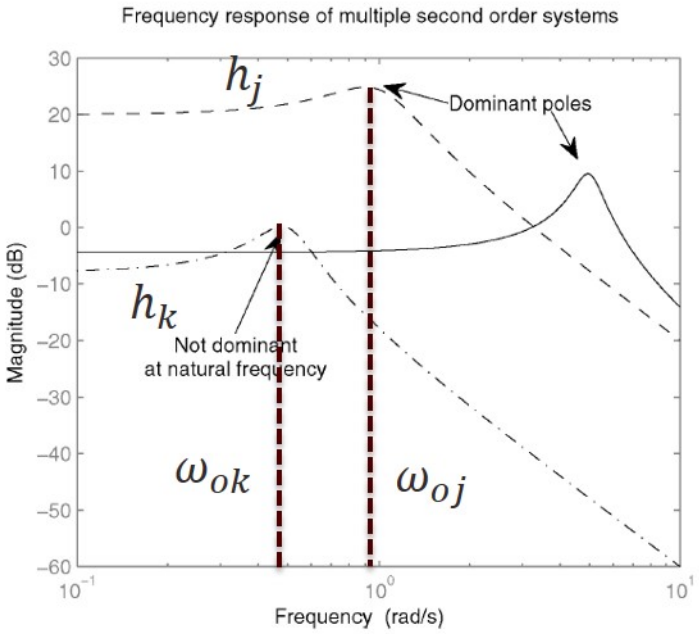

Figure 2.3: Several second order systems at a circuit node

Figure 2.3 shows the bode diagrams of several potentially noise-sensitive second order systems of a node impedance $Z(s)$. Let's have a look at the second order system h_j first. A peaking shows in its bode diagram at its natural frequency ω_{0j}. At the same time, other second order systems also have certain magnitudes at ω_{0j}, which are not peaking behaviors. The total impedance of the node at ω_{0j} is proportional to the summation of these magnitudes. Since the peaking behavior of h_j contributes the most to the value of impedance at ω_{0j}, the node is treated as noise-sensitive at ω_{0j}.

A concept called noise-sensitive pole, which refers to dominant pole in [3] [4] is

defined to describe the second order system like h_j in a node impedance transfer function.

Definition(Noise-sensitive pole): If a potentially noise-sensitive second order system h contributes the most to the node impedance at its corresponding natural frequency ω_0, the complex pole of it is called the noise-sensitive pole for the node impedance transfer function.

Based on this concept, the pole of h_k in Figure 2.3 is not a noise-sensitive pole. Even a peaking shows at ω_{0k} in its bode diagram, the value is much less than the magnitude of h_j at the same frequency. It means the peaking of h_k is not dominant to the node impedance value at ω_{0k}. The circuit node will still be considered as noise-insensitive at ω_{0k}.

Noise-sensitive poles are a subset of potentially noise-sensitive poles, and they are the real source of noise-sensitivity in any node impedance transfer functions. Each circuit node may have several noise-sensitive poles. They reflect the noise sensitivity property of this node at distinct natural frequencies separately.

2.3.3 Noise-sensitive loop in LTI circuits

Now, let's have a deeper look at what the noise-sensitive pole means to an LTI circuit. Consider p is a noise-sensitive pole to the $Z(s)$ of node A, and ω_p is the corresponding natural frequency. If a small signal input noise with frequency ω near ω_p is given to A, the output of A also has frequency ω. Since there is a peaking showing at ω_p, the magnitude of $Z(j\omega)$ is large at this frequency. It means the output of A will be much larger than input and has a ringing behavior in time domain.

As shown in Figure 2.4, a group of circuit nodes with the same noise-sensitive pole p_1 are considered together. If an input noise signal is given to any circuit nodes in this group, the signal will traverse through each other node and propagate in this

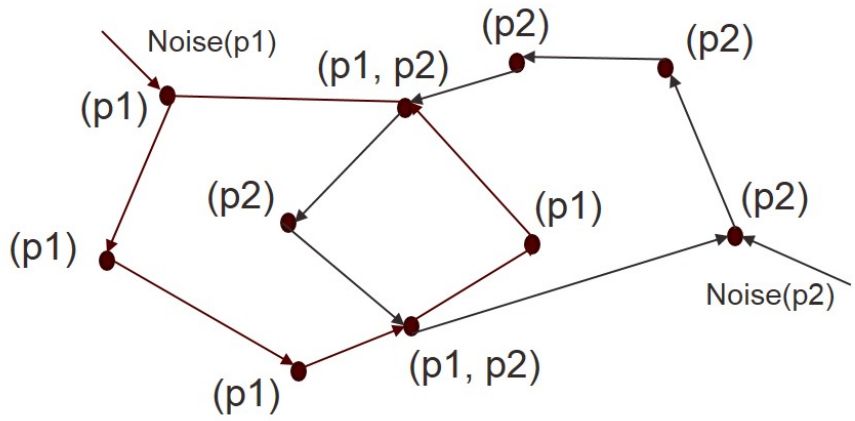

Figure 2.4: Noise-sensitive loops in LTI circuits

process. Finally, the signal will come back to node A and has much larger value than the input. In this case, the loop formed by these circuit nodes is said to be noise-sensitive since if a signal goes through the loop, its value will be growing and the ringing behavior will not decay. The definition of the noise-sensitive loop, which refers to the "unstable loop" [3] [4] in LTI circuits is shown as following.

Definition(Noise-sensitive loop): In an LTI circuit, a noise-sensitive loop is a group of circuit nodes which have the same noise-sensitive pole.

We can find that some other nodes have the same noise-sensitive pole p_2 in Figure 2.4. They are gathered to form another noise-sensitive loop. These two loops are coupled with each other since some circuit nodes have both p_1 and p_2 as their noise-sensitive poles.

One thing to be noted here is that the nodes in a noise-sensitive loop may not be physically connected. To be more accurate, they are actually a group of nodes which have a similar behavior corresponding to the input noise signal near a certain frequency. If the noise-sensitive loops are reported to analog designers, they may have some insights about the results and can make modifications and improvements

to the circuits.

2.3.4 Procedure for identifying noise-sensitive loops in LTI circuits

Noise-sensitive loops are the final output of the Loop finder algorithm. They represent the noise-sensitive part of an LTI circuit. To identify them, the following procedure can be followed:

- For a given LTI circuit, extract the G and C matrices via DC simulation.
- Compute all the poles from the linearized circuit and selectively pick out the potentially noise-sensitive poles.
- Compute residues and impedance value for all circuit nodes. To get the residue value, the following formula (2.16) can be used [4].

$$Z(s) = \sum_{j=1}^{n} \frac{k_j}{s - \lambda_j} \tag{2.16}$$

  In (2.16), $k_j = r_j \times l_j, l = X^H L, r = -(GX)^{-1}\Lambda B$. X is the eigenvector matrix in which each column corresponds to an eigenvalue λ.

- Ignore second order systems with very low DC impedance value. For a second order system with very low impedance value, even there may be a peaking behavior at the natural frequency, it contributes almost nothing to the node impedance, since its absolute value is so small. A threshold value r_{dc} can be set up to filter out such kinds of second order systems. If the DC impedance of a second order system is less than r_{dc}, it can be ignored. The value of r_{dc} may vary in different applications. For LTI circuits, r_{dc} are always set to be 0.1.
- At last, noise-sensitive complex poles for each node impedance transfer function can be determined. The circuit nodes with the same noise-sensitive poles can

be grouped to form noise-sensitive loops.

3. IDENTIFICATION OF NOISE-SENSITIVE LOOPS IN LINEAR TIME-VARYING CIRCUITS

In the previous chapter, the Loop Finder algorithm for LTI circuits is introduced. It can be treated as a foundation to our corresponding algorithm in LTV circuits, which will be discussed in this chapter. We shall start with the LTV transfer function [6] [7] [8], which is a generalized form of the LTI transfer function $H(s)$. We shall then introduce the Floquet Theory [12] [9], which is a classic theory on the stability property of LTV systems, and its application in frequency domain. The concepts of "noise-sensitive pole" and "noise-sensitive loop" are extended to the LTV case. What's more, the noise-sensitive loop detection algorithm for LTV circuits will be introduced. At last, we shall explain the procedure for the noise-sensitive loop identification algorithm in LTV circuits. Since almost all the widely used LTV circuits have periodic behaviors, the "LTV" in this chapter also refers to "LTPV".

3.1 System transfer function for LTV circuits

3.1.1 Definition of the LTV transfer function

Before introducing the LTV transfer function, we shall first review the definition of the LTI transfer function $H(S)$. No matter what kind of system(single-loop or multi-loop) $H(s)$ stands for, it can be defined using (3.1).

$$H(j\omega) = \int_{-\infty}^{\infty} h(t-p)e^{-j\omega(t-p)}dp \tag{3.1}$$

In (3.1), s is set to be $j\omega$. $H(j\omega)$ represents the value of $H(s)$ at frequency ω. $h(t-p)$ is the impulse response of the system. It can be applied to describe an LTI system because its value only depends on the difference between t and p. t is

the time that the system is observed, while p is the time that the input impulse $\delta(p' - p)$ occurs. The equation means that the LTI transfer function $H(s)$ is the Fourier transform of the impulse response of the corresponding system.

In a similar manner, the LTV transfer function can also be defined as the Fourier transform of the impulse response of a corresponding LTV system [6], which is described in (3.2).

$$H(j\omega) = \int_{-\infty}^{\infty} h(t,p)e^{-j\omega(t-p)}dp \tag{3.2}$$

In (3.2), $h(t,p)$ denotes the impulse response of the LTV system. It is different from $h(t-p)$ in LTI case since its value depends on t and p separately. The integration in (3.2) is performed on p, so a time-varying transfer function $H(j\omega,t)$ is produced as the result.

In fact, the LTV transfer function $H(j\omega,t)$ represents a natural extension of the LTI transfer function $H(j\omega)$ [6]. It possesses many of the basic properties of the LTI system. What's more important is that it can be used to describe the response of an LTV system to any prescribe input, just like the $H(j\omega)$ in LTI case.

3.1.2 Description of input/output relationships of LTV systems using $H(j\omega,t)$

In this subsection, we shall derive how to use the LTV transfer function $H(j\omega,t)$ to represent input/output relationship of a given LTV system. The conclusion is further used in the derivations of other useful results.

Let $x(t)$ and $u(p)$ to be the input and output separately. The relationship of $x(t)$ and $u(p)$ can be firstly written as (3.3), which is a fact to any linear systems.

$$x(t) = \int_{-\infty}^{\infty} h(t,p)u(p)dp \tag{3.3}$$

$$u(p) = \frac{1}{2\pi}\int_{-infty}^{infty} U(j\omega)e^{j\omega p}d\omega \tag{3.4}$$

$u(p)$ can be represented by the inverse Fourier transform of its frequency domain form $U(j\omega)$ as (3.4). Substitute (3.4) into (3.3), we can get (3.5). It is easy to be found that the last part of (3.5) is actually $H(j\omega, t)$ times an exponential part as described in (3.6). Finally, we can get the conclusion as shown in (3.7). It describes how to use $H(j\omega, t)$ to represent the output $x(t)$ [6].

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} U(j\omega) d\omega \int_{-\infty}^{\infty} h(t, p) e^{j\omega p} dp \tag{3.5}$$

$$\int_{-\infty}^{\infty} h(t, p) e^{j\omega p} dp = H(j\omega, t) e^{j\omega t} \tag{3.6}$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(j\omega, t) U(j\omega) e^{j\omega t} d\omega \tag{3.7}$$

For LTI systems, there is a similar classic conclusion, which is shown in (3.8). Since the LTV system is a more generalized form of the LTI system, we can treat (3.8) as a specific case of (3.7).

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(j\omega) U(j\omega) e^{j\omega t} d\omega \tag{3.8}$$

3.1.3 DAE for LTV circuits

Just like the LTI circuits, the small signal models of LTV circuits can also be expressed using DAE. For an LTV circuit, its DAE has the following form [7]:

$$\frac{d}{dt}(C(t)x(t)) = -G(t)x(t) + Bu(t) \tag{3.9}$$

$$y(t) = L^T x(t) \tag{3.10}$$

In (3.9) and (3.10), everything is the same as (2.3) and (2.4) except that $G(t)$

and $C(t)$ are time-varying. The reason for them to be time-varying is that the operating points of LTV circuits are always changing as time grows. At different time points, linearizing the circuit may result in different small signal models, which are represented by different G and C matrices. These matrices are combined to form $G(t)$ and $C(t)$ to describe the LTV circuits.

3.1.4 System transfer function for LTV systems

If we consider the state variable $x(t)$ in (3.9) as the output and $u(t)$ as the input, a corresponding transfer function $W(j\omega', t)$ can be used to represent $x(t)$, as shown in (3.11).

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} W(j\omega', t)U(j\omega')e^{j\omega' t}d\omega' \tag{3.11}$$

To get rid of the integration, let $U(j\omega')$ to be an impulse signal in frequency domain that occurs at $\omega' = \omega$ [7], which is shown in (3.12).

$$U(j\omega') = u_{imp}\delta(\omega' - \omega) \tag{3.12}$$

$$x(t) = \frac{1}{2\pi}u_{imp}W(j\omega, t)e^{j\omega t} \tag{3.13}$$

Substitute (3.12) into (3.11), $x(t)$ can be written as (3.13). It stands for the output of the LTV system descried by (3.9) when input is a small exponential signal with frequency ω.

Set $s = j\omega$ and substitute (3.13) into (3.9), $W(j\omega, t)$ has the form as shown in (3.14). Since $y(t)$ in (3.10) is the final output of the whole LTV system described by the DAE (3.9)(3.10), The transfer function of the whole system $H(j\omega, t)$ can be written as (3.15) [7].

$$W(s, t) = [G(t) + sC(t) + \frac{d}{dt}(C(t))]^{-1}B \tag{3.14}$$

$$H(s,t) = L^T W(s,t) = L^T[G(t) + sC(t) + \frac{d}{dt}(C(t))]^{-1}B \tag{3.15}$$

In (3.14) and (3.15), a derivative linear operator $\frac{d}{dt}(C(t))$ shows in the inversion parts. Its definition is as (3.16) [8].

$$\frac{d}{dt}(C(t))[x] = \frac{d}{dt}(C(t)x) \tag{3.16}$$

If we compare $H(s,t)$ with the LTI transfer function $H(s)$ in (2.7), there are mainly two differences. One is that C and G are time-varying in LTV case, while they are constant in LTI case. The other is that a derivative operator shows in the inversion part of the LTV transfer function $H(s,t)$. These differences are reasonable since that if $G(t)$ and $C(t)$ are constant with t, the derivative part will disappear and the LTV system will reduce to a corresponding LTI system. From this perspective, we can also conclude that LTV system is a more generalized form of the LTI system.

Just as mentioned before, almost all of the LTV circuits we concern are also periodic. To get a more specific form of the transfer function for LTPV circuits, further processing can be performed on $H(s,t)$.

For an LTPV system, $H(s,t)$ is periodic with t. What's more, $G(t)$ and $C(t)$ are also periodic. Let's denote the fundamental frequency of the LTPV system is ω_0, the above three can be written in their Fourier expansion forms, which are described in (3.17) [8].

$$H(s,t) = \sum_{i=-\infty}^{\infty} H_i(s)e^{ji\omega_0 t}, G(t) = \sum_{i=-\infty}^{\infty} G_i e^{ji\omega_0 t}, C(t) = \sum_{i=-\infty}^{\infty} C_i e^{ji\omega_0 t} \tag{3.17}$$

Substitute (3.17) into (3.15), the Fourier expansion form of $H(s,t)$ can be written as (3.18). Each $H_i(s)$ is the coefficient of corresponding harmonic, which is called the

harmonic transfer function of the LTPV system. $H_i(s)$ can be written as (3.19) [8].

$$H(s,t) = \sum_{i=-\infty}^{\infty} H_i(s) e^{ji\omega_0 t} \tag{3.18}$$

$$H_i(s) = {L_i}^T[(G_{FD} + \Omega C_{FD}) + sC_{FD}]^{-1} B_{FD} \tag{3.19}$$

All parts of $H_i(s)$ in (3.19) are listed in (3.20) $-$ (3.24) [8].C_{FD} and G_{FD} are called the Toeplitz form of $C(t)$ and $G(t)$ separately. They are two infinite matrices formed by the harmonic coefficients of $C(t)$ and $G(t)$. B_{FD} is an infinite vector with the central part to be B. LL is an infinite diagonal matrix in which the ith diagonal block is L_i.

$$G_{FD} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & G_0 & G_{-1} & G_{-2} & G_{-3} & \cdots \\ \cdots & G_1 & G_0 & G_{-1} & G_{-2} & \cdots \\ \cdots & G_2 & G_1 & G_0 & G_{-1} & \cdots \\ \cdots & G_3 & G_2 & G_1 & G_0 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{3.20}$$

$$C_{FD} = \begin{bmatrix} \ddots & \vdots & \vdots & \vdots & \vdots & \\ \cdots & C_0 & C_{-1} & C_{-2} & C_{-3} & \cdots \\ \cdots & C_1 & C_0 & C_{-1} & C_{-2} & \cdots \\ \cdots & C_2 & C_1 & C_0 & C_{-1} & \cdots \\ \cdots & C_3 & C_2 & C_1 & C_0 & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{3.21}$$

$$\Omega = j\omega_0 \begin{bmatrix} \ddots & & & & & & \\ & -2I & & & & & \\ & & -I & & & & \\ & & & 0 & & & \\ & & & & I & & \\ & & & & & 2I & \\ & & & & & & \ddots \end{bmatrix} \tag{3.22}$$

$$B_{FD} = [..., 0, 0, B^T, 0, 0, ...]^T \tag{3.23}$$

$$LL = \begin{bmatrix} \ddots & & & & & & \\ & L & & & & & \\ & & L & & & & \\ & & & L & & & \\ & & & & L & & \\ & & & & & L & \\ & & & & & & \ddots \end{bmatrix} \tag{3.24}$$

For each harmonic transfer function $H_i(s)$ in $H(s,t)$, we can also compare it with the LTI transfer function $H(s)$ in (2.7). If $(G_{FD} + \Omega C_{FD})$ is considered to be the equivalent G, while C_{FD} is considered to be the equivalent C, $H_i(s)$ and $H(s)$ actually have the same form. If all harmonic coefficients except the DC components of $G(t)$ and $C(t)$ are 0, the LPTV transfer function will reduce to an LTI transfer function automatically, since the LTI system is a specific case of the LTPV system.

Based on the above illustrations, some remarks about the LTPV system can be summarized as following:

- Any LTPV systems can be decomposed into LTI systems followed by memoryless multiplications with $e^{ji\omega_0 t}$ [8]. The LTI system is a specific case of the LTPV system.

- The reason for harmonics existing in the LTPV transfer function is that the LTPV system produces harmonic frequency shifts, while the LTI system doesn't produce new frequency.

- Ideally, an LTPV system has infinite number of poles shared by all harmonic transfer functions, since C_{FD} and G_{FD} are infinite. An important property about the poles of the LTPV system is that if p is a pole, then $p \pm ji\omega_0$ are also poles of the system, in which $i = 1, 2, 3, 4....$ This property is really helpful for computing the poles of LTPV systems and there will be a brief proof later.

- In real life, infinite dimension matrices cannot be computed and analyzed. Truncations can be performed on C_{FD} and G_{FD}, which means only the first n harmonics that we are interested in can be preserved. If N harmonics are considered, the size of G_{FD} and C_{FD} will be $(2N+1)n \times (2N+1)n$. The size of L_i and B_{FD} will be $(2N+1)n \times 1$ for single-input and single-output case. n is the size of $G(t)$ and $C(t)$. If the truncation is performed appropriately, the accuracy of stability analysis of the corresponding LTPV system can be ensured.

3.1.5 Procedure for computing the LTPV transfer function

To compute the transfer function of the LTPV system, the following procedure can be followed:

- For a given circuit, do transient simulation and record the G and C matrices for an amount of time after it's stabilized as $G(t)$ and $C(t)$.

- Do FFT on $G(t)$ and $C(t)$, extract our interested harmonic coefficients to form G_{FD} and C_{FD} with finite dimensions.

- By choosing proper L and B, the value of $H(s,t)$ at any frequencies and any time points can be computed.

3.2 Stability of LTPV systems

From the previous sections, we know that an LTPV system can also be represented with its transfer function $H(s,t)$. The poles of an LTPV system can be computed from $H(s,t)$, which is similar to the LTI system case.

Poles in LTI systems contain the stability information, that is, if the real parts of all poles are less than 0, the LTI system is treated as stable. In this section, we shall introduce the stability property of the LTPV system and demonstrate that the poles of LTPV systems also contain the stability information.

3.2.1 Floquet theory

The Floquet theory [9] is widely used in the analysis of stability of dynamical systems with periodic behaviors. The original Floquet theory is a branch of the theory of ordinary differential equations (ODE) relating the class of solutions to periodic linear differential equations of the form $\dot{x}(t) = Ax(t)$, in which $A(t)$ is a periodic matrix. In [12], the Floquet theory has been extended to deal with the DAE shown in (2.3)(2.4), and serves as the ground truth to our research.

The DAE for a single-input single-output LTPV circuit is re-written in (3.25)(3.26). We are giving some more detailed descriptions about them.

$$\frac{d}{dt}(C_{n\times n}(t)x(t)) = -G_{n\times n}(t)x(t) + B_{n\times 1}u(t) \tag{3.25}$$

$$y(t) = L^T_{n\times 1}x(t) \tag{3.26}$$

$C_{n\times n}(t)$ and $G_{n\times n}(t)$ are both $n \times n$ matrices. $L_{n\times 1}$ and $B_{n\times 1}$ are both constant $n \times 1$ vectors. $G_{n\times n}(t)$ is a non-singular matrix, which means for all $t \in R$, $G_{n\times n}(t)$ is non-singular. Since we are considering DAE instead of ODE, some equations of the system may be algebraic, and $C_{n\times n}(t)$ may not be non-singular. We set the rank of $C_{n\times n}(t)$ is r, in which $r <= n$ and it is true for all $t \in R$.

$$\frac{d}{dt}(C_{n\times n}(t)x(t)) = -G_{n\times n}(t)x(t) \tag{3.27}$$

The homogeneous part of (3.25) can be extracted and written in (3.27). Just like in LTI case, the homogeneous system in (3.27) is the core part of the corresponding system (3.25)(3.26). A concept for the LTPV system called fundamental matrix $X(t)$ [12] is defined in (3.28).

Fundamental matrix for the LTPV system:

$$X_{n\times n}(t) = [x_1(t), x_2(t), x_3(t)..., x_r(t), 0..., 0]_{n\times n} \tag{3.28}$$

In (3.28), each $x_i(t), i = 1, 2, 3.., r$ is a $n \times 1$ column vector and a solution to the system (3.27). All $x_i(t)$ are linear independent to each other. The last $(n - r)$ columns of $X(t)$ are zero-vectors since $C(t)$ has a rank of r. Actually all $x_i(t)$ form a set of basis to the solution space of (3.27), which means any solutions of (3.27) can be represented as a linear combination of the columns of $X(t)$.

A main contribution of the Floquet theory is the Floquet Theorem. The Floquet Theorem for periodic DAE case is described as following [12]:

Floquet Theorem: The fundamental matrix $X(t)$ of the DAE in (3.27) can be

written in the form:

$$X(t) = R(t) \begin{bmatrix} e^{Qt} & 0 \\ 0 & 0 \end{bmatrix} \tag{3.29}$$

$R(t)$ is a $n \times n$ bounded, non-singular, periodic matrix, in which period is T and $R(t) = R(t+T)$. Q is a $r \times r$ non-singular constant matrix.

The eigenvalues Λ of Q are called the characteristic exponents of the system [12]. They are the same as the poles in the LTI system. If the real parts of all characteristic exponents are less than 0, $X(t)$ has a decay behavior as time grows, which means the system is stable.

3.2.2 Extract characteristic exponents from the LTV transfer function

The characteristic exponents can also be computed using the LTPV transfer function $H(j\omega, t)$. To demonstrate it, another theorem needs to be introduced.

Theorem [12]: An equivalent realization of the LTPV system described by the DAE (3.25)(3.26) exists, the system parameters transform to $G' = \begin{bmatrix} -Q & 0 \\ 0 & I_{(n-r)\times(n-r)} \end{bmatrix}$, $C' = \begin{bmatrix} I_{r\times r} & 0 \\ 0 & 0 \end{bmatrix}$ via the periodic change of variable: $z(t) = R^{-1}(t)x(t)$. And the DAE of the equivalent system is:

$$\frac{d}{dt}(C'_{n\times n}x(t)) = -G'_{n\times n}z(t) + B_{n\times 1}R^{-1}(t)u(t) \tag{3.30}$$

$$y(t) = L^T_{n\times 1}R(t)z(t) \tag{3.31}$$

This is a beautiful theorem. It means that all LTPV systems have their own equivalent realizations that all time-varying behaviors are just showing at the input/output parts, while the core parts of them are constant and work as LTI systems.

If we do Fourier expansions to $R(t)$ and $R^{-1}(t)$, the LTPV transfer function $H'(s,t)$ representing this DAE can be computed as (3.32)(3.33).

$$H'(s,t) = \sum_{i=-\infty}^{\infty} H'_i(s) e^{ji\omega_0 t} \tag{3.32}$$

$$H'_i(s) = L_i'^T [(G'_{FD} + \Omega C'_{FD}) + sC'_{FD}]^{-1} B'_{FD} \tag{3.33}$$

Since G' and C' are constant, all harmonics except DC components of them are 0, which makes G'_{FD} and C'_{FD} diagonal matrices as shown in (3.34)(3.35). It gives benefits for computing the poles of this equivalent realization, which are also the poles of the original system.

$$G'_{FD} = \begin{bmatrix} \ddots & & & & & & \\ & G' & & & & & \\ & & G' & & & & \\ & & & G' & & & \\ & & & & G' & & \\ & & & & & G' & \\ & & & & & & \ddots \end{bmatrix} \tag{3.34}$$

$$C'_{FD} = \begin{bmatrix} \ddots & & & & & & \\ & C' & & & & & \\ & & C' & & & & \\ & & & C' & & & \\ & & & & C' & & \\ & & & & & C' & \\ & & & & & & \ddots \end{bmatrix} \tag{3.35}$$

To get the poles, the following generalized eigenvalue problem needs to be solved.

$$(G'_{FD} + \Omega C'_{FD})X = \lambda C'_{FD} X \tag{3.36}$$

$$(G' + ij\omega_0 C')X = \lambda C' X, i = ..., -3, -2, -1, 0, 1, 2, 3, ... \tag{3.37}$$

Since all matrices are diagonal, (3.36) can be split into several sub-problems, each of them has the form like (3.37). The only difference between these sub-problems is the value of i. It means that if p is a pole by solving a sub-problem like (3.37), $p \pm ji\omega_0$ will be poles in other sub-problems. Since the union of these poles are the poles of the system, we can conclude that if p is a pole of the system, $p \pm ji\omega_0$ are also the poles of the system. This result is really useful for computing and verifying the poles of an LTPV system.

To be more specific, we can write down (3.37) using their realistic forms, which is shown in (3.38).

$$\left(\begin{bmatrix} Q & 0 \\ 0 & I_{(n-r)\times(n-r)} \end{bmatrix} + ij\omega_0 \begin{bmatrix} I_{r\times r} & 0 \\ 0 & 0 \end{bmatrix}\right)X = \lambda \begin{bmatrix} I_{r\times r} & 0 \\ 0 & 0 \end{bmatrix} X \tag{3.38}$$

It can be concluded that the poles of an LTPV transfer function are the characteristic exponents of the system, which means that $H(j\omega, t)$ contains the stability information of the corresponding LTPV system. This property is similar to the LTI case.

3.2.3 Loop-based noise sensitivity analysis for LTPV circuits

As we described before, each LTPV transfer function can be decomposed into LTI systems followed by memory-less multiplications with $e^{ji\omega_0 t}$. Each harmonic transfer function $H_i(s)$ has the same form as the LTI transfer function $H(s)$ if the concept of

equivalent G and C is used. Actually the concept of "noise-sensitive pole" can also be applied to $H_i(s)$.

If $H(j\omega, t)$ represents the node impedance of an LTPV circuit, it can be written as (3.39)(3.40).

$$H(s,t) = \sum_{i=-\infty}^{\infty} H_i(s) e^{ji\omega_0 t} \tag{3.39}$$

$$H_i(s) = \sum_{j=1}^{N_C} \frac{res_j(s)}{s^2 + 2\zeta_j \omega_{0j} s + \omega_{oj}^2} \tag{3.40}$$

In (3.40), just like (2.13), each $H_i(s)$ can be factorized into the combination of several second order systems(first-order subsystems are ignored since they won't contribute to the noise-sensitive behaviors). If p is a noise-sensitive pole of $H_i(s)$, $H_i(s)$ will have a peaking at $\omega = \omega_p$, which is the natural frequency corresponding to p. However, it becomes much more complicated when we want to define "noise-sensitive loops" for LTPV systems. The main reason is that LTPV systems produce harmonic frequency shifts.

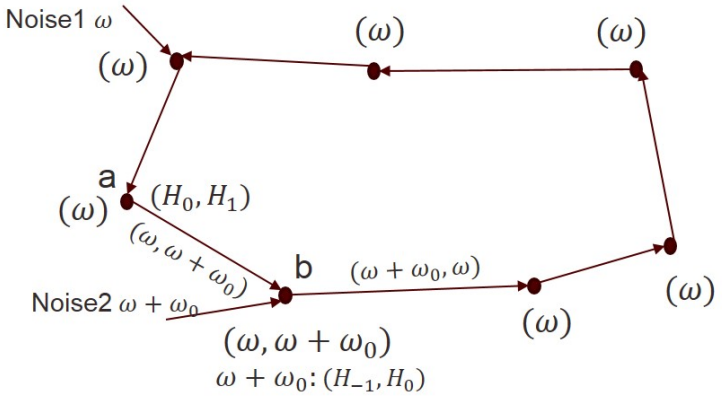

Figure 3.1: A noise-sensitive loop in an LTPV circuit

Figure 3.1 can be used to illustrate the problem. This graph shows a group of nodes in an LTPV circuit. The corresponding natural frequency of a noise-sensitive pole is used to represent the pole itself. At first, each circuit node has a noise-sensitive pole ω in their DC component $H_0(s)$, which means that if the input frequency is near ω, the output of each node has a noise-sensitive component with the same frequency. Thus, these circuit nodes can be grouped to form a "noise-sensitive loop", which is similar to the LTI case. Besides this noise-sensitive loop, there are more noise-sensitive loops in this LTPV circuit. If ω is also a noise-sensitive pole of $H_1(s)$ of node a, the output of node A will also have a noise-sensitive component with freqency near $\omega + \omega_0$. At the same time, let's denote $\omega + \omega_0$ is a noise-sensitive pole of $H_{-1}(s)$ of node B, which means that if the input frequency is near $\omega + \omega_0$, the output of B will have a noise-sensitive component near ω. In this case, the input noise signal will also have another "path" to traverse through node A and B: a signal with frequency ω is injected to A, its frequency becomes $\omega + \omega_0$ and serves as the input of node B, when it comes out from B, its frequency goes back to ω and the signal can traverse through other nodes. This process can be treated as the injected noise goes through another noise-sensitive loop with frequency shifting. Even the circuit node set is the same as the previous case, we say that there are two different noise-sensitive loops in the circuit.

This instance shows that for an LTPV system, a noise-sensitive loop may get involved with several different frequencies, that is, different noise-sensitive poles. The relationship of these frequencies is that they are harmonic frequencies to each other. A concept called "noise-sensitive pole group" is defined to clarify this fact.

Definition(Noise-sensitive pole group): A noise-sensitive pole group is a set of noise-sensitive poles in which any two of them are harmonic frequency to each other.

This concept is defined for the whole LTPV transfer function, a noise-sensitive

pole of any $H_i(s)$ will be grouped into a noise-sensitive pole group of the system.

Based on the definition of the noise-sensitive pole group, the noise-sensitive loop concept for LTPV systems can also be defined.

Definition(Noise-sensitive loop in LTPV circuits): a noise-sensitive loop is the maximum loop formed by the circuit nodes which have noise-sensitive poles in the same noise-sensitive pole group.

There are two kinds of noise-sensitive loops: one is the noise-sensitive loops without frequency shifting, which is similar to the LTI case; the other is the noise-sensitive loop with frequency shifting, it only exists in the noise-sensitive pole group with more than 1 noise-sensitive poles.

One thing needs to be noted is that even if two loops are using the same set of circuit nodes in some cases, if they are formed because of different frequency shifts, they are considered to be 2 different noise-sensitive loops.

Since the frequency shifting is considered for noise-sensitive loop identification in the LTPV system, simply grouping the nodes to form noise-sensitive loops cannot be applied. A more complicated algorithm is needed for detecting noise-sensitive loops in this case.

3.2.4 Noise-sensitive loop detection algorithm for LTPV systems

Let's restate the problem needs to be solved: Given a set of circuit nodes, find the maximum noise-sensitive loops for each noise-sensitive pole group.

To solve this problem, a circuit can be mapped to a graph, then the problem will transform into finding the maximum loops in a graph. How to map the circuit to an appropriate graph is the most critical thing. An algorithm will be discussed in the following. It gives the procedure of mapping a circuit to a graph $G(V, E)$ in which noise-sensitive loops can be detected. In the graph G, V is the set of nodes, while E

is the set of edges.

For a given circuit, a graph is built for each noise-sensitive pole group. To illustrate how the mapping process works, we consider a simple case of two circuit nodes A, B, and the noise-sensitive pole group is $(\omega, \omega + \omega_0)$. The two poles in this group are the noise-sensitive poles of $H_1(s)$ and $H_{-1}(s)$ for both node A and B.

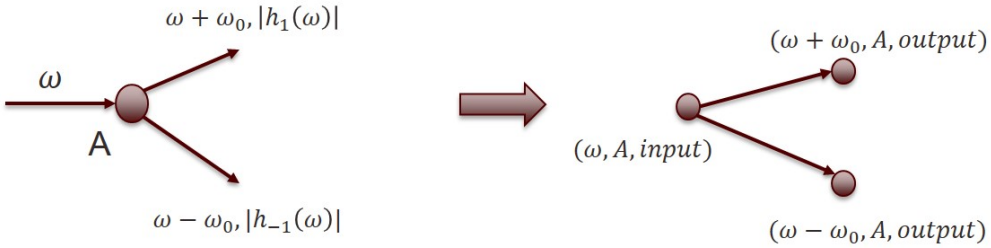

Figure 3.2: Node A with noise-sensitive pole ω

For circuit node A with input frequency ω, its mapping process is shown in Figure 3.2. It can be mapped to a small graph in $G(V, E)$. Each node in the graph has 3 properties: name, frequency and type. "name" is the name of the corresponding circuit node in the circuit. "frequency" is the input or output frequency of a circuit node. "type" describes which part of a circuit node the graph node is representing, that is, the input or output part. In a similar manner, the mapping of node A with noise-sensitive pole $(\omega + \omega_0)$ can be described in Figure 3.3.

Assuming node B is in the same situation as A, it can be mapped to small graphs in $G(V, E)$ using the same policy. There will be 4 small graphs in $G(V, E)$. Then the algorithm will connect those small graphs. The criteria is that if two nodes have the same frequency but different names and types, there will be an edge from the "output" node to the "input" node, which is represented by the dash line in Figure

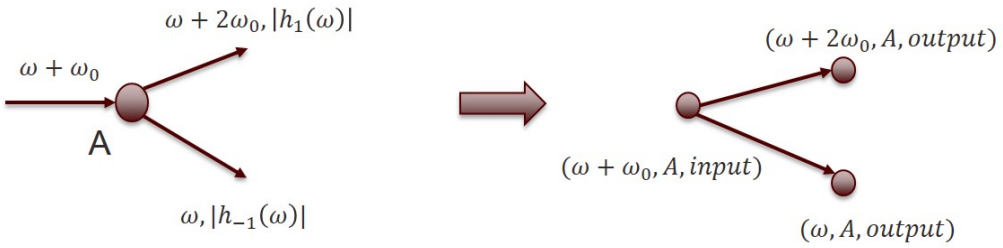

Figure 3.3: Node A with noise-sensitive pole $\omega + \omega_0$

3.4. In this example, there are two noise-sensitive loops formed by these two nodes with different frequency shifts.

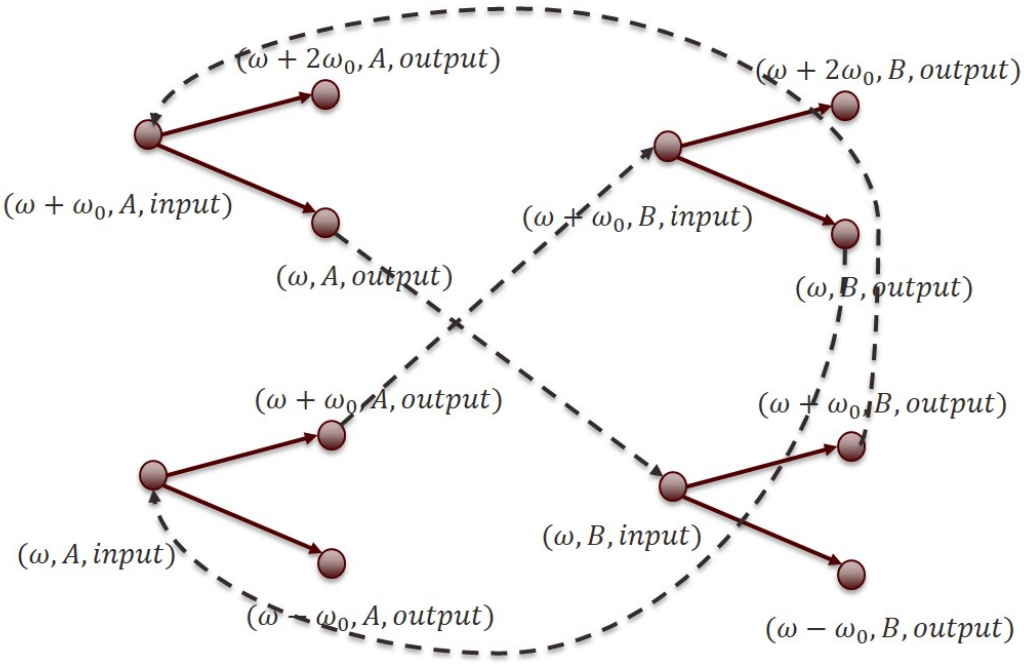

Figure 3.4: The whole graph for a noise-sensitive pole group

The pseudo code of the graph building algorithm is in Algorithm 1. It has two steps. Firstly, for each node in the circuit, a corresponding small graph is estab-

Algorithm 1 Graph building Algorithm

```
1: procedure NODE_GRAPH G(V, E) = GRAPH BUILDING ( NOISE-SENSITIVE
   POLE GROUP H, CIRCUIT A )
2:     for each node n in A do
3:         for each noise snsitive pole p in H do
4:             Build_small_graph(n, p);
5:         end for
6:     end for
7:     for each node n and n.type == output do
8:         for each node m and m.type == input do
9:             Connect(n, m);
10:        end for
11:    end for
12: end procedure
```

lished. Then the algorithm will connect small graphs as described before. The two sub-routines are shown in Algorithm 2 and Algorithm 3.

Algorithm 2 Small graph building

```
    procedure BUILD_SMALL_GRAPH(N,P)
2:      Add vi(input, n.name, ω_p) to V
        for each harmonic inpedance function do
4:          if p is a dominant pole for it
            Add vo(output, n.name, ω_har) to V
6:          Add edge < vi, vo > to E
            end if
8:      end for
        If no vo added, delete vi
10:     end if
    end procedure
```

In this way, the loops in G(V, E) are all noise-sensitive loops of the original circuit. Finding the largest loops in G is equivalent to detecting the noise-sensitive loops in the circuit for a noise-sensitive pole group.

Algorithm 3 Connect small graphs

procedure CONNECT(N,M)
    If $n.name! = m.name$ and $n.freq = m.freq$
3: Add $edge < vi, vo >$ to E
    end if
end procedure

The graph G(V,E) has some features which can help to improve the efficiency of finding loops in it. Firstly, the graph is a directed graph. Then the maximum length of loops in G(V,E) is $2n$, in which n is the circuit node number in the original circuit. Last but not the least, since all edges are between "input" and "output" nodes, the graph is a bipartite graph, the lengths of all loops are even.

An algorithm for finding the maximum loops in $G(V, E)$ is developed based on the work in [13]. The flow chart of the algorithm is described in Figure 3.5.

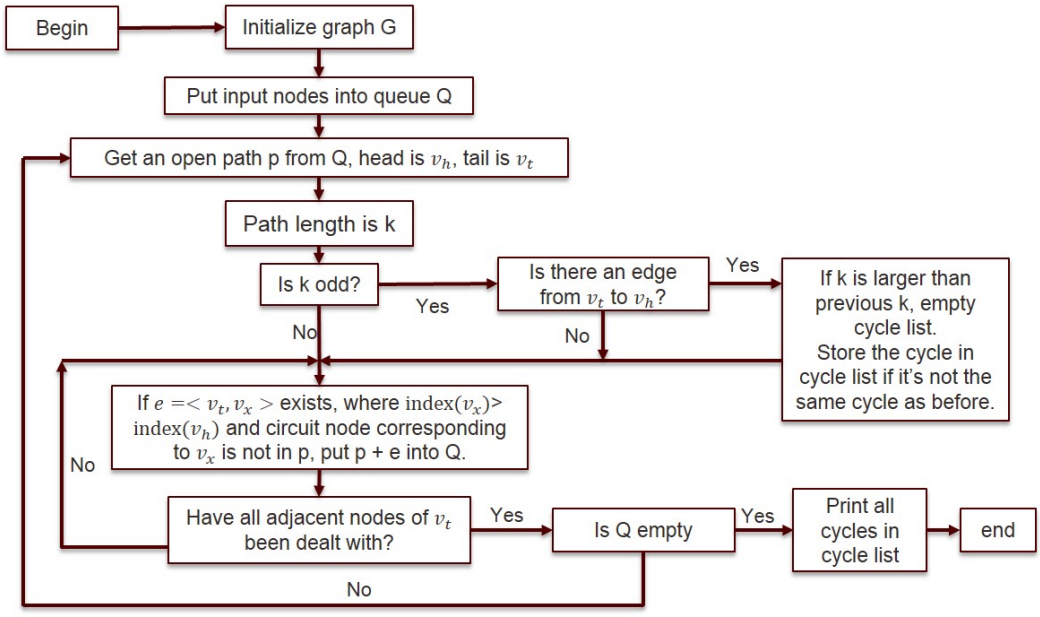

Figure 3.5: Flow chart of maximum loop finder algorithm

Before introducing the algorithm, some definitions need to be introduced first [13]. For a directed graph G(V,E), a path is an alternating of vertices and edges, beginning and ending with a vertex, i.e.: $v_1e_1v_3e_2v_4$. The length of a path is the number of edges in the path. A simple path is a path that all vertices except head and tail are distinct. A cycle is a simple path that head and tail are the same. An open path is a simple path that is not a cycle. a K-cycle is a cycle with length k.

The main idea of the algorithm is building k-cycles from $(k-1)$ simple paths iteratively. A queue is used to store identified paths in the graph. Nodes are ordered by their index in the graph to avoid detecting repeated loops. "input" nodes are put into the queue at first, then for each path(contains only one node at beginning) in the queue, the adjacent list of the tail node will be searched to form longer paths. At the same time, if there is an edge between the head and tail, a cycle is identified. The cycles are recorded in the *cycle_list*, which is a list. If a longer cycle is detected, the previous cycles can be discarded. The final cycles left in *cycle_list* are the maximum loops in the graph and can be mapped back to the loops in the circuit. These loops can be reported as noise-sensitive loops in a noise-sensitive pole group of the LTPV circuit.

A good feature of the algorithm is that it doesn't need to explore the whole graph so that number of cases needed to be numerated is reduced. What's more, the algorithm is good for the parallel realization since all paths can be handled independently. These features give the algorithm a great potential for finding noise-sensitive loops in the LTPV circuits efficiently.

3.3 Noise-sensitive loop identification algorithm for LTPV circuits

To sum up, the noise-sensitive loop identification algorithm for LTPV circuits can be described as following:

- Initialize the objective LTPV circuit and compute its G_{FD} and C_{FD} matrices. In our realization, this part is independent. Its efficiency is mainly determined by the circuit simulator we use. The produced circuit node name list and matrices will work as the input of the rest phases.

- Compute noise-sensitive poles, which are used to identify noise-sensitive pole groups for the whole system.

- Calculate harmonic node impedances for each node in the circuit.

- Build a graph for each dominant pole group of the circuit.

- Perform maximum loop detection algorithm on each graph and output all cycles left.

- Map the cycles back to noise-sensitive loops in the circuits, report all noise-sensitive loops to designers.

Designers may have some insights about the noise-sensitive loops reported and can make further improvements to the circuits.

3.4 Time complexity analysis

There are mainly 3 phases in our algorithm for noise-sensitive loops identification in LTPV circuits. They are pole computation, impedance computation and noise-sensitive loop identification.

For the pole computation, the classic QZ method [4] is used. The time complexity of the algorithm is $O(n^3)$, in which n is the size of the corresponding system. In our case, the size of the system is the actual size of G_{FD} and C_{FD}. If the size of $C(t)$ and $G(t)$, which is mainly decided by the circuit node number, is set to be N, while the harmonics we reserved in truncated G_{FD} and C_{FD} are m at each side of DC

component, the time complexity of QZ method in our application is $O([(2m+1)N]^3)$. Since the value of m are always fixed and set to be around 8, the time complexity of this phase is mainly determined by the circuit scale.

For the impedance computation part, we need to compute the impedances for each circuit node at each noise-sensitive pole in each harmonic impedance transfer function $Z_i(s)$. According to (2.16), each computation operation takes $O(n^2)$ time since the multiplication of a matrix and a vector needs to be performed. n is the size of matrices and can be expressed as $n = (2m + 1)N$ as described before. If the circuit node number is set to be num_cir, while the number of noise-sensitive pole is num_poles and number of harmonic transfer function is har_num. The time complexity of these phase is $O([(2m + 1)N]^2 \times num\_cir \times num\_poles \times har\_num)$. num_poles is mainly decided by the stability performance of the circuit and cannot be predicted, but we can expect that a well-designed circuit normally have just a few noise-sensitive poles. At the same time, har_num is set to be 2 in our case. It means the time complexity of the impedance computation is mainly determined by the circuit scale, which is the same as the previous phase.

For the noise-sensitive loop identification part, the worst case for time complexity will be $O(exp(V))$, which means it is an exponential time complexity problem. In the worst case, the graph is a complete graph and V is the number of vertices in the graph. However, since the sizes of vertices and edges in the graph we build are mainly decided by the noise-sensitive part of the circuit and usually small comparing with the circuit size itself, the worst case hardly happens. For a well-designed circuit, this phase is normally the least time-consuming part.

4. EXPERIMENT RESULTS

In this chapter, we shall present three applications of the noise-sensitive loop identification algorithm for LTPV circuits. We first adopt a simple LTPV RLC network, to intuitively demonstrate the application of the proposed algorithm and show what kinds of noise-sensitive loops we can expect in an LTV circuit. The second circuit example is a double-balanced mixer with parasitic effects, which is used to show that the algorithm can help to identify the noise-sensitive loops formed by the parasitics in a time-varying circuit network. The last circuit example is a switch capacitor gain stage with classic two-stage Op-amp and NMOS switches. This example is used to show that our algorithm is able to aid the design process for implementing stable LTPV circuit blocks. The last two applications are designed using a commercial 90nm technology with 1.2V supply.

4.1 Parameter settings

A C++ implementation of our algorithm has been realized in Linux environment. Before discussing the circuit examples, we may firstly define and set some parameters in our algorithm.

- $Freq\_range\ \omega_i$. This parameter describes the frequency range we are interested for detecting noise-sensitive poles. In our applications, $\omega_i = [1, 10G]rad/s$, which is about $(0.15 - 1.5GHz)$.

- $Har\_bound\ b$. This parameter indicates the number of harmonic frequencies we are concerning. In our applications, it is set to be 2, which means that we are considering 2 harmonics at each side of DC component of each node impedance transfer function.

- *Har_err_tol* e. This parameter is the threshold value for judging if two frequencies ω_1, ω_2 are harmonic to each other. If $\frac{\omega_1-(\omega_2+i\omega_0)}{min(\omega_1,\omega_2)} < e$, for any $i = -b, ...0, ...b$, ω_1 and ω_2 are treated as harmonic to each other. In our applications, $e = 0.01$.

There are some other parameters may vary among different cases, and will be discussed later.

4.2 A simple LTPV RLC circuit network

Before discussing the LTPV RLC network, we may first see an LTI circuit.

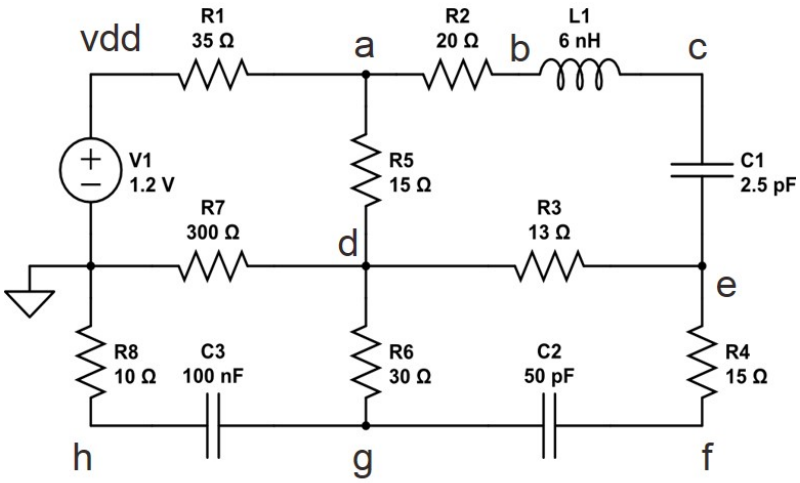

Figure 4.1: An LTI RLC network

Figure 4.1 depicts an LTI RLC ciruit. In this circuit, there are four small loops and a large loop formed by resistors, capacitors and inductors. The poles of this circuit are shown in Table 4.1. There are totally 4 poles, out of which two real poles p_3 and p_4 are mainly caused by $C3$, $C2$ and their surrounding resistors, while $C1$, $L1$ and the surrounding resistors mainly contribute to the complex pole pair $p_{1,2}$. The

natural frequency ω_0 of the complex pole pair $p_{1,2}$ is $1.302GHz$, and the damping factor ζ is 0.3947. Since $\omega_0 < 1.5GHz$ and $\zeta < 0.7$, $p_{1,2}$ is actually a dominant pole of this LTI circuit.

Table 4.1: Pole list of the LTI RLC network

Name	Re	Im
p_1	-3.2298e9	7.5183e9
p_2	-3.2298e9	-7.5183e9
p_3	-2.9575e8	0
p_4	-1.7239e5	0

By running the Loop Finder algorithm for LTI circuits [3], a noise-sensitive loop can be identified for the noise-sensitive pole $p_{1,2}$. The largest loop in the circuit (a, b, c, e, f, g, h) is the noise-sensitive loop. For the LTI case, our algorithm can also be used to detect noise-sensitive loops since LTI circuit is a specific case of LTPV circuit. And by using our algorithm, the same noise-sensitive loop is identified.

To create an LTPV case, we added some time-varying behaviors to the LTI circuit in Figure 4.1. Four constant resistors $R2$, $R6$, $R7$ and $R8$ are changed to resistors with sine wave time-varying behaviors. At first, weak time-varying behaviors are added to the circuit, in which $R2 = 20{+}2sin2\omega t$, $R6 = 30{+}2sin\omega t$, $R7 = 300{+}20sin2\omega t$, $R8 = 10{+}2sin\omega t$. ω is the basic operation frequency of the circuit and its value is $500MHz$. The purpose of adding such kinds of time-varying behaviors to some resistors is that we want the whole circuit to be operated at both the fundamental frequency ω and a harmonic frequency 2ω. Since the sine wave is periodic and circuit is formed by linear devices resistors, capacitors and inductors only, the circuit becomes a linear time-periodically varying circuit. For this case, since the time-varying behavior is really weak, we expect that the noise-sensitive loop should be the same as the LTI

case with a reasonable small error. Actually the result given by our algorithm is the same as what we expect. The only noise-sensitive pole in the circuit is as Table 4.2. We can find that the noise-sensitive pole is almost the same as the one in LTI case. This case demonstrates that our algorithm is accurate to capture the noise sensitivity information of LTPV circuits.

Table 4.2: Noise-sensitive pole list of the near LTI RLC network

Name	Re	Im	ω_0	ζ
p_1	-3.2296e9	7.5171e9	1302.1261MHz	0.3947

To make the time-varying behavior stronger, we increase the magnitude of the sine waves in resistors. For instance, we set $R_2 = 20 + 19.9sin2\omega t$, $R_6 = 30 + 29.9sin\omega t$, $R_7 = 300 + 299.9sin2\omega t$, $R_8 = 10 + 9.9sin\omega t$. Our algorithm can be further used to analyze the noise sensitivity of this circuit.

From the previous chapter we know that for an LTPV system, if p is a pole, then $p \pm ij\omega_0$ are also poles of it. If there is noise-sensitive pole p in the circuit, it's very likely that $p \pm ij\omega_0$ will also be noise-sensitive poles for the circuit. Compared to LTI circuits of the same size, LTPV circuits normally have more noise-sensitive poles and those poles can form different noise-sensitive pole groups.

For this LTPV case, there are two noise-sensitive pole groups and 2 noise-sensitive loops identified.

Firstly, there is a noise-sensitive pole group with 2 noise-sensitive poles, and their values are shown in Table 4.3. a noise-sensitive loop with frequency shifting exists in this group. It can be expressed as [h(503MHz, 503MHz),d(503MHz, 1GHz),a(1GHz, 503MHz),f(503MHz,503MHz),g(503MHz,503MHz),h]. In this noise-sensitive loop,

natural frequencies $1GHz$ and $503MHz$ are get involved and there are frequency shifts at node a and d.

Table 4.3: Pole list of the first noise-sensitive pole group

Name	Re	Im	ω_0	ζ
p_1	-4.5063e8	-6.2837e9	1002.6472MHz	0.0715
p_2	-4.5198e8	-3.1416e9	505.192MHz	0.1424

If we want to figure out the reason for the appearance of this noise-sensitive pole group and the corresponding noise-sensitive loop, we firstly have a look at the imaginary parts of these poles. We can write these poles as $p_r + jp_i$. Actually for all these poles, their p_i can be written as $p_i = i\pi \times \omega$, in which ω is the fundamental frequency of the system. According to the property of the poles in LTPV systems, there must be a real pole with value p_r in the system, too. It means that because of the special property of poles in the LTPV system, more potentially noise-sensitive poles may appear in the circuits. Besides, since the frequencies of these two poles are very close to the fundamental frequency $500MHz$ and the harmonic frequency $1GHz$ of the system, the peaking behaviors of the corresponding second order systems some nodes, such as a and d, can be dominant near the two frequencies. These two aspects are combined to make this noise-sensitive loop show up in the circuit.

Besides the above noise-sensitive pole group, there is another group with only one noise-sensitive pole in it. Its information is shown in Table 4.4. We can find that only noise-sensitive loop with no frequency shift exists. The noise-sensitive loop is actually corresponding to the original noise-sensitive loop in the LTI case.

Table 4.4: The second noise-sensitive pole group

Name	Re	Im	ω_0	ζ	Noise-sensitive loop
p_3	-3.2227e9	-7.3839e9	1282.2361MHz	0.4000	(a,b,c,e,f,g,h)

4.3 A double-balance mixer with parasitic effects

Double-balance mixers are widely used in RF communication systems for frequency conversion. Just as described before, the small signal model of mixers can be modeled as LTPV systems. A double-balanced mixer itself has no feedback loops. But because of the existence of parasitics, some feedback loops may be formed by transistors and parasitics together. If such a circuit is time-invariant, noise-sensitive loops can be easily identified using the existing algorithm. However, since there are time-varying behaviors in mixers, our method is really needed to help identify noise-sensitive behaviors for this kind of circuits.

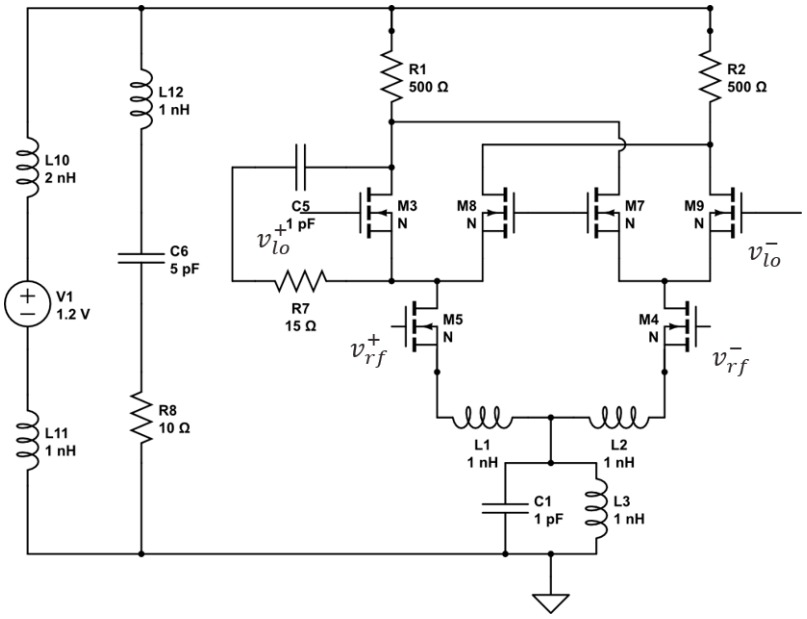

Figure 4.2: A double balanced mixer with parasitic effects

Figure 4.2 shows a Gilbert cell double balanced mixer with parasitic effects. The inductors that are series with the supply and the series RLC path are added to model the wirebonds used to connect an integrated circuit to the leadframe of a typical package [3]. The series resistor $R7$ and capacitor $C5$ are added to model the parasitics between two nodes in the circuit. Local signal V_{LO^+} and V_{LO^-} are sine signal with peak-to-peak value $600mV$ and frequency $500MHz$, which means the fundamental frequency of this LTPV system is $500MHz$.

By running our algorithm, there are two noise-sensitive loops without frequency shifting are identified. Their corresponding noise-sensitive pole information is shown in Table 4.5.

Table 4.5: Two noise-sensitive pole groups in the mixer

Name	Re	Im	ω_0	ζ
p_1	-1.2430e9	-6.8700e9	1.111GHz	0.1781
p_2	-4.6736e9	-7.8508e9	1.454GHz	0.5115

The first noise-sensitive loop of this circuit is highlighted in the schematic view in Figure 4.3. We can find that the supply-related nodes are in this loop, which means the inductor wirebonds and their parasitics can potentially cause noise-sensitive behavior in the circuit if noise injected. At the same time, some nodes in the mixer itself are also get involved. Since the loop contains time-varying behavior, it cannot be identified using previous methods. In this case, running our algorithm is a necessary. Normally in "real-life" cases, the dumping ratio of the noise-sensitive pole may be larger than 0.1781, but they may still result in excessive supply ringing in circuit designs and need careful consideration [3].

The second noise-sensitive loop is highlighted in Figure 4.4. It's caused mainly

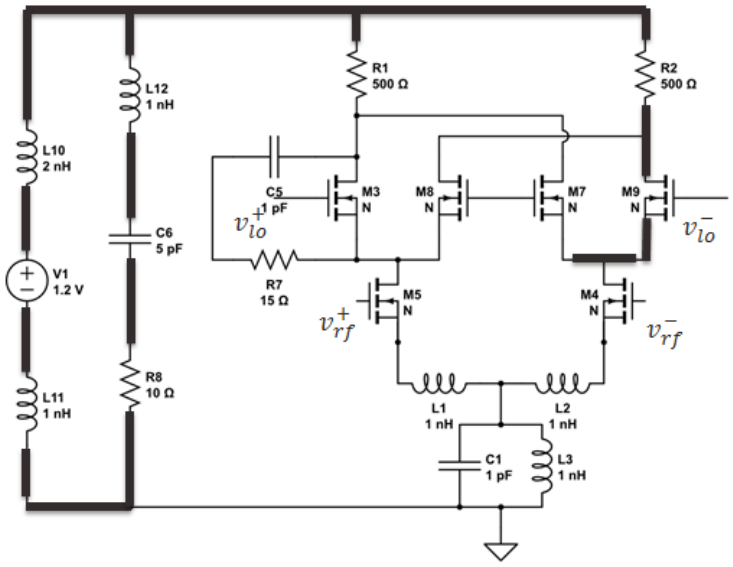

Figure 4.3: The first noise-sensitive loop

by the parasitic resistor $R7$ and capacitor $C5$. These kinds of parasitics may happen between the nodes in lots of LTPV circuits. Since there are time-varying behaviors between the two nodes a and b, only our method can be used to capture the noise-sensitive behaviors.

4.4 A switch capacitor gain stage

Figure 4.5 shows a switch capacitor gain stage. All switches are NMOS type, and the core part of this circuit is a classic two-stage Op-amp with miller compensation, which is shown in the dashed circle. In Figure 4.5, $C1$ is used for compensation to ensure that the Op-amp has enough phase margin so that the negative feedback loop in the circuit is stable, which refers to noise-sensitive in our background. Op-amps are also always used in LTI circuit blocks. This circuit is used to demonstrate that our loop-based noise sensitivity analysis algorithm can also help to identify if the phase margin of the Op-amp is enough for the negative feedback loop for this LTPV

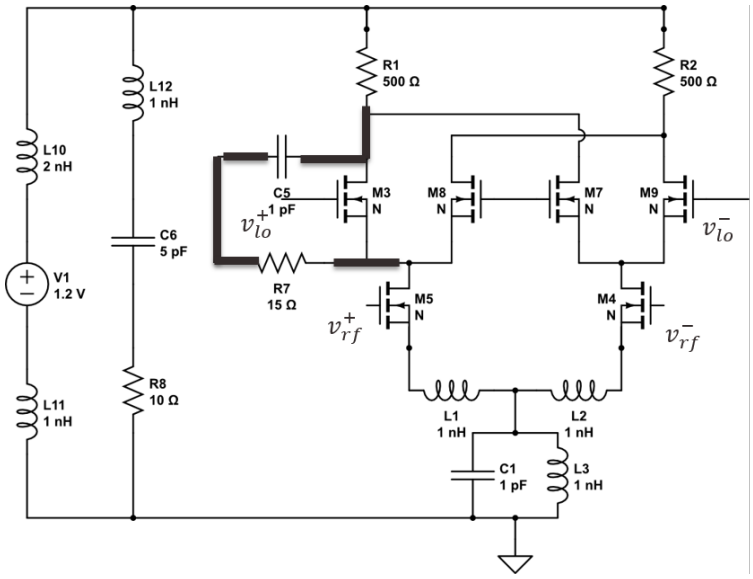

Figure 4.4: The second noise-sensitive loop

circuit to be noise-sensitive.

This circuit is time periodically-varying because of the switches. The NMOS switches in the circuit are driven by a two-phase non-overlapping clock signal as shown in Figure 4.6. The signal makes the state of the circuit periodically changing. The function of this circuit is sampling the input signal from vi at the rising edge of $phi1$ and output it at the rising edge of $phi2$ so that the input signal is discretized. The frequency of the clock signal is $1MHz$, while the rising and falling time of it are $10\mu s$.

To design this circuit, a proper Op-amp is needed to be designed first. In this design, what we care is whether the Op-amp has enough phase margin or not. By performing AC analysis on the Op-amp, we firstly got that the phase margin of this Op-amp is 62 degrees when the load capacitor is $1pF$. By running our algorithm on this switch-capacitor circuit, 2 noise-sensitive pole groups are identified and only one pole exists in each of them separately, which means there are 2 noise-sensitive loops

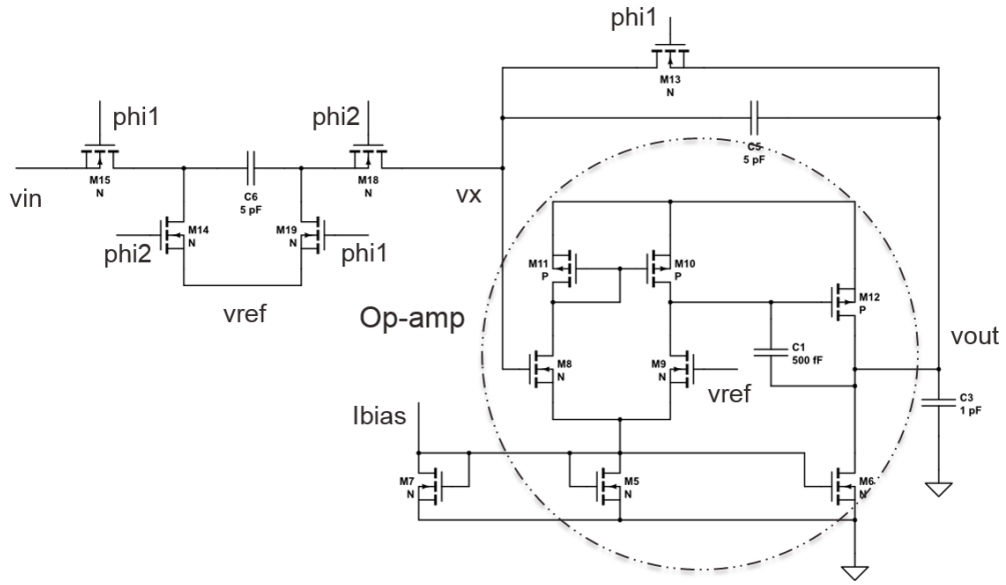

Figure 4.5: A switch capacitor gain stage

in the circuit.

Table 4.6: The first noise-sensitive pole group

Name	Re	Im	ω_0	ζ
p_1	-1.147e7	1.524e7	3.035MHz	0.6015

The first noise-sensitive pole group are shown in Table 4.6. The noise-sensitive loop with frequency shift in this group in highlighted in Figure 4.7. We can find that this noise-sensitive loop contains the nodes at the two sides of switches. It's reasonable since these nodes have dramatically impedance changes due to the varying magnitudes of the clock signals and the natural frequency of the pole are located at the vicinity of the harmonics of $1MHz$, which is the fundamental frequency of the circuit. To make the noise-sensitive behavior not that obvious, the clock signals with

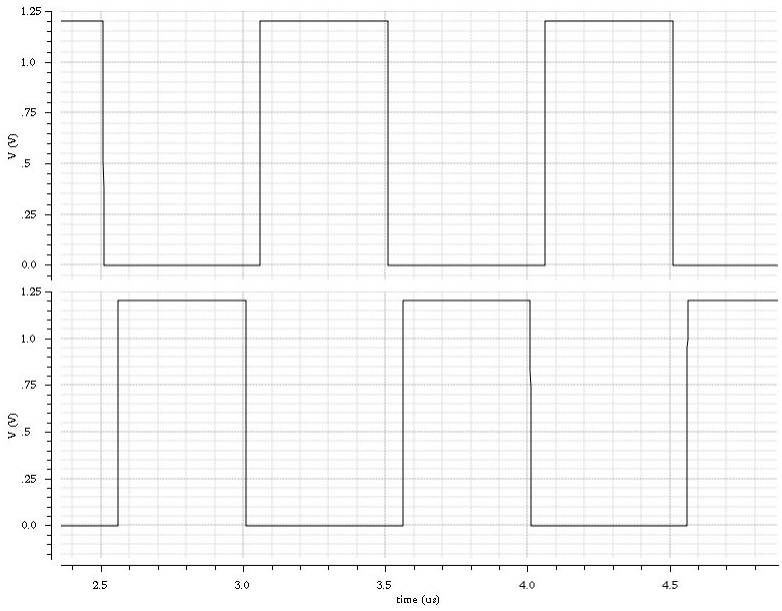

Figure 4.6: Non-overlap clock signal

relatively slow rising and falling edges may need to be applied.

The second noise-sensitive pole group and its noise-sensitive loop is different from the others. The dominant pole information and the loop is marked in Figure 4.8. We can find that the loop is actually the negative feedback loop formed by the Op-amp and the feedback capacitor $C5$. Since a switch is in parallel with $C5$, the loop has time-varying behavior. This noise-sensitive loop is formed because of its corresponding noise-sensitive pole. If we look into the pole deeper, we can find its damping factor ζ is 0.69, which is less but very close to 0.7.

If we recall the basic concept of the stability of second order systems, for LTI systems, the phase margin of the corresponding Op-amp can be expressed as (4.1) using ζ [4]. A damping factor ζ with value 0.7 corresponds to a phase margin of 65 degrees. A ζ with value 0.69 means that the phase margin is very close to 65 degrees but less than it. Actually the noise-sensitive loop identified by our algorithm can

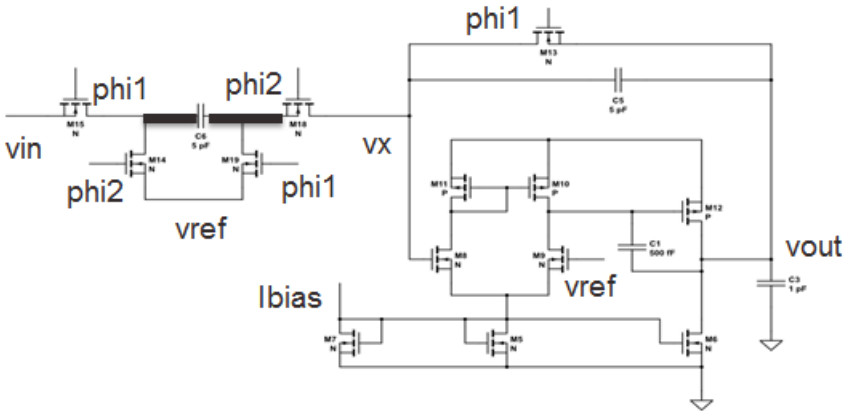

Figure 4.7: A noise-sensitive loop contains switch nodes

be treated as the LTI noise-sensitive part of the circuit. We can use the concept in LTI circuit to explain this noise-sensitive loop. Thus, to fix this noise-sensitive loop, the value of $C1$ can be slightly increased so that the phase margin of the Op-amp becomes larger to make the whole loop noise-insensitive.

$$\phi_{pm} = tan^{-1}(2\zeta(\frac{1}{(4\zeta^4+1)^{1/2}-2\zeta^2})^{1/2}) \tag{4.1}$$

After c increases to $600fF$, the second noise-sensitive loop disappears, and the only noise-sensitive behavior of the circuit is from the switches.

For the LTPV negative feedback loop as shown in Figure 4.8, it can be treated as a loop with time-varying loads. Traditionally, we cannot solve this kind of loop at one time since the circuit contains time-varying behaviors. However, by using our method, the switch capacitor circuit can be evaluated as a whole, the time-varying loop can be directly deal with and the noise-sensitive behavior can be measured easily. Designers can selectively adopt the information that we offer to further improve the circuit's performance.

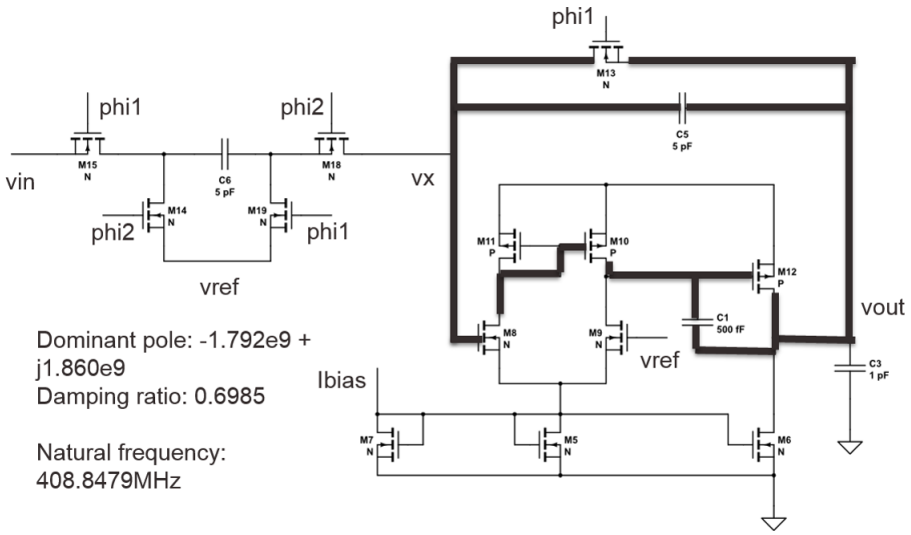

Figure 4.8: Noise-sensitive negative feedback loop

4.5 Verification of noise-sensitive loops in time domain

In the previous sections, we perform our algorithm on three circuit examples. However, to demonstrate their correctness, we need to verify these results from a different point of view. In this section, we do transient simulation and try to observe these noise-sensitive loops from time-domain directly to prove that our algorithm is correct. The double balanced mixer is used to illustrate the verification since its noise-sensitive behaviors are relatively apparent for observation.

Figure 4.9 shows the circuit we use. As we introduced before, there are two noise-sensitive loops without frequency shifts in the mixer. Their pole information is shown in Table 4.5. Node a an node b in Figure 4.9 are included in the first loop, while node c and node d are included in the second loop. It means that if we give a small injected current noise with frequency closed to the corresponding noise-sensitive pole to any nodes in the loop, output of all the nodes in the loop will have relatively large overshooting behavior in time domain. This phenomenon

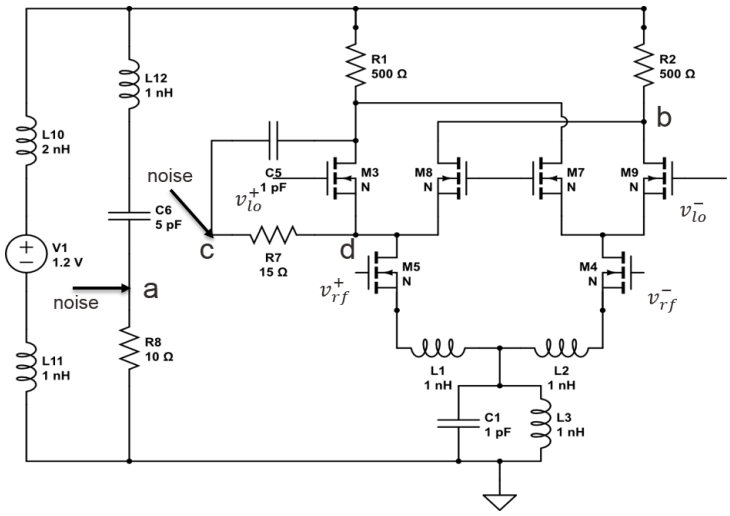

Figure 4.9: A double balanced mixer

can be observed if nodes in the noise-sensitive loop are also physically connected. In Figure 4.9, Small sine wave currents signal with $1mA$ magnitude and frequency closed to two noise-sensitive poles($1.11GHz$ and $1.454GHz$) are given to node a and node c separately. The output wave at node b and node$d$ are chosen to be observed for verification.

For the output voltage of node b, the plot is shown in Figure 4.10. Since node b is included in the noise-sensitive loop with natural frequency $\omega_0 = 1.111GHz$, we can find that the output waveform of b has larger variation when the input is $1.11GHz$(the solid line). When the noise frequency is far from $1.11GHz$(The dash and dot lines), the variation of magnitude is relatively small. For LTPV circuits, the voltage variation may still exist at each node when no noise is injected, the input noise will further increase the variation and introduce new frequency components to the varying state. When the noise frequency is near the natural frequency, this kind of variation of magnitude is more obvious.

In a similar manner, the output of node d has a larger variation of magnitude

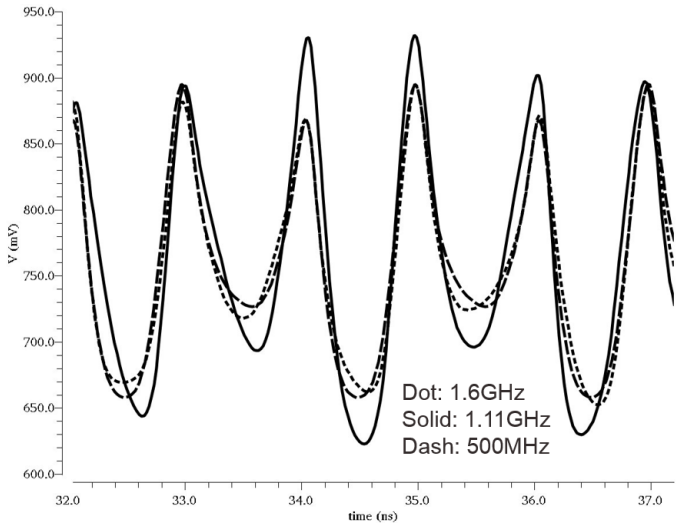

Figure 4.10: Output waveform of node b

with input frequency close to $1.454GHz$(the solid line in Figure 4.11). These two simple examples show that the noise-sensitive loops identified by our algorithm can be observed in time domain in the circuit. It means our algorithm can be used to capture the noise-sensitive behaviors in LTPV circuits efficiently.

4.6 Running time results

A conclusion of the running time of performing our algorithm on three circuit examples is shown in Table 4.7. The environment for running our algorithm is a Linux server with 4GB memory and Intel(R) Core(TM) i5 650 3.2GHZ CPU. We can find that the most time consuming pat is the second phase impedance computation. The most dominant factor to influence the runtime is the size of the system, which is partially determined by the circuit node number. Since usually the noise-sensitive part is small in the circuit, the loop identification phase is not that time-consuming.

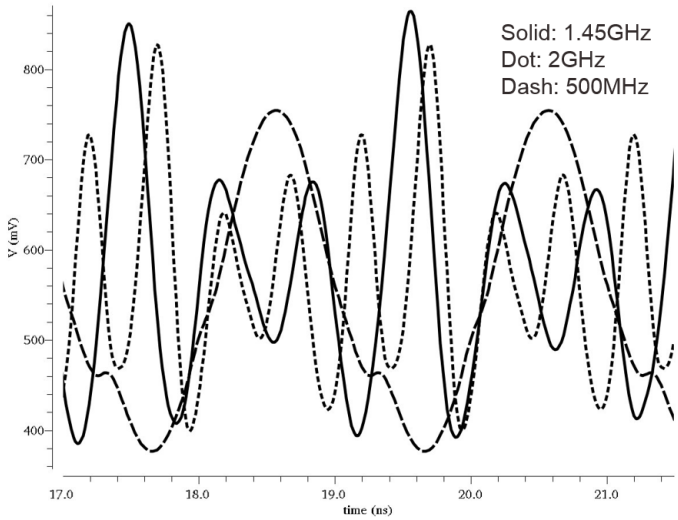

Figure 4.11: Output waveform of node d

Table 4.7: Running for different circuit examples

Application	node num.	sys. size	phase1 (s)	phase2 (s)	phase3 (s)	total (s)
RLC network	9	187	0.292	2.754	0.012	3.058
double-bal. mixer	19	510	7.355	142.333	0.001	149.689
switch-cap. gain stage	13	306	1.265	16.86	0.001	18.126

4.7 Summary

In this section, three circuit examples of our algorithm are discussed in detail. From these cases, we can conclude that our method is really needed for the noise-sensitive behavior detection of LTPV circuits. The noise-sensitive loops identified by our algorithm includes not only loops caused by parasitics, but also loops caused by inherent design problems. Combined with the insights of designers, noise-sensitive behaviors in LTPV circuits can be fixed.

Besides those common parameters, there is another crucial one needs to be discussed. From the previous discussion, we know that there is a threshold value (we

name it r_{dc}) in the noise-sensitive pole computing process. If the DC impedance of a second-order system is less than r_{dc}, we shall ignore this system since its low impedance value is not enough for the peaking behavior to be dominant.

Actually this value is crucial for identifying reasonable noise-sensitive loops in circuits and it may vary from one circuit to another. In LTI case, r_{dc} is always set to be 0.1 according to [4]. However, for the LTPV cases, it needs to be reset for different circuits. r_{dc}'s values for the three circuit examples is shown in Table 4.8.

Table 4.8: r_{dc} for different circuit examples

Application	r_{dc}
RLC network	0.15
double-balanced mixer	2
switch-capacitor gain stage	15

The main accordance for the setting of this value is that we want the algorithm to capture the most important noise-sensitive behaviors. For example, in the switch capacitor gain stage, the second noise-sensitive loop is mainly the negative feedback loop, the node impedances for these nodes are over 200. For some other circuit nodes, they have impedance value about 15 or below, which are much less than 200. Since for the same noise-sensitive pole, the cause of these nodes to have large impedance value are the same. If we can fix the most obvious part, the rest part also get fixed. In this case, we can set $r_{dc} = 15$ to get rid of those less-noise-sensitive nodes and capture the core part of this problem. In the design process, this parameter can be controlled by designers to determine how many nodes in a noise-sensitive loop and how many noise-sensitive loops will be reported. To get a more appropriate method for setting this value, a large number of "real-life" circuits may need to be tested

in the future, which can make our algorithm more robust and suitable for industry applications.

5. CONCLUSION

As can be seen, our new proposed algorithm meets the needs of identifying noise-sensitive behaviors, which can also be called unstable behaviors, in the circuits that can be modeled as LTPV systems. It can detect noise-sensitive loops not only caused by the coupling of parasitics and transistors, but also by inherent design issues, which makes it a helpful tool for designers during the whole circuit design process.

However, to make the algorithm more applicable to realistic products in industry, further works still need to be done. Firstly, a large number of test cases from the real world may need to be run to help tune the parameters and find the short-comes of our algorithm. Additionally, efficient computation technologies, such as model-order reduction, more advanced pole discovery algorithm, need to be researched and applied to our algorithm so that it can deal with very large scale LTPV circuits in real life. What's more, parallel computing is suitable to several phases of our algorithm, such as pole discovery, impedance computing and maximum unstable loop detection algorithm. And it can further speed up the whole noise-sensitive loop identification process.

To sum up, our algorithm for identifying noise-sensitive(unstable) loops in LTPV circuits has been demonstrated to be able to detect efficient unstable information from the LTPV circuit. It has a great potential for the further usage in industry and improving the quality of analog circuit designs.

REFERENCES

[1] Middlebrook, R. D. (1975). Measurement of loop gain in feedback systems. International Journal of Electronics Theoretical and Experimental, 38(4), 485-512.

[2] Dorf, R. C., & Bishop, R. H. (1998). Modern control systems. Addison-Wesley, Menlo Park, CA.

[3] Fang, G. P., Burt, R., & Dong, N. (2010, September). Loop finder analysis for analog circuits. In Custom Integrated Circuits Conference (CICC), 2010 IEEE (pp. 1-4). IEEE.

[4] Mukherjee, P., Fang, G. P., Burt, R., & Li, P. (2012). Efficient identification of unstable loops in large linear analog integrated circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 31(9), 1332-1345.

[5] Li, P., & Pileggi, L. T. (2005). Compact reduced-order modeling of weakly nonlinear analog and RF circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 24(2), 184-203.

[6] Zadeh, L. A. (1950). Frequency analysis of variable networks. Proceedings of the IRE, 38(3), 291-299.

[7] Phillips, J. R. (1998, November). Model reduction of time-varying linear systems using approximate multipoint Krylov-subspace projectors. In Computer-Aided Design, 1998. ICCAD 98. Digest of Technical Papers. 1998 IEEE/ACM International Conference on (pp. 96-102). IEEE.

[8] Roychowdhury, J. (1999). Reduced-order modeling of time-varying systems. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 46(10), 1273-1288.

[9] d'Angelo, H. (1970). Linear time-varying systems: analysis and synthesis. Allyn and Bacon, Boston, MA.

[10] Wereley, N. M., & Hall, S. R. (1990, December). Frequency response of linear time periodic systems. In Decision and Control, 1990., Proceedings of the 29th IEEE Conference on (pp. 3650-3655). IEEE.

[11] Zhou, J., Hagiwara, T., & Araki, M. (2002). Stability analysis of continuous-time periodic systems via the harmonic analysis. IEEE Transactions on Automatic Control, 47(2), 292-298.

[12] Lamour, R., Mrz, R., & Winkler, R. (1998). How floquet theory applies to index 1 differential algebraic equations. Journal of Mathematical Analysis and Applications, 217(2), 372-394.

[13] Liu, H., & Wang, J. (2006, February). A new way to enumerate cycles in graph. In AICT/ICIW (p. 57). IEEE.