AUTOMATIC TUNING OF
CONTINUOUS-TIME FILTERS

A Dissertation

by

TANER SUMESAGLAM

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

DOCTOR OF PHILOSOPHY

August 2004

Major Subject: Electrical Engineering

AUTOMATIC TUNING OF
CONTINUOUS-TIME FILTERS

A Dissertation

by

TANER SUMESAGLAM

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Approved as to style and content by:

Aydin I. Karsilayan (Chair of Committee)	
Edgar Sanchez-Sinencio (Member)	Costas N. Georghiades (Member)
Ibrahim Karaman (Member)	Chanan Singh (Head of Department)

August 2004

Major Subject: Electrical Engineering

ABSTRACT

Automatic Tuning of

Continuous-Time Filters. (August 2004)

Taner Sumesaglam, B.S., Bilkent University

Chair of Advisory Committee: Dr. Aydin I. Karsilayan

Integrated high-Q continuous-time filters require adaptive tuning circuits that will correct the filter parameters such as center frequency and quality factor (Q). Three different automatic tuning techniques are introduced. In all of the proposed methods, frequency and quality factor tuning loops are controlled digitally, providing stable tuning by activating only one loop at a given time. In addition, a direct relationship between passband gain and quality factor is not required, so the techniques can be applied to active LC filters as well as G_m-C filters.

The digital-tuning method based on phase comparison was verified with 1% tuning accuracy at 5.5 MHz for Q of 20. It uses phase information for both Q and center-frequency tuning. The filter output phase is tuned to the known references, which are generated by a frequency synthesizer. The core tuning circuit consists of D flip-flops (DFF) and simple logic gates. DFFs are utilized to perform binary phase comparisons. The second method, high-order digital tuning based on phase comparison, is an extension of the previous technique to high-order analog filters without depending on the master-slave approach. Direct tuning of the overall filter response is achieved without separating individual biquad sections, eliminating switches and their parasitics. The tuning system was verified with a prototype 6^{th} order bandpass filter at 19 MHz with 0.6 MHz bandwidth, which was fabricated in a conventional 0.5 μm CMOS technology. Analysis of different practical limitations

is also provided. Finally, the digital-tuning method based on magnitude comparison is proposed for second-order filters for higher frequency operations. It incorporates a frequency synthesizer to generate reference signals, an envelope detector and a switched comparator to compare output magnitudes at three reference frequencies. The theoretical analysis of the technique and the simulation results are provided.

To my parents Nimet and Kadir

ACKNOWLEDGMENTS

I would like to thank many people who have supported me during this work. First and foremost, I would like to thank and express my gratitude to my supervisor Dr. Aydin I. Karsilayan for his guidance and assistance in many respects of my Ph.D study. He has always been with me and has answered every single question that I might have had.

I also thank my committee members, Dr. Edgar Sanchez-Sinencio, Dr. Costas N. Georghiades, and Dr. Ibrahim Karaman, for their guidance. Dr. Sanchez-Sinencio deserves special acknowledgment. He always encouraged me with his guidance and enthusiasm.

I also would like to thank the Analog & Mixed Signal Center (AMSC) group for providing a good environment for research and academic study. In particular, special thanks go to Dr. Silva-Martinez, Dr. Ugur Cilingiroglu, and my colleagues Alberto Valdes-Garcia, Ari Yakov Valero Lopez, Timothy Wayne Fischer, Fikret Dulger, and Hengsheng Liu, for our valuable discussions and their help.

Finally, I would like to thank my family for their support and patience.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Motivation

Filters are essential components of many electrical systems. An electrical filter is a two-port device, which modifies magnitude and phase of the input signal in order to produce the desired signal at the output. A signal can be considered as a composition of signals at various frequencies. Pass frequency band (passband) and stop frequency band (stopband) are defined over the frequency of interest. From input to output, an ideal filter will transmit the signals in the passband, while it will attenuate or reject the signals in the stopband. Filters can be classified as lowpass, bandpass, highpass, and bandstop depending on the allocation of pass and stop frequency bands [1]. High-selectivity (or high-Q) bandpass filters will be the focus of this dissertation.

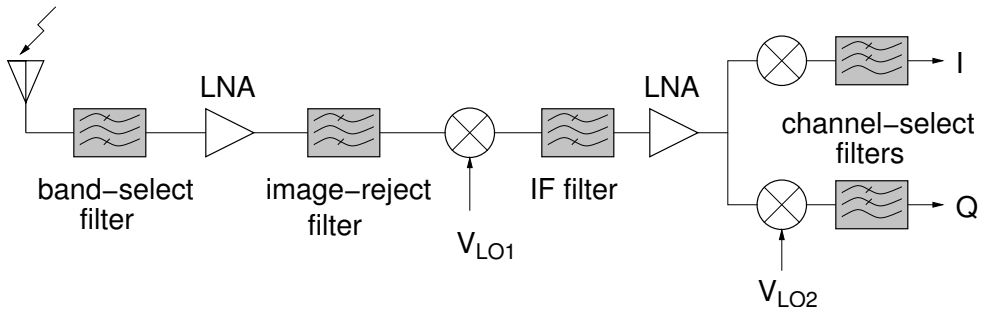

Fig. 1. RF receiver architecture.

In today's state-of-the-art designs, high-performance filters are required for sev-

This dissertation follows the style of *IEEE Journal of Solid-State Circuits*.

eral applications such as equalization filtering in high-speed links, radio frequency (RF), intermediate frequency (IF) filtering in RF receivers, anti-aliasing filtering in A/D (analog-to-digital) converters [2]. A typical RF receiver is shown in Fig. 1. After the antenna, RF band-select filter removes noise and interferers from the incoming signal. Image-reject filter is used before the mixer to remove the undesired signals at the image frequency. IF filter, on the other hand, removes the harmonics after the mixing operation. Finally, channel-select filter passes the desired channel while attenuating the other channels.

Many high-performance filters are required to operate at high frequencies, which can be from megahertz to gigahertz depending on the specific application. Another requirement is to have high dynamic range (DR), which can be defined as the maximum signal level that can be applied to a system divided by the minimum detectable signal level by the same system in the presence of noise. Growing wireless communication market is forcing the systems to be portable and cheaper. This can be accomplished by reducing the physical area and power consumption. High-frequency high-DR integrated filter designs become even more challenging under these constraints.

There are many different types of filters used depending on the frequency of interest and available technology. Figure 2 shows the choice of filter type as a function of the operating frequency range [1]. Based on implementation, filters can be broadly categorized into two as on-chip and off-chip, as shown in Fig. 3. Off-chip discrete filters such as SAW (Surface Acoustic Wave) and ceramic filters are widely used in commercial products due to their large dynamic ranges and reliable frequency characteristics. However, off-chip filters are expensive, bulky, and occupy large area. Off-chip components usually increase the chip pin count of the system. They also require matching circuits and buffers with challenging design specifications such as

capability of driving 50 Ω, which increases the power consumption.

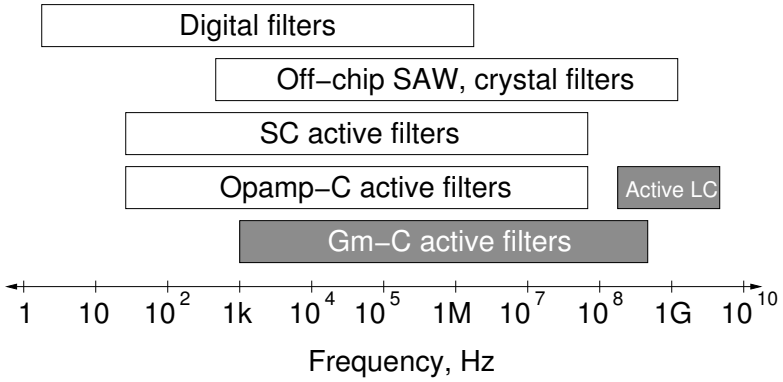

Fig. 2. Filter-type choice vs frequency of operation.

On-chip filters can be divided into three types depending on the input-output signal nature as digital, sampled (or discrete-time), and analog (or continuous-time). Digital filters have accurate responses and can be easily integrated. One of the main drawbacks of the digital filters is that they require analog-to-digital converters to be able to process the continuous-time input signal. This constraint limits their frequency of operation to a few megahertz. Another main problem is power consumption. Power, area, and cost constraints prohibit their usage in high-frequency applications [3]. Switched-capacitor (SC) filters, which process signals at discrete times, provide accurate filter responses. They are worse off in terms of linearity and noise with respect to discrete passive filters. Finite gain-bandwidth product, charge injection, and clock feed-through problems limit their performances at high frequencies [4]. Other types of filters using opamp such as active RC, G_m-C -opamp, and MOSFET-C have limited frequency responses as well due to finite gain-bandwith product. G_m-C filters can operate at relatively higher frequencies for a given power

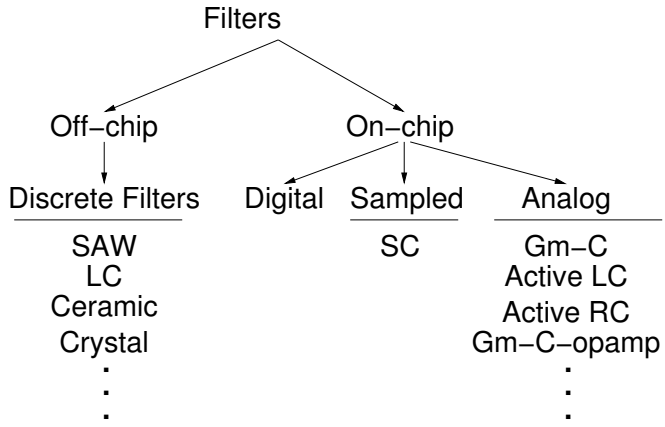

Fig. 3. Practical filter types.

consumption and area. With the advent of high-Q inductors in CMOS (Complementary Metal Oxide Semiconductor) technology, active-LC filters have become another alternative for high-frequency applications [5]. They are superior in terms of dynamic range compared to G_m-C filters. In summary, from different integrated filter options, G_m-C and active LC filters seem to be possible alternatives for high performance filters operating at high frequencies.

Main disadvantage of on-chip active filters is that they require automatic tuning circuits to correct the deviation of filter parameters [1]. Filter parameters such as center frequency and quality factor are typically a function of resistance R, capacitance C, inductance L, or transconductance G_m values. In conventional CMOS technology, on chip R, C, L, and G_m values may have $20-30\%$ variation due to process tolerances. Temperature changes and aging are also important factors causing those parameters to drift from the desired values.

A second-order bandpass filter can be characterized by its center frequency (f_0), where the peak gain occurs, and its 3-dB bandwidth. Essentially, it passes the

signals within 3-dB bandwidth and the filter gain drops tremendously as the input frequency deviates from f_0 as shown in Fig. 4. Quality factor (Q) is a measure of selectivity of the filter, which can be defined as the ratio of f_0 to bandwidth. The center-frequency accuracy required for a bandpass filter depends on Q. To illustrate the tuning accuracy required for high-Q filters, let us examine a filter with a Q of 100 shown in Fig. 4. Solid curve shows the desired magnitude response while the dashed one is with 1% frequency tuning error. Obviously, the desired frequency band is totally out of filter's passband, which leads to great loss in the input signal power.

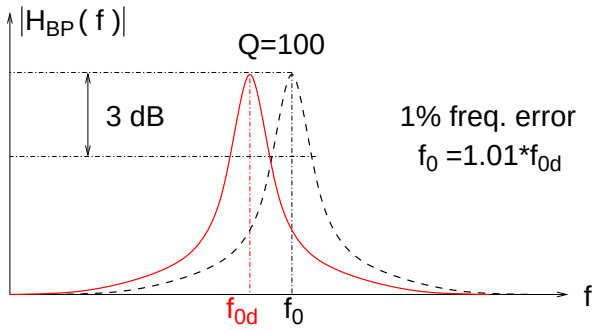

Fig. 4. Illustration of 1% tuning error on a high-Q biquadratic bandpass filter.

Automatic tuning circuits can also be used for applications employing programmable filters. For instance, third-generation wireless systems require receivers to support different wireless communication standards. Therefore, channel-select filters should exhibit programmable bandwidth to accommodate different standards [6].

Integration of continuous-time filters can be achieved with a reliable automatic tuning system that can operate at high frequencies. Most Q-tuning schemes, which have been reported so far, impose equality of passband gain and quality factor to filter

design. This constraint itself suffers from parasitics. Also Q-tuning circuits assumes that the center frequency is tuned correctly by another tuning circuit; however, having two active control loop on a high-Q filter may cause stability problems. In this dissertation, robust, digital, accurate and reliable automatic tuning techniques are proposed that overcome the problems with the existing tuning techniques.

B. Organization

The dissertation is organized as seven chapters including the introduction. In Chapter II, high frequency filters, namely, G_m-C and active LC filters are discussed briefly. In Chapter III, some of the reported tuning methods are revisited and their problems are addressed. In the following Chapters IV, V, and VI, the proposed tuning techniques Digital Tuning Method based on Phase Comparison (DTPC), High-order Digital Tuning Method based on Phase Comparison (HDTPC), and Digital Tuning Method based on Magnitude Comparison (DTMC) are introduced, respectively. DTPC and HDTPC techniques use the phase information of the filter output signal to tune the filter, on the other hand, DTMC uses the magnitude information. HDTPC is intended for direct tuning of high-order filters without separating individual biquad sections. Finally, Chapter VII concludes the dissertation with suggestions for further research work.

CHAPTER II

HIGH-FREQUENCY CONTINUOUS-TIME FILTERS

A. Introduction to Continuous-Time Bandpass Filters

A bandpass filter passes the components in input signal frequency spectrum within the passband to output, while it attenuates other frequency components. A second-order bandpass filter can be implemented with a two-integrator loop shown in Fig. 5. One of the integrators is lossy with a negative feedback factor of K_q [7]. There are two feedback loops in this configuration L_1 and L_2. The one including both integrators (L_1) determines the natural frequency or the center frequency of the system. The feedback loop including one integrator (L_2) determines the selectivity or Q factor of the filter. The transfer function from the input V_{in} to the bandpass output V_{bp} can be written as

$$H(s) = \frac{V_{bp}}{V_{in}} = \frac{K_g\, s}{s^2 + sK_q + K_w} \tag{2.1}$$

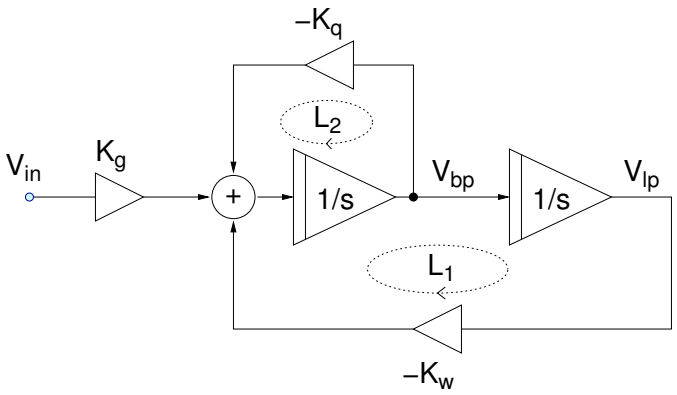

Fig. 5. Two-integrator loop.

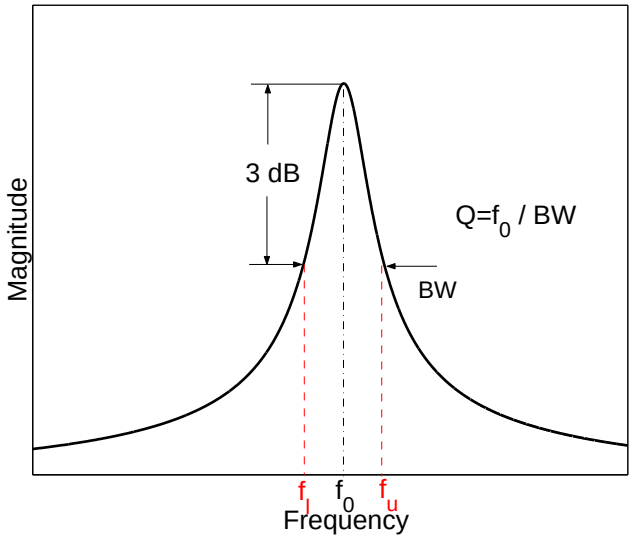

Fig. 6. Magnitude response of a typical second-order bandpass filter.

The magnitude response is shown in Fig. 6. The center frequency, which is the frequency where the peak gain occurs, is given as

$$\omega_0 = 2\pi f_0 = \sqrt{K_w} \tag{2.2}$$

3-dB frequencies, f_l and f_u, which is shown in Fig. 6, are defined as the frequencies where the peak gain drops by 3 dB. The 3-dB bandwidth (BW) is the difference between 3-dB frequencies. For the second-order case, it can be found as

$$BW = f_u - f_l = K_q \tag{2.3}$$

Quality factor is defined as the ratio of the center frequency to the bandwith.

$$Q = \frac{\sqrt{K_w}}{K_q} \tag{2.4}$$

The passband gain or peak gain of the filter can be found as

$$G = |H(j\omega)|\Big|_{\omega=\omega_0} \tag{2.5}$$

$$= \frac{K_g}{K_q} \tag{2.6}$$

Mainly, two parameters are required to represent an ideal second-order or bi-quadratic filter; center frequency (f_0) and quality factor (Q). Passband gain (G) is not a critical parameter in the sense that it does not directly change the frequency selectivity of filter.

Note that in general, there is no direct relation between G and Q unless imposed in the implementation. As it will be addressed in Chapter III, some tuning techniques assume $G = Q$ in order to tune Q.

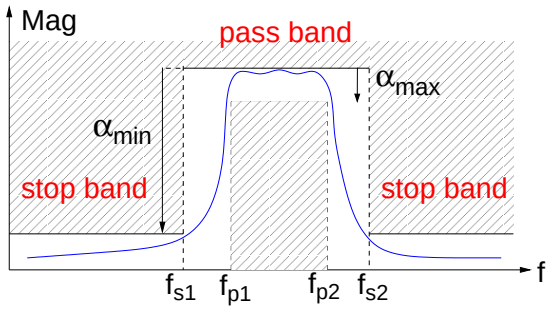

Fig. 7. High-order bandpass filter magnitude response.

In a second-order filter case, more attenuation at stopband cannot be achieved for a given bandwidth and center frequency. In other words, when Q is increased for more attenuation, the bandwidth will also get smaller. A high-order filter, on the other hand, can achieve more attenuation in stopband for a fixed bandwidth. A bandpass filter can be specified as shown in Fig. 7. Attenuation in the passband and

stopbands have to satisfy

$$\begin{aligned} \alpha_{max} &\geq \alpha(f) \qquad \text{for } f_{p1} \leq f \leq f_{p2} \\ \alpha_{min} &\leq \alpha(f) \qquad \text{for } f \leq f_{s1}, f \geq f_{s2} \end{aligned} \tag{2.7}$$

where $\alpha(f)$ is defined as the attenuation with respect to maximum gain (in the passband). It is usually expressed in dB unit. f_{p1} and f_{p2} are passband frequencies while f_{s1} and f_{s2} are stopband frequencies. α_{max} and α_{min} are maximum allowed attenuation in the passband and the minimum required attenuation in the stopbands, respectively. A high-order bandpass filter can be designed by using different magnitude approximation techniques such as Butterworth, Chebyshev, inverse Chebyshev, and elliptical, based on the allocation of poles and zeros [1]. There are many software tools available to synthesize filter with different approximation methods.

B. G_m-C Filters

A typical G_m-C biquadratic filter implementation is shown in Fig. 8. It has two outputs as lowpass (V_{LP}) and bandpass (V_{BP}) available. Ideally, G_m is voltage controlled current source.

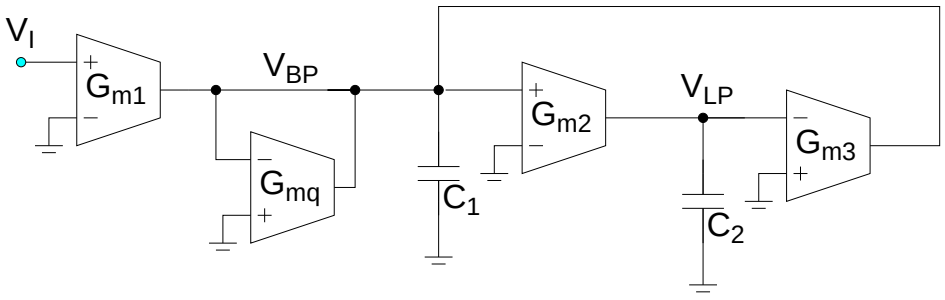

Fig. 8. A typical G_m-C implementation of a biquadratic filter.

The transfer functions from input to bandpass and lowpass outputs can be written as,

$$\frac{V_{BP}}{V_I}(s) = \frac{s\frac{G_{m1}}{C_2}}{s^2 + s\frac{G_{mq}}{C_2} + \frac{G_{m2}G_{m3}}{C_1C_2}} \tag{2.8}$$

$$\frac{V_{LP}}{V_I}(s) = \frac{\frac{G_{m1}G_{m2}}{C_1C_2}}{s^2 + s\frac{G_{mq}}{C_2} + \frac{G_{m2}G_{m3}}{C_1C_2}} \tag{2.9}$$

Assuming all the transconductor values and capacitor values are the same ($G_m = G_{m1} = G_{m2} = G_{m3}$ and $C = C_1 = C_2$), the center frequency, the quality factor, and the passband gain can be written as

$$\omega_0 = \frac{G_m}{C} \tag{2.10}$$

$$Q = \frac{G_m}{G_{mq}} \tag{2.11}$$

$$G = \frac{G_m}{G_{mq}} \tag{2.12}$$

Usually Q is not directly measurable quantity. With this implementation, Q is set equal to a measurable quantity, which is the passband gain. However, in the analysis parasitic capacitances and losses associated with V_{LP} and V_{BP} nodes are ignored. The circuit shown in Fig. 9 is more reliable model. G_{o1} and G_{o2} represent resistive losses associated with bandpass on lowpass outputs, respectively. C_{p1} and C_{p2} are parasitic capacitors. Including the parasitics, the bandpass transfer function becomes

$$H(s) = \frac{\frac{G_{m1}}{\hat{C}_2}\left(s + \frac{G_{o2}}{\hat{C}_1}\right)}{s^2 + s\left(\frac{G_{o2}}{\hat{C}_1} + \frac{G_{o1}}{\hat{C}_2} + \frac{G_{mq}}{\hat{C}_2}\right) + \left(\frac{G_{o2}(G_{o1} + G_{mq})}{\hat{C}_1\hat{C}_2} + \frac{G_{m2}G_{m3}}{\hat{C}_1\hat{C}_2}\right)} \tag{2.13}$$

where $\hat{C}_1$ and $\hat{C}_2$ defined as

$$\hat{C}_1 = C_1 + C_{p1} \tag{2.14}$$

$$\hat{C}_2 = C_2 + C_{p2} \tag{2.15}$$

Approximate Q and G can be written as

$$Q \approx \frac{\sqrt{\hat{C}_1\hat{C}_2[G_{m2}G_{m3} + G_{o2}(G_{o1} + G_{mq})]}}{G_{o2}\hat{C}_2 + (G_{o1} + G_{mq})\hat{C}_1} \tag{2.16}$$

$$G \approx \frac{G_{m1}}{G_{o2}\dfrac{\hat{C}_2}{\hat{C}_1} + G_{o1} + G_{mq}} \left(1 + \frac{G_{o2}\hat{C}_2}{\hat{C}_1\left(G_{o1} + G_{mq} + \dfrac{G_{m2}G_{m3}}{G_{o2}}\right)}\right)^{1/2} \tag{2.17}$$

Apparently, certain assumptions have to be made to claim $Q = G$. First of all, the parasitic conductances G_{o1} and G_{o2} should be very small compared to G_{mq}. Also, matching between transconductors and capacitors is required so that $G_{m1} = G_{m2} = G_{m3}$ and $\hat{C}_1 = \hat{C}_2$ are satisfied. Due to unknown parasitics, these conditions are difficult to achieve especially at high frequencies. Parasitic values (C_{p1}, C_{p2}, G_{o1}, and G_{o2}) become more dominant and comparable to actual values (C_1, C_2, G_{mi}) at high frequencies.

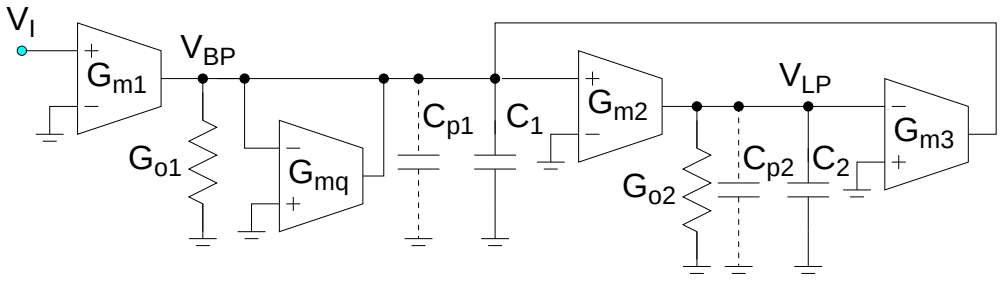

Fig. 9. G_m-C type biquadratic filter with parasitics included.

C. Active-LC Filters

In current CMOS technology, on-chip inductors are realized with relatively low Q values ($\sim$ 5). High-Q can be achieved by cancelling the losses with a negative resistor. This type of filters are also referred as Q-enhanced LC filters [8], [9]. They provide better DR and consume much less power compared to G_m-C filters at high frequencies [10].

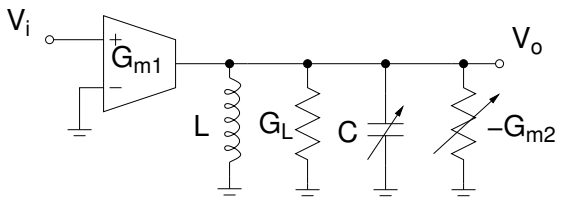

Fig. 10. Q-enhancement LC filters.

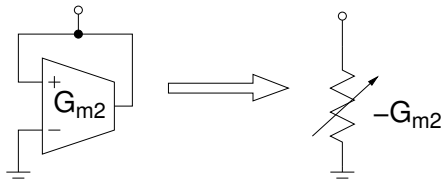

Fig. 11. Negative resistor.

Figure 10 depicts the conceptual circuit diagram of active-LC filters. The conductance G_L shown in Fig. 10 represents the losses of inductor L. Negative resistor is achieved by employing positive feedback with G_{m2} as shown in Fig. 11. In actual implementation, transconductors G_{m1} and G_{m2} are simply realized by a couple of

transistors. The transfer function of the representative active-LC filter can be found as,

$$\frac{V_o}{V_i}(s) = \frac{s\frac{G_{m1}}{C}}{s^2 + s\frac{G_L - G_{m2}}{C} + \frac{1}{LC}} \tag{2.18}$$

Filter parameters can be found as

$$\omega_0 = \frac{1}{\sqrt{LC}} \tag{2.19}$$

$$Q = \frac{G_{m1}}{G_L - G_{m2}} \tag{2.20}$$

$$G = \frac{1}{G_L - G_{m2}}\sqrt{\frac{C}{L}} \tag{2.21}$$

Tunability of the center frequency can be achieved with a varactor while Q can be tuned by changing G_{m1}. Note that Q and G are arbitrarily set by process-dependent circuit parameters. Note that it is more difficult to match different type of circuit elements than the same type, which was the case for G_m-C filters.

CHAPTER III

EXISTING TUNING METHODS

Integrated analog filters require continuous automatic tuning circuits to correct deviation of the filter parameters due to aging, temperature and process variations. Section A mentions more general tuning approaches such as direct, indirect, and switching filters. Then, the following Sections B, C, and D summarize some of the center frequency, quality factor, and high-order filter tuning techniques which have been already reported in the literature. It is difficult to make fair comparison of tuning techniques; nevertheless, each technique is presented by its weakness and superiority in terms of accuracy and practicality at high frequencies.

A. Filtering and Tuning

Filters with practical tuning systems cannot process the actual input and tuning reference signals simultaneously. There are different approaches to resolve the problem such as master/slave (or indirect) tuning, direct tuning and switching filters which will be presented shortly.

1. Master/Slave (or Indirect) Tuning Approach

The conceptual master/slave tuning architecture is shown in Fig. 12 [11],[12]. The master filter stays connected to the tuning circuit and is not used for signal processing. The tuning circuit applies a reference signal to the master filter, processes the output signal coming from the filter, then adjusts the voltage control signals accordingly. The same voltage control signals are applied to both master and slave filters. Slave filter is available for signal processing all the time. Essentially, this technique indirectly tunes the slave filter. The master can be a VCO (voltage con-

trolled oscillator), which is made out of the same type of the slave filter with positive feedback, or a replica of the slave filter (see Section B). This approach wastes area with the additional filter. More importantly, this approach extensively depends on the matching of master and slave filters. The accuracy of the tuning is limited by the matching of the two filters. With today's CMOS technology, the component matching (capacitor-capacitor or resistor-resistor) is within 0.1%. In component matching, different layout techniques such as inter-digitizing and common centroid can be used for better matching; however, some of the techniques are difficult to apply to match the whole circuit. Evidently, matching of the whole filters will have worse figures. Also note that both master and slave will have different loadings and and different signal paths. Although master filter can be tuned correctly, due different loadings, matching of master and slave and consequently tuning accuracy will degrade.

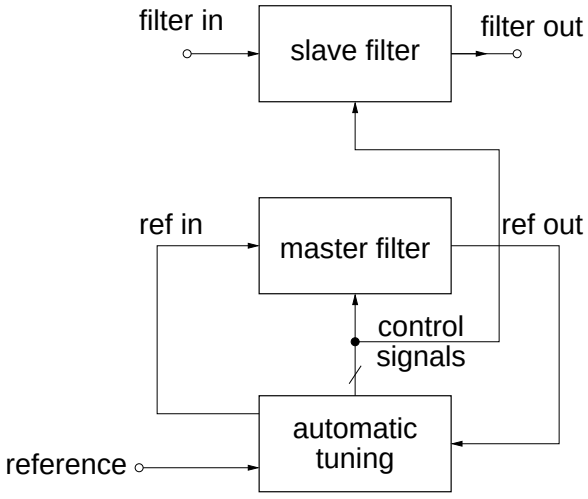

Fig. 12. Master-slave tuning approach.

2. Direct-Tuning Approach

Direct-tuning approach is depicted in Fig. 13. The filter is tuned only when the tuning is enabled, i.e., the filter is connected to the tuning circuit when the control enable signal EN is high. When the tuning is disabled, the voltage control signals are kept at the same level as tuned. This approach is more appropriate for time division multiplexed systems such as TDMA (Time Division Multiple Access). The filter is tuned in idle times [13].

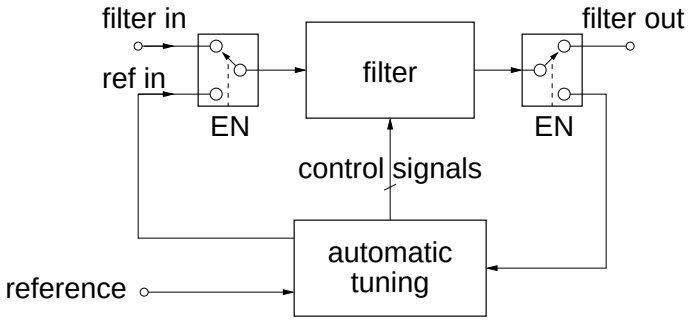

Fig. 13. Direct-tuning approach.

3. Switching-Filters Tuning Approach

Direct tuning may not be applicable for applications that require continuous filtering such as CDMA (Code Division Multiple Access) systems [14]. As shown in Fig. 14, this methods employs two filters. While *filter 1* is tuned, *filter 2* processes incoming signals. Then, *filter 2* is tuned while *filter 1* is used. When EN signal is high, *filter 2* is connected to the tuning circuit. During this time, the voltage controls of *filter 1* are kept the same. When the EN signal is low, *filter 1* is tuned and the voltage controls of *filter 2* are kept the same. This approach eliminates the matching of

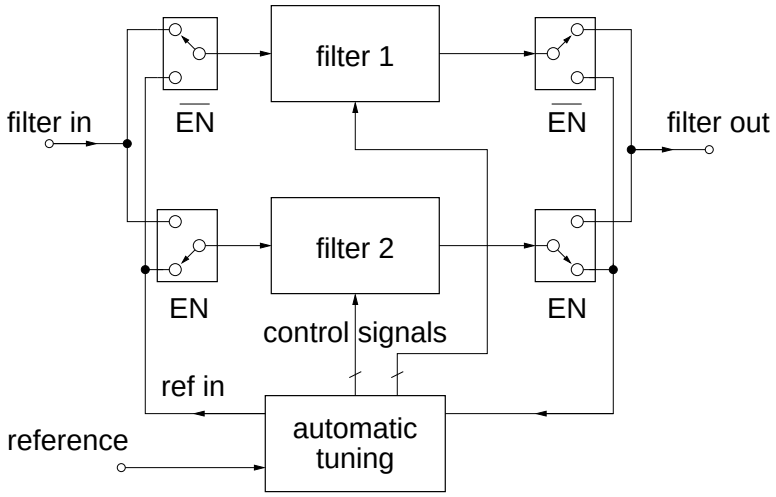

Fig. 14. Switching-filters tuning approach.

filters requirement, which is the case in master and slave approach.

B. Frequency-Tuning Techniques

1. Tuning of Integrator Time Constant with Magnitude-Locked Loop

Tuning of integrator time constant with Magnitude-Locked Loop (MLL) architecture is given in Fig. 15 [15], [16]. The loop forces the gain of G_m-C integrator to be unity. When the gain of the integrator is unity, the following holds

$$\left|\frac{G_m}{s\,C}\right|_{s=j\omega_R} = 1 \tag{3.1}$$

$$\frac{G_m}{C} = \omega_R \tag{3.2}$$

The value of G_m/C, which is process dependent, is forced to be equal to the reference signal $\omega_R = 2\pi f_R$.

The loop tunes the master integrator and applies the same control voltages

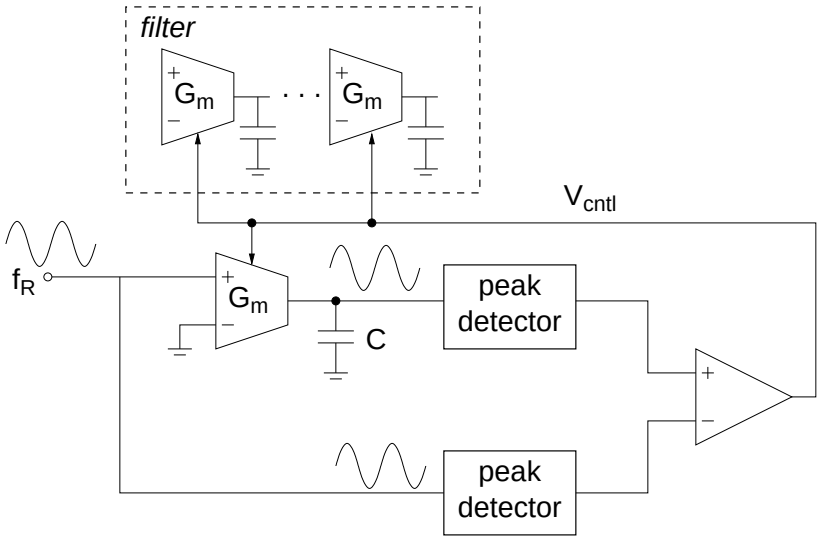

Fig. 15. Tuning of integrator time constant with magnitude-locked loop.

to other integrator blocks in the filter. This technique has limited accuracy for several reasons. First, it depends on the matching of master and slave G_m-C cells. Moreover, peak detectors have to match. Furthermore, DC offset of the amplifier will directly degrade the accuracy. Note also that (3.1) ignores parasitics due to finite output impedance of G_m block and any possible internal node in the transconductor causing parasitic poles and zeros. Finally, this technique requires a perfect sinusoidal reference which is hard to achieve. Any harmonics in the reference signal will be accumulated by the integrator and result in inaccuracies.

2. Tuning Method Based on Phase-Locked Loop Using Voltage-Controlled Filter

This technique is based on the fact that the ideal biquadratic filter will have zero phase shift from input to bandpass output [17]-[18]. The system architecture is shown in Fig. 16. The phase difference between the reference signal and the filter output is detected and lowpass-filtered to generate the control voltage for the center frequency.

When the center frequency of the filter is tuned to f_R, the phase difference will be zero. The accuracy of the technique is limited by the offsets of the phase detector and lowpass filter (LPF).

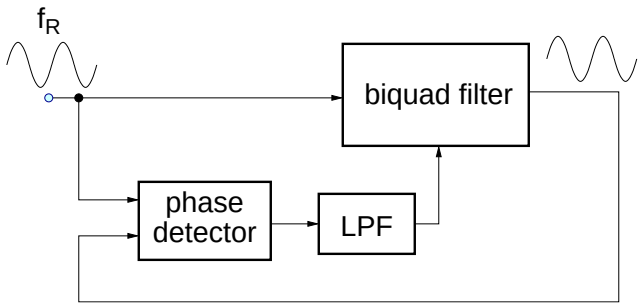

Fig. 16. Tuning method based on phase-locked loop using voltage-controlled filter.

3. Tuning Method Based on Phase-Locked Loop Using Voltage-Controlled Oscillator

The tuning method is illustrated in Fig. 17. At steady state, the master Voltage-Controlled Oscillator (VCO) oscillates at the reference frequency, which is the desired center frequency f_R. Phase-Locked Loop (PLL) locks to the reference frequency f_R and some constant phase instead of 0° phase, which is the case in the previous technique. This, in turn, eliminates the absolute phase accuracy required so the phase detector design is more relaxed. Master VCO has to be the same type of slave filter. A biquadratic filter can be formed by inverting and non-inverting integrators while one can obtain an oscillator with two non-inverting integrators. This technique has been applied for G_m-C filters in [19]-[22] and MOSFET-C filters in [12].

The technique relies on the matching of the filter and the master VCO. Oscillation amplitude has to be limited to ensure the linear operation. Note that the filter

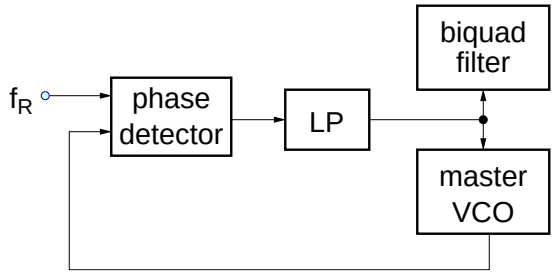

Fig. 17. Phase-locked loop for VCO tuning method.

is used in its linear region, naturally for better matching, VCO has to be in linear region as well. To achieve better matching, the filter and VCO should be physically as close as possible; however, the performance of the filter is degraded in terms of noise due to the feed-through from VCO. The method involves a trade-off between matching and isolating the filter from the noise of VCO.

4. Tuning Method Based on Charge-Comparison

The block diagram of this technique is shown in Fig. 18 [23]-[26]. ϕ_1 and ϕ_2 are two non-overlapping clocks that determine the charge transfer phases of the system. In phase one (when ϕ_1 is high), the capacitor C is charged up to I_R/G_m. In phase two (when ϕ_2 is high), however, the charge is transfered to the holding capacitor C_H. At steady state, the averages of the voltages V_1 and V_2 are constant. Therefore, the transferred charge will be balanced by the extracting current source NI_R,

$$\frac{I_R}{G_m} C = NI_R T \tag{3.3}$$

where T is period of the clock, i.e., total duration of phase one and two. Then (3.3) can be rewritten as

$$\frac{G_m}{C} = \frac{f_{clk}}{N} \tag{3.4}$$

Note that the value of G_m/C is fixed by the precise clock signal frequency (f_{clk}) and the current mirror ratio (N). One advantage of this technique is that the reference signal is at a frequency of NG_m/C, which eliminates the clock interference in the band of interest. The control voltage is copied to G_m blocks in the filter. This scheme suffers from inaccuracies caused by mismatch between the master G_m-C block and the slaves due to different loadings and different parasitics. The offset of the OTA (G_m) circuit increases the tuning error. Moreover, finite gain of OTA and opamp produces tuning errors almost equal to the reciprocal of the individual gains.

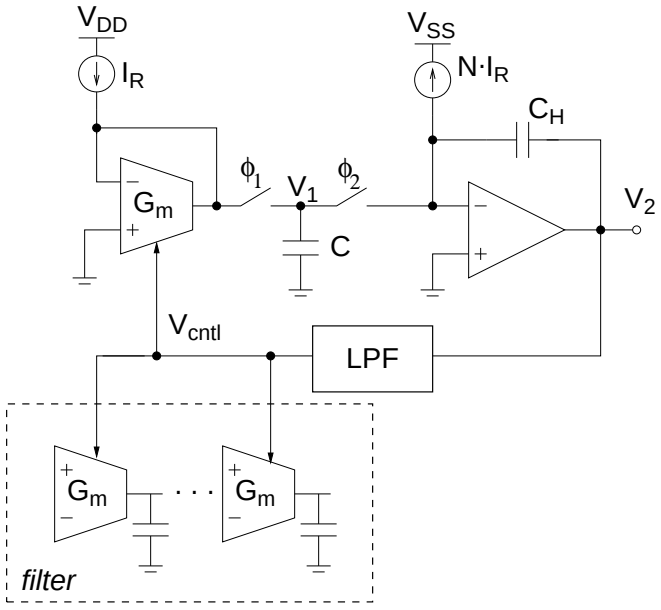

Fig. 18. Charge-comparison based tuning.

A similar charge-comparison technique for the tuning of MOSFET-C filters is reported in [27]. This technique suffers from timing inaccuracies of the tuning clocks in addition to the previously mentioned problems. Both techniques are limited to low-frequency operations.

5. Maximum-Gain Search Tuning

This tuning scheme relies on the fact that the peak gain of a second-order filter occurs at the center frequency [28]. A reference signal is applied to the filter at the desired center frequency (f_R). The control voltage of the master filter (V_{fm}) is swept within a reasonable range covering process variation. The magnitude of the filter output is detected by an envelope detector as shown in Fig. 19. The value of the frequency control voltage yielding the maximum gain (V_{fs}) is sampled and applied to the slave filter. This scheme is verified at 120 MHz with less than 0.3% tuning error [28]. One drawback is that continuous sweeping the frequency control voltage makes Q-tuning difficult.

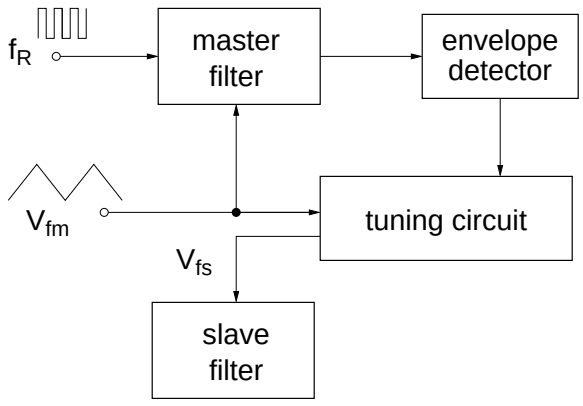

Fig. 19. Maximum-gain search tuning.

C. Quality-Factor Tuning Techniques

1. Envelope Matching Tuning

The envelope matching tuning method exploits the similarity of step responses of a first-order and a second-order filters [25]. The step-response envelope of a unity-gain first-order filter with a 3-dB frequency of f_{3dB} is equivalent to a second-order filter with a bandwidth of $2f_{3dB}$. The block diagram of the scheme is shown in Fig. 20. The reference signal V_R is a pulse train operating at lower frequency with a period of T. Envelope detectors and a difference integrator are used to extract the envelope information and to generate the Q-control voltage (V_{cntl}). Desired Q value is set by the ratio of OTAs, which are used in the implementation of the filters. One advantage of the technique is that the reference signal is at a lower frequency. Thus, the interference with the main filter is minimized. Also, it does not assume passband gain and quality factor equality. However, the accuracy is limited due to several reasons. First of all, the matching of OTAs are limited to 1-3%. Second, the offset of the OTA degrades the performance. Another design issue is good matching requirement for the envelope detectors. The work in [29] reports a theoretical error of 8% for an OTA with offset of 10 mV and T-BW product of 8.

2. Magnitude-Locked Loop for Q Tuning

This technique uses the peak gain (or passband gain) of the filter to tune Q [11],[28]. It assumes that Q of the filter always tracks the peak gain of the filter (G). The tuning system is depicted in Fig. 21. A sinusoidal reference signal is applied to the filter at the desired center frequency. Assuming the center frequency is tuned correctly, the output signal magnitude will be the same as the reference signal with employing a divide-by-Q circuit. Using peak detectors and difference integrator, the

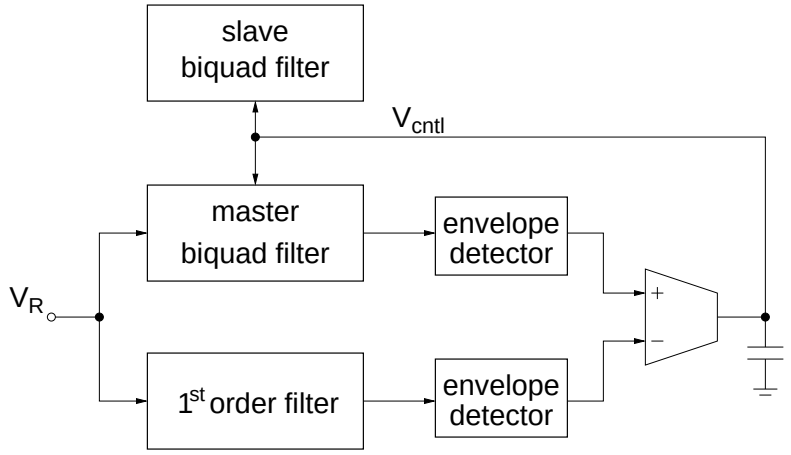

Fig. 20. Envelope matching tuning.

control voltage is generated. At steady state the following holds

$$\frac{Q}{Q_d} = 1 \tag{3.5}$$

where Q_d is the desired Q value set by the divider ratio.

One of the main problems of this approach is that the reference signal is assumed to be sinusoidal. Thus, any harmonics in the reference signal will introduce error. In case a square wave is used as the reference for instance, the division ratio has to be modified to $\pi Q_d/4$. Another problem is the error in the assumption of $G = Q$. It ignores parasitics and requires perfect matching for transconductors and capacitors (discussed in Chapter II). Implementing an accurate divide-by-Q circuit is also challenging issue at high frequencies. Note that any inaccuracy in this block directly degrades the tuning accuracy. Observe also that the matching is required for peak detectors. In addition, this technique assumes that the center frequency is already tuned correctly by another tuning circuit. Therefore, any error in the center frequency will degrade the performance of MLL for Q-tuning technique. Furthermore,

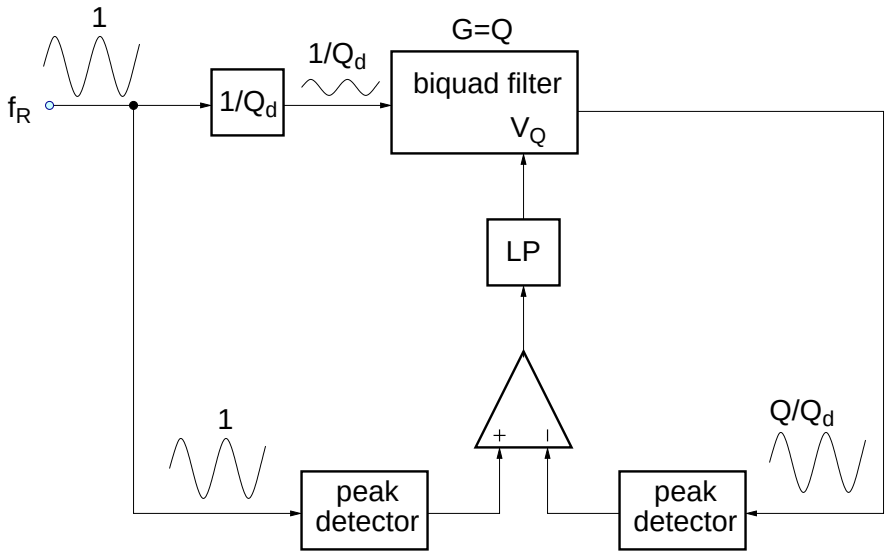

Fig. 21. MLL for Q tuning.

requirement of another tuning circuit for frequency tuning degrades the reliability of this technique because having two tuning loops operational at the same time may cause the filter to oscillate. Note that the filter becomes sensitive to control voltages at high-Q values.

There are other Q-tuning techniques based on MLL with different implementations. Charge pumps and comparators are used instead of peak detectors in [18]. The work in [30], however, proposes a digital implementation based on a latched comparator and counters. This technique suffers from the undesired phase shifts in the latched comparator in addition to common drawbacks of MLL technique.

3. Adaptive Gain Control Based on LMS Tuning

In this tuning technique, filter output signal is matched to desired output by minimizing the mean-square error between them [31]. A general block diagram is given

in Fig. 22. The LMS algorithm essentially implements the following equation

$$\frac{V_{wi}}{dt} = \mu(V_d - V_o)V_{gi}, \qquad i = 1, 2, ..., n \tag{3.6}$$

where V_o and V_d are filter output and the desired output signals, while V_{wi} and V_{gi} are the tuning and the gradient signals, respectively. The gradient signals, which are the derivative of V_o with respect to V_{wi}, determine the direction of the tuning. At steady state, time derivative of the tuning signals becomes zero. As a result, the filter yields the desired output signal implying that the filter is tuned.

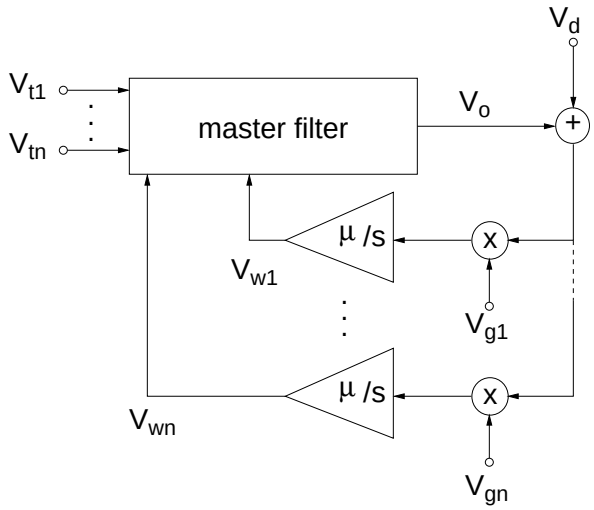

Fig. 22. LMS tuning.

A similar LMS approach proposed in [32] is implemented by using delta-sigma oscillators [33]. The tuning input signals are generated by delta-sigma oscillators within the passband of the filter with certain amplitudes and frequencies. These values are stored in digital registers. Main problem of [32] and [33] is the system complexity and the requirement of DACs operating at a higher frequency greater

than the passband of the filter itself.

More practical and simple methods based on combination of LMS and MLL tuning techniques are reported in [34] and [21]. The the number of reference signals is reduced to one. They are similar to the MLL Q tuning in that Q is assumed to have the same value as the filter passband gain; however, the use of peak detectors are eliminated. Furthermore, they do not require perfect sinusoidal reference signals. The architecture is shown in Fig. 23. The modified LMS equation is given as

$$\frac{dV_Q}{dt} = \mu(V_R - V_o)V_o \tag{3.7}$$

These techniques also suffer from the same problems as MLL technique such as implementing high frequency divide-by-Q circuit and inaccuracies due to $Q = G$ assumption. Note also that accurate multiplier and summer circuits at high frequencies can be challenging and power consuming.

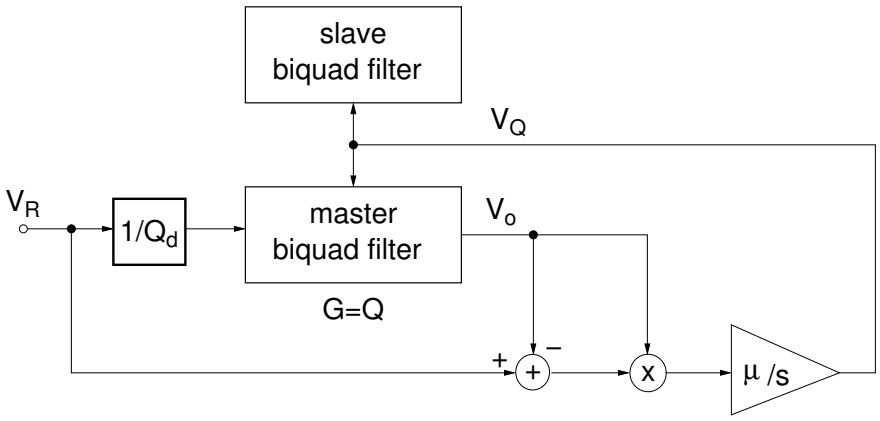

Fig. 23. LMS-MLL tuning.

4. Magnitude Comparison Tuning

Magnitude comparison tuning technique applies reference signals at three frequencies $f_R - \Delta$, f_R, and $f_R + \Delta$ and forces the gains at the edge reference frequencies ($f_R \pm \Delta$) to be half of the gain at the middle reference frequency (f_R). At steady state the following equation holds,

$$|H(j2\pi f_R)| = \frac{1}{2}|H(j2\pi(f_R \pm \Delta))| \tag{3.8}$$

From (3.8), the tuned Q and f_0 values can be found as

$$f_{0t} \approx f_R \tag{3.9}$$

$$Q_t \approx \frac{\sqrt{3} f_R}{2\Delta} \tag{3.10}$$

The architecture is shown in Fig. 24. It uses a peak detector and switched-capacitor integrators to compare the magnitudes at different frequencies. A selective attenuator is utilized in order to eliminate the non-linearity of the peak detector. This technique is verified with 3% accuracy by a second-order bandpass filter having center frequency of 200 MHz and Q of 29 [35].

It provides an accurate tuning method by eliminating $Q = G$; however, simultaneous operation of frequency and Q tuning loops still exists. Also, the switched-capacitor integrators used occupy large area.

D. High-Order Filter Tuning

Necessity of tuning circuits for second-order filters is a well-known problem and many different frequency and Q tuning techniques are reported as mentioned in previous sections. On the other hand, there is almost no reported general tuning method for high-order filters. One convenient way of implementing high-order filters is to

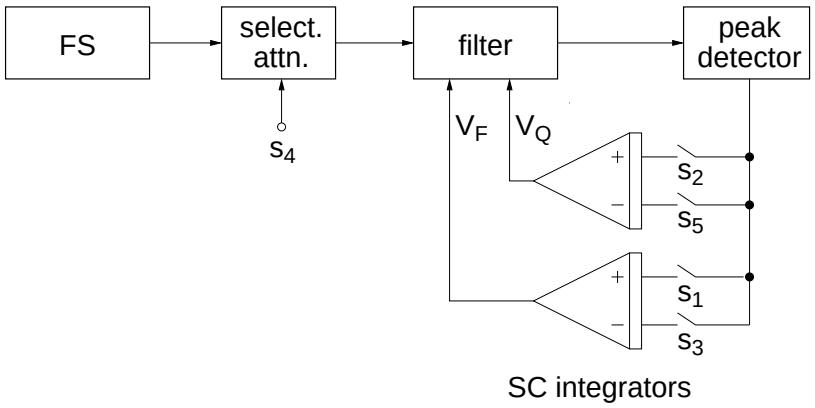

Fig. 24. Magnitude comparison tuning.

cascade biquadratic sections. The tuning techniques mentioned previously can be applied to cascade of biquads with master-slave approach [11],[22]. It can also be applied to ladder type G_m-C filters, where each G_m-C integrator is indirectly tuned by copying the tuned control voltage of the master. However, the tuning accuracy is limited by the matching of components (Section A-1).

A tuning technique for magnetically-coupled Q-enhanced LC type 4^{th}-order bandpass filters is reported in [36]. It employs a VCO of the same type as the filter in the frequency synthesizer (PLL VCO technique); thereby the frequency tuning accuracy is limited by the matching of master and slave filters. A general block diagram of the tuning system is shown in Fig. 25. It indirectly tunes the filter response by making Q of the LC resonator infinite. The bandwidth of the filter is determined by mutual magnetic coupling of the inductors, which is set by the physical layout of the inductors. Although the shunt losses are cancelled out by the positive feedback, the series losses cause ripples in the passband. This technique does not actually tune Q, i.e, it does not provide the tuning of individual Q values to certain target Q(s).

The tuning method in [37] is proposed for leap-frog type filters. LC resonators

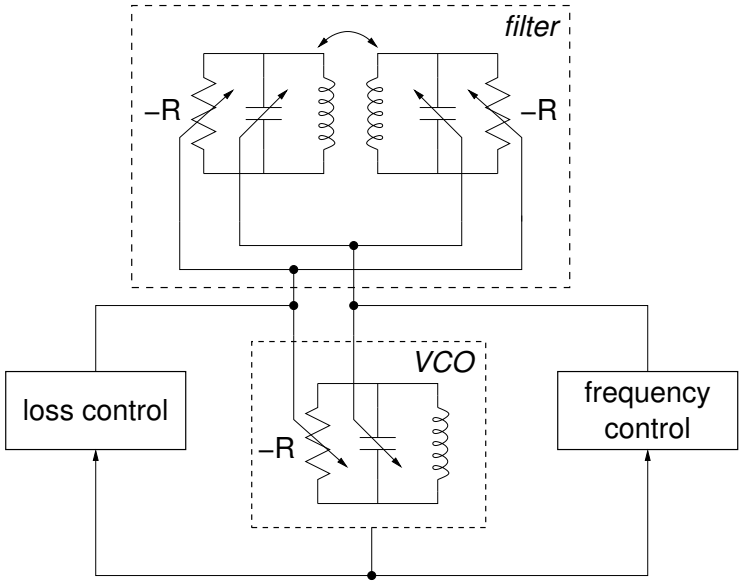

Fig. 25. Loss-control tuning.

are realized by G_m-C biquads. The switches are employed in order to isolate each shunt and series biquad sections. The parasitics associated with the switches will limit the use of this technique to low frequencies and will degrade the performance of the filter. More importantly, the technique does not completely define the transfer function of the filter. Also, it relies on the matching of transconductors forming the biquad, which is known to be poorer than simple component matching. Furthermore, it assumes that G_m ratios do not change with the change of the control voltages.

CHAPTER IV

DIGITAL-TUNING METHOD BASED ON PHASE COMPARISON

An automatic tuning method for second-order continuous-time filters is proposed in this chapter. The digital-tuning method based on phase comparison (DTPC) uses phase information of the filter output signal to tune center frequency and quality factor [38],[39]. Basic principle, circuit implementation, simulation results, prototype design, and experimental results are presented, as well as limitations on the tuning accuracy.

A. Principle

In general, a second-order bandpass filter has lowpass output available. Transfer function for a typical second-order lowpass and bandpass outputs can be written as

$$H_{LP}(s) = \frac{K_1}{s^2 + s\omega_0/Q + \omega_0^2} \tag{4.1}$$

$$H_{BP}(s) = \frac{K_2 s}{s^2 + s\omega_0/Q + \omega_0^2} \tag{4.2}$$

where the lowpass phase response can be calculated from

$$\angle H_{LP}(\omega) = \phi(\omega) = -\tan^{-1}\left(\frac{\omega_0\,\omega/Q}{\omega_0^2 - \omega^2}\right) \tag{4.3}$$

Magnitude response of the bandpass and phase response of the lowpass outputs are shown in Fig. 26; where f_a and f_b are 3-dB frequencies. The gain is maximized at the center frequency, $f_0 = \omega_0/2\pi$. The phases corresponding to 3-dB frequencies at the lowpass output are given as

$$\phi(f_a) = -45^\circ \tag{4.4}$$

$$\phi(f_b) = -135^\circ \tag{4.5}$$

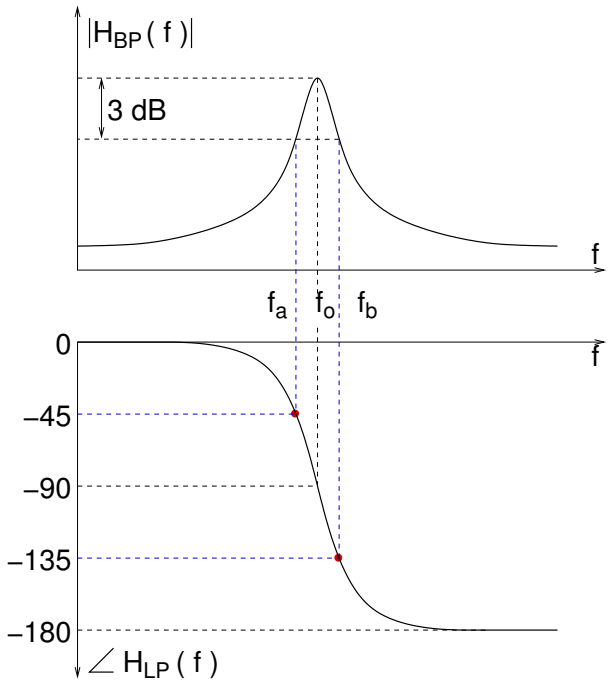

Fig. 26. Bandpass magnitude and lowpass phase responses of a typical biquadratic filter.

After fabrication, the filter will have arbitrary f_0 and Q values, and consequently f_a and f_b are unknown. Assume that the tuning circuit generates reference signals at f_1 and f_2. The tuning circuit changes the filter parameters in such a way that the following conditions are satisfied,

$$\phi(f_1) = -45^\circ \tag{4.6}$$

$$\phi(f_2) = -135^\circ \tag{4.7}$$

Once (4.6) and (4.7) are achieved, the tuned f_0 and Q values can be found as

$$f_{0t} = \sqrt{f_1 f_2} \tag{4.8}$$

$$Q_t = \frac{\sqrt{f_1 f_2}}{f_2 - f_1} \tag{4.9}$$

Note that the lowpass output is chosen for convenience since it has relatively larger gain at low frequency. The tuning can also be achieved by using bandpass output. If this is the case, the reference phases are needed to be modified to $\pm 45°$.

An algorithm has to be implemented to achieve the tuning conditions given in (4.6) and (4.7). Define two binary numbers A and B to indicate whether the phases at reference signal is larger or smaller than the reference phases

$$A = \begin{cases} 0 \quad , \quad \phi(f_1) > -45° \\ 1 \quad , \quad \phi(f_1) < -45° \end{cases} \tag{4.10}$$

$$B = \begin{cases} 0 \quad , \quad \phi(f_2) > -135° \\ 1 \quad , \quad \phi(f_2) < -135° \end{cases} \tag{4.11}$$

Based on the values of A and B, filter's lowpass output phase can be one of the four different cases as shown in Fig 27. Dashed lines show the reference window; reference frequencies and phases. When the filter is tuned, the phase curve will pass through the cross-section points. Phase transition from $-45°$ to $-135°$ occurs within the bandwidth of the filter. At the center frequency of the filter, the corresponding phase is $90°$. Fig. 27(a) corresponds to AB =11, the current center frequency of the filter needs to be increased. In Fig. 27(b), the center frequency needs to decreased (AB =00). In (c), the current Q is larger than the desired Q. Note that larger the Q, the sharper the phase transition from $-45°$ to $-135°$. In a way, the slope of the phase curve at the center frequency will give an idea about how large the Q is. In Fig. 27(d), the Q has to be increased. The tuning actions for each case are summarized in Table I.

This tuning strategy assigns only one tuning action for each case. The digital

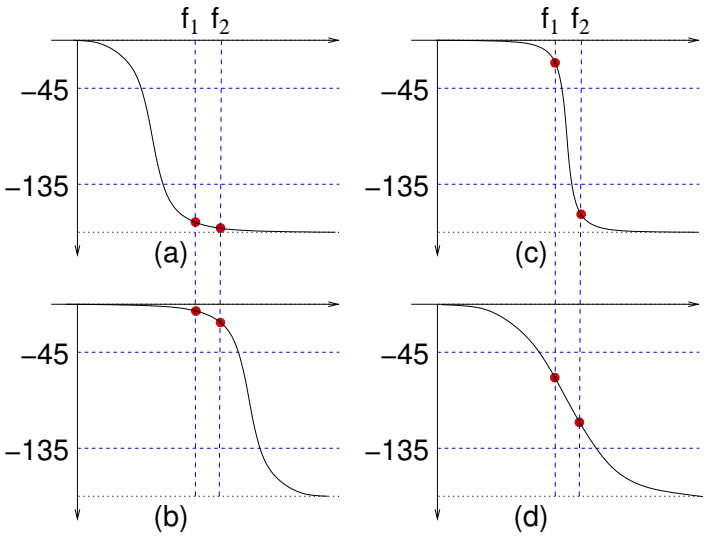

Fig. 27. Four different cases of filter based on A and B.

TABLE I

TUNING PROCESS OF DTPC

Case	**A**	**B**	**Action**
(a)	1	1	$f_0 \Uparrow$
(b)	0	0	$f_0 \Downarrow$
(c)	0	1	$Q \Downarrow$
(d)	1	0	$Q \Uparrow$

approach allows using digital logic circuits and binary phase comparators. Note that in general digital circuits are more reliable and robust. Another important advantage is that Q tuning is enabled only when the center frequency of the filter is within the reference window, namely $f_1 < f_0 < f_2$. Therefore, Q-tuning loop will not cause the filter oscillate before the center frequency is tuned close to the desired value. Other tuning techniques, where Q and frequency tuning circuits simultaneously enabled, do not have good control on this issue (discussed in Chapter III).

B. Circuit Implementation

The overall tuning system can be implemented as shown in Fig. 28. The biquad filter has two control voltages V_F and V_Q to adjust f_0 and Q. In most practical cases, f_0 and Q cannot be changed independently, herein, independent tunability is not assumed. The two reference frequencies can be generated using a programmable synthesizer. When CLK is low, the frequency of the reference signal is $f_1 = (N-1)/Mf_R$, otherwise it is $f_2 = (N+1)/Mf_R$. V_0, V_{-45}, and V_{-135} are pulse signals with 50% duty cycle which are available from frequency synthesizer (FS). V_0 is applied to the filter input, $-45°$ and $-135°$ delayed signals and the output of the filter are processed by the tuning circuit. Then, the control voltages are generated and applied to the filter. Note that this tuning technique does not require perfect sinusoidal reference signals. On the other hand, the reference signals should have fundamental frequency at the reference frequency and 50% duty cycle. Therefrom, distorted signals other than pulse and sinusoidal are also acceptable.

The block, FS, shown in Fig. 28 is a standard frequency synthesizer with programmable dividers (see Fig. 29). A fixed input signal at a frequency of f_R coming from an accurate generator (such as a crystal oscillator) is applied to the input. The

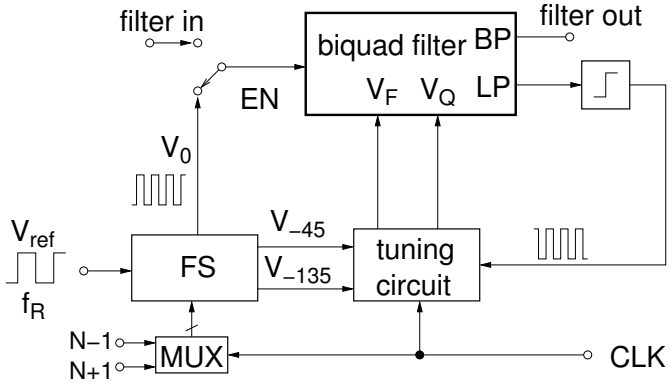

Fig. 28. Overall tuning system.

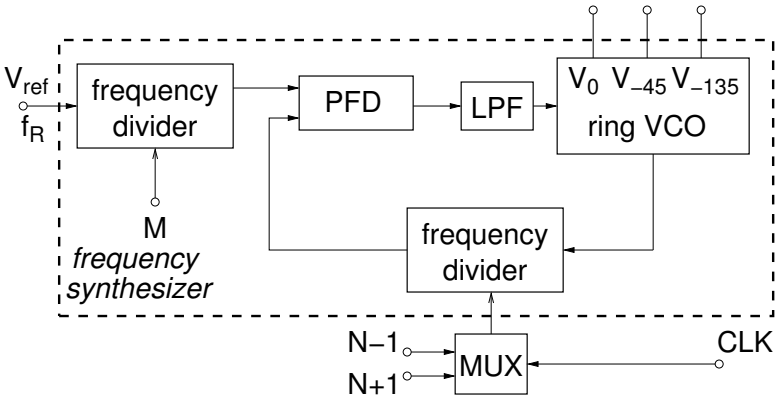

Fig. 29. Frequency synthesizer.

division ratio of frequency divider in the loop can be changed by CLK signal. Either $(N-1)$ or $(N+1)$ is multiplexed depending on the value of CLK. A ring voltage-controlled oscillator (VCO) provides the delayed clock signals (see Fig. 30). It is also possible to generate 45° delayed reference clock signals by using a ring counter circuit in case a ring VCO is not available in FS.

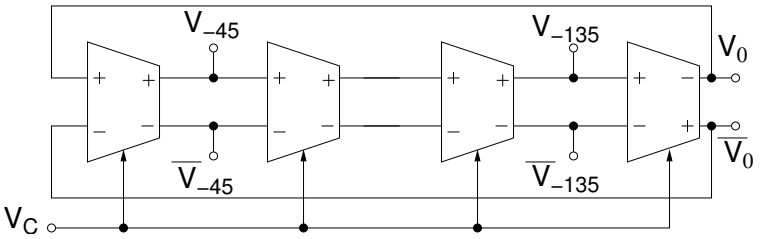

Fig. 30. Ring oscillator.

Binary phase comparison of reference signals (V_{-45} and V_{-135}) and the lowpass output signal V_{LP} can be achieved by using a single D flip-flop (DFF) as shown in Fig. 31. Depending on whether S_2 signal is delayed or not with respect to S_1, DFF yields output digital 1 or 0. One problem in using DFF is that the output signal of the filter is expected to be sinusoidal due to high-Q. Therefore, a comparator has to

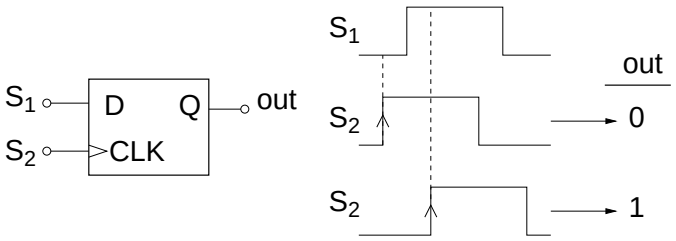

Fig. 31. Binary phase comparison by using a DFF.

be used to convert sinusoidal filter output to rail-to-rail clock/pulse signal so that DFF can be used for phase comparison.

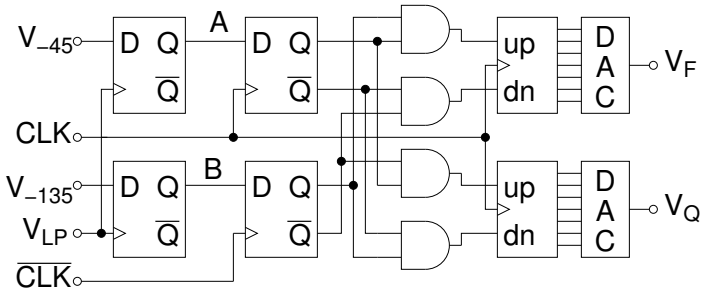

Fig. 32. Core tuning circuit.

The core tuning circuit is shown in Fig. 32. The first set of DFFs accomplish the binary phase comparison. The second set is used to store A and B since both reference frequencies cannot be applied at the same time. When the CLK is low, f_1 is applied and A is generated. When it is high, f_2 is applied and B is generated. Both A and B are available right before CLK signal makes low to high transition. Then, frequency and Q counters are updated. Digital output of counters are converted to analog by using a simple digital-to-analog converter (DAC). From Table I, the up-down control signals can be extracted as

$$f_u = AB \tag{4.12}$$

$$f_d = \bar{A}\bar{B} \tag{4.13}$$

$$q_u = A\bar{B} \tag{4.14}$$

$$q_d = \bar{A}B \tag{4.15}$$

With this implementation of DTPC, f_1 and f_2 are given as

$$f_1 = \frac{(N-1)}{M} f_R \tag{4.16}$$

$$f_2 = \frac{(N+1)}{M} f_R \tag{4.17}$$

By substituting (4.16) and (4.17) into (4.8) and (4.9), the tuned values can be found as

$$f_0 = \sqrt{N^2-1}\frac{f_R}{M} \tag{4.18}$$

$$Q = \frac{\sqrt{N^2-1}}{2} \tag{4.19}$$

As it can be seen, the tuned values are digitally set by the choice of f_R, N and M. Note that since N is an integer number, the tuned Q value will can have certain discrete values.

The tuning is enabled as long as EN signal is high. When the tuning has converged, the tuning circuit can be disabled while holding the tuned control voltages. Convergence speed of the tuning circuit depends on the settling time of the FS and the resolution of the DAC-counter pair. The tuning will converge roughly in $2^{L_c} \cdot t_s$, where L_c and t_s are the resolution of the DAC-counter pair and the settling time of FS, respectively.

C. Limitations

In this section, the limitations of the tuning system will be investigated. The effect of the parasitics will be analyzed. Pre-distortion technique will be proposed for very high-frequency applications where the parasitics are more dominant. In addition, the effect of the resolution of the DAC-counter pair on the tuning accuracy will be analyzed.

1. Parasitic-Pole Effect

In the analysis so far, the filter has been assumed to have only two poles. In reality, there might be parasitic poles and zeros located at higher frequencies. For high-frequency filters, parasitics become more effective on the filter response. In the presence of parasitics, the reference phases at reference frequencies shift slightly. Consequently, some error is involved in the tuning scheme. The effect of parasitic pole on a G_m-C bandpass filter is analyzed in this section.

The transfer function of a typical G_m-C filter as seen at bandpass and lowpass outputs are given as

$$H_{BP}(s) = \frac{sKG_m/C}{s^2 + s\,G_{mq}/C + (G_m/C)^2} \tag{4.20}$$

$$H_{LP}(s) = \frac{K(G_m/C)^2}{s^2 + s\,G_{mq}/C + (G_m/C)^2} \tag{4.21}$$

A transconductor with a parasitic pole at frequency ω_p can be modeled by [40], [41],

$$G_m = \frac{G_{m0}}{1 + \frac{s}{\omega_p}} \tag{4.22}$$

Substituting (4.22) into (4.21) results in

$$H'_{LP}(s) = \frac{K\left(\frac{G_{m0}/C}{1+\frac{s}{\omega_p}}\right)^2}{s^2 + s\frac{G_{mq0}/C}{1+\frac{s}{\omega_p}} + \left(\frac{G_{m0}/C}{1+\frac{s}{\omega_p}}\right)^2} \tag{4.23}$$

Defining $\omega_{oo} = G_{m0}/C$ and $Q_o = G_{m0}/G_{mq0}$ yields

$$H'_{LP}(s) = \frac{K\omega_{oo}^2}{s^2\left(1+\frac{s}{\omega_p}\right)^2 + s\frac{\omega_{oo}}{Q_o}\left(1+\frac{s}{\omega_p}\right) + \omega_{oo}^2} \tag{4.24}$$

Rewriting the H'_{LP} by substituting $s = j\omega$ yields

$$H'_{LP}(j\omega) = \frac{K\omega_{oo}^2}{j\left(\omega\frac{\omega_{oo}}{Q_o} - \omega^3\frac{2}{\omega_p}\right) + \left(\omega^4\frac{1}{\omega_p^2} - \omega^2(1 + \frac{\omega_{oo}}{Q_o\,\omega_p}) + \omega_{oo}^2\right)} \tag{4.25}$$

Defining

$$R(\omega) = \omega^4\frac{1}{\omega_p^2} - \omega^2(1 + \frac{\omega_{oo}}{Q_o\,\omega_p}) + \omega_{oo}^2 \tag{4.26}$$

$$I(\omega) = \omega^3\frac{2}{\omega_p} - \omega\frac{\omega_{oo}}{Q_o} \tag{4.27}$$

(4.25) simplifies to

$$H'_{LP} = \frac{K\omega_{oo}^2}{\sqrt{R^2 + I^2}}(R + jI) \tag{4.28}$$

From (4.28), the lowpass phase can be written as,

$$\phi'(\omega) = \tan^{-1}\left(\frac{I(\omega)}{R(\omega)}\right) \tag{4.29}$$

After the tuning has converged, assuming $M = 1$ for simplicity, the circuit parameters ω_{oo} and Q_o will take their values to satisfy the following tuning conditions

$$\phi'[(N-1)\omega_R] = -45^\circ \tag{4.30}$$

$$\phi'[(N+1)\omega_R] = -135^\circ \tag{4.31}$$

From (4.29)-(4.31),

$$I[(N-1)\omega_R] = R[(N-1)\omega_R] \tag{4.32}$$

$$I[(N+1)\omega_R] = -R[(N+1)\omega_R] \tag{4.33}$$

Define X as the relative location of the pole as $X = \omega_p/\omega_{oo}$. For a fixed target center frequency (keeping $N\omega_R$ constant), equations (4.32) and (4.33) are numerically solved by using Maple (see Appendix A). Calculated ω_{oo} and Q_o values are substituted into

(4.28) in order to find the tuned effective center frequency and quality factor values. Note that the filter is not second-order anymore with the additional parasitic poles. Frequency and Q-tuning errors generated for different target Qs and Xs are shown in Figs. 33 and 34 (see Appendix B for Matlab code). For larger values of X, the tuning error gets smaller as expected. The Q-tuning error is less than 1 % for $X > 30$.

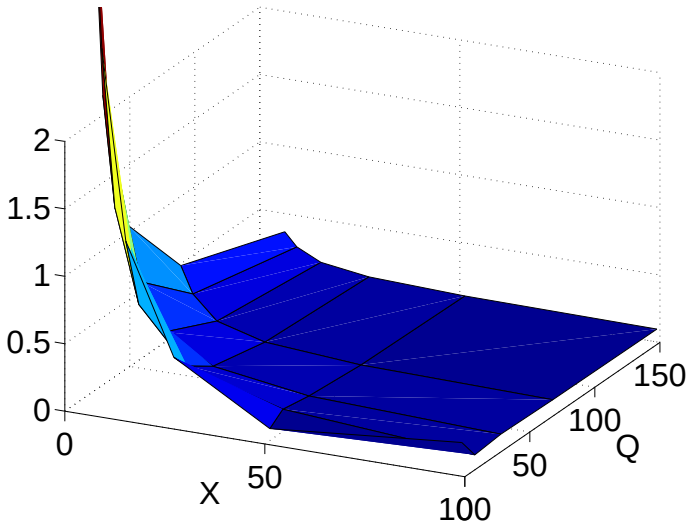

Fig. 33. Frequency-tuning error with a parasitic pole at $X\, f_0$.

2. Predistortion of the Parasitic-Pole Effect

When the frequency of operation is very high for a given technology, having parasitic poles close to ω_0 is unavoidable. In such cases, the tuning error can be decreased by modifying the reference frequencies according to the estimated value of the parasitic pole location. Assume that the filter is designed but the parasitic pole could not be pushed high enough to minimize the error. Using simulation results, the reference frequencies that give $-45°$ and $-135°$ phases from the lowpass phase response can be

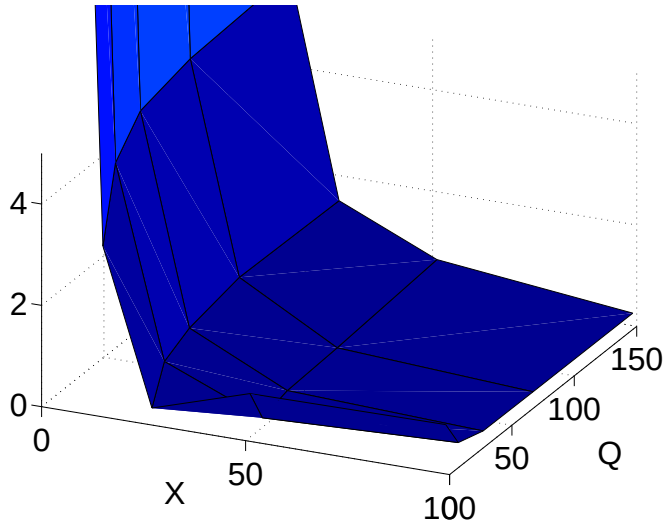

Fig. 34. Q-tuning error with a parasitic pole at X f_0.

found. Note that they will be slightly different from the 3-dB frequencies. The tuning circuit is programmed to generate the frequencies $N_1\omega_R$ and $N_2\omega_R$ (with $M = 1$) as the two references. When there is no parasitic pole, $N_1\omega_R$ and $N_2\omega_R$ are symmetrical (at high-Q) around ω_0 as $(N-1)\omega_R$ and $(N+1)\omega_R$.

A bandpass filter with $Q = 40$ and $f_0 = 40$ MHz having a parasitic pole at 480 MHz ($X = 12$) is investigated as an example. The Q-tuning error without compensation is 5%. Theoretically, compensation decreases the error to zero if the pole is located exactly at the predicted value. However, due to unpredictable behavior of the parasitic pole, the actual error may be different. Nevertheless, smaller errors can be achieved using a good layout extraction tool in the estimation of X. Figures 35 and 36 show the tuning errors with and without compensation for a variation of X from -50% to 100%. This method essentially shifts the error curve in such a way that the error around the estimated value of X is minimized. The predistortion approach can be employed for any G_m modeled with arbitrary poles and zeros.

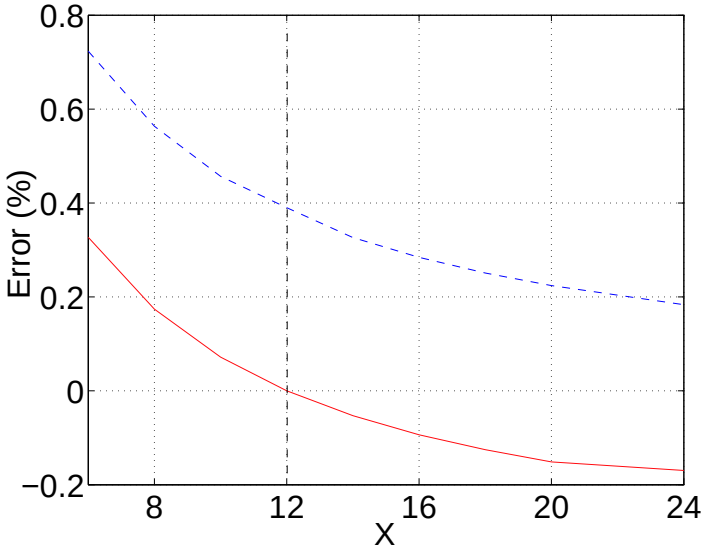

Fig. 35. Parasitic predistortion effect on the frequency-tuning error for a filter with $Q = 40$ and $f_0 = 40$ MHz, solid line is the compensated case.

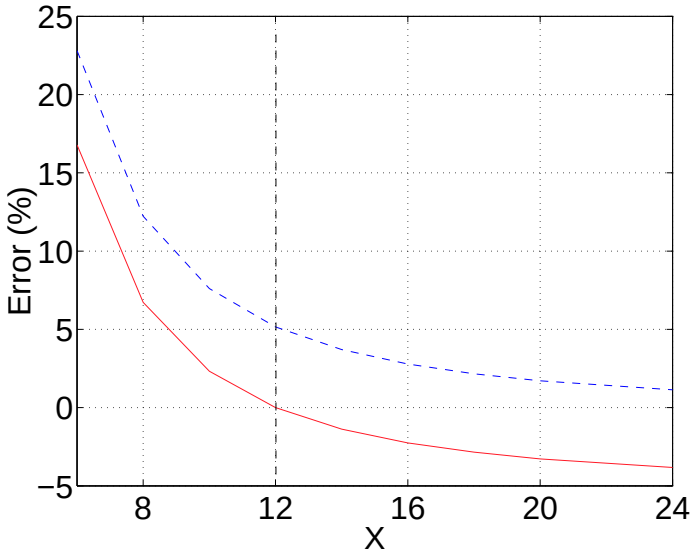

Fig. 36. Parasitic predistortion effect on the Q-tuning error for a filter with $Q = 40$ and $f_0 = 40$ MHz, solid line is the compensated case.

3. Parasitic-Zero Effect

The effect of a parasitic zero on the tuning performance will be examined in this section. As in Section 1, the transconductor element is assumed to have a parasitic zero located at $X \cdot \omega_{oo}$,

$$G_m = G_{m0}\left(1 + \frac{s}{X\omega_{oo}}\right) \tag{4.34}$$

Substituting (4.34) into (4.20), the modified transfer function is obtained as following,

$$H'(s) = \frac{\dfrac{s^2}{X} + s\,\omega_{oo}}{\left(1 + \dfrac{1}{X^2}\right)s^2 + s\left(\dfrac{\omega_{oo}}{Q_o} + \dfrac{2\omega_{oo}}{X}\right) + \omega_{oo}^2} \tag{4.35}$$

where $\omega_{oo} = G_{m0}/C$ and $Q_o = G_{mq}/G_{m0}$ as defined previously. Note that $1/X^2$ term can be ignored since X is expected to be large. From (4.35), the following can be written,

$$H'(j\omega) = \frac{j\omega\omega_{oo} - \omega^2/X}{j\omega\omega_{oo}\left(\dfrac{1}{Q_o} + \dfrac{2}{X}\right) + \omega_{oo}^2 - \omega^2} \tag{4.36}$$

$$= \frac{\left(j\omega_{oo} - \dfrac{\omega}{X}\right)\left(-j\omega_{oo}(\dfrac{1}{Q_o} + \dfrac{2}{X}) + \dfrac{\omega_{oo}^2}{\omega} - \omega\right)}{\text{denominator}} \tag{4.37}$$

$$= \frac{\left(\omega_{oo}^2(\dfrac{1}{Q_o}+\dfrac{2}{X}) - \dfrac{\omega_{oo}^2}{X} + \dfrac{\omega^2}{X}\right) + j\left(\dfrac{\omega_{oo}^3}{\omega} - \omega\omega_{oo} + \dfrac{\omega\omega_{oo}}{X}(\dfrac{1}{Q_o}+\dfrac{2}{X})\right)}{\text{denominator}} \tag{4.38}$$

Phase of $H'(j\omega)$ is given as

$$\angle\, H'(j\omega) = \tan^{-1}\left(\frac{\mathrm{Im}[H'(j\omega)]}{\mathrm{Re}[H'(j\omega)]}\right) \tag{4.39}$$

At the steady state, the tuning conditions given in (4.6) and (4.7) will be satisfied, which can be shortly expressed as

$$\frac{\mathrm{Im}[H'(j\omega)]}{\mathrm{Re}[H'(j\omega)]} = \pm 1 \tag{4.40}$$

Note that (4.40) is +1 for one reference value of ω and -1 for the other. By substituting real and imaginary parts of (4.36),

$$\left(\omega_{oo}^2(\frac{1}{Q_o}+\frac{2}{X})-\frac{\omega_{oo}^2}{X}+\frac{\omega_1^2}{X}\right)=+\left(\frac{\omega_{oo}^3}{\omega_1}-\omega_1\omega_{oo}+\frac{\omega_1\omega_{oo}}{X}(\frac{1}{Q_o}+\frac{2}{X})\right) \tag{4.41}$$

$$\left(\omega_{oo}^2(\frac{1}{Q_o}+\frac{2}{X})-\frac{\omega_{oo}^2}{X}+\frac{\omega_2^2}{X}\right)=-\left(\frac{\omega_{oo}^3}{\omega_2}-\omega_2\omega_{oo}+\frac{\omega_2\omega_{oo}}{X}(\frac{1}{Q_o}+\frac{2}{X})\right) \tag{4.42}$$

By eliminating $1/X^2$ terms, Q_o can be found from (4.41) and (4.42), Note that there are two unknowns (Q_o and ω_{oo}) and two equations:

$$Q_o=\frac{\omega_{oo}^2-\dfrac{\omega_1\,\omega_{oo}}{X}}{\left(\dfrac{\omega_{oo}^3}{\omega_1}-\omega_1\,\omega_{oo}-\dfrac{\omega_{oo}^2}{X}-\dfrac{\omega_1^2}{X}\right)} \tag{4.43}$$

$$Q_o=\frac{-\omega_{oo}^2-\dfrac{\omega_2\,\omega_{oo}}{X}}{\left(\dfrac{\omega_{oo}^3}{\omega_2}-\omega_2\,\omega_{oo}-\dfrac{\omega_{oo}^2}{X}-\dfrac{\omega_2^2}{X}\right)} \tag{4.44}$$

where ω_1 and ω_2 are the reference frequencies. By dividing (4.43) by (4.44), one of the unknowns, Q_o, can be eliminated.

$$\omega_{oo}^3\left(\frac{\omega_1+\omega_2}{\omega_1\,\omega_2}\right)-\omega_{oo}^2\frac{1}{X}\left(2+\frac{\omega_1^2-\omega_2^2}{\omega_1\,\omega_2}\right)-\omega_{oo}(\omega_1+\omega_2)-\frac{\omega_1^2+\omega_2^2}{X}=0 \tag{4.45}$$

The value of ω_{oo} will be one of the roots of (4.45). Then, Q_o can be found either from (4.43) or (4.44). As X goes to infinity, (4.45) yields $\omega_{oo}=\sqrt{\omega_1\,\omega_2}$ while Q_o becomes $\omega_{oo}=\sqrt{\omega_1\,\omega_2}/(\omega_2-\omega_1)$ as expected. Unfortunately, the roots of (4.45) is not exact solution of (4.41) and (4.42). Therefore, (4.43) and (4.44) yield two different Q_o values. Better approximations can be made by finding roots of (4.41) and (4.42) in terms of Q_o, then finding Q_o value that makes the roots equal.

By substituting Q_o and ω_{oo} back to the modified transfer function given in (4.35), the tuned (effective) quality factor and center frequency values can be found (most likely numerically). Figures 37 and 38 show the tuning errors calculated by using

Matlab for various X and desired Q values (see Appendix C). The results show that the frequency tuning error is less than 0.1%. For Q-tuning the error is less than 1%. Note that the actual tuning errors are expected to be less than the ones shown in Fig. 37 and Fig. 38 since part of the error can attributed to the error due the approximation of Q_o and ω_{oo}. The irregularity in Fig. 38 can also be attributed to the error in the approximation.

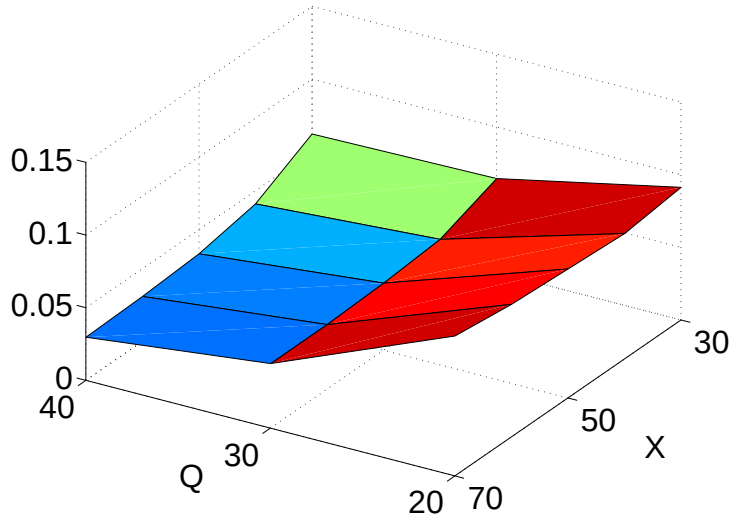

Fig. 37. Frequency-tuning error with a parasitic zero at X f_0.

4. Resolution of DAC-Counter Pair

One limitation of the proposed tuning circuit architecture in Section B is the effect of the finite resolution of the DAC-counter pair (L_c) on the tuning accuracy (see Fig. 32). First of all, the output voltage range of DAC should be large enough to cover the process variation. For a given filter design, the required voltage range can be determined by Monte-Carlo and corner simulations. The next step is to

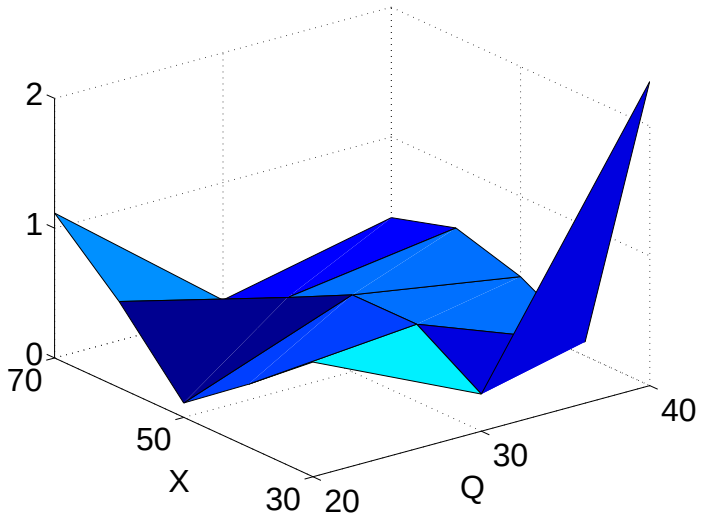

Fig. 38. Q-tuning error with a parasitic zero at X f_0.

determine the minimum required resolution so that the error due to finite resolution is reasonably small.

Let the amount of the expected center frequency variation is Δf_0. Assume the variation is exactly covered by the output voltage range of DAC and the center frequency, f_0, is linearly proportional to the control voltage. Based on these assumptions and ignoring other limitations, the frequency-tuning error can be written as

$$f_0 \text{ error} = \frac{\Delta f_0 / 2^{L_c}}{f_{d0}} \cdot 100\% \tag{4.46}$$

where f_{d0} is the desired center frequency. Note that $\Delta f_0 / 2^{L_c}$ is the center frequency shift corresponding to one increment of the control voltage. If $\pm 50\%$ variation for the center frequency is assumed, i.e., $\Delta f_0 = f_{d0}$, (4.46) simplifies to

$$f_0 \text{ error} = \frac{1}{2^{L_c}} \cdot 100\% \tag{4.47}$$

For 1% frequency tuning error, L_c should be at least 7 bits from (4.47).

For Q-tuning, bandwidth variation as well as f_0 variation play role on the tuning accuracy. Considering only the error due to finite resolution, Q can be written as

$$Q = \frac{f_{d0} \pm \dfrac{\Delta f_0}{2^{L_c}}}{BW \pm \dfrac{\Delta f_0}{2^{L_c}}} \tag{4.48}$$

where BW is the desired bandwidth value. (4.48) can be further simplified;

$$Q = \frac{f_{d0}\left(1 \pm \dfrac{\Delta f_0/f_{d0}}{2^{L_c}}\right)}{BW\left(1 \pm \dfrac{\Delta f_0/BW}{2^{L_c}}\right)} \tag{4.49}$$

$$= \frac{f_{d0}\left(1 \pm \dfrac{\Delta f_0/f_{d0}}{2^{L_c}}\right)}{BW\left(1 \pm \dfrac{Q\Delta f_0/f_{d0}}{2^{L_c}}\right)} \tag{4.50}$$

$$= Q_d\left(1 + \frac{\dfrac{\Delta f_0/f_{d0}}{2^{L_c}}(1 \mp Q_d)}{1 \pm Q_d\dfrac{\Delta f_0/f_{d0}}{2^{L_c}}}\right) \tag{4.51}$$

where Q_d is the desired quality factor ($Q_d = f_{0d}/BW$) .

From (4.51), Q-tuning error can be written as

$$Q \text{ error} = \frac{\dfrac{\Delta f_0/f_{d0}}{2^{L_c}}(1 \mp Q_d)}{1 \pm Q_d\dfrac{\Delta f_0/f_{d0}}{2^{L_c}}} \cdot 100\% \tag{4.52}$$

For $\pm 50\%$ center frequency variation, the Q-error expression simplifies to

$$Q \text{ error} = \frac{1 \mp Q_d}{2^{L_c} \pm Q_d} \cdot 100\% \tag{4.53}$$

$$\approx \frac{Q_d}{2^{L_c} \pm Q_d} \cdot 100\% \tag{4.54}$$

Using (4.47) and (4.54), one can determine the resolution required for DAC-

counter pair for ±50% variation of center frequency. Note that in the worst case, the sign of Q_d in the denominator in (4.54) will be minus. Obviously, for a given resolution, larger Q_d increases the error. Fig. 39 shows the minimum required resolution to achieve 1% Q-tuning accuracy in the worst case for ±50% center-frequency variation.

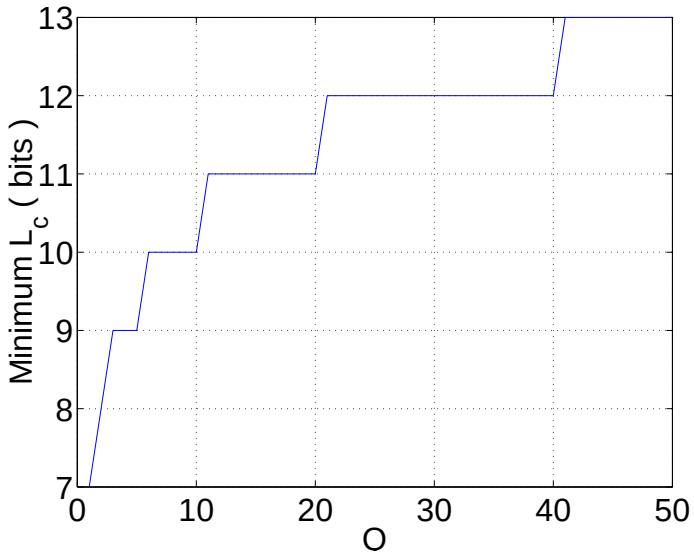

Fig. 39. Minimum required resolution L_c for 1% Q-tuning error (worst case).

D. Simulation Results

The tuning system is simulated using Matlab. Figure 40 shows the snapshots of bandpass magnitude and lowpass phase responses during simulation. Initially, the filter has low Q and low f_0 as indicated green solid curve. The tuning start with increasing f_0, note that the control voltage V_F is inversely proportional to f_0 (Fig. 41). Once f_0 is between f_1 and f_2, Q starts increasing. Eventually, both parameters

converge to desired values. The system is simulated in Cadence using ideal filter and tuning circuit. Figure 42 shows the convergence of control voltages. Zoomed version clearly shows that V_F and V_Q do not change simultaneously.

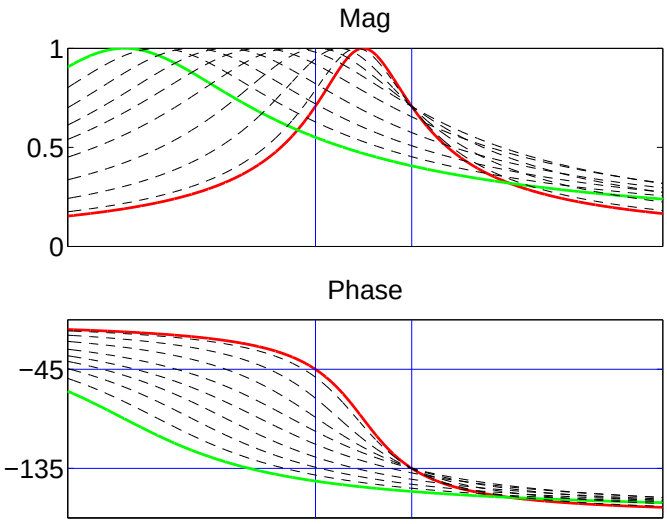

Fig. 40. Snapshots of filter response during tuning.

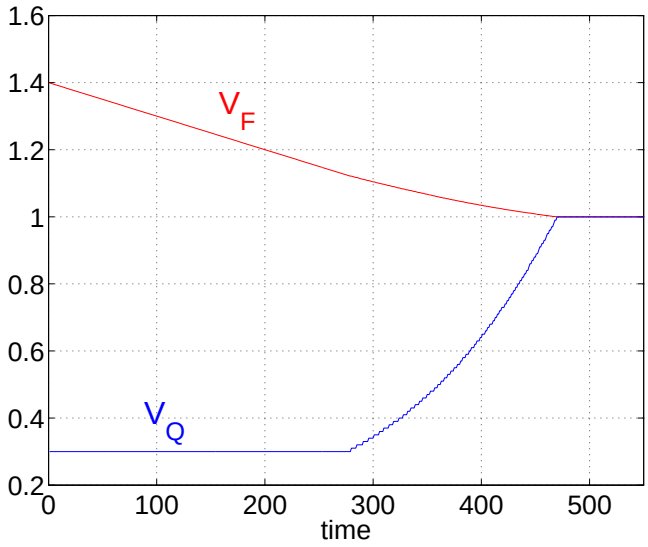

Fig. 41. Transient response of control voltages (Matlab simulation).

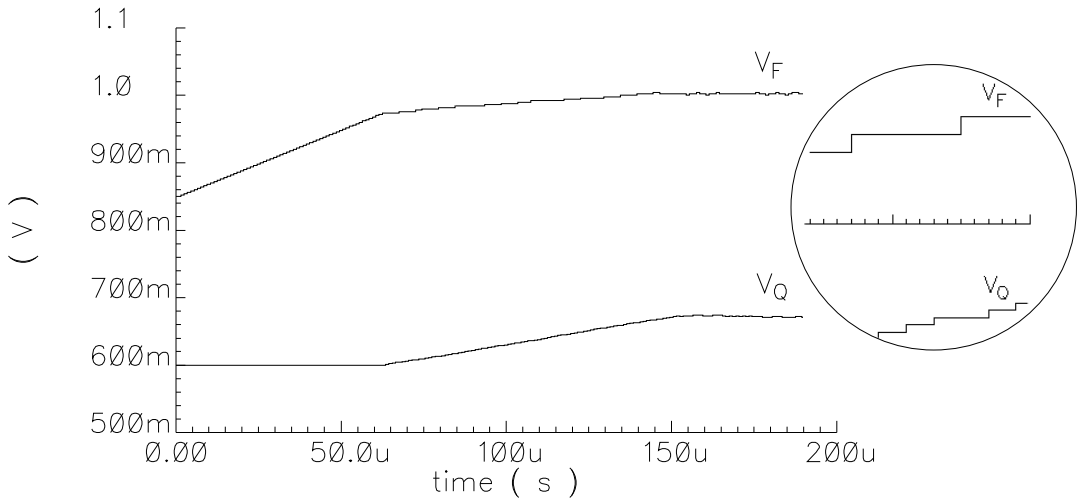

Fig. 42. Transient response of control voltages (Cadence simulation).

E. Prototype Design

Discrete version of automatic tuning circuit is built on a Printed Circuit Board (PCB) to verify the system experimentally.

1. Filter

A discrete LC-type filter shown in Fig. 43 is used as tunable high-Q filter. A chip inductor and a chip capacitor are used to achieve high-Q values. Frequency tuning is achieved through voltage variable capacitance diodes (NTE 612, 614 and 616 with 33 pF, 12 pF and 5 pF diode capacitances connected in parallel). Q-tunability is accomplished by JFET N-channel transistor connected to the output node. The current drawn from the output node is controlled by V_Q control voltage, consequently, it changes the quality factor of the filter. The component values are given in Table II. It is not explicitly shown, but large bypass capacitors and a buffer (MAX4005) at the output are used for filter characterization. Note that this filter has only one output which is bandpass. Therefore, the reference signals are slightly modified.

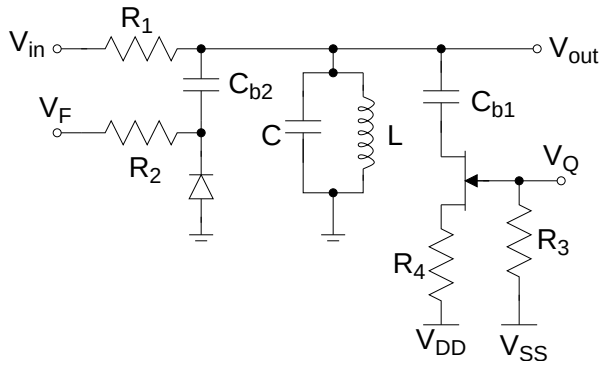

Fig. 43. Tunable discrete LC-type filter.

TABLE II

COMPONENT VALUES USED FOR LC-TYPE TUNABLE FILTER

L	0.56 μH
C	330 pF
C_{b1}, C_{b2}	10 nF
R_1	10 kΩ
R_2	1 kΩ
R_3	150 kΩ
R_4	56 kΩ

2. Tuning Circuit

Discrete version of the tuning circuit is constructed using various commercial chips. Three MAX4223 chips are used as comparators. One is used in the signal path to convert sinusoidal signal to rail-to-rail signal, and the other two are used in the reference path for delay compensation.

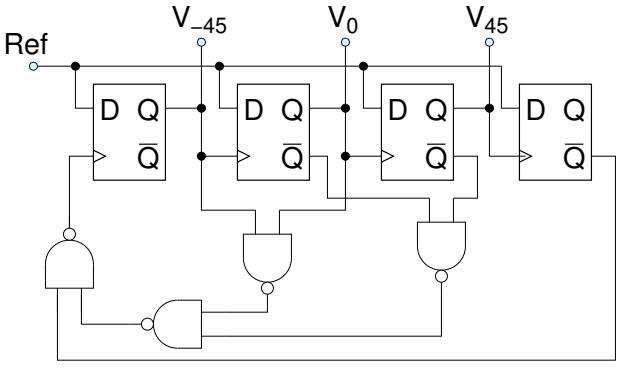

Fig. 44. Reference phase generator circuit.

Quad D-type flip-flop chips (VHC175) are used for the generation of the reference clock and 45° delayed versions (see Fig. 44) and binary phase comparators (see Fig. 32). Since the filter does not have lowpass output, V_{-45} and V_{45} clock signals are used as references instead of V_{-135} and V_{-45}. Up/down counter and logic circuit is constructed by using 74 family integrated circuits. Table III shows the complete list of commercial ICs used in the test setup.

TABLE III

COMMERCIAL ICS USED IN THE TEST SETUP OF PROTOTYPE TUNING SYSTEM

MAX4223	1 GHz Current feedback amplifier
MAX4005	950 MHz FET-input buffer with 75 Ω output
VHC175	Quad D-type flip-flop chip
SN74F00N	Quad 2-input nand-gate
SN74F04N	Hex inverters
SN74F08N	Quadruple 2-input positive and-gate
SN74F74N	Dual positive-edge-triggered DFFs with clear and preset
74LS14N	Hex inverter schmitt trigger
74F191N	Up/Down binary counter with reset and ripple clock

A simple 8-bit digital-to-analog converter shown in Fig. 45 is built with LM741 opamps. The second opamp stage allows the adjustment of output control voltage level through V_1 and V_2 bias voltages.

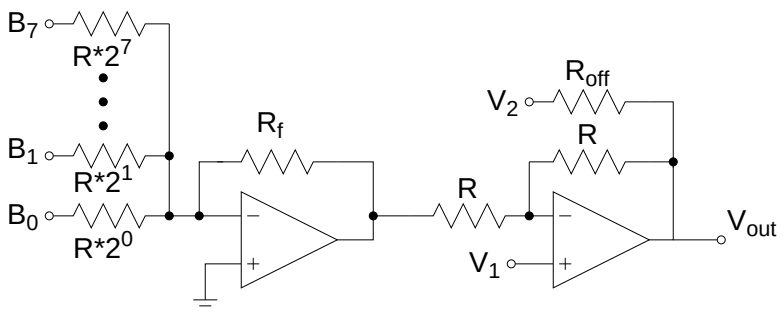

Fig. 45. Simple DAC circuit based on summing amplifier.

F. Experimental Results

The discrete prototype circuit discussed in the previous section is constructed and the tuning system is verified experimentally. Agilent 81110A-pulse generator is used as the frequency synthesizer. Synchronization of FS and the tuning circuit is achieved through a computer using Labview, which sends a signal to FS to switch frequency generated f_1 or f_2. Right before sending the switch-frequency signal, it sends a rising edge to tuning core circuit (see Fig. 32) and the result of binary phase comparison is sampled to the succeeding DFF. Once both phase comparisons performed, i.e., A and B are determined, Labview sends update signal to counters.

Since the system is digital, the tuning speed can be as slow as desired. With this setup, the duration of a single tuning slot is limited by slow GPIB connection. In general the speed of the tuning circuit is limited by the settling time of FS and resolution of DAC-counter pair. The control voltages V_F and V_Q are observed for a while, when they converged, the tuning circuit is disabled. Then, using spectrum analyzer, the tuned magnitude response is measured for different target center frequency and Q values. Since an external FS is used, target f_0 and Q values do not have to be fixed. Therefore, the experiment was repeated for various target values

within the tuning range of the filter. Figure 46 shows the measured bandpass filter magnitude response for a fixed desired Q ($Q_d = 18$) and varying reference frequency. Table IV summarizes the results. In Fig. 47, the reference frequency is fixed at f_R = 5.4 MHz and the desired Q is varied. Table V summarizes the measurement data for this plot. Due to discrete design, offline tuning method is preferred in order to verify the operation.

TABLE IV

FREQUENCY-TUNING RANGE FOR $Q_d = 18$

Desired		Tuned		Error (%)	
f_R (MHz)	Q_d	f_0 (MHz)	Q	F Error	Q Error
5.2	18	5.20	18.1	0.0	+0.6
5.4	18	5.39	18.2	-0.2	+1.1
5.9	18	5.86	18.2	-0.7	+1.1

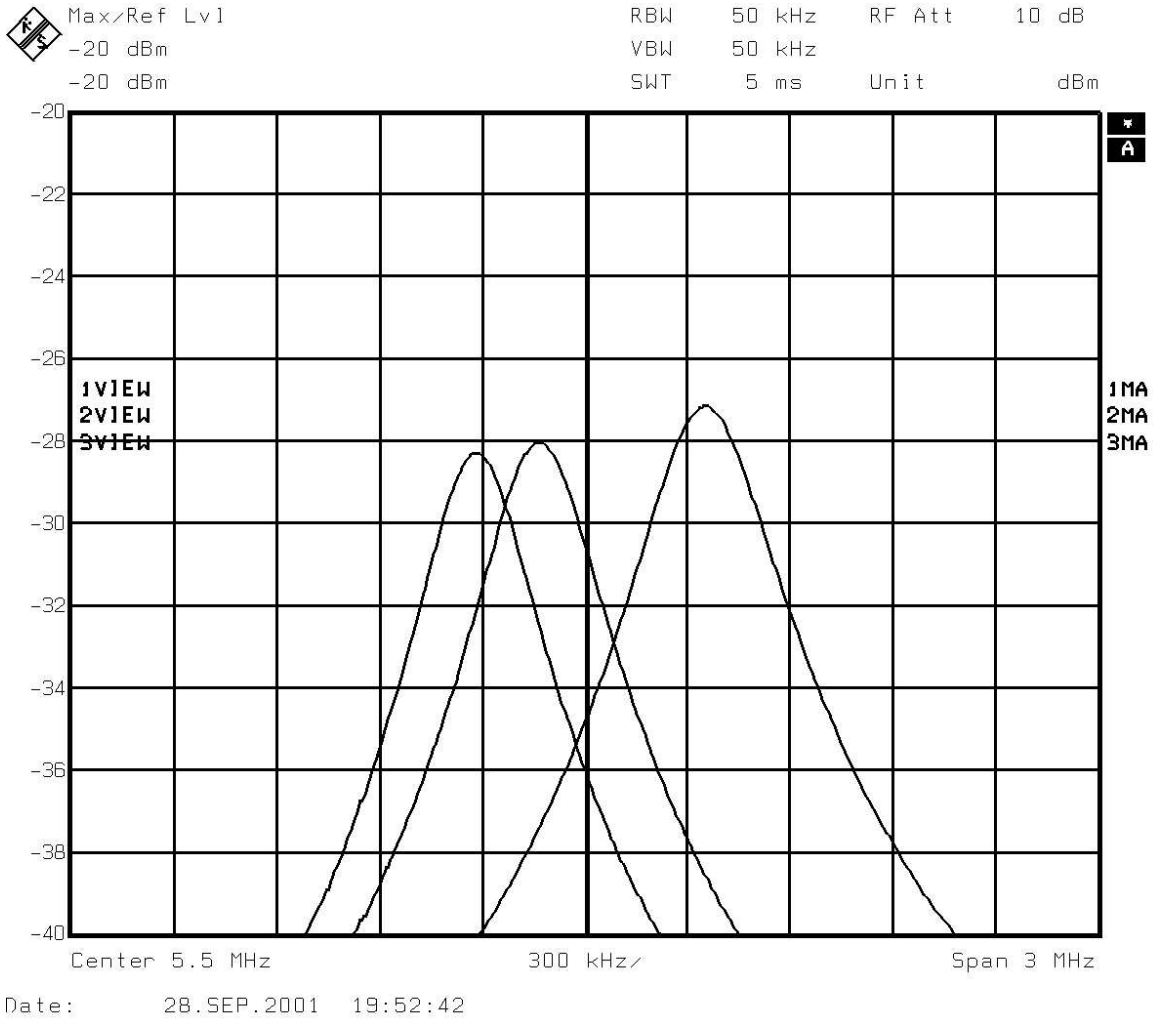

Fig. 46. Measured bandpass filter magnitude response for fixed target Q of 18 and target center frequencies of 5.2, 5.4, and 5.9 MHz.

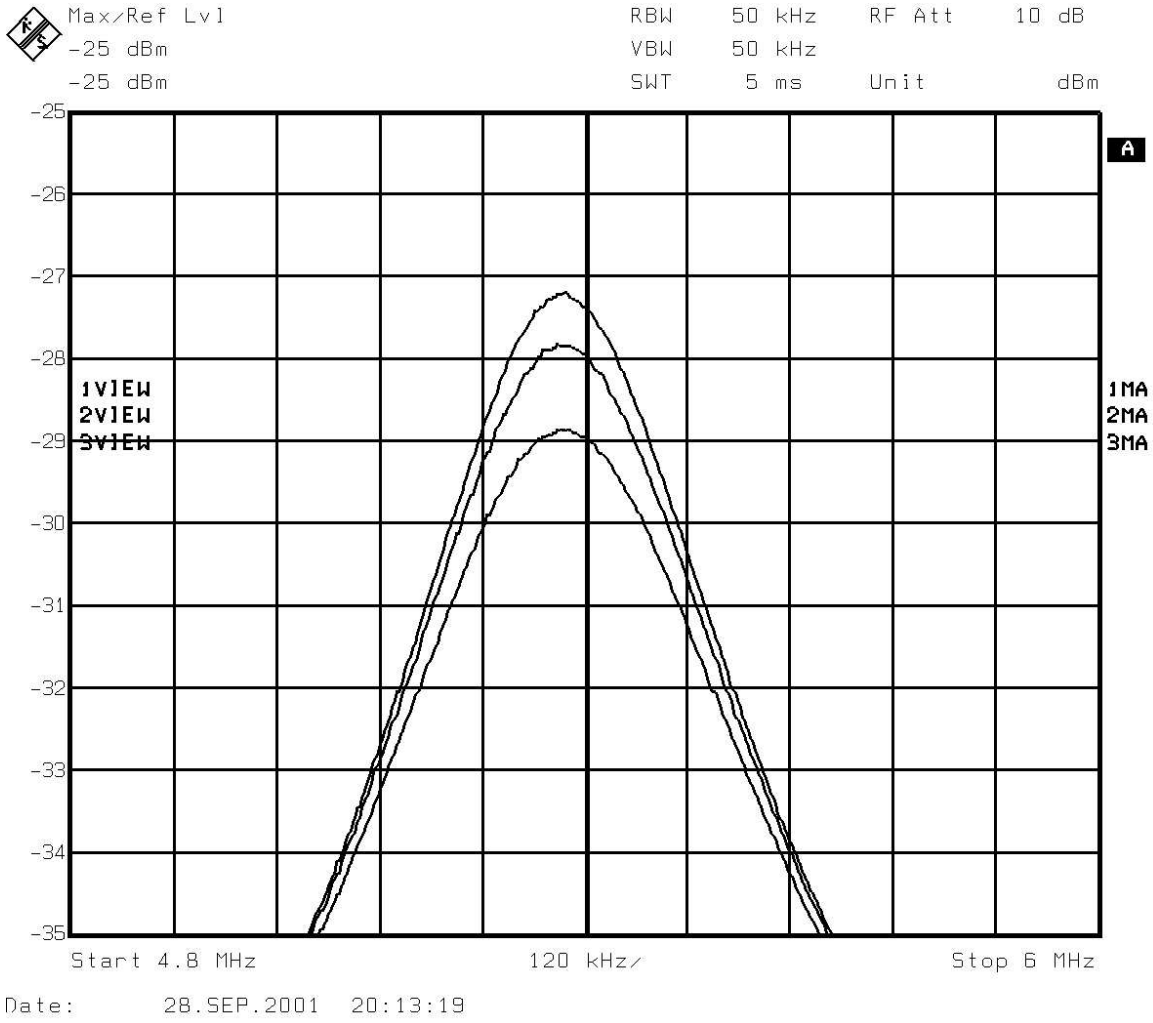

Fig. 47. Measured bandpass filter magnitude response for fixed center frequency of 5.4 MHz and target Qs of 16, 18, and 20.

TABLE V

Q-TUNING RANGE AT $f_o = 5.4$ MHz

Desired		Tuned		Error (%)	
f_R (MHz)	Q_d	f_0 (MHz)	Q	F Error	Q Error
5.4	16	5.39	15.8	-0.2	-1.2
5.4	18	5.39	18.2	-0.2	+1.1
5.4	20	5.38	20.0	-0.3	0.0

CHAPTER V

HIGH-ORDER DIGITAL-TUNING METHOD BASED ON PHASE COMPARISON

In this chapter, an automatic tuning technique is proposed for the tuning of high-order filters [42]-[43]. As mentioned in Chapter III, efficient high-order automatic tuning techniques with direct tuning approach are required while no practical method has been reported in the literature yet. High-order digital-tuning method based on phase comparison (HDTPC) is an extension of DTPC to high-order filters. The principle, practical limitations, circuit implementation, simulation, prototype design, and experimental results are presented.

A. Principle

A high-order filter can be constructed by cascading biquads as shown in Fig. 48. Each biquad section has its own desired center frequency (f_{0di}) and quality factor (Q_{di}). Lowpass outputs (V_{lp}) are used for convenience as in the case of DTPC since it has relatively large gain at low frequencies. Note that only difference between lowpass and bandpass outputs is the 90° phase shift in terms of phase response.

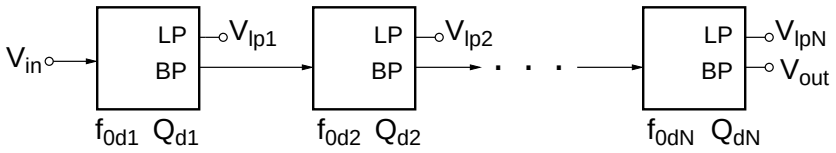

Fig. 48. A bandpass filter formed by cascading biquads.

Define the phase response seen at the i^{th} lowpass output as,

$$\phi_i(f) = \angle \frac{V_{lpi}}{V_{in}} \tag{5.1}$$

The tuning circuit applies two reference frequencies for each biquad and calibrates each lowpass output phase to satisfy the following

$$\phi_{di}(f_{ai}) = \phi_{ai} \tag{5.2}$$

$$\phi_{di}(f_{bi}) = \phi_{bi} \tag{5.3}$$

where $\phi_{di}(f)$ is the desired phase response at the output V_{lpi} and f_{ai}, f_{bi}, ϕ_{ai} and ϕ_{bi} are reference frequencies and phases, respectively. How to determine reference frequencies are explained in the following section (Section B). Note that for $\phi_{a1} = -45°$ and $\phi_{b1} = -135°$, the situation simplifies to second-order case (see Chapter IV).

Analytical expressions for tuned f_{0ti} and Q_{ti} can be found as follows. Lowpass phase response of a single biquad is given as

$$\phi_{lp}(f) = -\tan^{-1}\left(\frac{f\ f_0/Q}{f_0^2 - f^2}\right) \tag{5.4}$$

while bandpass phase is given as

$$\phi_{bp}(f) = 90° - \tan^{-1}\left(\frac{f\ f_0/Q}{f_0^2 - f^2}\right) \tag{5.5}$$

When the systems are connected in cascade, the phase responses of the individual systems are added. Using (5.5), the phase at i^{th} output can be written as

$$\phi_i(f) = \angle \frac{V_{lpi}}{V_{in}} = (i-1)\,90° - \sum_{j=1}^{i} \tan^{-1}\left(\frac{f_{0j} f/Q_j}{f_{0j}^2 - f^2}\right) \tag{5.6}$$

Solving (5.3), (5.2), and (5.6) simultaneously for $f_{ai} = k_{ai} f_R$ and $f_{bi} = k_{bi} f_R$, where f_R is a fixed reference frequency and k_{ai} and k_{bi} are scaling factors, the expressions

for tuned values f_{0ti} and Q_{ti} can be found as

$$f_{0ti} = f_R\sqrt{\frac{k_{bi}^2 - k_{ai}^2}{1-\Lambda_i} + k_{ai}^2}\ , \tag{5.7}$$

$$Q_{ti} = \frac{k_{ai}}{\Gamma_i(k_{bi}^2 - k_{ai}^2)}\sqrt{(\Lambda_i k_{ai}^2 - k_{bi}^2)(\Lambda_i - 1)} \tag{5.8}$$

where

$$\Lambda_i = \frac{k_{bi}\Gamma_i}{k_{ai}\tan\left((i-1)\ 90^\circ - \phi_{bi} - \sum_{j=1}^{i-1}\tan^{-1}\left(\frac{f_{0j}k_{bi}f_R/Q_j}{f_{0j}^2 - f_R^2k_{bi}^2}\right)\right)} \tag{5.9}$$

and

$$\Gamma_i = \tan\left((i-1)\ 90^\circ - \phi_{ai} - \sum_{j=1}^{i-1}\tan^{-1}\left(\frac{f_{0j}k_{ai}f_R/Q_j}{f_{0j}^2 - f_R^2k_{ai}^2}\right)\right) \tag{5.10}$$

An algorithm has to be implemented that will converge and achieve the tuning conditions given in (5.2) and (5.3) at the steady state. Define two binary numbers A_i and B_i to indicate whether the phases at the reference frequencies is larger or smaller than the reference phases

$$A_i = \begin{cases} 0 & , \quad \phi_i(f_{ai}) > \phi_{ai} \\ 1 & , \quad \phi_i(f_{ai}) < \phi_{bi} \end{cases} \tag{5.11}$$

$$B_i = \begin{cases} 0 & , \quad \phi_i(f_{bi}) > \phi_{ai} \\ 1 & , \quad \phi_i(f_{bi}) < \phi_{bi} \end{cases} \tag{5.12}$$

Assuming the parameters of previous biquads f_{0i} and Q_i are tuned close to the desired values, based on the definitions of A_i and B_i, four different plots can be generated for i^{th} lowpass output as shown in Fig. 49. Only one tuning action is assigned for each case as summarized in Table VI. The previous discussion on DTPC can be repeated here for high-order tuning as well. In Fig. 49(a) and (b), f_{0i} has to be increased and decreased, respectively. In Fig. 49(c) and (d), however, Q_i has to be decreased or increased, respectively. Dashed lines indicating reference frequencies

and phases show the tuning reference window. When the filter is tuned, the phase curve will pass through the cross-section points.

TABLE VI

TUNING PROCESS FOR THE i^{th} BIQUAD, ENABLED ONLY IF $A_{i-1} \oplus B_{i-1} = 1$

Case	$\mathbf{A_i}$	$\mathbf{B_i}$	**Action**
(a)	1	1	$f_0 \Uparrow$
(b)	0	0	$f_0 \Downarrow$
(c)	0	1	$Q \Downarrow$
(d)	1	0	$Q \Uparrow$

The tuning for i^{th} biquad should be enabled when the center frequency of the $(i-1)^{th}$ biquad is tuned close to the target value for reliable tuning. The tuning for i^{th} biquad can be conditioned on the status of the previous biquad. Note that when the Q tuning is enabled for i^{th} biquad, its center frequency is within the reference window, i.e., $f_{a(i-1)} \leq f_{0t(i-1)} \leq f_{b(i-1)}$. From the truth table given in Table VI, it can be seen that Q tuning for $(i-1)^{th}$ biquad is enabled when $A_{i-1} \oplus B_{i-1} = 1$[1]. Note that the tuning of the first biquad is always enabled.

The flowchart describing the overall tuning algorithm can be seen in Fig. 50. The tuning circuit performs all the phase comparisons in one tuning cycle starting from the first lowpass output, i.e., A_i and B_i are determined, and consequently the tuning logic determines the tuning actions whether to increase or decrease Q_i or f_{0i}. After processing the last output, at the end of the tuning cycle, all the control voltages (V_{fi} and V_{qi}) are updated. Then the next tuning cycle starts.

[1] $\oplus$ represents the XOR operation

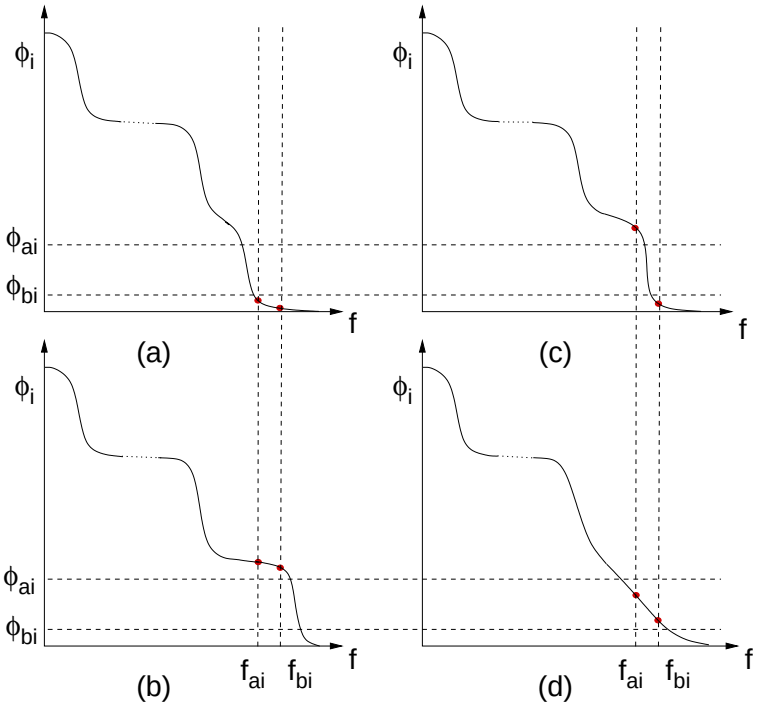

Fig. 49. Phase response at the output V_{lpi}, $\phi_i(f)$, for different states of the i^{th} biquad.

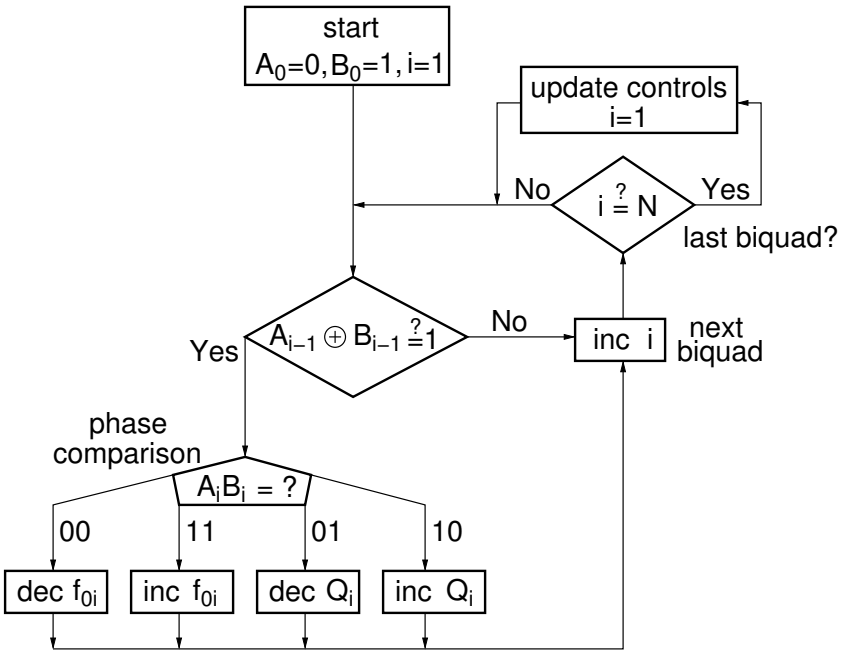

Fig. 50. A flowchart showing the overall tuning procedure for high order filters.

Initially, there will be some error since the previous biquads are not tuned correctly; however, the errors will converge to zero as the outputs are tuned. Eventually, all the outputs and the overall output response of the filter will be tuned correctly. As long as the tuning is enabled, the tuning loops will be functional indefinitely. After certain point in time, all the loops converge, the tuning can be disabled and the filter can be used to process signals. The convergence of the tuning loops can also be sensed by digital means since up-down control signals will toggle between 1 and 0 at steady state.

B. Determination of Reference Frequencies and Phases

For a $(2N)^{th}$ order filter, the reference frequencies (f_{ai}, f_{bi}) and the reference phases (ϕ_{ai}, ϕ_{bi}) are needed to be determined. The following graphical approach can used. Each lowpass output phase, $\phi_i(f)$, is plotted by varying Q_i around the desired value

Q_{di} with previous biquads having the desired values, i.e, $Q_j = Q_{jd}$ and $f_{0j} = f_{0jd}$ for $1 \leq j \leq i-1$. As an example, the third output of an arbitrary filter is plotted in Fig. 51 with $\pm 50\%$ variation of Q_i from the desired value. All the curves with

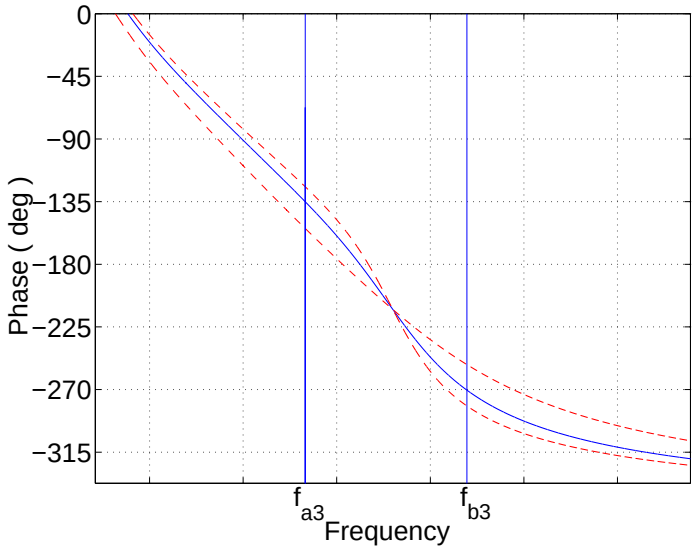

Fig. 51. $\phi_i(f)$ with $\pm 50\%$ variation of Q_i from the desired value (for i=3).

different Qs cross at certain critical phase ϕ_{ci}. The references maximizing $\Delta\phi/\Delta Q$ and satisfying the following conditions should be chosen.

$$\phi_{ai} > \phi_{ci} > \phi_{bi} \tag{5.13}$$

$$\phi_{ai} = -P_{ai}\frac{180^\circ}{M} \tag{5.14}$$

$$\phi_{bi} = -P_{bi}\frac{180^\circ}{M} \tag{5.15}$$

where P_{ai}, P_{bi}, and M are integers numbers. Being multiple of $180^\circ/M$ serves certain circuit implementation purposes, which will be addressed in Section E. Also having larger variation of phase at the reference frequency is desired to have less sensitivity to any possible phase offsets in the circuit implementation (see Sections C and E). ϕ_{ci}

can be different for different types of filters depending on individual quality factors and center frequencies of the previous sections. The condition in (5.13) guarantees proper operation of the tuning as shown in Table VI. For the particular example shown in Fig. 51 -135° and -270° are chosen, and corresponding frequencies are extracted from ideal filter response either graphically or numerically.

C. Limitations

In this section, practical limitations of the tuning technique such as resolution of frequency synthesizer, phase offset of the detector, and parasitics of the filter are analyzed.

1. Resolution of Frequency Synthesizer

The reference frequencies eventually will be generated by using a single frequency synthesizer. The resolution of the synthesizer may become an issue, especially when many different reference frequencies are needed to be generated.

A conventional frequency synthesizer with a resolution of L generates signals at frequencies $f_R, 2f_R, 3f_R, \cdots 2^L f_R$, where f_R is the fixed input signal to the synthesizer. f_R can be chosen as the maximum reference signal to be generated divided by 2^L as

$$f_R = \frac{\max(f_{ai}, f_{bi})}{2^L} \tag{5.16}$$

As previously defined, scaling factors k_{ai} and k_{bi} are given as

$$k_{ai} = \frac{f_{ai}}{f_R} \tag{5.17}$$

$$k_{bi} = \frac{f_{bi}}{f_R} \tag{5.18}$$

k_{ai} and k_{bi} can be approximated by rounding them to the closest integer,

$$K_{ai} = \text{round}(k_{ai}) \tag{5.19}$$

$$K_{bi} = \text{round}(k_{bi}) \tag{5.20}$$

Substituting (5.19) and (5.20) into (5.7)-(5.10), exact tuning errors can be found for each Q and f_0. The errors are not necessarily additive from one biquad section to another; an error in a particular biquad may compensate errors in succeeding biquad(s) or worsen. This depends on particular design, i.e, type of the filter, choice of L and f_R. One rough measure of error is the deviation of K_{ai} or K_{bi} from the desired value,

$$\text{Error} = \frac{|K_i - k_i|}{k_i} \tag{5.21}$$

Worst case rounding error can be $|K - k| = 0.5$. Ignoring the effects of the previous filter sections, the largest frequency error for a biquad section is encountered when the rounding errors are in the same direction such as

$$K_{ai} = k_{ai} + 0.5 \tag{5.22}$$

$$K_{bi} = k_{bi} + 0.5 \tag{5.23}$$

or

$$K_{ai} = k_{ai} - 0.5 \tag{5.24}$$

$$K_{bi} = k_{bi} - 0.5 \tag{5.25}$$

However, the following gives the largest Q error

$$K_{ai} = k_{ai} + 0.5 \tag{5.26}$$

$$K_{bi} = k_{bi} - 0.5 \tag{5.27}$$

or

$$K_{ai} = k_{ai} - 0.5 \tag{5.28}$$

$$K_{bi} = k_{bi} + 0.5 \tag{5.29}$$

The worst-case frequency-tuning error with respect to L is plotted for different target Q values in Fig. 52. The error is found to be less than 0.2% for $L > 8$ bits. The frequency error shows negligible dependence on quality factor. Figure 53 shows that the worst case Q tuning error increases as Q_d increases as expected since it requires more resolution with decreasing bandwith of the biquad section. 11 bits is required to achieve 1% tuning accuracy for Q_d of 40 (in the worst case).

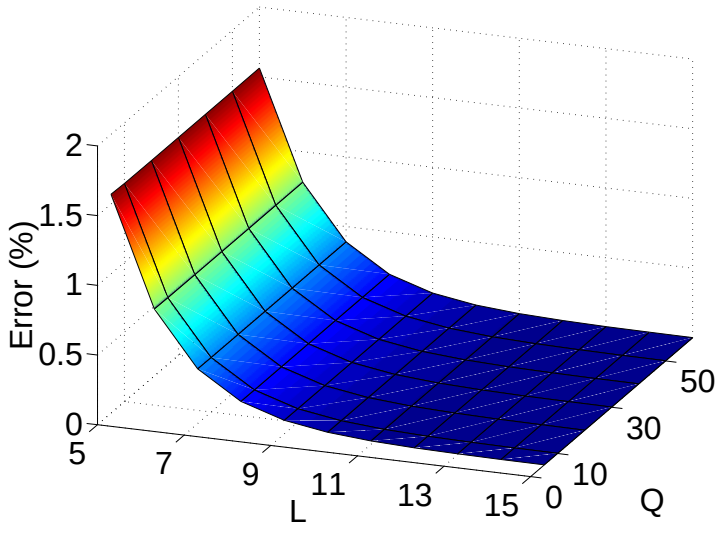

Fig. 52. Worst-case frequency-tuning error for a fixed center frequency.

The effect of resolution on a 6^{th}-order prototype filter (with Qs of 33, 16, 33) is illustrated in Figs. 54 and 55. Lower precision causes ripples in the passband region due to relatively large Q-tuning errors, while the center frequency error is less. For

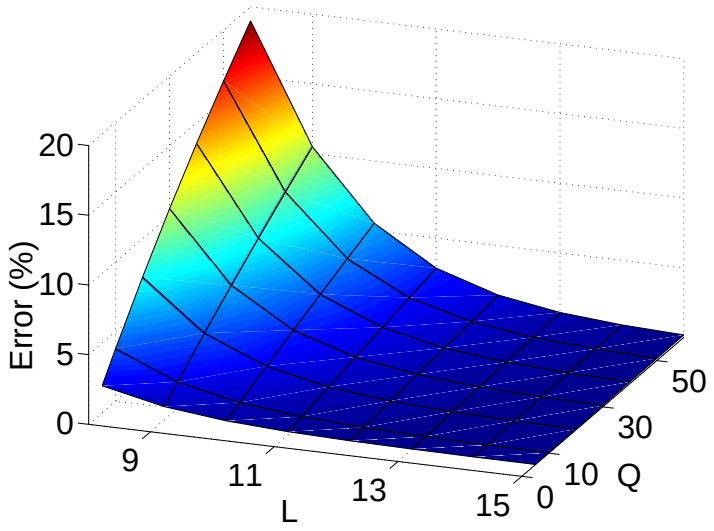

Fig. 53. Worst case Q-tuning error for a fixed center frequency.

this particular case, the tuned magnitude response and the desired response are in a good agreement for 8-bit resolution although the worst case analysis yields 20% Q-tuning error for the prototype filter.

The performance of the tuning circuit can be improved by employing a narrow-band frequency synthesizer [44]. In the FS model, the frequency synthesizer covers low-frequency band unnecessarily. Note that high-Q filters are usually narrow devices. A narrow-band frequency synthesizer can be used to achieve better tuning accuracy for a given resolution.

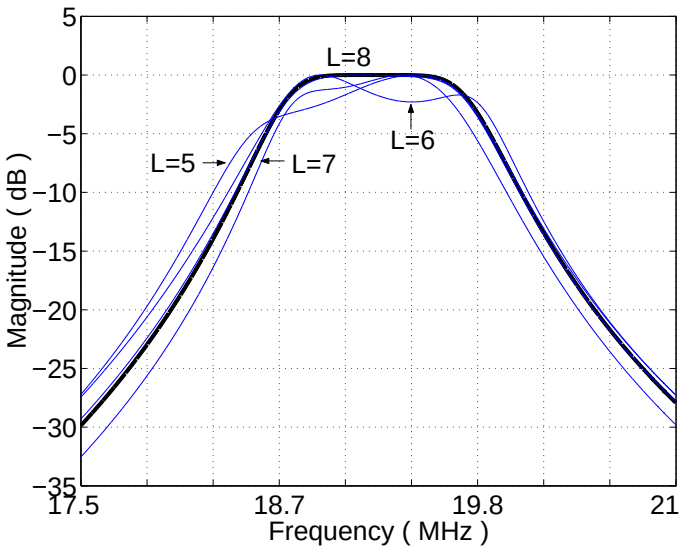

Fig. 54. L vs tuned (normalized) magnitude response.

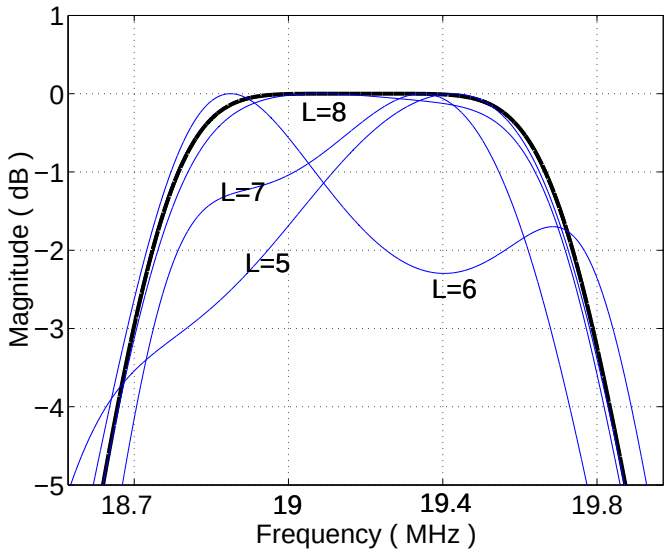

Fig. 55. L vs tuned (normalized) magnitude response, zoomed.

2. Phase Offset Error

The tuning method does not require the detection of absolute phase values. Hence, a binary phase comparator can be employed; however, any possible phase offset in the comparison or any additional delay in the reference path or signal path will degrade the performance of the overall system.

Figures 51 and 56 illustrate the phase variation at references with a variation of Q by $\pm 50\%$ and center frequency by $\pm 10\%$, respectively. The phase change at references are more sensitive to the center frequency variation. Consequently, the frequency tuning is expected to be more robust to the phase offset errors.

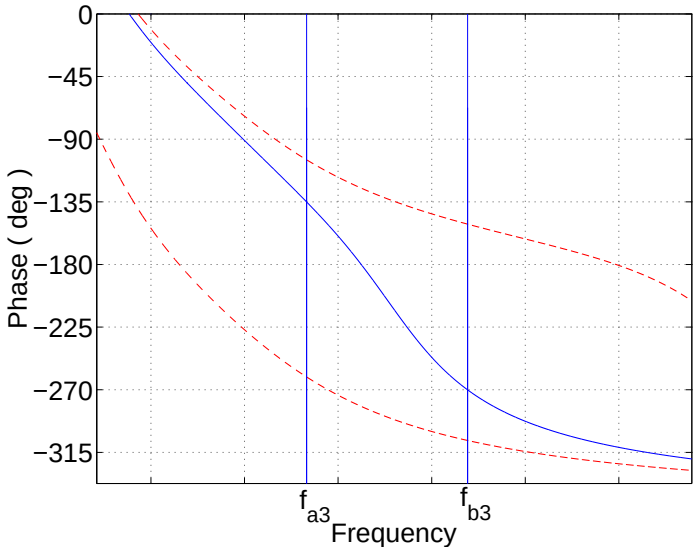

Fig. 56. $\phi_i(f)$ with $\pm 10\%$ variation of f_{0i} from the desired value (for i=3).

Assume $\Delta\phi_{ai}$ and $\Delta\phi_{bi}$ to be the phase offsets, then (5.2) and (5.3) become

$$\phi_{di}(f_{ai}) = \phi_{ai} + \Delta\phi_{ai} \tag{5.30}$$

$$\phi_{di}(f_{bi}) = \phi_{bi} + \Delta\phi_{bi} \tag{5.31}$$

By solving (5.30) and (5.31), the tuned values f_{0ti} and Q_{ti}, and the error can be found. The previous results given in Eqs. (5.7) - (5.10) are still valid with the exception that ϕ_{ai} and ϕ_{bi} are replaced with $\phi_{ai} + \Delta\phi_{ai}$ and $\phi_{bi} + \Delta\phi_{bi}$, respectively. For a given design, assuming different phase offsets, the tuning errors for each f_0 and Q can be calculated by using the resulting equations. For high-order filters, there is no commonly accepted tuning-error measure. One way to define the tuning accuracy would be in terms of accuracy of the center frequency or bandwidth of the overall filter response.

To see the overall effect of individual f_0 and Q-tuning errors, the filter response can be plotted and compared with the desired response to evaluate the performance of the tuning. Assuming $|\Delta\phi_{ai,bi}| = 0°$, $5°$, $10°$, and $15°$, the tuned responses of a 6^{th}-order Butterworth (with Qs of 33.2, 16.6, and 33.2) and a 6^{th} order Chebyshev filters (with Qs of 71.4, 35.7, and 71.4) are shown in Figs. 57 and 58, respectively. In both cases, additional phase offsets increase the ripple in the passband regions while the center frequencies of the filters are affected relatively less. The ripple error due to phase offset is relatively larger in the case of Chebyshev due to larger Q values.

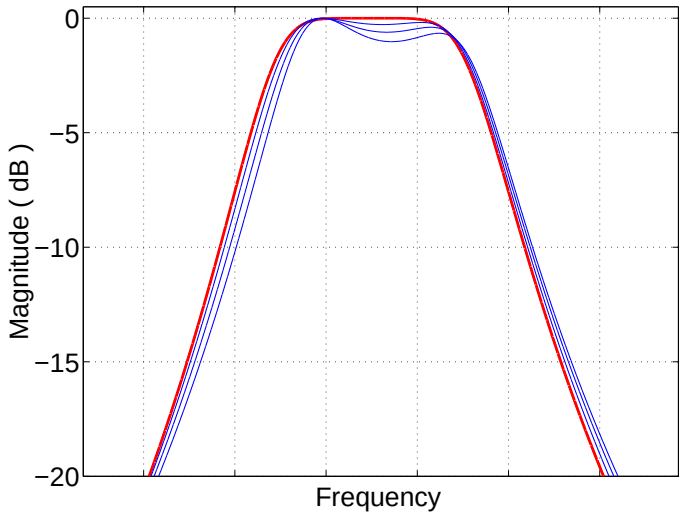

Fig. 57. Effect of phase offset error on tuned magnitude response of a 6^{th}-order Butterworth filter.

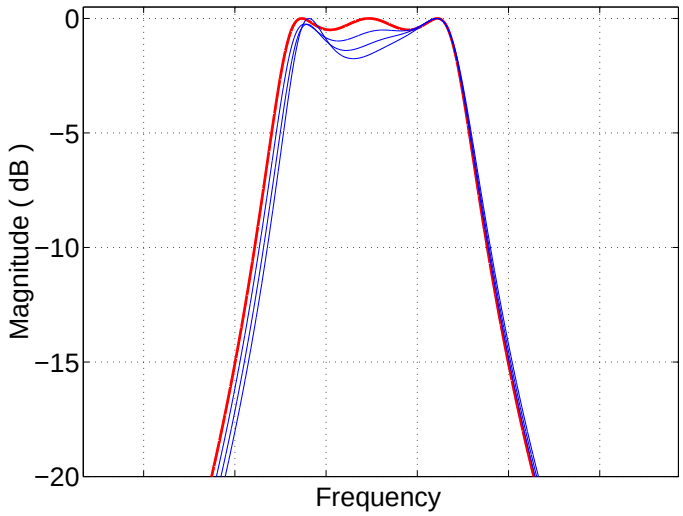

Fig. 58. Effect of phase offset error on the tuned magnitude response of a 6^{th}-order Chebyshev filter.

3. Parasitic Pole and Zero Effect

In the ideal case, a biquad has only two poles. In reality, there might parasitic poles and zeros at higher frequencies. To illustrate the effect of the parasitics on the tuning performance, a G_m-C filter shown in Fig. 59 will be analyzed. The transfer function

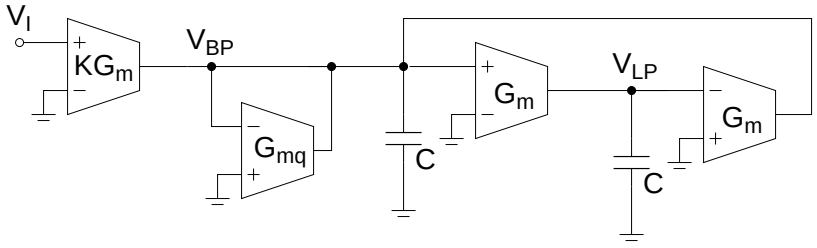

Fig. 59. A second-order G_m-C filter.

seen at bandpass output can be written as

$$H(s) = \frac{V_{LP}}{V_I} = \frac{sKG_m/C}{s^2 + s\,G_{mq}/C + (G_m/C)^2} \tag{5.32}$$

A G_m having a parasitic pole at a frequency ω_p can be modeled as [40], [41]

$$G_m = \frac{G_{m0}}{1 + \dfrac{s}{\omega_p}} \tag{5.33}$$

X is defined to represent the relative location of parasitic pole,

$$X = \frac{\omega_p}{G_{m0}/C} \tag{5.34}$$

Equation (5.33) can be rewritten in terms of X,

$$G_m = \frac{G_{m0}}{1 + s\dfrac{C}{XG_{m0}}} \tag{5.35}$$

By substituting (5.35) into (5.32), the modified transfer function can written as a function of X

$$H_p(s) = K\frac{s\dfrac{G_{m0}/C}{1+s\,C/XG_{m0}}}{s^2+s\,G_{mq}/C+\left(\dfrac{G_{m0}/C}{1+s\,C/XG_{m0}}\right)^2} \tag{5.36}$$

Similarly, a G_m having a parasitic zero at a frequency ω_z can be modeled as,

$$G_m = G_{m0}\left(1+\frac{s}{\omega_z}\right) \tag{5.37}$$

and X can be defined as

$$X = \frac{\omega_z}{G_{m0}/C} \tag{5.38}$$

In this case, the modified transfer function can be found as

$$H_z(s) = K\frac{s^2/X+s\,G_{m0}/C}{s^2+s\left(\dfrac{G_{mq}/C+2G_{m0}/XC}{1+1/X^2}\right)+\dfrac{(G_{m0}/C)^2}{1+1/X^2}} \tag{5.39}$$

In the presence of parasitics (one pole and one zero cases), the modified transfer functions are found as in (5.36) and (5.39). Note that Q and f_0 is only defined for 2^{nd}-order systems and with the inclusion of a parasitic pole, the modified transfer function is not biquadratic. Instead, effective center frequency can be defined as the frequency where the gain is maximized,

$$\hat{\omega}_0 = \arg\max_{\omega}\left(|H_{p,z}(j\omega)|\right) \tag{5.40}$$

Parasitic poles and zeros will cause slight phase shift at the reference frequencies. To simplify the analysis, the phase shifts at the references can be assumed to be equal to the phase shift at the effective center frequency of the filter. Hence, equivalent phase offset can be expressed as

$$\text{Equivalent phase offset} = \angle H_{p,z}(j\omega)\Big|_{\omega=\hat{\omega}_0} \tag{5.41}$$

Equivalent phase offsets versus relative location of the parasitic pole or zero (X) is plotted[2] in Fig. 60. Both pole and zero cause almost the same amount of phase shifts but in the opposite direction. In case G_m has both parasitic pole and zero, their effect may cancel each other.

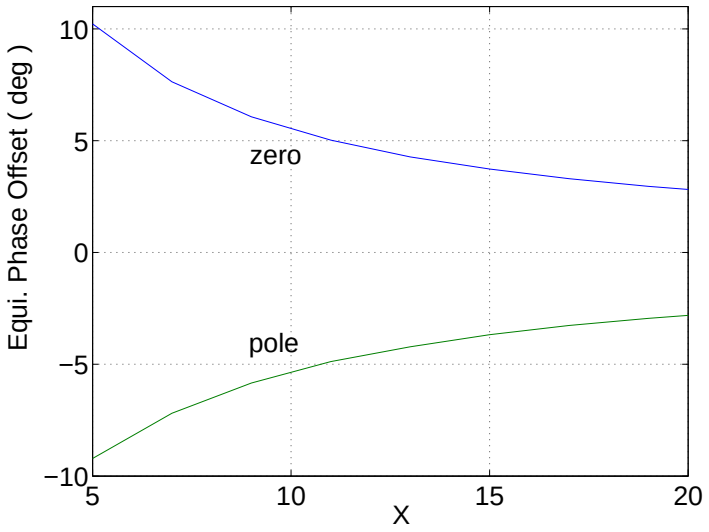

Fig. 60. Equivalent phase offset due to parasitic pole and zero in the transconductor: equivalent phase offset vs. relative location of parasitic.

Once the equivalent phase offset is found, as described in Section C-2, the tuned filter response can be superimposed with the ideal response in order to see the effect. Alternatively, a specification (X) for the transconductor element can be extracted by finding the maximum tolerable phase offset error.

D. Design Procedure and a Design Example

A general design flow diagram for HDTPC is given in Fig. 61, which is implemented in Matlab (refer to Appendix E). Bandpass filter specification may include the

[2]Refer to Appendix D for the Matlab code

passband (f_{p1}, f_{p2}) and the stopband frequencies (f_{s1}, f_{s2}), maximum attenuation in the passband (α_{max}), minimum required attenuation in the stopband (α_{min}), and filter type. The first step is to approximate the magnitude response of the filter satisfying the given specifications. A prototype Butterworth bandpass filter is chosen as specified in Table VII.

TABLE VII

Specifications of the prototype filter

Filter Type	Butterworth	
Passband frequencies (MHz)	f_{p1},f_{p2}	18.81, 19.67
Stopband frequencies (MHz)	f_{s1},f_{s2}	18.03, 20.52
Max. attenuation in passband (dB)	α_{max}	3
Min. attenuation in stopband (dB)	α_{min}	20

The block "Filter approximation" in Fig. 61 produces the minimum number of required biquads, desired center frequencies (f_{d0i}) and quality factors (Q_{di}) as shown in Table VIII. The order of the biquad sections can be arranged arbitrarily. To optimize the dynamic range of the overall filter, it is desirable to order biquads from minimum to maximum Q value in an increasing fashion. N is found as 3 so the prototype is a 6^{th}-order bandpass filter with three biquads.

Next step is to "Determine references" as described in Section B. M is chosen as 4 so that all the reference phases will be multiple of 45°. The phase response seen at the first output is shown in Fig. 62. As in DTPC case, the optimum references are $-45°$ and $-135°$. The phases at the second and the third outputs are shown in Figs. 63 and 64. All the reference phases and corresponding frequencies are summarized in

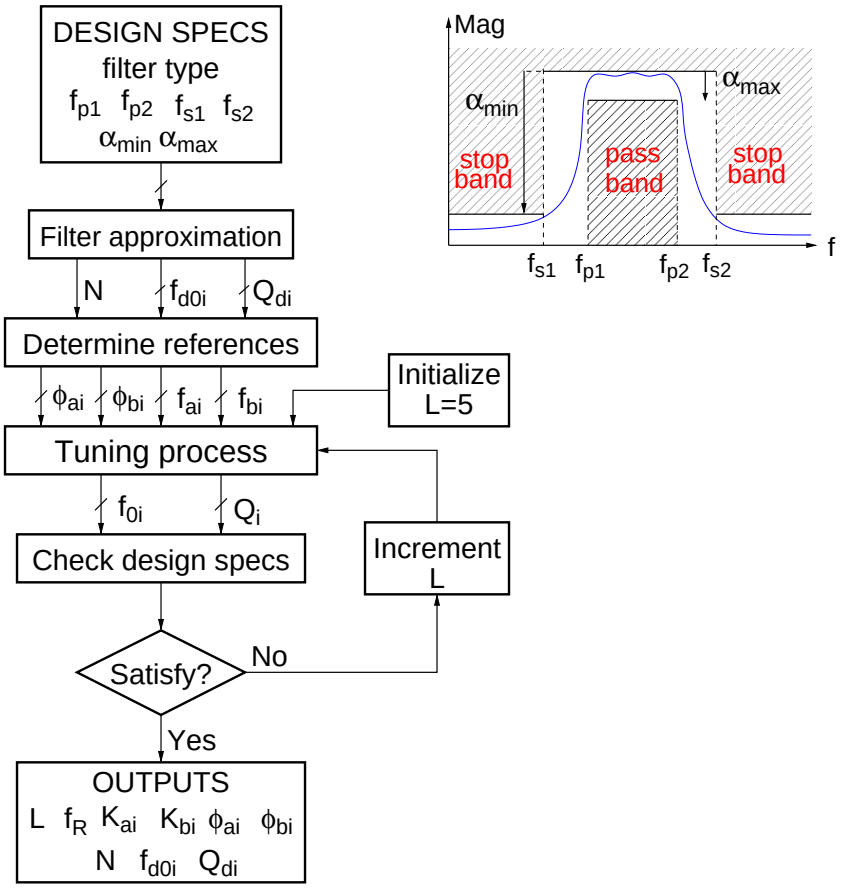

Fig. 61. Design flow diagram.

TABLE VIII

DESIRED f_0 AND Q VALUES FOR THE EXAMPLE

Biquad #	1	2	3
f_{d0i} (MHz)	18.74	19.23	19.75
Q_{di}	33.23	16.61	33.23

Table IX.

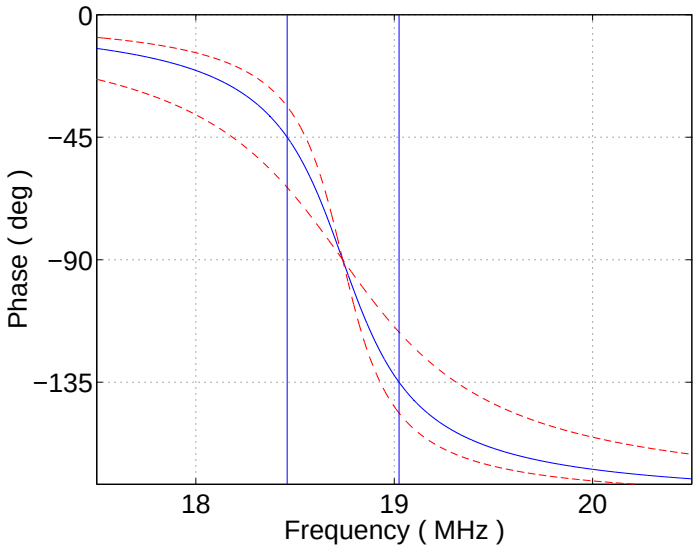

Fig. 62. Phase at the first output, $\phi_1(f)$.

Ignoring the other effects, if all the references could be generated exactly, the tuned filter response would be same as the desired response. The function of the "Tuning process" block in Fig. 61 is to determine required resolution of the frequency synthesizer (L). The process starts with an initial small value of L such as 5. f_{ai} and

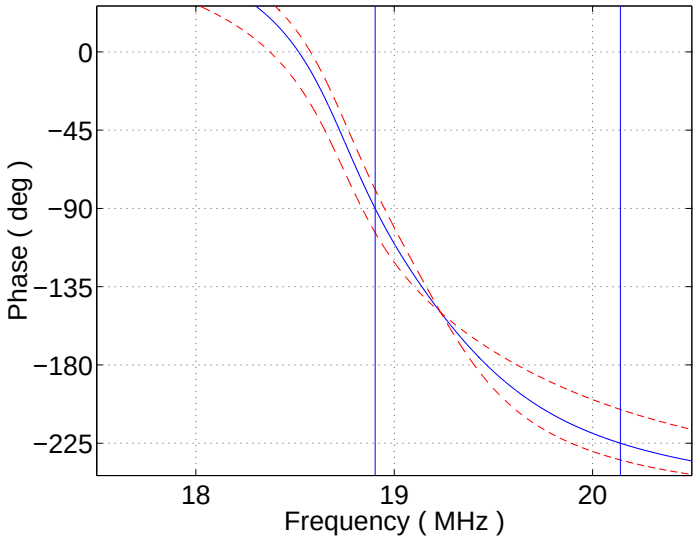

Fig. 63. Phase at the second output, $\phi_2(f)$.

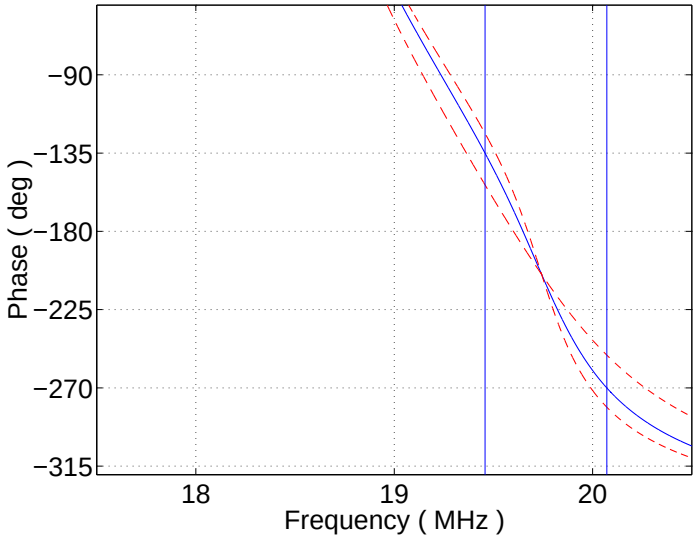

Fig. 64. Phase at the third output, $\phi_3(f)$.

TABLE IX

REFERENCE FREQUENCIES AND PHASES FOR THE PROTOTYPE DESIGN

i	1	2	3
f_{ai} (MHz)	18.488	18.881	19.432
f_{bi} (MHz)	19.039	20.140	20.061
ϕ_{ai} (deg)	-45	-90	-135
ϕ_{bi} (deg)	-135	-225	-270

f_{bi} are approximated with the given resolution as (see Section C-1 for details)

$$f_R = \frac{\max(f_{ai}, f_{bi})}{2^L} \tag{5.42}$$

$$\hat{f}_{ai} = f_R \cdot K_{ai} \tag{5.43}$$

$$\hat{f}_{bi} = f_R \cdot K_{bi} \tag{5.44}$$

Substituting (5.43) and (5.44) into (5.7) and (5.8), tuned values of f_{0ti} and Q_{ti} are calculated.

The filter response is reconstructed with tuned values, and checked to see whether all the initial filter specification given are satisfied. If not, L is incremented by 1. This loop continues till the design specs are satisfied.

Besides N, f_{d0i}, and Q_{di}, the overall output of the design procedure is L, f_R, K_{ai}, K_{ai}, ϕ_{ai} and ϕ_{bi} for $1 \leq i \leq N$ as given in Table X for the prototype design. The design procedure, although seem to be burdensome at first, can be easily implemented with any programming language. Refer to Appendix E for the Matlab code.

TABLE X

REFERENCE FREQUENCIES, PHASES, AND K VALUES FOR THE PROTOTYPE DESIGN

Resolution L	8		
Number of biquad N	3		
i	1	2	3
K_{ai}	235	240	247
K_{bi}	242	256	255
ϕ_{ai} (deg)	-45	-90	-135
ϕ_{bi} (deg)	-135	-225	-270

E. Circuit Implementation

Circuit implementation of the overall high-order tuning system is shown in Fig. 65. The filter consists of N biquadratic sections. Each biquad has control voltages V_f and V_q to adjust the center frequency and Q. All the reference phases and frequencies are generated by a frequency synthesizer (FS). V_0 is applied to the filter. Delayed versions of V_0, $V_{\phi_{ji}}$ j = a, b and $i = 1, 2, \ldots, N$, are also available from FS. Each lowpass output, after converting to rail-to-rail clock signal by using a comparator, is processed by the tuning circuit. The control voltages are generated by the tuning circuit and fed back to filter. A clock generator circuit is used to achieve the synchronization of FS and the tuning circuit.

Frequency synthesizer, which is shown in Fig. 66, generates reference signals at frequencies of $f_R K_{ji}$ for j = a, b and $i = 1, 2, \ldots, N$. The frequency applied is controlled by clock signals (see Fig. 67), which can be obtained from the clock generator circuit shown in Fig. 68. When CLK_{b1} is high, for instance, the frequency

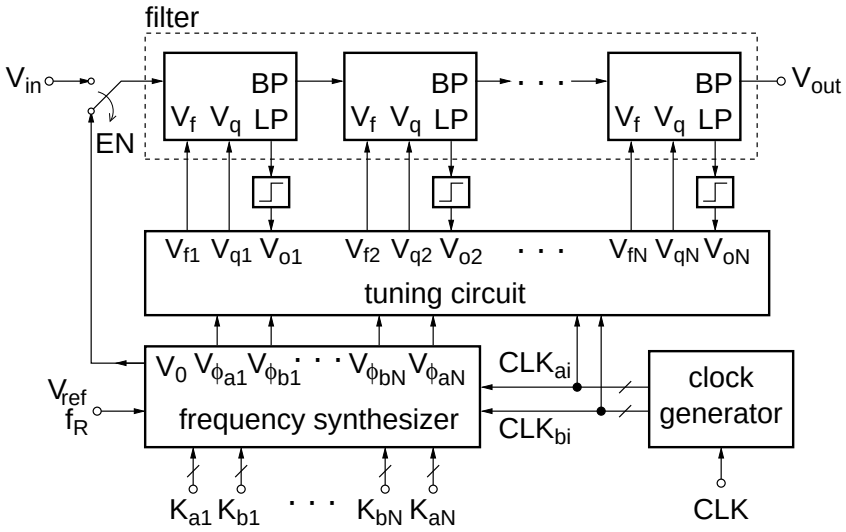

Fig. 65. Complete high-order tuning system.

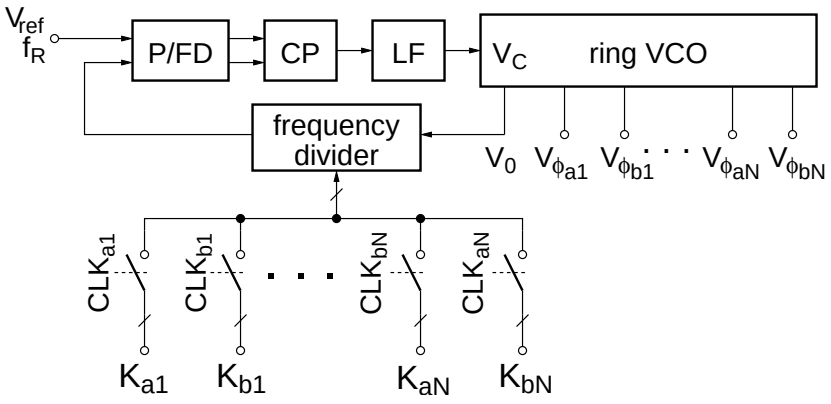

Fig. 66. Programmable frequency synthesizer.

$f_R \cdot K_{b1}$ is generated.

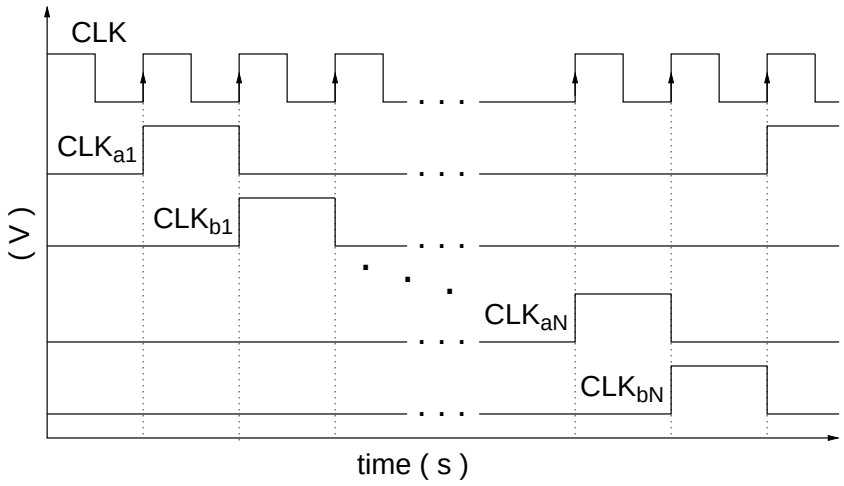

Fig. 67. Synchronizing clock signals.

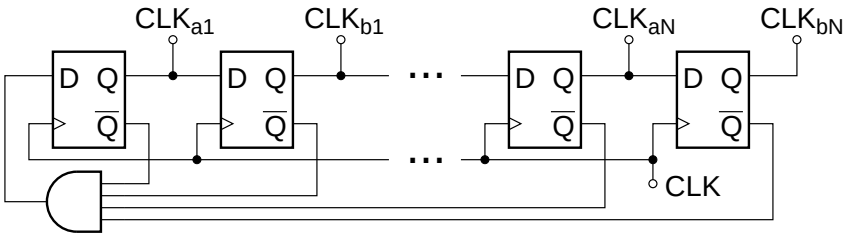

Fig. 68. Clock generator.

A ring voltage-controlled oscillator (VCO) is employed to generate the reference phases as shown in Fig. 69. The number of OTAs depends on the choice of M (see Section B for details). $\Delta\phi$ is the minimum possible phase difference between two references and defined as

$$\Delta\phi = \frac{180^\circ}{M} \tag{5.45}$$

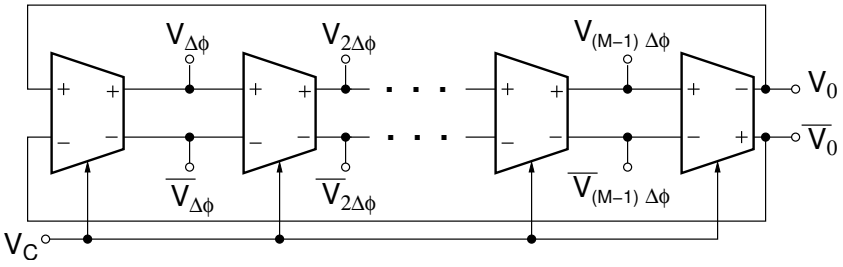

Fig. 69. Ring oscillator (assuming M is even).

The tuning circuit for the i^{th} biquad section is shown in Fig. 70. The first DFFs perform binary phase comparisons. The second set of DFFs stores A_i and B_i until CLK_{bN} makes high-to-low transition. All the up-down signals, which are generated by AND-gates, will have correct values right before CLK_{bN} makes high-to-low transition. Then, frequency and Q counters are updated, DACs converts digital output of counters to analog signals to generate V_{fi} and V_{qi}.

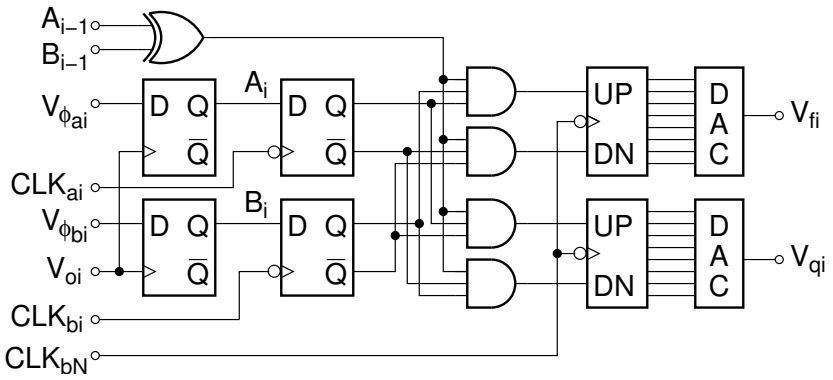

Fig. 70. Tuning circuit for the i^{th} biquad.

F. Simulation Results

The prototype 6^{th}-order Butterworth bandpass filter specified in Table VIII is simulated in Cadence with parameters given in Table X. In the simulation, an ideal

FS and filter are used to reduce the simulation overhead. Transient responses of the control voltages, V_{fi} and V_{qi}, are shown in Figs. 71 and 72, respectively. Tuning circuit first starts increasing V_{f1}, while all the other controls are kept the same. Once the center frequency of the first biquad is tuned to the reference window (around $t = 200\ \mu$s), Q tuning is enabled, i.e, V_{q1} decreases. Since f_{01} is tuned to the reference window ($f_{a1} \leq f_{01} \leq f_{b1}$), frequency tuning for the second biquad is enabled, i.e, V_{f2} decreases. Then, Q tuning for second biquad and frequency tuning for the third biquad are enabled. Eventually, all the control voltages are tuned correctly. With the converged control voltages, the tuned magnitude response is plotted together with the ideal response in Fig. 73. As it can be seen, the tuned and the desired responses closely match although largest Q tuning error is 2.5% due to finite resolution of FS ($L = 8$).

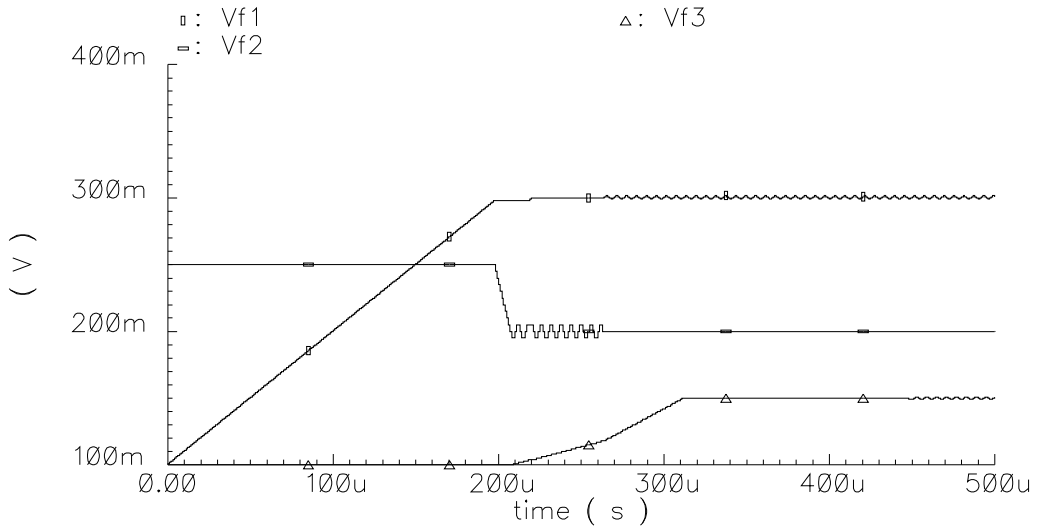

Fig. 71. Transient response of frequency control voltages.

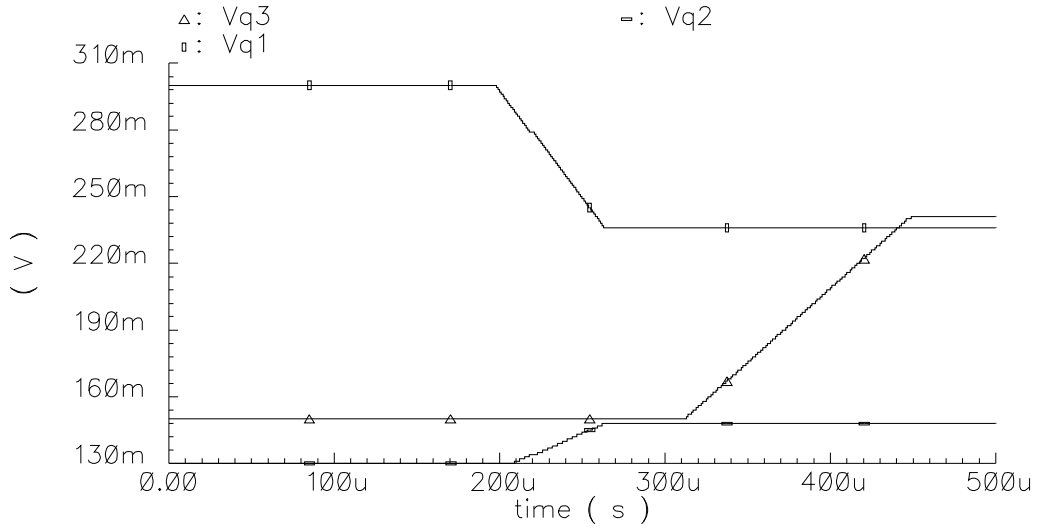

Fig. 72. Transient response of Q control voltages.

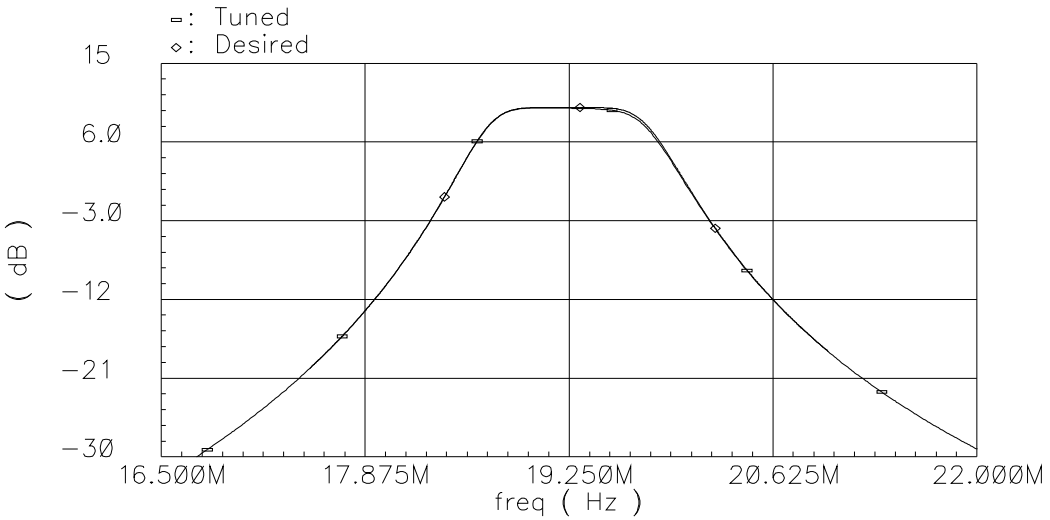

Fig. 73. Desired and tuned filter magnitude responses (simulation result for $L = 8$).

G. Prototype Design

This section summarizes the prototype tuning system for a 6^{th}-order bandpass filter designed in transistor-level and fabricated in conventional 0.5 μm CMOS technology.

1. Filter

The biquad sections of the filter are realized by the G_m-C topology shown in Fig. 74 with $C = 2.5$ pF. Figure 75 shows the transistor-level operational transconductor amplifier (OTA), where source degeneration improves the linearity [45]. Resistors are realized by poly 1 layer. Overall transconductance of this architecture is given by

$$G_m = \frac{g_m}{1 + g_m R} \tag{5.46}$$

where g_m is the transconductance of the input transistors. It can be tuned by changing the tail current through V_{cntrl}. R values are chosen as 800 Ω, 10 kΩ, and 10 kΩ for G_m and G_{mi}, and G_{mq}, respectively.

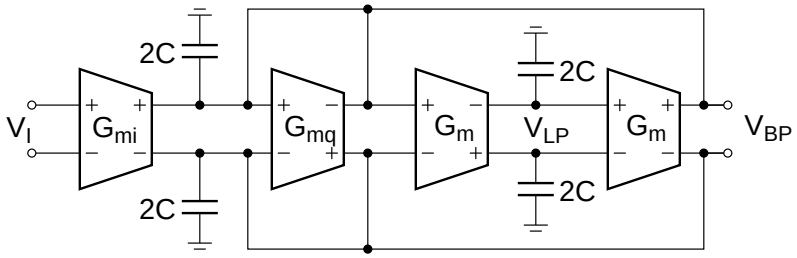

Fig. 74. G_m-C type biquad section of the filter.

Note that OTA circuit has high output impedance so DC level at the output is not well defined. A common-mode feedback (CMFB) circuit is required to adjust the DC level at the output. The circuit shown in Fig. 76 is employed as CMFB circuit [35]. The output common-mode level can be sensed from V_{cm} of the next

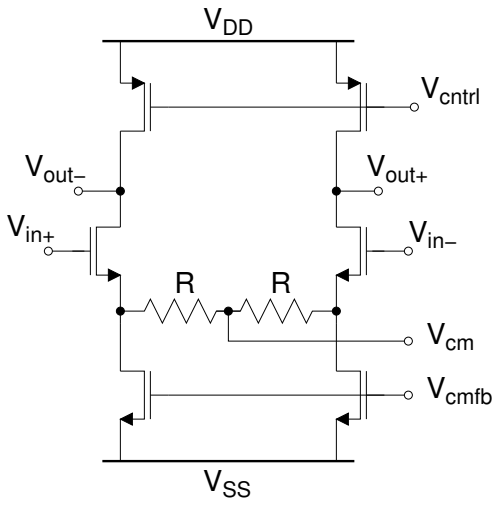

Fig. 75. Source-degenerated differential OTA circuit realizing G_m.

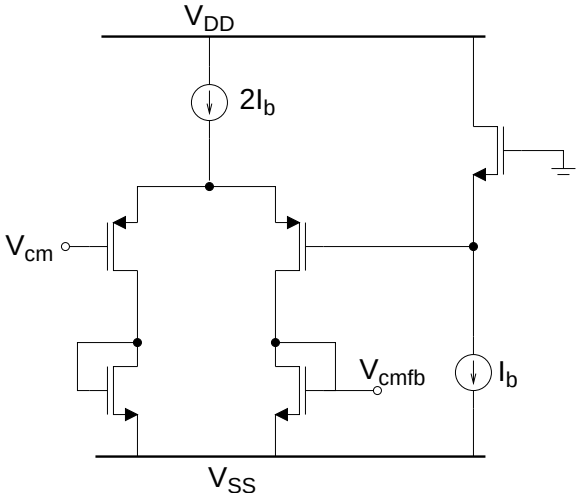

Fig. 76. Common-mode feedback circuit.

OTA circuit [15]. Then, V_{cm} is compared with reference voltage level, which is ground. Hence, V_{cmfb} voltage is generated and applied to the tail transistors.

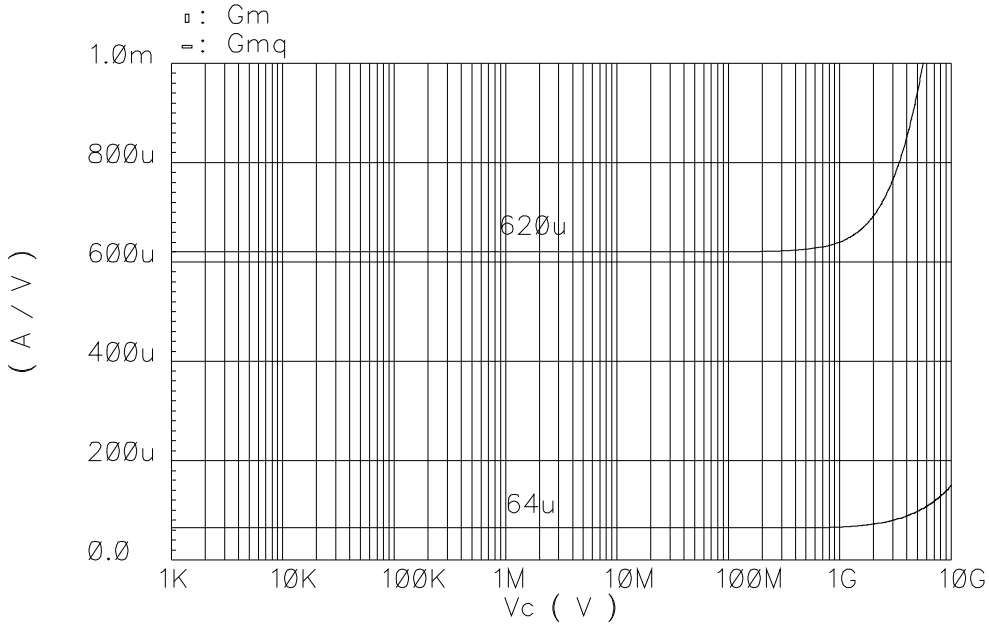

Fig. 77. AC magnitude responses of G_m and G_{mq}.

The AC magnitude responses of G_m and G_{mq} are shown in Fig. 77. As it can be seen, nominal values are 620 μA/V and 64 μA/V. OTA has a parasitic zero located at 4.286 GHz. That implies an X of about 200 for a desired center frequency of 20 MHz. Figure 78 shows tuning range of G_m and G_{mq}. Approximately $\pm$100% tuning range can be achieved.

Although the tuning circuit corrects the process variations, the tuning circuit may need to be optimized regarding control voltage range, sensitivity of the filter parameters to the control voltages, and required resolution in order to achieve better accuracy. Monte Carlo and corner parameter simulations can be performed to see the effect of the process variations on the design parameters. In Monte Carlo simulations,

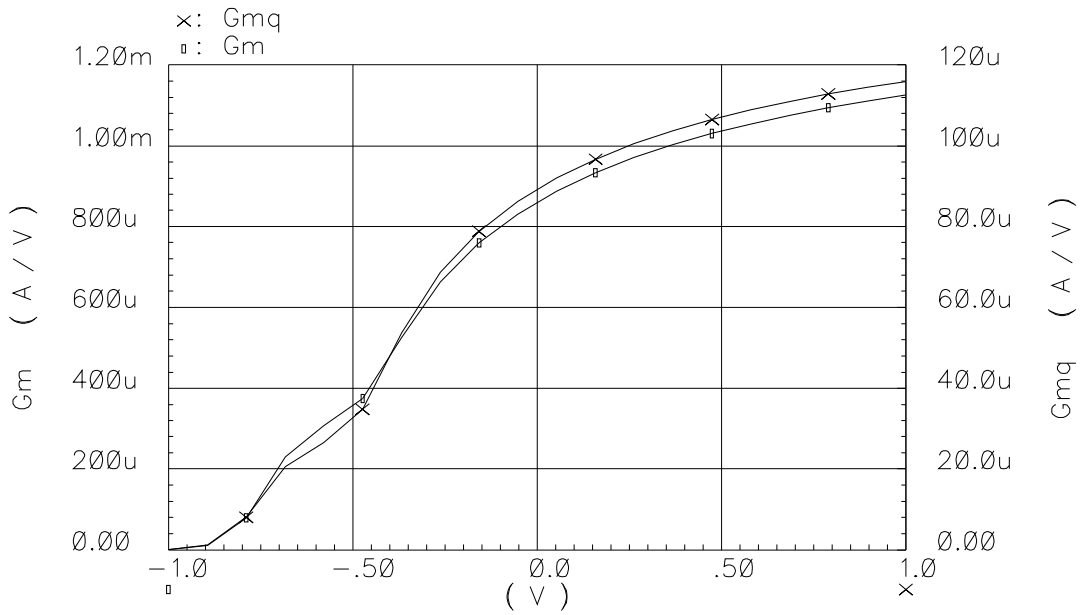

Fig. 78. Tuning range of transconductors.

some of the parameters such as threshold voltage and mobility are defined by random variables with certain mean and standard deviation. In corner simulations, on the other hand, model files with fast (F), slow (S), and typical (T) corner parameters, which are provided by MOSIS, are used for transistors. The actual circuit behavior would be most likely within the simulated results. Simulations are performed with different corner parameters to see the variation of nominal transconductor values for the prototype design. Variation of G_m AC magnitude response is shown in Fig. 79. Transconductor and parasitic-zero frequency (f_z) values are summarized in Table XI.

2. Tuning Circuit

At steady state, the filter output signals will have the same phase as the reference phases. DFF is supposed to perform binary phase comparison; however, there is a

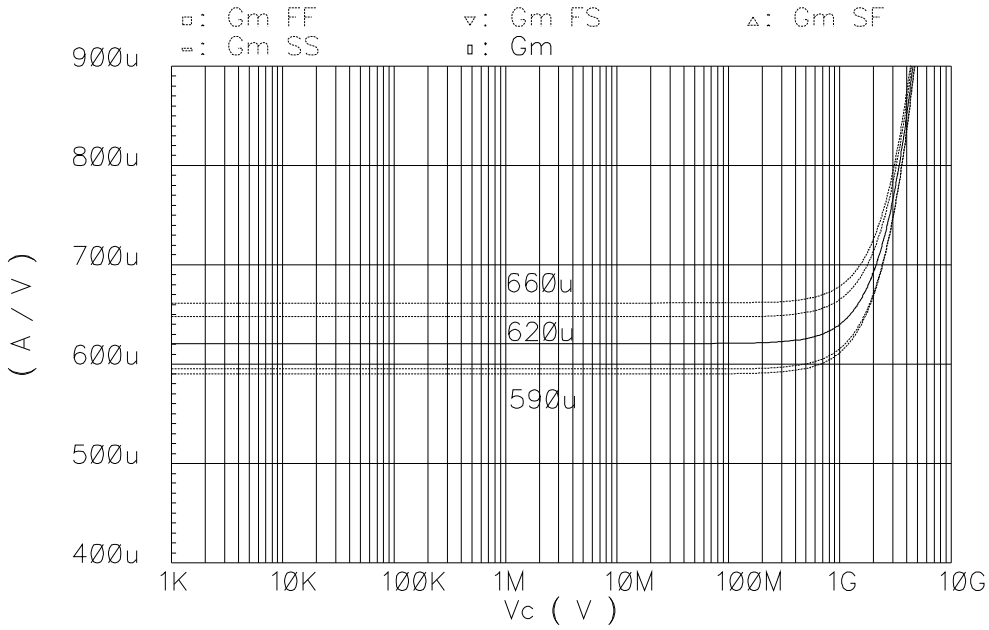

Fig. 79. Corner simulations for AC magnitude response.

dead zone that DFF output is not correct because positive edges of two inputs (D and ϕ) are too close. Therefore, fast DFF with minimal dead zone are needed to minimize phase offset errors. True Single Phase Clock (TSPC) type DFFs shown in Fig. 80 are employed as phase comparator.

The comparator circuit is given in Fig. 81. The comparators are used to convert sinusoidal outputs of filter to rail-to-rail clock signals. It is crucial that the compara-

TABLE XI

CORNER SIMULATIONS RESULTS FOR OTA

Parameter	TT	SS	SF	FS	FF
G_m (μ A/V)	620	595	660	590	650
f_z (GHz)	4.29	4.56	3.98	4.67	4.06

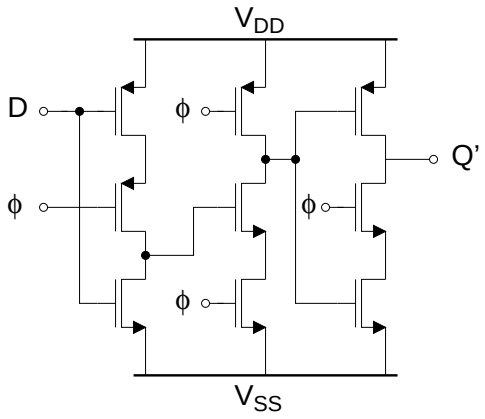

Fig. 80. TSPC-type fast DFF.

tors are fast so that rise and fall times of the output signal are small. M_3 provides positive feedback and boosts the gain. M_5 and M_7 forms the second gain stage. Note that the path through M_4'-M_6'-M_7 provides additional gain. Since the comparators are used in open loop, additional gain stages can also be added. Input stage needs to be carefully designed to minimize the offset; using common-centroid, inter-digitized input transistors methods in the layout.

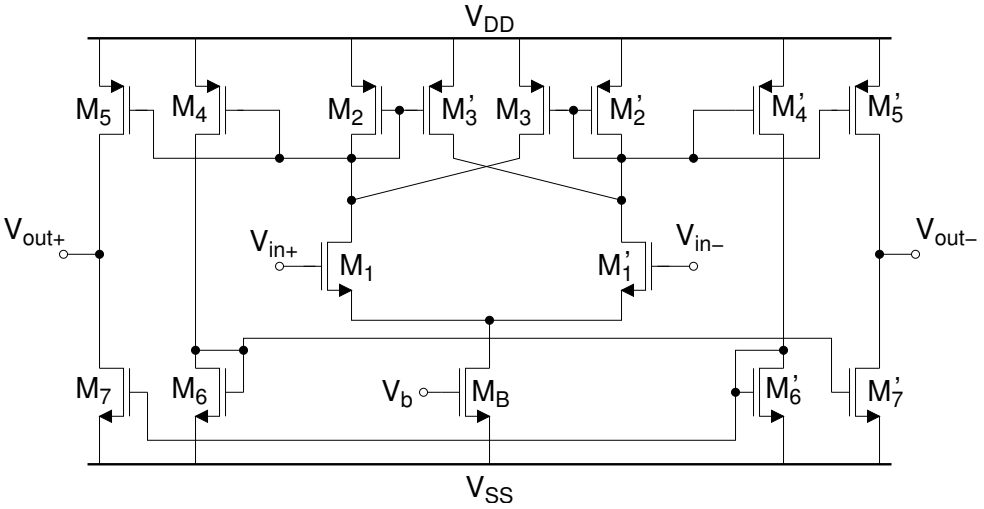

Fig. 81. Comparator circuit.

An external FS is used for the prototype design. Reference signals are generated on chip by utilizing the frequency-divider circuit shown in Fig. 82. This circuit divides the frequency of input signal Ref by 8 to achieve 45°-delayed clock signals. Therefore, the reference signal generated by the external FS has to be 8 times higher frequency than the actual reference frequencies.

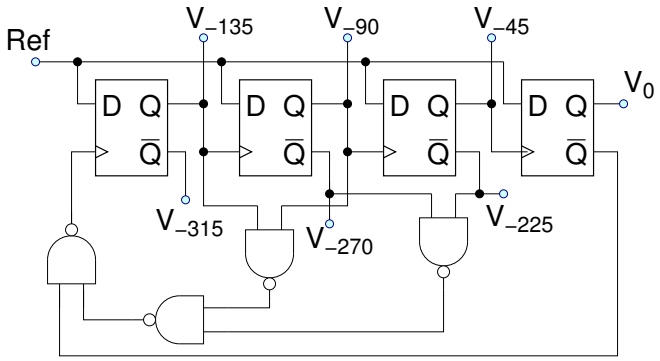

Fig. 82. On-chip reference phase generator circuit.

H. Experimental Results

The prototype filter designed in Sections F and G was fabricated in AMI 0.5 μm CMOS technology with two poly and three metal layers. Layout, DRC (design-rule-checker), extraction, LVS (layout-versus-schematic) tools of Cadence were used to design and verify the circuits before fabrication.

The micrograph of the designed chip is shown in Fig. 83. Agilent 81110A 165/330 MHz Pulse/Pattern Generator was used to generate the reference signal Ref. The tuning loops were completed on a PCB (printed circuit board) with the external counters and DACs. Upon convergence of the tuning loops, the control voltages were recorded. Then, the tuning was disabled, and the same control voltages were applied to the filter. The tuned magnitude response of the filter is captured from

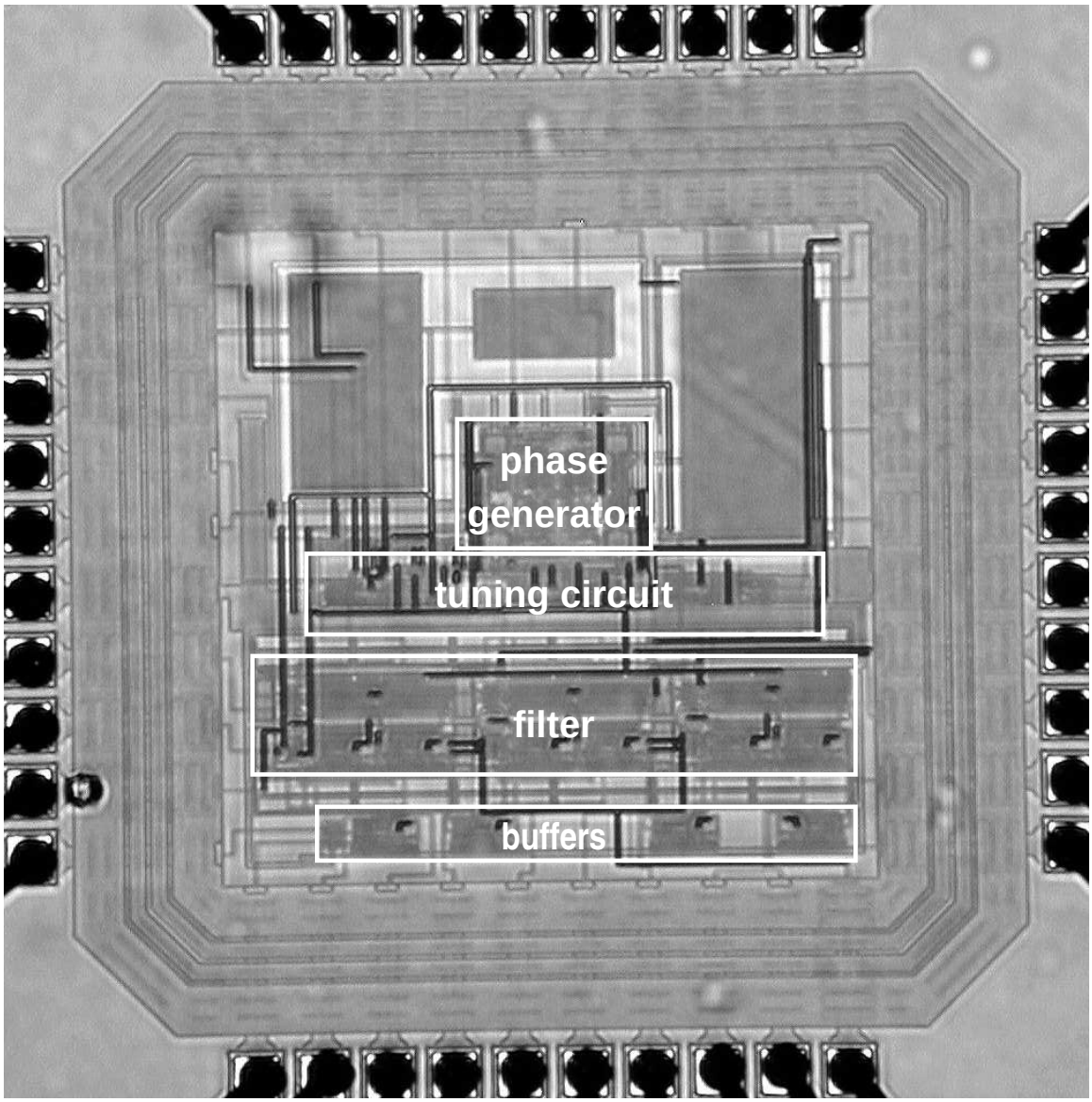

Fig. 83. Micrograph of the chip.

network analyzer as shown in Fig. 84. Synchronization of the FS and the tuning circuit was achieved through a computer using Labview. During the measurements, the speed of the tuning circuit was kept low. In fact, arbitrarily large tuning period (set by CLK in Fig. 67) can be chosen since the phase information is stored on D latches. The lower limit of the tuning period is set by the settling time (t_s) of the FS. It also depends on the resolution of the counter (L_c) and the order of the filter (N). The tuning will converge roughly in $N\ 2^{L_c} t_s$. For $N = 3$, $L_c = 8$, and $t_s = 10\ \mu$s, the convergence time is less than 8 ms.

The center frequency was tuned with 0.7% error while the bandwidth was smaller than expected. Ideal filter response is indicated as dashed lines on the measured response (Fig. 84). The error can be partially attributed to the comparators used at the lowpass outputs, although the comparator circuits used are expected to have negligible delays. One solution can be using faster comparators. Another solution would be to develop a better phase comparator technique to eliminate the use of comparators. Another source of error was the precision of the external FS used. Externally generated signal at around 160 MHz is divided on chip by 8. It was not possible to measure how accurately reference signals are generated in terms of duty cycle, phase noise, rise and fall times. On-chip FS is expected to achieve better tuning accuracy.

To see the adaptation of the tuning circuit, the temperature was increased after all tuning loops converged. The magnitude response of the tuned filter were captured at different temperatures (approximately within the range of 20 °C – 100 °C) and plotted in Fig. 85. For this particular case, temperature increase degraded the ripple in the passband, while it improved the bandwidth.

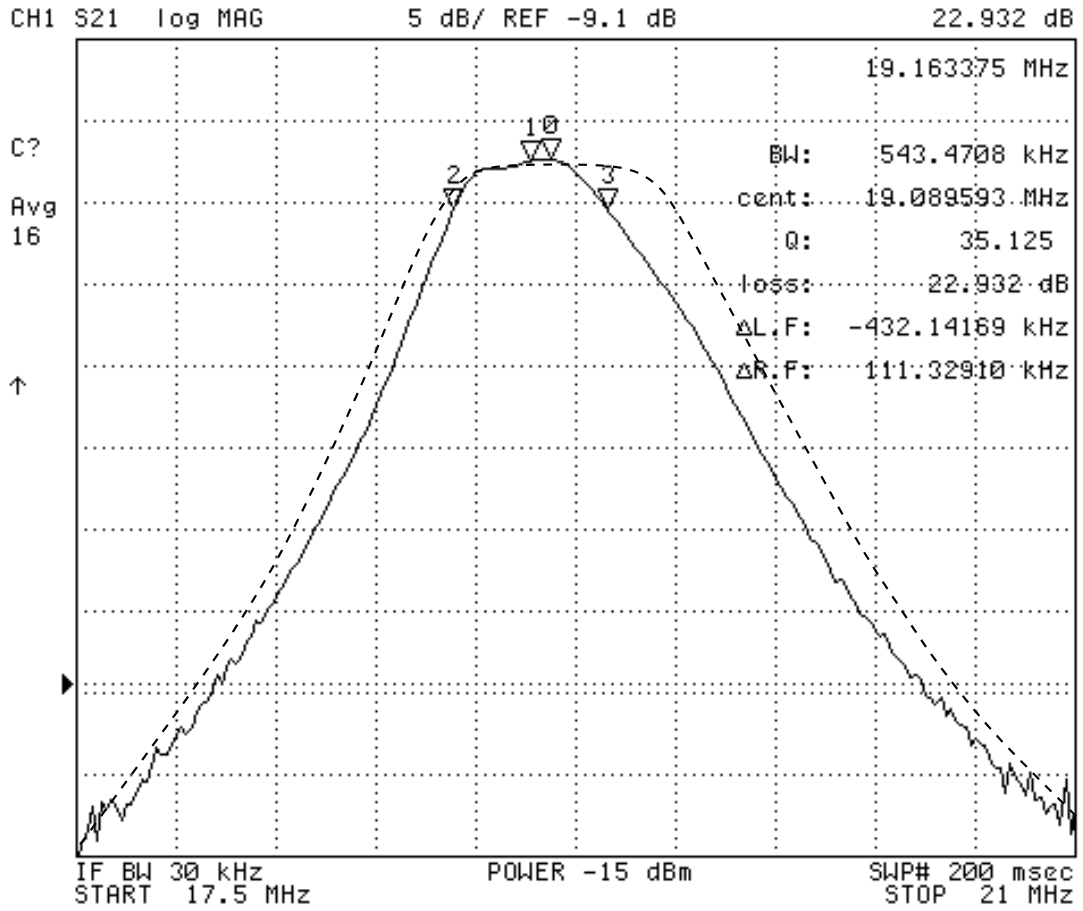

Fig. 84. Measured tuned magnitude response with the target response superimposed.

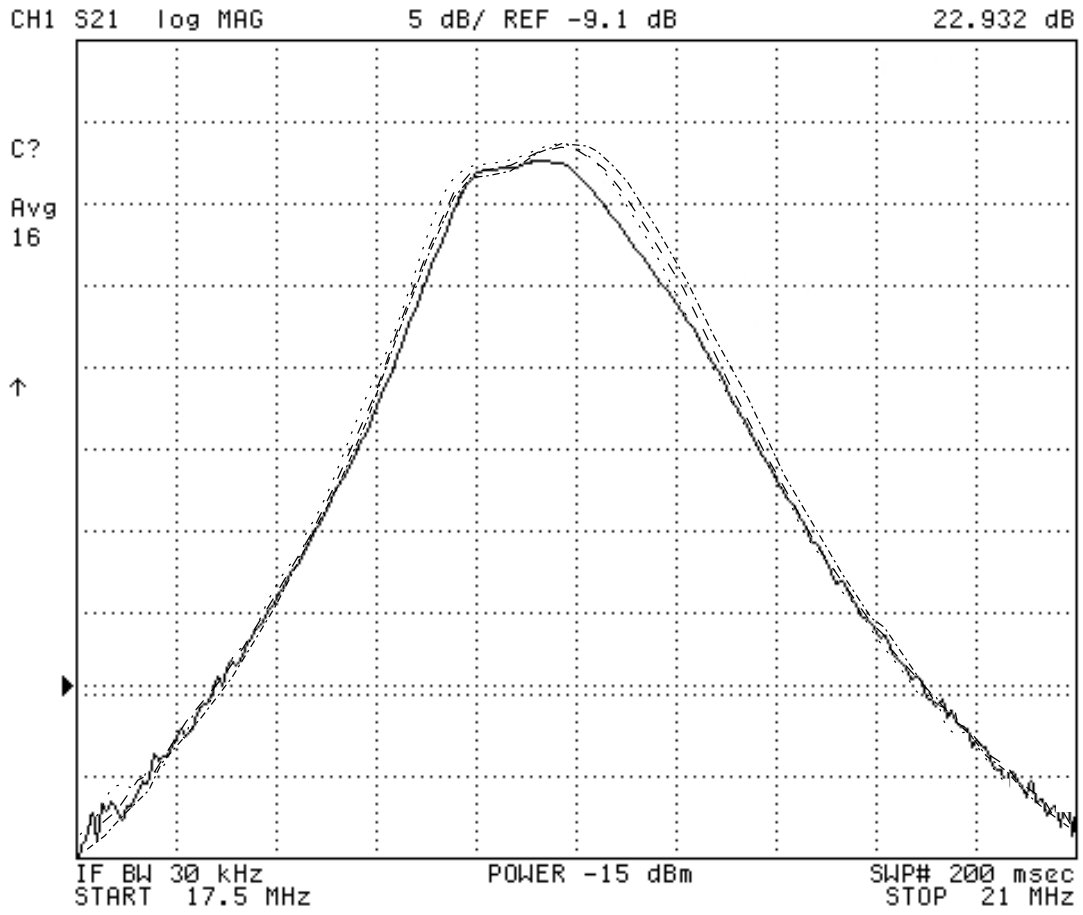

Fig. 85. Adaptation of the tuning circuit to temperature increase.

CHAPTER VI

DIGITAL-TUNING METHOD BASED ON MAGNITUDE COMPARISON

The automatic tuning methods discussed in previous chapters use the phase information to tune the filter parameters f_0 and Q. The magnitude response of the biquadratic filter can also be used to tune the filter.

One drawback of the phase-based tuning techniques is the difficulty of precise phase comparison as frequency increases. Although DTPC does not require detecting the exact phase shift caused by the filter, the sinusoidal filter output has to be compared with the reference signals, which are most likely clock signals. The proposed circuit architecture for DTPC in Chapter IV employs comparators to convert sinusoidal filter output to rail-to-rail clock signal. At high frequencies, implementing those comparators may lead to high power consumption. In this chapter, digital-tuning method based on magnitude comparison is proposed [46].

A. Principle

A second-order continuous-time bandpass filter can be represented by the following transfer function,

$$H(s) = G\frac{s\,\omega_0/Q}{s^2 + s\,\omega_0/Q + {\omega_0}^2} \tag{6.1}$$

where G is the pass band gain, Q and ω_0 are the quality factor and the center frequency, respectively. Magnitude response is shown in Fig. 86. Peak gain or passband gain G occurs at f_0. Lower and upper half gain frequencies, f_L and f_U, are defined as the frequencies where the peak gain drops by half (6 dB).

The tuning circuit applies reference signals at three frequencies f_1, $\sqrt{f_1 f_2}$, and f_2, and forces the gains at f_1 and f_2 to be the half of the gain at the geometric mean

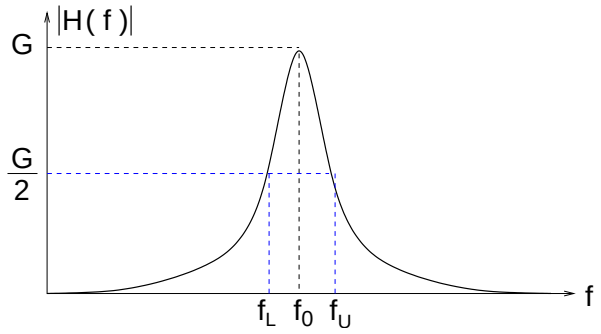

Fig. 86. Magnitude response of a second-order bandpass filter.

frequency ($f_m = \sqrt{f_1 f_2}$) [35]. At steady state, the following equations hold

$$|H(j\,2\pi f_1)| = \frac{1}{2}|H(j\,2\pi f_m)| \tag{6.2}$$

$$|H(j\,2\pi f_2)| = \frac{1}{2}|H(j\,2\pi f_m)| \tag{6.3}$$

The tuned values can be found by solving (6.2) and (6.3) as

$$f_0 = \sqrt{f_1 f_2} \tag{6.4}$$

$$Q \approx \frac{\sqrt{3}}{2}\frac{f_m}{f_2 - f_1} \tag{6.5}$$

Now a tuning algorithm has to be developed to achieve the conditions given in (6.2) and (6.3). Define the gains at frequencies f_1, f_2, and f_m as G_1, G_2, and G_0, respectively. Normalized gains with respect to G_0 can be written as

$$g_1 = \frac{G_1}{G_0} \tag{6.6}$$

$$g_2 = \frac{G_2}{G_0} \tag{6.7}$$

There are two critical values for g_i; 0.5 and 1. g_i can take values from three possible ranges [0 0.5], [0.5 1], and [1 ∞]. Based on the definition of g_1 and g_2, an

arbitrary filter can be in 9 different cases, one of which is not possible due to typical shape of bandpass filter. In order to represent each range, define two binary numbers A_i and B_i $(i = 1, 2)$ as

$$A_i = \begin{cases} 0, & g_i < 0.5 \\ 1, & g_i > 0.5 \end{cases} \tag{6.8}$$

$$B_i = \begin{cases} 0, & g_i < 1 \\ 1, & g_i > 1 \end{cases} \tag{6.9}$$

Note that $A_1B_1A_2B_2$ can have 16 values; however, the rest of the cases are redundant. Relevant eight cases are given in Table XII. Each case is illustrated in Fig. 87. The horizontal lines 0.5 and 1 are drawn to distinguish the regions of g_1 and g_2, whereas the vertical lines are the reference frequencies. The dashed curve shows the desired normalized response. In Fig. 87(a), the magnitudes at the reference frequencies are larger than 0.5, so the best tuning action should be to increase Q; whereas in Fig. 87(b) is just the opposite. The rest of the possible states are shown in Fig. 87(c) and (d) where f_0 needs to be increased or decreased, respectively.

One immediate observation from Table XII is that only one tuning action either to change f_0 or Q is assigned for each case in order to avoid the simultaneous operation of the two loops. Moreover, the Q tuning can only be enabled when the current center frequency of the filter is between the reference frequencies f_1 and f_2, i.e., when $f_1 < f_0 < f_2$ holds. Note also that exact values of the gains do not have to be detected but their relative values with respect to the gain at the middle frequency are compared. Note that detecting exact magnitude/gain requires accurate envelope detector, magnitude-comparison approach relaxes this requirement.

TABLE XII

TUNING PROCESS

Case	A_1	B_1	A_2	B_2	Tuning action
1	0	0	0	0	$Q \Downarrow$
2	0	0	1	0	$f_0 \Downarrow$
3	0	0	1	1	$f_0 \Downarrow$
4	1	0	0	0	$f_0 \Uparrow$
5	1	0	1	0	$Q \Uparrow$
6	1	0	1	1	$f_0 \Downarrow$
7	1	1	0	0	$f_0 \Uparrow$
8	1	1	1	0	$f_0 \Uparrow$

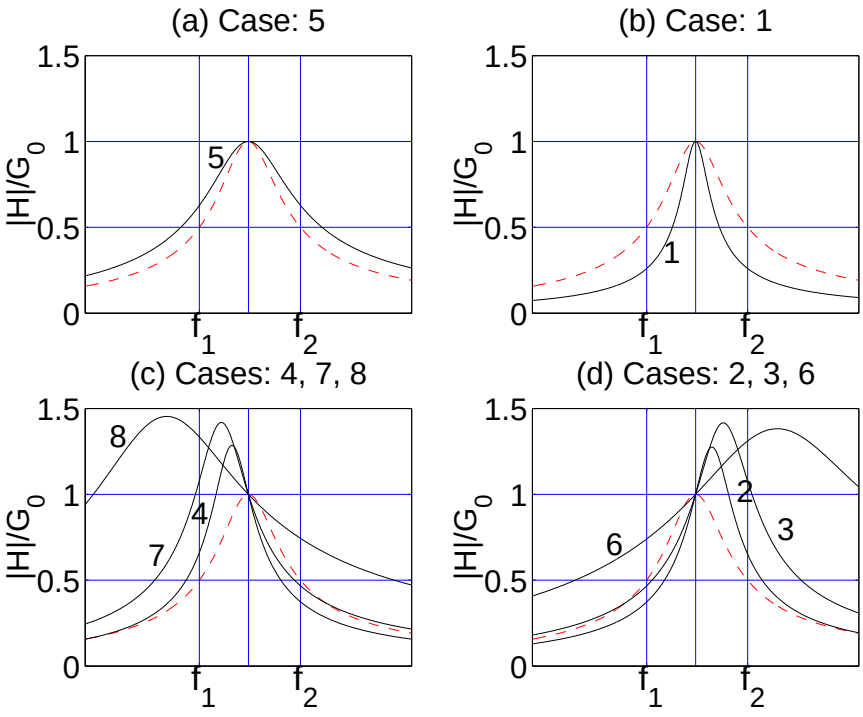

Fig. 87. Possible states of the filter.

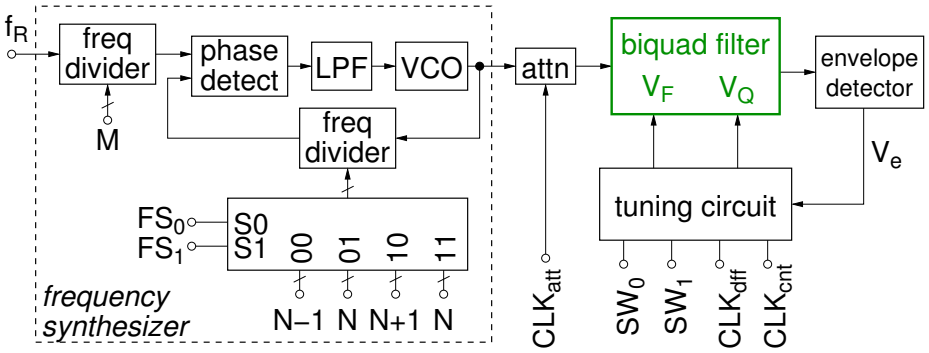

Fig. 88. DTMC tuning system.

B. Circuit Implementation

System-level implementation of DTMC is shown in Fig. 88. Reference frequencies are generated by a programmable frequency synthesizer. Applied frequency is controlled by FS_0 and FS_1 signals. Four magnitude comparisons have to be performed to determine A_1, A_2, B_1, and B_2 values in four different tuning slots shown in Table XIII.

TABLE XIII

APPLIED FREQUENCY, ATTENUATOR GAIN, AND CONTROL VOLTAGES IN EACH TUNING SLOT

	FS_0	FS_1	Frequency	Attenuator gain
T_1	1	1	$\frac{N}{M} f_R$	$\frac{G_{att}}{2}$
T_2	0	1	$\frac{(N+1)}{M} f_R$	G_{att}
T_3	1	0	$\frac{N}{M} f_R$	G_{att}
T_4	0	0	$\frac{N-1}{M} f_R$	G_{att}

To eliminate the non-linearities of the envelope detector, a selective attenuator is employed. When CLK_{att} is low, attenuator has a gain of G_{att}, whose value may

vary with process variation. The gain becomes $G_{att}/2$ when CLK_{att} is high. With the use of the selective attenuator, the filter output magnitude will have the same values for different reference signals at steady state.

The four tuning slots can be described as follows. For simplicity, G_{att} is assumed to be unity. Note that G_{att} and other losses will be eliminated in magnitude comparison.

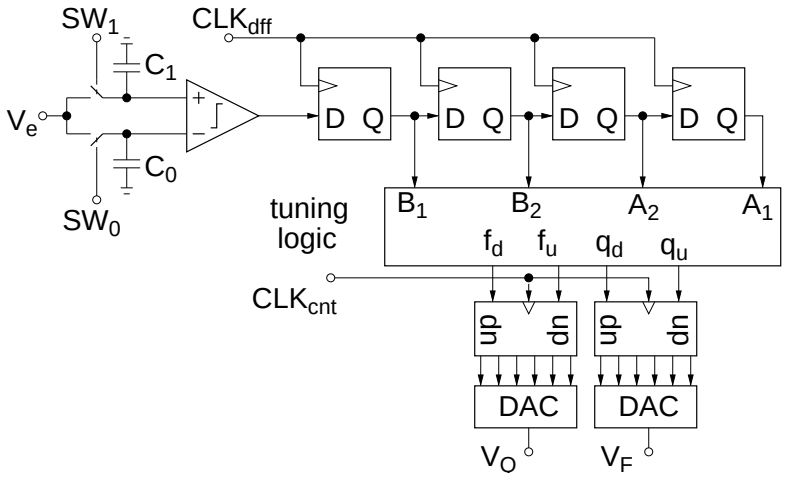

Fig. 89. Tuning circuit.

$\mathbf{T_1}$: The applied frequency is $f_m = f_R N/M$ and the attenuator gain is 0.5. $G_0/2$ is sampled on C_0 (see Fig. 89) by the clock SW_0. C_1 holds G_1 from step $\mathbf{T_4}$ (see below), therefore the comparator output is A_1.

$\mathbf{T_2}$: The applied frequency is $f_2 = f_R(N+1)/M$ and the attenuator gain is 1. G_2 is sampled on C_1 by the clock SW_1. C_1 holds $G_0/2$ from step $\mathbf{T_1}$, therefore the comparator output is A_2.

$\mathbf{T_3}$: The applied frequency is $f_m = f_R N/M$ the the attenuator gain is 1. G_0 is sampled on C_0 by the clock SW_0. C_1 holds G_2 from step $\mathbf{T_2}$, therefore the comparator output is B_2.

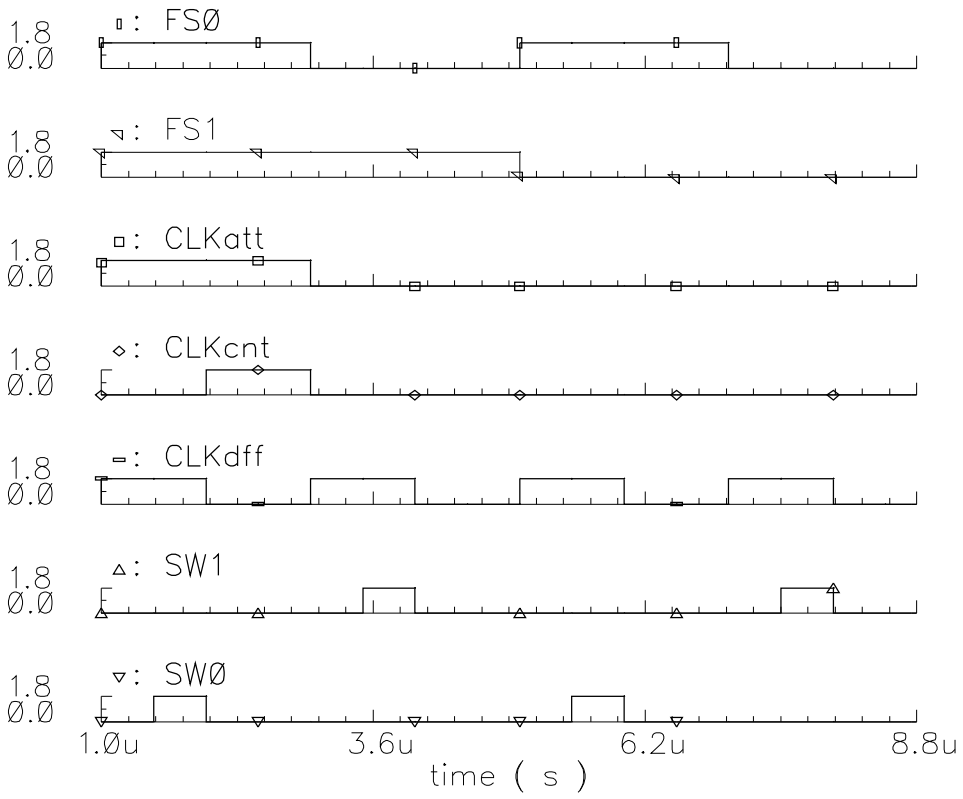

Fig. 90. Clock signals.

$\mathbf{T_4}$: The synthesized frequency is $f_1 = f_R(N-1)/M$ and the attenuator gain is still 1. G_1 is sampled on C_1 by the clock SW_1. C_0 holds G_0 from step $\mathbf{T_3}$, therefore the comparator output is B_1.

The tuning periods and synchronizing clock signals are shown in Fig. 90. These signals are generated by the clock generator circuit in Fig. 91. Each time CLK_{dff} goes high, the output of the comparator is shifted serially into D flip-flops in Fig. 89. A_i and B_i become available for the tuning logic at the end of $\mathbf{T_4}$. Depending on the values of A_i and B_i, the tuning logic generates update signals for the up-down counters for frequency and Q control. The tuning logic functions are derived from

Table XII, and can be given as follows:

$$f_u = B_1 A_2 \bar{B}_2 + A_1 \bar{A}_2 \tag{6.10}$$

$$f_d = A_1 \bar{B}_1 B_2 + \bar{A}_1 A_2 \tag{6.11}$$

$$q_u = A_1 \bar{B}_1 A_2 \bar{B}_2 \tag{6.12}$$

$$q_d = \bar{A}_1 \bar{A}_2 \tag{6.13}$$

Figure 92 shows an implementation with AND and OR gates. Counter outputs are increased if (up,dn) = (1,0), decreased if (up,dn) = (0,1), and remains the same if (up,dn) = (0,0). Note that (1,1) is avoided by the logic circuit. At the rising edge of CLK_{cnt}, the counters are updated. When the tuning circuits converged to the steady state, the filter output at T_1, T_2, and T_4 will have the same amplitude at steady state, so the envelope detector output will have the same voltage value. As a result, the envelope detector does not have to be precise and linear as long as it has a monotonic response to the magnitude of its input.

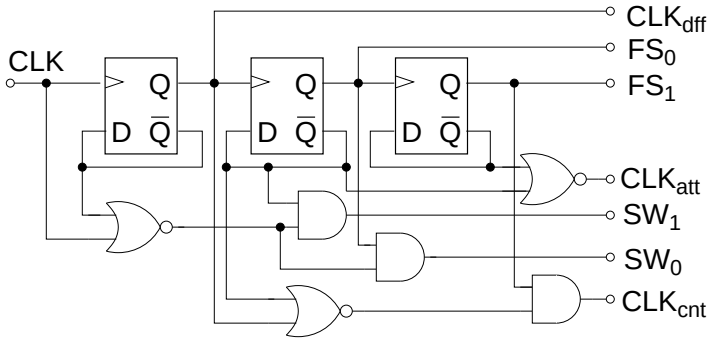

Fig. 91. Clock generator circuit.

The circuit topology shown in Fig. 93 is used as selective attenuator. It employs resistors for voltage division. When CLK_{att} is high, V_{1+} and V_{1-} differential signals are amplified to the outputs. Otherwise, the amplifier on the left is switched on to

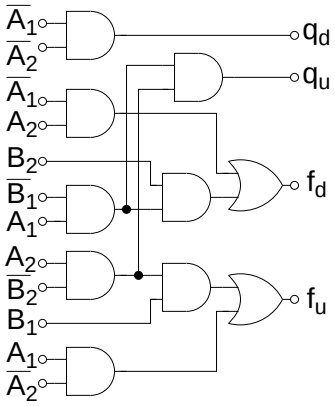

Fig. 92. Tuning logic circuit.

amplify $V_{2+} - V_{2-}$, which has twice as much signal magnitude.

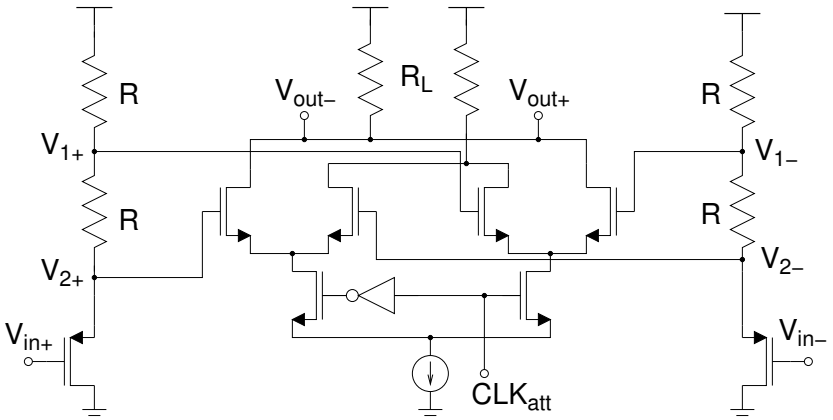

Fig. 93. Selective attenuator.

The tuned f_0 and Q can be found by substituting $f_1 = f_R(N-1)/M$, $f_2 = f_R(N+1)/M$, and $f_m = f_R N/M$ into (6.4) and (6.5),

$$f_{0t} = f_R\sqrt{\left(\frac{N^2-1}{M^2}\right)} \approx f_R\frac{N}{M} \tag{6.14}$$

$$Q_t \approx \frac{\sqrt{3}}{2}N \tag{6.15}$$

(6.14) and (6.15) are set by the input reference frequency, f_R, and division ratios, N and M, of frequency dividers.

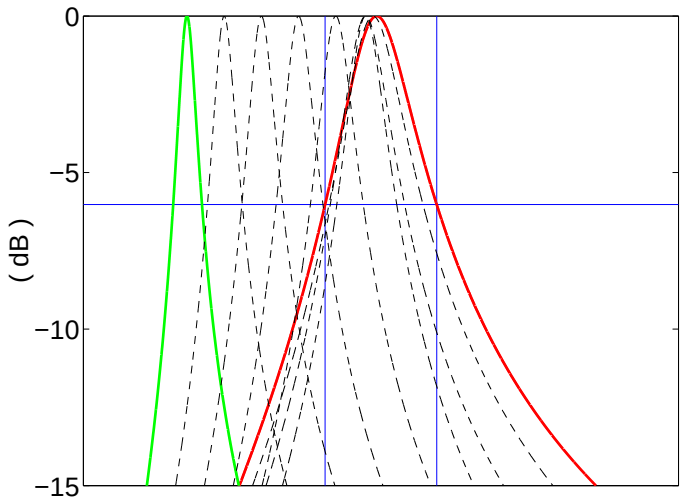

Fig. 94. Normalized filter magnitude response as tuning progresses.

C. Simulation Results

The tuning system is simulated in Matlab (the code is included in Appendix F). The normalized magnitude responses of the filter during tuning is captured as shown in Fig. 94. For this particular case, the filter initially has a higher Q and a lower f_0 (green solid curve). The reference frequencies, f_1 and f_2, are indicated as vertical solid lines, whereas half gain magnitude level is shown as a vertical line. The tuning first increases f_0. The control voltages of the ideal filter is shown in Fig. 95. Once the peak enters to the reference window ($f_0 \geq f_1$), Q tuning is enabled and Q decreases. Eventually, the control voltages converge, the magnitude response becomes red solid curve in Fig. 94. The transistor-level circuit is simulated using Cadence. The control voltages and tuned and desired filter responses are shown in Figs. 96 and 97.

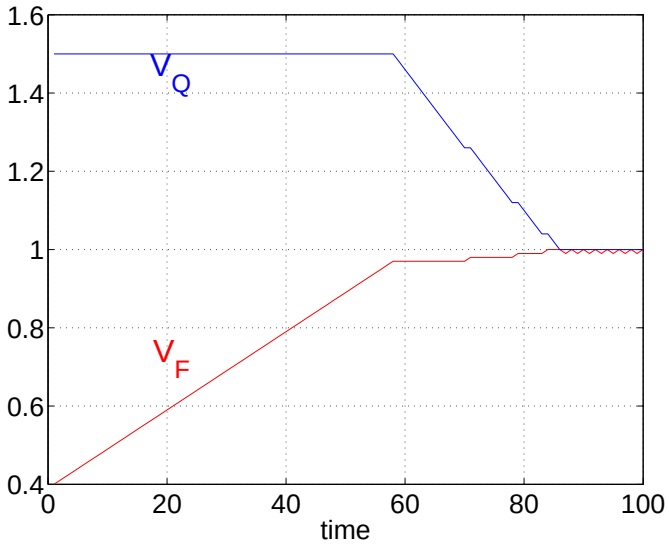

Fig. 95. System-level simulation: transient response of control voltages.

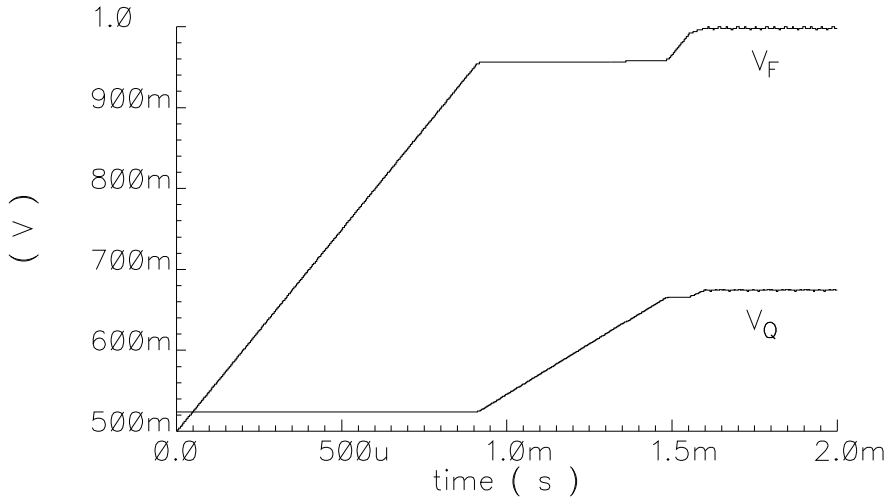

Fig. 96. Transistor-level simulation: transient response of control voltages.

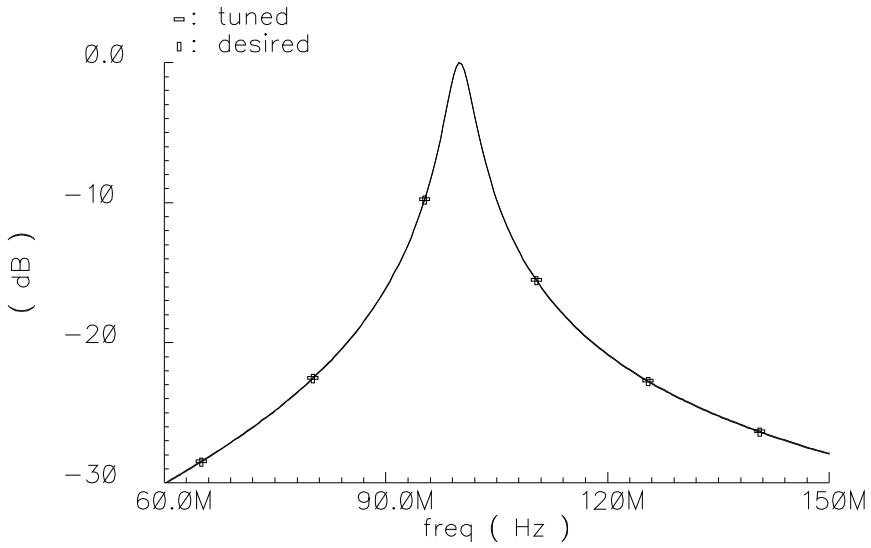

Fig. 97. Tuned and desired magnitude responses.

D. Limitations

The accuracy of the tuning circuit mostly depends on how accurately the filter output signal is stored at different time periods and compared. In the first section, a better comparator architecture with offset cancellation is presented. In the second section, the performance of DTMC is investigated in the presence of parasitics.

1. Offset-Free Magnitude Comparison

The circuit shown in Fig. 98 is used to sample the input signal at different times which is controlled by non-overlapping clock signals SW_0 and SW_1. When the signal SW_0 is high, the input $V_{in} = V_0$ is stored to capacitor C_0. Similarly, when SW_1 is high, the input $V_{in} = V_1$ is stored to capacitor C_1. Once the output of the comparator, V_{out}, has converged to the correct value, the output signal is sampled.

This architecture is vulnerable to the offset of the comparator. Several comparators with offset cancellation techniques are reported in [47], [48]. They are usually

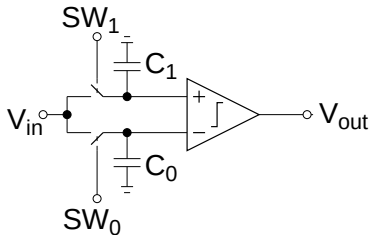

Fig. 98. Comparator without offset cancellation.

applied for the cases where the input needs to be compared with a reference signal. In one phase the offset is stored to the capacitor, in the second phase the input and reference subtracted offset free and the result is amplified. Both reference and input signals are required to be available at the time of comparison. For this particular application, another circuit topology needs to be implemented. Offset of the comparator and previous input (V_0) are required to be stored and compared with V_1.

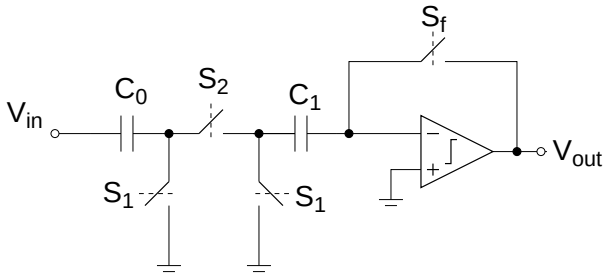

Fig. 99. Proposed comparator with offset cancellation.

The circuit shown in Fig. 99 efficiently implements the comparison with offset cancellation. Non-overlapping clock signals S_1, S_2, and S_f are shown in Fig. 100. Note that S_f is advanced version of S_1, to minimize the charge injection [48]. The offset can be modeled as a voltage source at one of the inputs of the comparator as

shown in Fig. 101. In the first phase, S_f is high, assuming high gain for comparator, the inputs of the comparator will have the same potential, which is ground since V_+ is connected to ground. Then, the offset is stored to C_1 while V_0 is stored to C_0 as shown in Fig. 101. In the second phase, the voltage at the negative input of the comparator is $V_1 - V_0$, as depicted in Fig. 102. The output V_{out} is either low or high depending on the sign of $V_1 - V_0$.

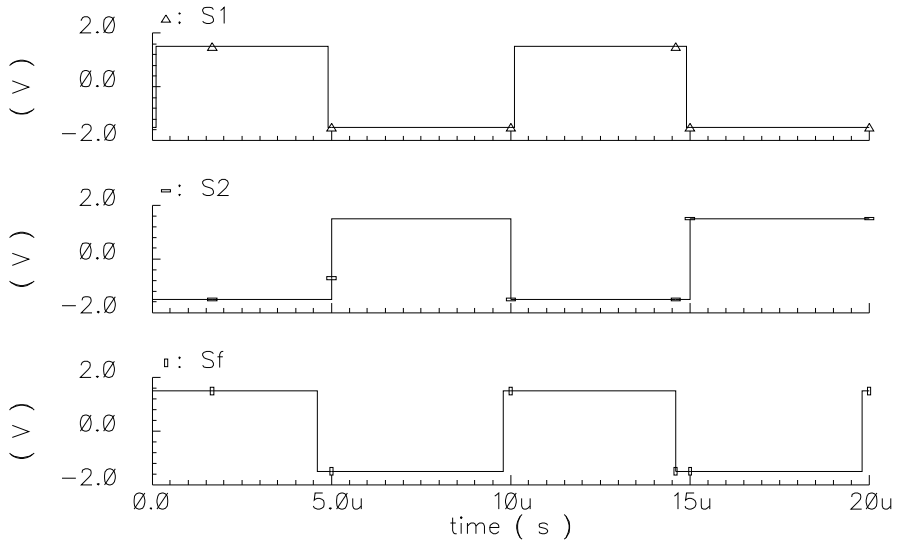

Fig. 100. Clock signals for comparator.

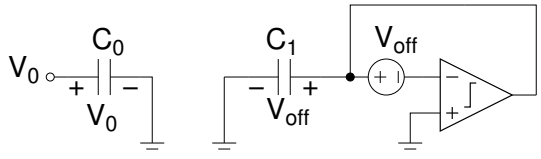

Fig. 101. Sampling phase.

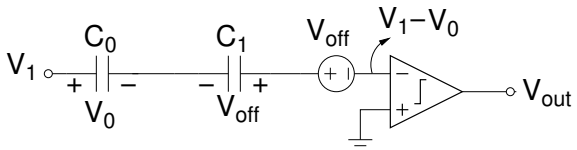

Fig. 102. Comparison phase.

2. Parasitic Pole and Zero Effect

In the presence of parasitic poles, half gain at reference frequencies will shift slightly. Consequently, some error will be involved in the tuning scheme. For illustration, a G_m-C filter is chosen.

The transfer function seen at the bandpass output of a typical G_m-C filter is given by

$$H(s) = \frac{sKG_m/C}{s^2 + sG_{mq}/C + (G_m/C)^2} \tag{6.16}$$

A transconductor with a parasitic zero at frequency ω_z can be modeled by,

$$G_m = G_{m0}\left(1 + \frac{s}{\omega_z}\right) \tag{6.17}$$

Substituting (6.17) into (6.16) results in

$$H(s) = \frac{K\left(\frac{G_{m0}}{C}\left(1 + \frac{s}{\omega_z}\right)\right)^2}{s^2 + s\frac{G_{mq}}{C} + \left(\frac{G_{m0}}{C}\left(1 + \frac{s}{\omega_z}\right)\right)^2} \tag{6.18}$$

Defining $\omega_{00} = G_{m0}/C$ and $b_0 = G_{mq}/C$ yields

$$H(s) = K\omega_{oo}^2 \frac{s\left(1 + \frac{s}{\omega_z}\right)}{s^2 + s\left(\frac{b_o + 2\omega_{oo}^2/\omega_z}{1 + \omega_{oo}^2/\omega_z^2}\right) + \frac{\omega_{oo}^2}{1 + \omega_{oo}^2/\omega_z^2}} \tag{6.19}$$

Note that with $1 >> \omega_{oo}^2/\omega_z^2$, (6.19) can be simplified to

$$H(s) = K' \frac{s\left(1 + \frac{s}{\omega_z}\right)}{s^2 + s\left(b_o + 2\frac{\omega_{oo}^2}{\omega_z}\right) + \omega_{oo}^2} \tag{6.20}$$

After the tuning has converged, the circuit parameters ω_{oo}^2 and b_o will take their values to satisfy the following tuning conditions,

$$|H(j\omega_1| = \frac{1}{2}|H(j\omega_m)| \tag{6.21}$$

$$|H(j\omega_2)| = \frac{1}{2}|H(j\omega_m)| \tag{6.22}$$

The tuned ω_{oo}^2 can be found by using the following

$$|H(j\omega_1| = |H(j\omega_2)| \tag{6.23}$$

In the calculation of ω_{oo}^2, the term b_o in (6.20) is ignored to simplify the analysis. Using (6.20), we obtain

$$\omega_1 \left(\frac{\omega_z^2 + \omega_1^2}{(\omega_{oo}^2 - \omega_1^2)^2\omega_z^2 + 4\omega_1^2\omega_{oo}^4}\right)^{1/2} = \omega_2 \left(\frac{\omega_z^2 + \omega_2^2}{(\omega_{oo}^2 - \omega_2^2)^2\omega_z^2 + 4\omega_2^2\omega_{oo}^4}\right)^{1/2} \tag{6.24}$$

(6.24) can be manipulated to obtain

$$A\omega_{oo}^4 + B\omega_{oo}^2 + C = 0 \tag{6.25}$$

where

$$A = (D-1)\omega_z^2 + 4D\,\omega_2^2 - 4\omega_1^2 \tag{6.26}$$

$$B = 2\omega_z^2(\omega_1^2 - D\,\omega_2) \tag{6.27}$$

$$C = \omega_z^2(D\,\omega_2^4 - \omega_1^4) \tag{6.28}$$

$$D = \frac{4\omega_1^2}{\omega_2^2}\frac{\omega_z^2 + \omega_1^2}{\omega_z^2 + \omega_2^2} \tag{6.29}$$

From (6.25), ω_{oo}^2 is found as

$$\omega_{oo}^2 = (\omega_{oo1}^2 \omega_{oo2}^2)^{1/2} = (C/A)^{1/2} \tag{6.30}$$

where ω_{oo1}^2 and ω_{oo2}^2 are the roots of the second-order polynomial given in (6.25).

In order to find b_o, (6.21) can be used with (6.20), which yields

$$\frac{2\omega_1}{\omega_m} \cdot \sqrt{\frac{\omega_z^2 + \omega_1^2}{(\omega_{oo}^2 - \omega_1^2)^2 \omega_z^2 + 4\omega_1^2 \left(b_o + \dfrac{2\omega_{oo}^2}{\omega_z}\right)^2}} = \sqrt{\frac{\omega_z^2 + \omega_m^2}{(\omega_{oo}^2 - \omega_m^2)^2 \omega_z^2 + \omega_m^2 \left(b_o + \dfrac{2\omega_{oo}^2}{\omega_z}\right)^2}} \tag{6.31}$$

b_o can be extracted from (6.31) as

$$b_o = \left(\frac{D'(\omega_{oo}^2 - \omega_1^2)^2 - (\omega_{oo}^2 - \omega_m^2)^2}{\omega_m^2 - D'\omega_1^2} \right)^{1/2} - 2\frac{\omega_{oo}^2}{\omega^z} \tag{6.32}$$

where ω_{oo}^2 is as given in (6.30) and D' as

$$D' = \frac{\omega_1^2}{\omega_m^2} \frac{\omega_z^2 + \omega_1^2}{\omega_z^2 + \omega_m^2} \tag{6.33}$$

With the proposed tuning circuit, the frequencies are approximated by a frequency generator as $\omega_1 = (N-1)\omega_R$, $\omega_m = N\omega_R$, and $\omega_2 = (N+1)\omega_R$. The tuned circuit parameters, ω_{oo}^2 and b_o, are found by substituting those values into (6.30) and (6.32). Once ω_{oo}^2, b_o and consequently the transfer function of the filter are determined, the tuned center frequency, ω_{ot}, and the quality factor, Q_t, can be found. The center frequency is the frequency where the peak gain occurs, while Q is the center frequency divided by 3-dB bandwidth.

The whole procedure described here is implemented using a Matlab code, which is provided in Appendix G. Define X as the relative location of the zero as $X = \omega_z/\omega_m$. For a fixed target center frequency (keeping $N\omega_R$ constant), frequency and Q-tuning errors are generated as shown in Figs. 103 and 104 for different target Qs

and Xs. Frequency-tuning error is relatively smaller in comparison with Q-tuning. The effect of a parasitic zero depends on the desired Q value. The Q-tuning error is larger for higher Q values. For high Qs (> 80), the Q-tuning error can be as high as 5% if $X \approx 30$.

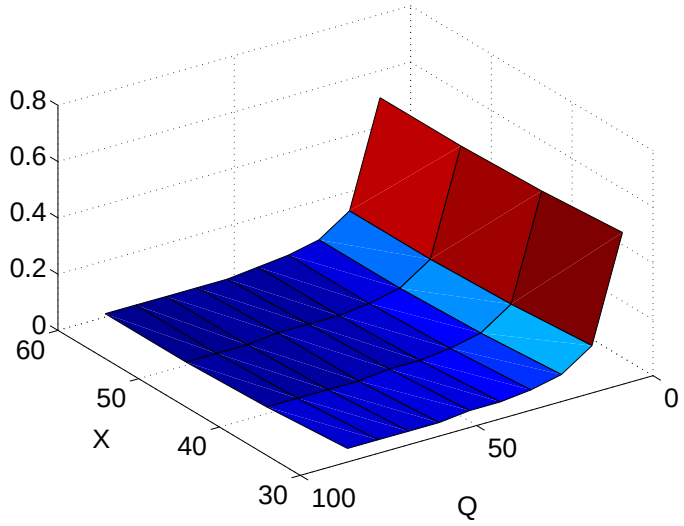

Fig. 103. Frequency-tuning error.

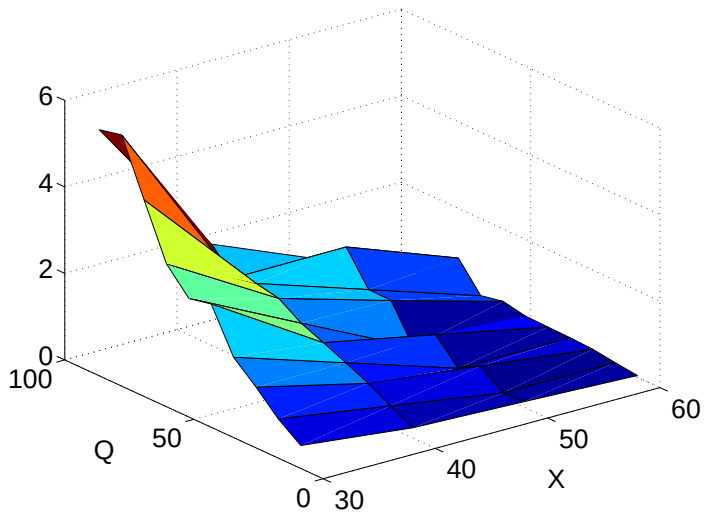

Fig. 104. Q-tuning error.

CHAPTER VII

CONCLUSION

A. Conclusion

Continuous-time filters provide more efficient solutions in terms of power, area, and, cost at high frequencies; however, on-chip realization of these filters necessitates automatic tuning circuits since component values drift due aging, process, and temperature variations. Major existing automatic tuning techniques reported in the literature are summarized and advantages and disadvantages in terms of accuracy and practicality are also provided. Most Q-tuning methods assume that quality factor is equal to passband gain of the filter, and this condition is imposed to filter design; however, this requirement cannot be achieved for some filter types such as active LC.

Novel digital automatic tuning techniques based on phase and magnitude comparisons (DTPC and DTMC) are proposed for second-order continuous-time filters to provide better solutions. In order to tune the center frequency and quality factor, the techniques use certain characteristics of the magnitude and phase responses of a generic second-order filter transfer function regardless of filter implementation. These techniques also do not make any direct assumption on the relation between passband gain and quality-factor of the filter. Digital tuning algorithm enables only Q-tuning or frequency-tuning loops at a given time. In most practical cases, the center frequency and quality factor cannot be changed independently. Therefore, having both tuning loops active on the filter simultaneously may cause the filter start oscillating at high-Q values since Q becomes very sensitive to control voltages. This approach substantially relieves common problem of existing tuning techniques,

and makes the tuning system more reliable.

The circuits realizing DTPC and DTMC are also discussed with possible limitations due to parasitics. The DTPC system is verified with 1% tuning accuracy at 5.5 MHz with discrete prototype on a LC-type tunable filter for Q of 20. While DTPC technique provides accurate and reliable tuning, power requirement of comparators used in circuit implementation may limit its operation at gigahertz ranges. On the other hand, DTMC can operate at very high frequencies since it uses a simple high-frequency envelope detector with relaxed linearity requirement.

The high-order digital tuning method (HDTPC) is proposed for the tuning of high-order filters (cascade of biquads). This technique does not require the use of master-slave method, achieves efficient direct tuning without separating the individual biquad sections. Possible limitation of the system due to parasitics, phase offsets, and finite resolution of synthesizer are also discussed. HDTPC is verified with a 6^{th}-order G_m-C -type bandpass filter (center frequency of 19 MHz and bandwidth of 0.6 MHz).

B. Future Work

Both second-order and high-order digital tuning methods based on phase comparison prove to be accurate and reliable tuning techniques; however, their operation at gigahertz range are limited due to use of comparators. Note that comparator is employed to convert sinusoidal output of filter to rail-to-rail clock signal in order to compare it with the reference clock signal by using a DFF. Alternative circuit techniques are needed to be explored to achieve more efficient comparison of the filter output and the reference signals.

The tuning technique based on magnitude comparison (DTMC) yields better

results in simulation compared to the techniques based on phase comparison (DTPC and HDTPC) in terms of frequency of operation, since it does not use comparators but employs a simple envelope detector. A high-order tuning technique based on a similar magnitude comparison approach can be explored in future for gigahertz-range high-order filters.

Proposed high-order tuning technique can only tune filters formed by cascading biquads. Although cascade topologies are easy to build and have larger tuning ranges, ladder-type filters are also attractive due to their less sensitivity to variation of component values. A digital tuning technique specific to ladder-type filters may be explored as a future work.

REFERENCES

[1] R. Schaumann and M. E. V. Valkenburg, *Design of Analog Filters*. Oxford: Oxford University Press, 2001.

[2] Y. P. Tsividis and J. O. Voorman, *Integrated Continuous-Time Filters*. Piscataway, NJ: IEEE Press, 1993.

[3] E. A. Vittoz, "Future of Analog in the VLSI Environment," in *IEEE Proc. ISCAS*, 1990, vol. 2, pp. 1372–1375.

[4] Y. P. Tsividis, "Integrated Continuous-Time Filter Design-An Overview," *IEEE J. of Solid-State Circuits*, vol. 29, no. 3, pp. 166–176, Mar. 1994.

[5] W. B. Kuhn, D. Nobee, and D. Kelly, "Dynamic Range Performance of On-Chip RF Bandpass Filters," *IEEE Trans. on Circuits Syst. II*, vol. 50, no. 10, pp. 685–694, Oct. 2003.

[6] H. Elwan, H. Alzaher, and M. Ismail, "A New Generation of Global Wireless Compatibility," *IEEE Circuits and Devices Magazine*, vol. 17 , no. 1, pp. 7–19, Jan. 2001.

[7] R. Schaumann, M. S. Ghausi, and K. R. Laker, *Design of Analog Filters: Passive, Active RC and Switched-Capacitor*. Englewood Cliffs, NJ: Prentice-Hall, 1990.

[8] W. Gao and W. M. Snelgrove, "A Linear Integrated LC Bandpass Filter with Q Enhancement," *IEEE Trans. on Circuits Syst. II*, vol. 45, no. 5, pp. 635–639, May 1998.

[9] F. Dulger, E. Sanchez-Sinencio, and J. Silva-Martinez, "A 1.3-V 5-mW Fully Integrated Tunable Bandpass Filter at 2.1 GHz in 0.35 μm CMOS," *IEEE J. Solid-State Circuits*, vol. 38, no. 6, pp. 918–928, June 2003.

[10] W. B. Kuhn, F. W. Stephenson, and A. Elshabini-Riad, "Dynamic Range of High Q OTA-C and Enchanced-Q LC RF Bandpass Filters," in *IEEE Proc. MSCAS*, 1994, vol. 2, pp. 767–771.

[11] R. Schaumann and M. A. Tan, "The Problem of On-Chip Automatic Tuning in Continuous-Time Integrated Filters," in *IEEE Proc. ISCAS*, 1989, vol. 1, pp. 106–109.

[12] J. I. Osa, A. Carlosena, and A. J. Lopez-Martin, "MOSFET-C Filter with On-Chip Tuning and Wide Programming Range," *IEEE Trans. on Circuits Syst. II*, vol. 48, no. 10, pp. 944–951, Oct. 2001.

[13] T. Salo, S. Lindfors, and K. Halonen, "Direct Digital Tuning for Continuous-Time Filters," in *IEEE Proc. MWSCAS*, 2000, vol. 1, pp. 216–219.

[14] Y. P. Tsividis, "Self-Tuned Filters," *IEE Electronics Letters*, vol. 17, pp. 406–407, June 1981.

[15] A. Lopez-Martinez, R. Antonio-Chavez, and J. Silva-Martinez, "A 150 MHz Continuous-Time Seventh Order 0.05° Equiripple Linear Phase Filter with Automatic Tuning System," in *IEEE Proc. ISCAS*, 2001, vol. 1, pp. 156–159.

[16] M. Chen, J. Silva-Martinez, S. Rokhsaz, and M. Robinson, "A 1.8V CMOS, 80-200MHz Continuous-Time 4th Order 0.05° Equiripple Linear Phase Filter with Automatic Tuning System," in *IEEE Proc. ISCAS*, 2002, vol. 5, pp. 173–176.

[17] H. Khorramanabadi and P. R. Gray, "High-Frequency CMOS Continuous-Time Filters," *IEEE J. Solid-State Circuits*, vol. SC-19, no. 12, pp. 939–948, Dec. 1984.

[18] H. Liu and A. I. Karsilayan, "A High-Frequency Bandpass Continuous-Time Filter with Automatic Frequency and Q-factor Tuning," in *IEEE Proc. ISCAS*,

2001, vol. 1, pp. 328–331.

[19] F. Krummenacher and N. Joehl, "A 4-MHz CMOS Continuous-Time Filter with On-Chip Automatic Tuning," *IEEE J. Solid-State Circuits*, vol. 23, no. 3, pp. 750–758, June 1988.

[20] J. Khoury, "Design of a 15-MHz CMOS Continuous-Time Filter with On-chip Tuning," *IEEE J. Solid-State Circuits*, vol. 26, no. 12, pp. 1988–1997, Dec. 1991.

[21] P. Kallam, E. Sanchez-Sinencio, and A. I. Karsilayan, "An Enhanced Adaptive Q-Tuning Scheme for a 100-MHz Fully Symmetric OTA-Based Bandpass Filter," *IEEE J. Solid-State Circuits*, vol. 38, no. 4, pp. 585–593, Apr. 2003.

[22] J. Shin, S. Min, S. Kim, J. Choi, S. Lee, H. Park, and J. Kim, "3.3-V Baseband Gm-C Filters for Wireless Transceiver Applications," in *IEEE Proc. ISCAS*, 2003, vol. 1, pp. 457–460.

[23] T. R. Viswanathan, S. Murtuza, V. Syed, J. Berry and M. Staszel, "Switched-Capacitor Frequency Control Loop," *IEEE J. Solid-State Circuits*, vol. 17, no. 8, pp. 775–778, Aug. 1982.

[24] Z. Y. Chang, D. Haspeslagh, and J. Verfaillie, "A Highly Linear CMOS Gm-C Bandpass Filter with On-Chip Frequency Tuning," *IEEE J. Solid-State Circuits*, vol. 32, no. 3, pp. 388–397, Mar. 1997.

[25] J. Silva-Martinez, M. S. J. Steyaert, and W. Sansen, "A 10.7-MHz 68-dB SNR CMOS Continuous-Time Filter with On-Chip Automatic Tuning," *IEEE J. Solid-State Circuits*, vol. 27, no. 12, pp. 1843–1853, Dec. 1992.

[26] E. Raisanen-Ruotsalainen, K. Lasanen, M. Sijander, and J. Kostamovaara, "A Low-Power 5.4 kHz CMOS Gm-C Bandpass Filter with On-Chip Center Frequency Tuning," in *IEEE Proc. ISCAS*, 2002, vol. 4, pp. 651–654.

[27] J. van der Plas, "MOSFET-C Filter, with Low Excess Noise and Accurate Automatic Tuning," *IEEE J. Solid-State Circuits*, vol. 26, no. 7, pp. 922–929, July 1991.

[28] A. I. Karsilayan and R. Schaumann, "Mixed-Mode Automatic Tuning Scheme for High-Q Continuous-Time Filters," in *IEE Proc. Circuits, Devices and Syst.*, 2000, vol. 147, no. 1, pp. 57–64.

[29] J. Silva-Martinez, "Design Techniques for High-Performance CMOS Continuous-Time Filters," Ph.D. dissertation, Katholieke Universiteit Leuven, Leuven, Belgium, 1992.

[30] H. Huang and E. K. F. Lee, "Design of Low-Voltage CMOS Continuous-Time Filter with On-Chip Automatic Tuning," *IEEE J. Solid-State Circuits*, vol. 36, no. 8, pp. 1168–1178, Aug. 2001.

[31] B. Widrow and M. Hoff, "Adaptive Switching Circuits," *IRE WESCON Convention Rec.*, vol. 4, pp. 96–104, 1960.

[32] D. A. Johns, W. M. Snelgrove, and A. S. Sedra, "Continuous-Time LMS Adaptive Recursive Filters," *IEEE Trans. on Circuits and Syst.*, vol. 38, no. 7, pp. 769–778, July 1991.

[33] K. Kozma, D. Johns, and A. Sedra, "An Approach for Tuning High-Q Continuous-Time Bandpass Filters," in *IEEE Proc. ISCAS*, 1995, vol. 2, pp. 1037–1040.

[34] J. M. Stevenson and E. Sanchez-Sinencio, "An Accurate Quality Factor Tuning Scheme for IF and High-Q Continuous-Time Filters," *IEEE J. Solid-State Circuits*, vol. 33, no. 12, pp. 1970-1978, Dec. 1998.

[35] Liu H. and A. I. Karsilayan, "An Accurate Automatic Tuning Scheme for High-

Q Continuous-Time Bandpass Filters Based on Amplitude Comparison," *IEEE Trans. on Circuits and Syst. II*, vol. 50, no. 8, pp. 415–423, Aug. 2003.

[36] D. Li and Y. P. Tsividis, "Design Techniques for Automatically Tuned Integrated Gigahertz-Range Active LC Filters," *IEEE J. Solid-State Circuits*, vol. 37, no. 3, pp. 967–977, Aug. 2002.

[37] J.R. Moritz and Y. Sun, "Automatic Tuning of High Frequency, High Q, Multiple Loop Feedback Bandpass Filters," in *IEEE Proc. ISCAS*, 2002, vol. 5, pp. 605–608.

[38] A. I. Karsilayan and T. Sumesaglam, "Digital Tuning of Continuous-Time High-Q Filters," in *IEEE Proc. ISCAS*, 2002, vol. 5, pp. 601–604.

[39] T. Sumesaglam and A. I. Karsilayan, "A Digital Approach for Automatic Tuning of Continuous-Time High-Q Filters," *IEEE Trans. on Circuits and Syst. II*, vol. 50, no. 10, pp. 755–761, Oct. 2003.

[40] P. Bowron, H.M. Dahir, "Modelling of Nonideal Active Devices in Continuous-Time OTA-C Filters," in *IEEE Proc. ECCTD*, 1989, pp. 128–131.

[41] J. Silva-Martinez, "Design Issues for UHF OTA-C Filter Realizations," in *IEEE Proc. SSMSD*, 2001, pp. 93–98.

[42] T. Sumesaglam and A. I. Karsilayan, "Digital Tuning of High-Order High-Q Continuous-Time Filters," *IEE Electronics Letters*, vol. 38, no. 19, pp. 1076–1078, Sep. 2002.

[43] T. Sumesaglam and A. I. Karsilayan, "A Digital Automatic Tuning Technique for High-Order Continuous-Time Filters," to appear in *IEEE Trans. on Circuits and Syst. I.*

[44] M. T. Hill, A. Cantoni "An Integrated High-Frequency Narrow-Band High Resolution Synthesizer," *IEEE Trans. on Circuits and Syst. II*, vol. 46, no. 9, pp. 1171–1178, Sep. 1999.

[45] E. Sanchez-Sinencio and J. Silva-Martinez, "CMOS Transconductance Amplifiers, Architectures and Active Filters: a Tutorial," in *IEE Proc. Circuits, Devices, and Syst.*, 2000, vol. 147, no. 1, pp. 3–12.

[46] T. Sumesaglam and A. I. Karsilayan, "Digital Tuning of Analog Bandpass Filters Based on Envelope Detection," in *IEEE Proc. ISCAS*, 2004.

[47] P. Deval, V. Valence, F. Anghinolfi, "Low-Voltage (3.3 V)/Low-Power (100 μW), 2 MHz CMOS Comparator for 12-bit ADCS," in *IEEE Proc. ISCAS*, vol. 5, pp. 501–504, May 1994.

[48] D. A. Johns and K. Martin, *Analog Integrated Circuit Design.* New York: John Wiley & Sons, 1997.

APPENDIX A

MAPLE CODE: PARASITIC POLE EFFECT ON DIGITAL TUNING METHOD BASED ON PHASE COMPARISON

```
gms having parasitic pole
gm= gmo/ ( 1+s/wp ); pole located at wp = X woo
Solve two equations (comes from tuning steady state conditions)
obtain woo, Qo which are two parameters in TF
the desired center freq is arbitrarily  chosen as 40MHz
w1 and w2 are the reference frequencies
Output format is either
target Q, X, Qo, woo
or
target Q, X, woo, Qo

> for N in [20,30, 40, 80, 160, 320] do
> wr:=40e6/N:
> for X in [12, 16, 20, 24, 30, 48, 60, 96] do
> eq1:=w1*woo/Q-2*w1^3/wp=woo^2-w1^2+w1^4/wp^2-woo*w1^2/Q/wp:
> eq2:=-w2*woo/Q+2*w2^3/wp=woo^2-w2^2+w2^4/wp^2-woo*w2^2/Q/wp:
> eq11:=subs(w1=(N-1)*wr,wp=woo*X,eq1):
> eq22:=subs(w2=(N+1)*wr,wp=woo*X,eq2):
> mysol:=solve({eq11,eq22},{woo,Q}):woox:=rhs(mysol[5,1]):
> Qox:=rhs(mysol[5,2]):
> print(N/2,X,Qox,woox);
> od;
> od;

                10, 12, .4057458449 10e8 , 3.808047929
                10, 16, .4039618707 10e8 , 4.481912845
                10, 20, 5.024151963, .4029596709 10e8
```

```
10, 24, 5.469538050, .4023210188 10e8
10, 30, .4017065327 10e8 , 6.005822378
10, 48, 7.051876669, .4008311683 10e8
10, 60, 7.489526111, .4005519776 10e8
10, 96, 8.261569479, .4001451954 10e8

15, 12, .4044145467 10e8 , 4.349780464
15, 16, .4030196308 10e8 , 5.263506857
15, 20, .4022557068 10e8 , 6.034083779
15, 24, .4017777320 10e8 , 6.692136839
15, 30, 7.516412303, .4013252672 10e8
15, 48, 9.233257468, .4006952183 10e8
15, 60, .4004983514 10e8 , 9.998097406
15, 96, .4002155217 10e8 , 11.42108612

20, 12, 4.679929662, .4037020649 10e8
20, 16, .4025037995 10e8 , 5.762844750
20, 20, .4018600439 10e8 , 6.704424756
20, 24, .4014629825 10e8 , 7.529903246
20, 30, .4010920074 10e8 , 8.592874184
20, 48, 10.91680077, .4005851459 10e8
20, 60, 12.00303982, .4004295652 10e8
20, 96, .4002088472 10e8 , 14.11421319

40, 12, 5.276753686, .4025747434 10e8
40, 16, 6.713529777, .4016741719 10e8
40, 20, .4012119818 10e8 , 8.038652195
40, 24, 9.263487592, .4009370199 10e8
40, 30, 10.93534517, .4006888858 10e8
40, 48, .4003674732 10e8 , 15.01506574
40, 60, 17.15365313, .4002739776 10e8
40, 96, 21.82206390, .4001465973 10e8

80, 12, .4019846847 10e8 , 5.633944565
```

```
 80, 16, .4012342065 10e8 , 7.314325550
 80, 20, .4008633986 10e8 , 8.923613384
 80, 24, .4006498186 10e8 , 10.46471824
 80, 30, .4004633803 10e8 , 12.65646464
 80, 48, 18.47795480, .4002349946 10e8
 80, 60, .4001726121 10e8 , 21.83134539
 80, 96, 30.00754725, .4000919774 10e8

160, 12, .4016830488 10e8 , 5.830740105
160, 16, .4010079337 10e8 , 7.656218638
160, 20, .4006829685 10e8 , 9.442549414
160, 24, 11.18923978, .4005001633 10e8
160, 30, .4003446424 10e8 , 13.73630073
160, 48, 20.88456548, .4001628462 10e8
160, 60, .4001160379 10e8 , 25.27562032
160, 96, 36.93129227, .4000587952 10e8
```

APPENDIX B

MATLAB CODE: PARASITIC POLE EFFECT ON DIGITAL TUNING BASED ON PHASE COMPARISON

```
% Tuning error in the presence of parasitic pole

% Qo and woo data calculated by Maple
datwoo=1e4*[4057.5,4039.6,4029.6,4023.2,4017.1,4008.3,4005.5,4001.4,
            4044.1,4030.2,4022.6,4017.8,4013.2,4006.9,4005.0,4002.1,
            4037.0,4025.0,4018.6,4014.6,4010.9,4005.8,4004.3,4002.1,
            4025.7,4016.7,4012.1,4009.4,4006.9,4004.6,4002.7,4000.9,
            4019.8,4012.3,4008.6,4006.5,4004.6,4002.3,4001.7,4000.9,
            4016.8,4010.1,4006.8,4005.0,4003.5,4001.6,4001.2,4001];

datQ=[3.80805,4.48191,5.02415,5.4695,6.00582,7.05188,7.4895,8.26157,
      4.34978,5.26351,6.03408,6.6921,7.51641,9.23326,9.9981,11.4211,
      4.66799,5.76284,6.70442,7.5299,8.59287,10.9168,12.003,14.1142,
      5.27675,6.71353,8.03865,9.2635,10.9353,15.0151,17.154,21.8221,
      5.63394,7.31432,8.92361,10.465,12.6565,18.4780,21.831,30.0075,
      5.83074,7.65622,9.44255,11.189,13.7363,20.8846,25.275,36.931];

% Desired Q values
Qdes=[10, 10, 10, 10, 10, 10, 10, 10,
      15, 15, 15, 15, 15, 15, 15, 15,
      20, 20, 20, 20, 20, 20, 20, 20,
      40, 40, 40, 40, 40, 40, 40, 40,
      80, 80, 80, 80, 80, 80, 80, 80,
     160,160,160,160,160,160,160,160];
indN=0;

for N=[20,30,40,80,160,320],
```

```
indN=indN+1;
indX=0;
for X=[12,16,20,24,30,48,60,96],
   N
   X;
   indX=indX+1;
   woo=datwoo(indN,indX);
   Q=datQ(indN,indX);
   wp=X*woo;

   % Initialize
   maxval=0;
   wo=0;

   % Find the tuned center frequency
   for w=38e6:300:42e6,
     dum=abs(woo^2*wp^2*Q/(-w^2*Q*wp^2-j*2*w^3*Q*wp+w^4*Q+j
         *w*woo*wp^2-w^2*woo*wp+woo^2*wp^2*Q));
     if (dum>=maxval), maxval=dum;wo=w; end
   end

   % Find the tuned Q
   wlfound=0;
   whfound=0;
   wl=0;wh=0;

   for w=37e6:300:43e6,
      dum=abs(woo^2*wp^2*Q/(-w^2*Q*wp^2-j*2*w^3*Q*wp+w^4*Q+j
          *w*woo*wp^2-w^2*woo*wp+woo^2*wp^2*Q));

      if ((dum>=maxval/sqrt(2)) & (wlfound==0)),
         wl=w;
         wlfound=1;
      end
```

```
            if ((dum<=maxval/sqrt(2)) & (whfound==0) &  (wlfound==1) ),
               wh=w; whfound=1;
            end

         end

         Qa=wo/(wh-wl);
         %Frequency error matrix
         errwo(indN,indX)=abs(40e6-wo)/40e6*100;
         Qarr(indN,indX)=Qa;
      end %X
      % Q error matrix
      Qerr(indN,:)=abs( Qarr(indN,:)-Qdes(indN,:))/ Qdes(indN,1)*100;
end %N

% Plot 3D tuning errors
x=[12 16 20 24 30 48 60 96]; q=[10 15 20 40 80 160];
surf(x,q,Qerr), xlabel('X'),ylabel('Q'),title('Q error (%)');
%surf(x,q,errwo),xlabel('X'),ylabel('Q'),title('Wo error (%)');
```

APPENDIX C

MATLAB CODE: PARASITIC ZERO EFFECT ON DTPC

```
% Parasitic zero effect (located at X*woo) on tuning accuracy
clear;
Nrange=20:10:100;
Xrange=50:10:120;
woo=zeros(length(Xrange),length(Nrange));
Qo=zeros(length(Xrange),length(Nrange));

% Calculate woo, Qo values satisfying tuning condition
i=0;
for X=Xrange,
    i=i+1;
    j=0;
    for N=Nrange,
        j=j+1;
        w1=(N-1);
        w2=(N+1);
        diff=w1-w2;
        sum=w1+w2;
        mul=w1*w2;
        A=1+1/X^2;
      sol=roots([sum/mul -2/X-(sum*diff)/sum/X A*(-sum+w1/X^2-w2/X^2)
            -A*(w1^2+w2^2-2*mul)/X A*(w1*w2^2/X^2-w1^2*w2/X^2)]);
        woo(i,j)=sol(2);
        dum=woo(i,j);
        Qo(i,j)=(dum^2-w1*dum/X)  /
                (dum^3/w1-w1*dum*(A-2/X^2)-dum^2/X-w1^2/X*A);
        dum=Qo(i,j)
    end
end
```

```
% Calculate actual (or tuned) Q and wo and tuning errors
% Desired center frequencies
wo_des=sqrt(Nrange.^2-1);
% Desired Q values
Q_des=wo_des/2;
dw=0.01;

i=0;
for X=Xrange,
   i=i+1;
   j=0;
   for N=Nrange,
      j=j+1;
      % next woo, Qo
      woo_nx=woo(i,j);
      Qo_nx=Qo(i,j);

      % Initialize
      maxval=0;
      wo=0;
      % Transfer function with zero
      % calculated over frequency range of wrange
      wrange=(N-10:dw:N+10);
      h=freqs([1/X woo_nx 0],[1+1/X/Qo_nx
        woo_nx*(1/Qo_nx+2/X)/(1+1/X^2) woo_nx^2/(1+1/X^2)],wrange);

      % Find the tuned center frequency
      for w=wrange,
        dum=abs(h(round((w-wrange(1))/dw+1)));
        if (dum>=maxval), maxval=dum;wo=w; end
      end

      % Find the tuned Q
```

```
        wlfound=0;
        whfound=0;
        wl=0;wh=0;

        for w=wrange,
            dum=abs(h(round((w-wrange(1))/dw+1)));
            if ((dum>=maxval/sqrt(2)) & (wlfound==0)),
                wl=w;
                wlfound=1;
            end
            if ((dum<=maxval/sqrt(2)) & (whfound==0) & (wlfound==1)),
                wh=w;
                whfound=1;
            end
        end

        Qa=wo/(wh-wl);
        %Frequency error matrix
        errwo(i,j)=abs(wo_des(j)-wo)/wo_des(j)*100;
        % Q error matrix
        errQ(i,j)=abs(Q_des(j)-Qa)/Q_des(j)*100;

    end %X
end %N

% Results, figures
errwo
errQ
figure(1)
set(gca,'Xgrid','on','Ygrid','on','FontSize',24)
surf(Q_des,Xrange,errQ),ylabel('X'),xlabel('Q'),title('Q error(%)');
set(gca,'XTick',[10 20 30 40 50])
set(gca,'YTick',[50 70 90 110])
axis([10 50 50 120 0.5 4.5])
```

```
figure(2)
set(gca,'Xgrid','on','Ygrid','on','FontSize',24)
surf(Q_des,Xrange,errwo),ylabel('X'),xlabel('Q'),title('Ferror(%)');
set(gca,'XTick',[10 20 30 40 50])
set(gca,'YTick',[50 70 90 110])
axis([10 50 50 120 0.5 4.5])
```

APPENDIX D

MATLAB CODE: PHASE SHIFT CAUSED BY PARASITICS

```
% Phase shift caused by a parasitic
% pole or a zero around the effective center frequency
% calculated for second-order case.

clear
dw=1e3;
w=1e6:dw:20e6;
%define s
s = j.*w;

%Define circuit parameters
C=10e-12;   %capacitor
K=10;
gmq0=50e-6; %low frequency value of transconductance
gm0=100e-6;

i=0;
Xrng=5:10:100;
for X=Xrng,
    i=i+1;
    woo=gm0/C;
    wz=X*woo;% Location of parasitic zero
    wp=X*woo;% Location of parasitic pole

    %Transfer function of gm-C biquadratic filter with zero
    gm=gm0*(1+s/wz); %gm with zero
    gmq=gmq0*(1+s/wz);
    H=(s.*K.*gm/C)./(s.^2+s.*gmq/C+(gm/C).^2);
    % finds the center freq. (where the maximum gain is.)
```

```
    [G I]=max(H);
    ph=phase(G)/pi*180
    ph_off(i)=ph;

    %Transfer function of gm-C biquadratic filter with pole
    gm=gm0./(1+s/wp);  %gm with pole
    gmq=gmq0./(1+s/wp);
    H=s.*(K*gm/C)./(s.^2+s.*gmq/C+(gm/C).^2);
    [G I]=max(H);
    ph=phase(G)/pi*180
    ph_offp(i)=ph;

end
plot(Xrng,ph_off,Xrng,ph_offp)
set(gca,'Xgrid','on','Ygrid','on','FontSize',20)
axis([5 100 -10 13 ])
```

APPENDIX E

MATLAB CODE: DESIGN TOOL FOR HIGH-ORDER TUNING METHOD

```
% A Design Tool for High-order high-Q bandpass filters
% with automatic high-order tuning method.

% Overall outputs of the program for the design:
% N, f0, Q, Ka, Kb, ph_a, ph_b, L, fR

% It consists of 3 parts

%  1- Filter approximation (or synthesis)
%  2- Determination of Reference frequencies and phases
%  3- Finding the minimum required resolution L
%      and other parameters (fr, Ka, Kb)

%  Part-1 Synthesize the filter
%  Inputs      :fp1, fp2, fs1, a_min, a_max, ripple (filter specs )
%  Outputs      :N, f0, Q  (Desired values)

answer=input
  ('Enter new filter [n] specs or use the default [d]: ','s');
%  Get the filter specifications
if answer=='n'
    fp1=input('Lower end passband frequency : ');
    fp2=input('Higher end passband frequency : ');
    fs1=input('Lower end stop frequency : ');
    fs2=fp1*fp2/fs1;  % Required by the filter synthesis method
    a_max=input('Max attenuation in passband (dB): ');
    a_min=input('Min required attenuation at stop band (dB): ');
    type=input
      ('Filter appr. type Butterworth [b], Chebyshev [c]: ','s');
```

```
    %  Other types of filters can also be added.
    if type=='b',
        ripple=a_max;
    else
        ripple=input('Max ripple in passband (dB): ');
    end
else                        % Default filter specs
    fp1=18.81e6;            % Passband frequencies
    fp2=19.67e6;
    fs1=18.03e6;            % Stop band frequencies
    fs2=fp1*fp2/fs1;
    a_max=3;                % Max attenuation in passband
    a_min=20;               % Min required attenuation at in stopband
    type='b';
end
df=(fp2-fp1)/10000;   % Frequency increment,
                      %may affect accuracy if not small enough
f=fs1*0.7:df:fs2*1.3; % Freq axis

global xstart xstop  dx; % Global declaration for index scale
xstart=min(f);         % function sc.m, it helps to convert freq.
xstop=max(f);          % to matrix index for convenience
dx=df;

% Find the lowpass equivalent order (N) and pole-zeros
fs_lp=(fs2-fs1)/(fp2-fp1);
if type=='b',                                  % Butterworth
    n=log10((10^(a_min/10)-1)/(10^(a_max/10)-1))/(2*log10(fs_lp));
    N=ceil(n);                                 % Round up
    a_max_opt=10*log10((10^(a_min/10)-1)/(10^(2*N*log10(fs_lp)))+1);
    f3dB_lp=1/(10^(a_max_opt/10)-1)^(1/2/N); % 3dB freq. of LP equiv.
    BW=(fp2-fp1)*f3dB_lp;                  % Bandwidth of BP
    fo=sqrt(fp1*fp2);                      % Center freq. of BP
    [zl,pl,kl]=buttap(N);                 % 1st order LP zeros, poles
```

```
elseif type==’c’,                          % Chebyshev
    n=acosh(sqrt((10^(a_min/10)-1)/(10^(a_max/10)-1)))/acosh(fs_lp);
    N=ceil(n);                             % Round up
    a_max_opt=10*log10((10^(a_min/10)-1)/(cosh(N*acosh(fs_lp))^2)+1);
    fo=sqrt(fp1*fp2);                      % Center freq.
    BW=(fp2-fp1);                          % Bandwidth of BP
    [zl,pl,kl]=cheb1ap(N,a_max_opt);       % 1st order LP zeros, poles

% Other types of filters can be added here.
end

% Find the transfer function of overall filter
numl=poly(zl);                          % Numerator of lp func
denl=poly(pl);                          % Denominator of lp func
[numb denb]=lp2bp(numl,denl,fo,BW);     % Transform lp to bp
hb=freqs(numb,denb,f);                  % Bandpass TF calculated along f
hb_db=20*log10(abs(hb))-20*log10(max(abs(hb)));% Normalize gain to 0dB
figure(1)

plot(f,hb_db,’r’);              % Plot the magnitude of overall filter
xlabel(’Frequency ( MHz )’)
ylabel(’Magnitude ( dB )’)

% Find Qs and f0s from the denominator of bandpass transfer function
root_b=roots(denb);
count=1;
f0=zeros(1,N);
Q=zeros(1,N);
for j=1:2*N,
    for i=j+1:2*N,
        if round(real(root_b(j)))==round(real(root_b(i))),
            if round(imag(root_b(j)))==-round(imag(root_b(i))),
                f0(count)=real(sqrt(root_b(i)*root_b(j)));
```

```
                Q(count)=real(f0(count)/(-root_b(i)-root_b(j)));
                count=count+1;
            end
        end
    end
end
temp=[f0;Q]';
temp=sortrows(temp);

% The outputs of the program, so far the filter is approximated
sprintf('Number of biquads required: %d \n', N)
f0=temp(:,1)';
Q=temp(:,2)';
disp('Center Frequencies:')
disp(f0)
disp('  Q values:')
disp(Q)

% Part-2   Determine the references
% Inputs:  N, f0, Q (From part-1), ph_a, ph_b
%       (user will be asked to choose them from phase response)
% Outputs: fre_a,fre_b

% initialize
hi=-pi/2*ones(1,length(f));
fre_a=zeros(1,length(f0));
fre_b=zeros(1,length(f0));
ph_a=zeros(1,length(f0));
ph_b=zeros(1,length(f0));

choice=input
  ('Use the refs for the default filter[d] or continue[c]: ','s');
if choice=='c',
  for i=1:N,        % For each output 1 to N
```

```
    % Phase response with Q=Qd*1.5
    hi_p=phase(freqs(1, [1 f0(i)/(Q(i)*1.50) f0(i)^2],f))+hi+pi/2;
    % Phase response with Q=Qd*0.5
    hi_n=phase(freqs(1, [1 f0(i)/(Q(i)*0.50) f0(i)^2],f))+hi+pi/2;
    % Desired phase response
    hi=phase(freqs(1, [1 f0(i)/Q(i) f0(i)^2],f))+hi+pi/2;

   % Plot all phase curves for the ith output
   figure(2);
   plot(f,180/pi*hi,'b -',f,180/pi*hi_p,'r --',f,180/pi*hi_n,'r --')
   xlabel('Frequency ( MHz )')
   ylabel('Phase ( deg )')
   axis([min(f) max(f) min(hi)/pi*180 min(hi)/pi*180+270]);

    confirm='n'; % Initialize
    while (confirm=='n'),
      sprintf('Output %d ',i)
      ph_a(i)=input
        ('Choose the reference phase ph_a (deg): ')/180*pi;
      [value ind]=min(abs(hi-ph_a(i)));
      fre_a(i)=ind*df+min(f);  % Finds the corresponding frequency
      % Draws a line on the graph
      line([fre_a(i) fre_a(i)],[min(hi)/pi*180 min(hi)/pi*180+270]);

      ph_b(i)=input
        ('Choose the reference phase ph_b (deg): ')/180*pi;
      [value ind]=min(abs(hi-ph_b(i)));
      fre_b(i)=ind*df+min(f);  % Finds the corresponding frequency
      % Draws a line on the graph
      line([fre_b(i) fre_b(i)],[min(hi)/pi*180 min(hi)/pi*180+270]);

      confirm=input('please confirm y/n :','s')
    end
end
```

```
else% Default reference phases and frequencies for the default filter
  fre_a =[18460400    18903816    19457742];
  fre_b =[19024388    20140152    20071610];
  ph_a=[-45   -90  -135]/180*pi;
  ph_b=[-135  -225  -270]/180*pi;
  f0 =1e6 *[18.74 19.235 19.743];
  Q =[33.2288 16.6088 33.2288];
end

% Part-3 Find the minimum required L, Ka, Kb and fR,
%              plot the tuned magnitude responses
% Outputs:  L, Ka, Kb, fR, err_f, err_Q (Individual tuning Errors)

Qt=zeros(1,N);  % Tuned values
f0t=zeros(1,N);
L=4; % Initial value
cont_tune=1;
figure(1);
hold on
while cont_tune==1,  % Increment L till the specs are satisfied
  L=L+1;
  fR=max([fre_a fre_b])/2^L;
  Ka=round(fre_a/fR);
  Kb=round(fre_b/fR);
  suma=0;sumb=0;
  % This part implements the equations found for Q and center freq
  for i=1:N,
    suma=0;sumb=0;
    if (i >1)
      for j=1:i-1,
        suma=suma+atan(f0t(j)*Ka(i)*fR/Qt(j)/(f0t(j)^2-fR^2*Ka(i)^2));
        sumb=sumb+atan(f0t(j)*Kb(i)*fR/Qt(j)/(f0t(j)^2-fR^2*Kb(i)^2));
      end
    end
```

```
  di=tan((i-1)*pi/2-ph_a(i)-suma);
  ci=Kb(i)*di/(Ka(i)*tan((i-1)*pi/2-ph_b(i)-sumb));
  f0t(i)=fR*sqrt((Kb(i)^2-Ka(i)^2)/(1-ci)+Ka(i)^2);
  Qt(i)=
       Ka(i)/di/(Kb(i)^2-Ka(i)^2)*sqrt((ci*Ka(i)^2-Kb(i)^2)*(ci-1));
end

% Find and plot the tuned magnitude response of the overall filter
ht=freqs([f0t(1)/(Qt(1)) 0],[1 f0t(1)/(Qt(1)) f0t(1)^2],f);
for i=2:N,
    dum=freqs([f0t(i)/(Qt(i)) 0],[1 f0t(i)/(Qt(i)) f0t(i)^2],f);
    ht=ht.*dum;
end
% Normalized magnitude
ht_db=20*log10(abs(ht))-max(20*log10(abs(ht)));
plot(f,ht_db);

% Find the ripple in the pass band
max_h=max(ht_db);
min_h=max_h;
for i=sc(fp1):sc(fp2),
    if ( ht_db(i) <= min_h)
        min_h=ht_db(i);
        min_index=i;
    end
end
min_index=min_index*df+min(f); % Convert vector index to frequency
ripplet=abs(min_h-max_h);      % Ripple found in dB

% Check the specifications
if(ht_db(sc(fs1)) >= -a_min )
    cont_tune=1;
elseif (ht_db(sc(fs2)) >= -a_min )%
    cont_tune=1;
```

```
  elseif ( ht_db(sc(fp1)) <= -a_max )
      cont_tune=1;
  elseif (ht_db(sc(fp2)) <= -a_max )
      cont_tune=1;
  elseif (a_max <= ripplet )
      cont_tune=1;
  else
      cont_tune=0;     % Tuning is done
  end
  disp('Press any key to continue')
  sprintf('Current L : %d',L)
  pause
end % while
err_Q=abs(Q-Qt)./Q*100;
err_f=abs(f0-f0t)./f0*100;

disp('Individual Q tuning errors :'),disp(err_Q)
disp('Individual f tuning errors :'),disp(err_f)
sprintf('Required bit-resolution for synthesizer: %d', L)
sprintf('Reference frequency (fR) for synthesizer: %d', fR)
disp('Coefficients Ka: '),disp(Ka)
disp('Coefficients Kb: '),disp(Kb)
disp('References phases ph_a:'),disp(ph_a*180/pi)
disp('References phases ph_b:'),disp(ph_b*180/pi)
```

APPENDIX F

MATLAB CODE: DTMC

```
% fo current center freq
% Q current Q
% Qd desired Q
% fod desired center frequency
% A1,B1,A2,B2; logic
% g1,g2 gains at f1 and f2

mq=10;     % Sensitivity of Q to vq   dQ/dV
mf=10e6;   % Sensitivity of fo to vf   dfo/dV
dt=1;      % Time increment, one tuning step
tf=100;    % Final time, should be large enough for convergence
off=0;     % Offset in the comparator
t=0:dt:tf;
f1=8.5e6;    % Reference frequencies
f2=11.5e6;
freq=2e6:1e3:18e6;  % Frequency range over which TF calculated
fod=sqrt(f1*f2)     % Target center freq. and Q
Qd=sqrt(3)*fod/(f2-f1)

% Following vectors keeps the values at each tuning step
vf=zeros(1,length(tf));  % Freq. control voltage
vq=zeros(1,length(tf));  % Q control voltage
fo=zeros(1,length(tf));  % fo values
Q=zeros(1,length(tf));   % Q values
g1=zeros(1,length(tf));
g2=zeros(1,length(tf));

dvf=10e-3;   % Freq. control voltage increment
dvq=20e-3;   % Q control voltage increment
```

```
vf(1)=0.4;   % Initialize
vq(1)=1.5;

figure(1)
set(gca,'NextPlot','replace')
plot(1)
hold on
line([min(freq) max(freq)],20*log10([0.5 0.5]))
line([f1 f1],[-15 0])
line([f2 f2],[-15 0])
set(gca,'FontSize',20)
set(gca,'XTick',[])
set(gca,'Xgrid','on','Ygrid','on')

ylabel('( dB )')
plot(freq,20*log10(bp_mag(Qd,fod,freq)),'r');
axis([min(freq) max(freq) -15 0])
for i=1:dt:length(t)-1,

    fo(i)=fod+mf*(vf(i)-1); %current f
    Q(i)=Qd+mq*(vq(i)-1); %current Q

 G1=abs(f1)/sqrt((fo(i)*fo(i)-f1*f1)^2+f1*f1*fo(i)*fo(i)/Q(i)/Q(i));
 G2=abs(f2)/sqrt((fo(i)*fo(i)-f2*f2)^2+f2*f2*fo(i)*fo(i)/Q(i)/Q(i));
 G0=abs(fod)/sqrt((fo(i)*fo(i)-fod*fod)^2+fod*fod*fo(i)
                                              *fo(i)/Q(i)/Q(i));

    g1(i)=G1/G0;%normalize
    g2(i)=G2/G0;%normalize

    A1=( g1(i) >= 0.5-off );
    B1=( g1(i) >= 1-off) ;
    A2=( g2(i) >= 0.5+off);
    B2=( g2(i) >= 1+off);
```

```
    fu=(B1&A2&not(B2))|(A1&not(A2));
    fd=(A1&not(B1)&B2)|(A2&not(A1));
    qu=A1&not(B1)&A2&not(B2);
    qd=(not(A2)&not(A1));

    vf(i+1)=vf(i)+fu*dvf-fd*dvf; % Update voltages
    vq(i+1)=vq(i)+qu*dvq-qd*dvq;
    if ( mod(i,10)==0),
        Go=1;
        plot(freq,20*log10(bp_mag(Q(i),fo(i),freq)/Go),'k--');
    end
end

% Plot control voltages
t=1:dt:length(t);
figure(2)
set(gca,'FontSize',20)
xlb=[0 100 200 300 400 500];
plot(t,vf,'r --',t,vq,'b')
title('vf & vq')
set(gca,'Xgrid','on','Ygrid','on')
xlabel('time')
set(gca,'XTick',xlb)
set(gca,'XTickLabel',xlb)

Qt=(Q(length(t)-1))      % Tuned Q value at tf
fot=(fo(length(t)-1))    % Tuned fo value at tf
err_q=(Qd-Qt)/Qd*100     % Q tuning error
err_f=(fod-fot)/fod*100  % Frequency tuning error
```

APPENDIX G

MATLAB CODE: PARASITIC EFFECT ON DTMC

```
% Zero effect on the magnitude based tuning is approximated

wm=10e6;                % Middle reference frequency
wstart=0.4*wm;          % Start frequency
wstop=1.6*wm;           % Stop frequency
dw=(wstop-wstart)/10000;% Frequency increment
w=wstart:dw:wstop;
xrange=30:10:60;    % Range of X, the location of parasitic
Nrange=10:10:100;   % Range of N, freq. division ratio for mid freq.

%Initialize variables
Qt=zeros(length(Nrange),length(xrange));
wot=zeros(length(Nrange),length(xrange));
Qd=zeros(length(Nrange),length(xrange));
wod=zeros(length(Nrange),length(xrange));

j=0;
for N=Nrange,
  j=j+1;
  wr=wm/N;
  w1=(N-1)*wr;
  w2=(N+1)*wr;
  Qd(j,:)=sqrt(3)/2*N;   % Desired Q
  wod(j,:)=wm;           % Desired wo

  i=0;
  for x=xrange,
    i=i+1;
    wz=x*wm;
```

```
    D=w1^2/w2^2*(w1^2+wz^2)/(w2^2+wz^2);
    A=wz^2*(D-1)+4*D*w2^2-4*w1^2;
    B=2*wz^2*(w1^2-D*w2^2);
    C=wz^2*(D*w2^4-w1^4);
    % Tuned circuit parameter woo
    woo=abs(C/A)^(1/4);

    Db=1/4*wm^2/w1^2*(wm^2+wz^2)/(w1^2+wz^2);
    % Tuned circuit parameter bo
    bo=sqrt((Db*(woo^2-w1^2)^2-(woo^2-wm^2)^2)/(wm^2-Db*w1^2))
       -2*woo^2/wz;
    num=[1/wz 1 0];
    den=[1 bo+2*woo^2/wz woo^2];
    hb=freqs(num,den,w);
    hbn=hb/max(abs(hb)); % Normalized transfer function of the filter

    % Find tuned wo and Q
    [m windex]=max(abs(hbn));
    wot(j,i)=wstart+dw*windex;
    htemp=hbn(1:windex);
    [m wli]=min(abs(abs(htemp)-0.707));
    wl=wstart+dw*wli;
    htemp=hbn(windex:length(hbn));
    [m wui]=min(abs(abs(htemp)-0.707));
    wu=wstart+dw*(wui+windex);
    Qt(j,i)=wot(j,i)/(wu-wl);

  end % x

end   %N

% Plot Error Figures
figure(1)
Qerr=abs(Qt-Qd)./Qd*100
```

```
Ferr=abs(wot-wod)./wod*100
surf(xrange,sqrt(3)/2*Nrange,Qerr);
set(gca,'FontSize',20)
xlabel('X'),ylabel('Q')

figure(2)
surf(xrange,sqrt(3)/2*Nrange,Ferr);
set(gca,'FontSize',20)
xlabel('X'),ylabel('Q')
```

VITA

Taner Sumesaglam received his Bachelor of Science degree in electrical electronics engineering from Bilkent University, Turkey in 2000. He joined the Doctor of Philosophy program of Texas A&M University in September 2000. He worked on automatic tuning of high frequency filters under the supervision of Dr. Aydin I. Karsilayan. His research interests include continuous-time filters, automatic tuning circuits and RF/analog mixed-signal integrated circuits for communication applications.

Permanent address: c/o Dr. Aydin I. Karsilayan
Electrical Engineering
3128 TAMU
College Station, TX 77843
USA