Parameter-Varying Low Pass Filters Based on Switched-Capacitor Circuits

by

Timoteo Cayetano Antonio

Thesis submitted as the partial fulfillment
of the requirement for the degree of:

MASTER OF SCIENCE IN ELECTRONIC

at

Instituto Nacional de Astrofísica, Óptica y Electrónica

May, 2014
Tonantzintla, Puebla

Advisor:

Dr. Miguel Ángel Gutiérrez de Anda

Parameter-Varying Low Pass Filters Based on Switched-Capacitors Circuits

Timoteo Cayetano Antonio

Contents

Acknowledgments

This work and the realization of my graduate studies would not have been possible without the financial support of CONACyT under the scholarship grant 322618.

I sincerely would want to give to Instituto Nacional de Astrofísica, Óptica y Electrónica many thanks for all the support provided, particularly to Instrumentation Laboratory, Department of Electronics and the library "Luis Enrique Erro". Likewise, many thanks go to the researchers from the Department of Electronics in particular to the group of IC Design.

Finally, I would like to express my unlimited gratitude to Dr. Miguel Ángel Gutiérrez de Anda who was not only my thesis advisor but also my friend. Many thanks for all his advises and his limitless patience shown to me during this time.

Algo sobre la muerte del Mayor Sabines
(Jaime Sabines)

XV

Papá por treinta o por cuarenta años,
amigo de mi vida todo el tiempo,
protector de mi miedo, brazo mío,
palabra clara, corazón resuelto,

te has muerto cuando menos falta hacías,
cuando más falta me haces, padre, abuelo,
hijo y hermano mío, esponja de mi sangre,
pañuelo de mis ojos, almohada de mi sueño.

Te has muerto y me has matado un poco.
Porque no estás, ya no estaremos nunca
completos, en un sitio, de algún modo.

Algo le falta al mundo, y tú te has puesto
a empobrecerlo más, y a hacer a solas
tus gentes tristes y tu Dios contento.

In memory of Dr. De Anda

Abstract

In some applications, filtering circuits whose response have a short transient time are necessary. In the classical approach of filter design, the designer has no much options to manipulate the filter characteristics in the time domain because the design formulae affects the frequency-domain specifications. Parameter-varying filters posses as a chief characteristic their short transient time duration, which makes them suitable for applications in which speed is a mandatory need. The short transient is the result of a temporary variation of one or more of their descriptive parameters. Despite this advantageous feature, that has been demonstrated with solid theoretical analyses, there is still a bottle-neck in the practical implementation of these kinds of filters.

In this work, an implementation of a parameter-varying filter based on switched-capacitors (SC) is presented. This implementation is a second-order low pass SC filter on a FPAA (Field-Programmable Analog Array) development board. Theoretical aspects related to this implementation are reported, in particular stability and homogeneous response analysis. Experimental results of the parameter-varying implemented filter are presented and compared with the time-invariant counterpart.

In summary, the theoretical analysis and the experimental results support the feasibility of implementing parameter-varying filters with switched-capacitor techniques.

Abstract

Resumen

En algunas aplicaciones, es necesario que los circuitos de filtraje presenten una respuesta cuyo tiempo transitorio sea de corta duración. En los filtros clásicos, dado que el diseño se hace en el dominio de la frecuencia, se cuenta con poca libertad para manipular sus características en el dominio del tiempo pues sus ecuaciones de diseño inciden sólo en las especificaciones del dominio de la frecuencia. Los filtros con parámetros variantes en el tiempo son una clase de filtros cuya principal característica es la corta duración del tiempo transitorio en su respuesta, producto de una variación temporal de uno o más de sus parámetros descriptivos. Aunque se han hecho varios estudios sobre este tópico, existen pocas implementaciones físicas de esta clase de filtros.

En este trabajo se presenta una implementación electrónica de un filtro con parámetros variantes en el tiempo basado en circuitos con capacitores conmutados. La implementación presentada aquí es de un filtro pasabajas de segundo orden hecha en una tarjeta de desarrollo del tipo FPAA (Field-Programmable Analog Array). Junto con los resultados de las pruebas hechas, se muestra analíticamente que el filtro implementado es exponencialmente estable y que su respuesta transitoria es más corta en comparación con los filtros lineales invariantes en el tiempo.

Finalmente, es preciso apuntar que el análisis teórico y los resultados experimentales permiten demostrar la factibilidad de realización de esta clase de filtros con capacitores conmutados.

List of Figures

LIST OF FIGURES

Chapter 1

Introduction

A filter can be defined as an electric or electronic circuit which allows to discern the spectral content present in a given frequency band from another. This kind of circuits constitutes an essential part of communication and control systems [1]. They can be classified according to diverse characteristics such as the band of frequencies they can manage, the kind of elements they are composed of, the type of signal they can handle and so on.

Another classification of filters is made attending to nature of their descriptive parameters. A descriptive parameter of a filter is a quantity that define its behavior. This quantity is generally defined in the frequency domain. Quality factor, gain and cut-off frequency are examples of a descriptive parameter. If the descriptive parameters of a given filter are subject to variations in their values, it is possible to distinguish between time-invariant filters and parameter-varying filters. The study of parameter-varying filters is the topic covered in this thesis.

A parameter-varying filter (or PVF, for short) is a system whose transient behavior is reduced in duration through the temporary variation of one or more of its descriptive parameters compared against the response of a linear time-invariant (LTI) filter with the same input-output transfer function in the frequency domain. This subject has been studied from the mid-nineties. Since the seminal work presented in [2], there are many works reported in the literature dealing with this topic (see, for instance, [2–11]). Most works has been focused on analyzing different types

of filters subject to the variation of some of its parameters. In this context, these works have focused their efforts treating and analyzing parameter-varying filters from a theoretical point of view. Other works have been focused in other directions, that is, they have been focused on implementations of this type of filters. It should be emphasized that there have been few works reported in the aforementioned sub-field.

This chapter is organized as follows. In Section 1.1, classical filter design is reviewed. This review is required in order to show the main differences between classical filters and parameter-varying filters. In Section 1.2, the motivation behind this work is fully expounded. Finally, in Section 1.4, it is mentioned how this document is organized.

1.1 Design of Classical Filters

Before discussing what is a parameter-varying filter, it is necessary to talk about filters based on LTI systems. For reference purposes, these filters will be called classical filters in this work. This is necessary in order to obtain a clear comparison between a PVF and its LTI counterpart and to know specific advantages of the former.

The classical approach of filter design consists in a set of very well-known steps. It is based in the approximation of its idealized transfer function by a polynomial. This filter design starts with specifications in the frequency domain. The most common approximations are Butterworth, Chebyshev, Bessel and elliptical. As it was stated before, all the design process consists in approximating the filter performance in the frequency domain. Time domain is thus relegated to a secondary plane. In general, it is not possible to modify the time-domain performance of a filter which has been designed with a classical approach, i.e. if the filter under consideration has been synthesized as a LTI system.

In some applications, the time domain response of the filter has to be taken into account to ensure a good performance. An example of them is the filter used in an static automatic catchweighing instrument [12]: in a production line, products must be placed consecutively on a scale pan and weighted. However, to get a reliable measurement obtained by the instrument, it should be obtained when the transducer

used to estimate the weight of a given system is already in its equilibrium state. For this aim, a low pass filter may be used to supress the transient behavior generated by the transducer as such. Given that the catchweighting instrument must attain a given rate of measurements per unit of time (which is ultimately imposed by the speed at which the production line operates), a fast operation of the low pass filter is highly desirable. In this case, it can be seen that the filter response in the time domain is an important factor and cannot be neglected. Therefore, filters whose transient time response are shorter than the offered by classical filters are preferred.

1.2 Motivation Behind This Work

As it was mentioned earlier, there are few works on the implementation of continuous-time parameter-varying filters. An implementation of a first-order parameter-varying filter based in the static and dynamic translinear principles of transistors was presented in [13]. A general strategy for the implementation of these filters using operational transconductance amplifiers (OTAs) and capacitors is presented in [14]. In [15], a parameter-varying low pass filter based on current mirrors implemented in a 0.5 μm CMOS technology is reported. In [16], a parameter-varying notch filter implemented in a 0.35 μm CMOS technology is introduced. Finally, two circuits implementing, respectively, a first-order and a second-order parameter-varying low-pass filters are presented in [17]. These circuits were implemented with discrete components.

In this thesis work, an implementation of a discrete-time parameter-varying low pass filter is shown. The proposed filter is based on switched-capacitor circuits. This kind of circuits have a distinctive feature, namely, the possibility of building circuits which may be readily configured through the use of arrays of switches and capacitors. It is shown from an analytical point of view that the discrete-time implementation of the aforementioned class of systems is possible. It is also shown that a parameter-varying filter can be physically implemented with switched-capacitor circuits without losing its stability properties.

1.3 Objective of this Work

The primary objective of this work is to analyze the feasibility of the implementation of parameter-varying low pass filter with switched-capacitor circuits while keeping its stability properties. In order to achieve this objective a filter prototype build in a design framework constituted by a Field-Programmable Analog Array is analyzed and tested.

1.4 Structure of this Document

This thesis is organized as follows.

- Chapter 2 contains preliminary concepts about systems theory. Definitions and classifications of systems are also presented in this chapter. Notions of stability and state-space variables are treated too.
- In Chapter 3, principles of switched-capacitor circuits are commented. The fundamentals behind the operation of switched-capacitor filters and the main approach used to design them are also mentioned.
- Chapter 4 introduces the theory behind parameter-varying filters from an evolutionary point of view. This review comprises the first works reported in the literature to the latest advances in this subject. In this chapter, the analysis of stability and the homogeneous response of a parameter-varying filter are also treated.
- In Chapter 5, the synthesis of parameter-varying filters with switched-capacitor circuits is presented.
- In Chapter 6, tests of a parameter-varying filter implemented on a Field Programmable Analog Array (FPAA) are presented. The methodology used and the results obtained are also discussed.
- After Chapter 6 the conclusions of this work are presented.

- At the end of this thesis, four appendices are included. Appendix A describes the AN231K04-DVLP3 development board. This board contains the AN-231 FPAA chip. Appendix B presents and describes the Chip-Kit development board. It contains a 32-bit microcontroller which was used to generate the control signals used to induce the variation of parameters in the switched-capacitor filter. Appendix C discusses how to generate the data sequences used to reconfigure the FPAA in order to change a specific circuit parameter on the fly. Finally, Appendix D shows the source code used in this work.

Chapter 2

Dynamical Linear Systems

2.1 Introduction

Nature, in particular a natural phenomenon, can be modeled with a mathematical equation. Depending on the analyzed phenomenon, its model can show a high degree of complexity. However, in some cases it is only necessary analyze the phenomenon under consideration in some restricted conditions. By doing that, its model can be simplified considerably. In this case, concepts as linearity and time-invariance may come into play. In order to study and analyze a natural phenomenon, it is preferable to represent it by an abstract object. This mathematical representation is called a system. A system is a convenient form of abstraction which may be used to analyze an arbitrary phenomenon.

Considering the latter, concepts of systems theory will be discussed in this chapter. Definitions, terms and notions which will be used in the remaining chapters of this thesis are introduced. It is important to mention that some notions are treated in this chapter without entering into many details because they are put here only to introduce the reader to the aforementioned topics and with the aim that this thesis could be self-contained. The reader interested in learning in more detail the concepts dealt in this chapter may review the references at the end of this document, specially [18] and [19] for systems theory.

This chapter is organized as follows: In Section 2.2 concepts about dynamical linear systems will be introduced. The properties of this class of systems are discussed as well. The response of continuous-time second-order LTI systems is also considered.

2.2 Definition and Classification of Systems

A basic definition of system is a set of objects which acts in conjunction to obtain a certain goal [18]. According to [19], a system is an interconnected set of abstract objects named its components. These components may be oriented, not oriented, finite in number or not and each of them may be associated with a certain number of terminal variables.

In this work a system is defined as in [20]:

A system is a triad of sets

$$\sum = \{T, W, B\}$$

where:

- T stands for a set of time instants, or even time intervals, in which the behavior of the system is observed.
- W represents the set of signals which are used to characterize its behavior. This set is also called alphabet of signals.
- B is a set containing all the possible behaviors of the system described in the set of signals.

This definition is taken from a mathematical point of view. It may be applied to any class of systems (economical, physical, biological, etc.). T refers to the time interval the system is modeled or analyzed. This set defines whether the behavior of a given system is defined either in the continuous-time or in the discrete-time domain. W is a set which contains all the signals relevant to the dynamical system, that is, variables which are being studied. This set contains input and output signals (variables which interact with the system's environment), as well as state variables.

Finally, B represents a family of time trajectories and takes its values from W. In fact, B contains all the behaviors which the system can generate. This set is defined in terms of the relationships between the signals contained in W. These relations are the consequence of a given set of laws which define as such the operation of the system.

Classifications of Systems

Taking into account different criteria, a system can be classified in different groups or categories. In the following, some classes of systems are introduced. This list is not exhaustive but it shows the most important classes studied in systems theory.

- **Linear Systems.** A system is considered as linear if it exhibits the homogeneity and additivity properties. If u_1 and u_2 are two different inputs of a system S and y_1 and y_2 are the outputs for those inputs respectively, then S possesses the additivity property if the output of S is $y_1 + y_2$ when the input is $u_1 + u_2$. In addition to that, if α is a real number, S exhibits the homogeneity property if the output of S is αy_1 when its input is αu_1.

- **Non-linear Systems.** A nonlinear system is one which does not exhibit the homogeneity and additivity properties found in a linear system.

- **Static System.** A static system is also called a *memoryless* system because its main feature is that its response depends only on its input in a specific instant of time. The mathematical model of this kind of systems is an algebraic equation of the form $\mathbf{y}(t) = \mathbf{f}(\mathbf{u}(t))$, where $\mathbf{u}(t)$ and $\mathbf{y}(t)$ stand, respectively, for the sets of inputs and outputs to the system. A good example of a static system is an electronic circuit which is only composed of resistors.

- **Dynamical System.** Dynamical systems are one of the most important classes of systems. In a first approximation, they can be viewed as the counterpart of static systems. These systems possess a *memory*. The output of a dynamical system in a specific time instant depends not only on the value of its inputs but also on the value of latent variables of the system which define its

state. The mathematical model of a dynamical system is a differential equation in the continuous-time domain or a difference equation in the discrete-time domain.

- **Time-Invariant System.** A system is considered as time-invariant if its output depends only on its input (or on its inputs and state variables in the case of dynamical systems) and not on the instant of time that this input is applied. Put it in a simplistic manner, the output (or outputs) of a time-invariant system for a given input at a specific time instant will be the same if this input is applied at another time instant. Casted in mathematical terms, if a system S has an output $y_1(t)$ when its input is $u_1(t)$, then S is a time-invariant system if $u_1(t+\tau)$ produces $y_1(t+\tau)$ for the same set of initial conditions. Constant τ is a real number and represents a translation in time.

- **Time-Varying System.** A time-varying system is one whose output depends not only on its input (or on its inputs and state variables in the case of dynamical systems) but also on the instant of time that this input is applied. In the case of a continuous-time dynamical system, its mathematical model is a differential equation with time-dependent coefficients.

As it was mentioned earlier, this list does not include all the existing systems. For example, it did not include stochastic and deterministic systems or causal and non-causal systems. Again, the reader is encouraged to review the references [18] and [19] for a more complete classification.

2.3 Analysis of Dynamical Systems

In the following, a description of behavior of dynamical linear systems will be presented. Firstly, the time-invariant case is shown.

LTI Systems

The mathematical model for a dynamical LTI system is a differential equation with constant coefficients. This kind of systems is customarily studied in undergraduate courses dealing with dynamical systems.

Usually, in order to discuss the basic properties of LTI systems and show their characteristics, second-order systems are often considered. The main reason is that a higher-order system can be represented by a set of second-order systems connected in cascade (if the system considered is of even order) or by a set of second-order systems and a first-order system connected in cascade (if the system is of odd order). Another reason is that a simplified representation of a higher-order system can be formulated in terms of a second-order or a first-order system with the pole-dominant concept [18].

A second-order LTI system can be represented by the following transfer function in the Laplace domain

$$H(s) = K\frac{s^2 + as + b}{s^2 + cs + d}. \tag{2.1}$$

In the previous expression a, b, c, d and K are real constants. Similarly, second-order systems can also be described in terms of their poles and zeros as given below

$$H(s) = K\frac{(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)}. \tag{2.2}$$

Either expression (2.1) or expression (2.2) can be used to model the behavior of an arbitrary second-order LTI system.

In general terms, the transient response of a second-order system to a unit step signal has the shape shown in figure 2.1. This response can be characterized in the time domain through different measures. Some of them are defined below.

- **Overshoot**. It is the difference between the magnitude of the output signal in the steady-state regime and its maximum value attained during the transient response. It is usually denoted by letter M and it is usually expressed in terms of a percentage.

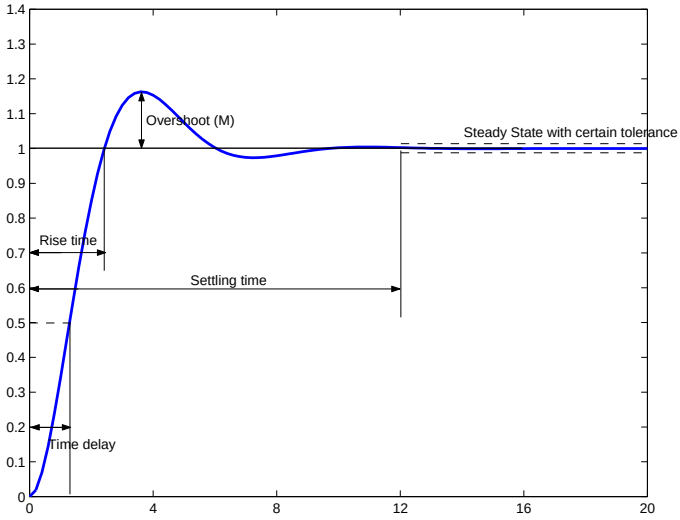


Figure 2.1: Response of a second-order low pass filter of the form given in Equation (2.3) to a unit step signal.

- **Rise time**. It is the time required by the response of the system to rise from 0% to 100% of the magnitude of the steady-state response. In some cases, percentages from 5% to 95% are considered instead.

- **Settling time**. It is the time needed by the response of the system to reach the steady-state and stay in there with a given tolerance. Tolerances of 2% or 5% of the steady-state value are usually considered in control applications. In the design of operational amplifiers for switched-capacitor filters, tighter tolerances (0.1% or even 0.01%) are taken as design constraints.

From this list, the settling time is the most important time-domain specification in the context of parameter-varying filters. This concept will be clarified later in this document.

Because in this work the reduction of the duration of the transient response of a filter is of particular interest — due to reasons which will be clear in Chapter 4 — the time-domain response of a second-order low pass filter to a unit step input will be now considered. The transfer function of a continuous-time second-order LTI low pass filter in the Laplace domain is given by:

$$H(s) = \frac{K\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{2.3}$$

where:

K is the gain of the filter.

ω_n is the frequency of the system's undamped oscillations.

ζ is the damping ratio.

The transient response of (2.3) depends both of ω_n and ζ. Because of that, diverse types of responses can be identified:

- When $\boldsymbol{\zeta = 0}$, the filter's response is called *undamped*. In this kind of response, the filter output will not tend to a finite value and its poles are located on the imaginary axis.

- When $\boldsymbol{0 < \zeta < 1}$, the filter's response is called *underdamped*. The shape of this response is characterized by decaying oscillations in amplitude which die out when the steady-state is reached after a certain time.

- When $\boldsymbol{\zeta = 1}$, the filter's response is called *critically damped*.

- When $\boldsymbol{\zeta > 1}$, the filter's response is called *overdamped*. There is a slight similarity between an overdamped response and the response of a first-order system. Both of them do not have an overshoot.

Figure 2.2 shows different responses of second-order LTI low pass filter described by the transfer function given in expression (2.3) to a unit step for different values of ζ, while Figure 2.3 shows the response of the filter for different values of ω_n. It is observed from Figure 2.2 that the overshoot depends strongly on the damping ratio: the greater the ζ, the lower the overshoot. At the same time, the time required by the filter in order to reach the steady-state is related to the undamped natural frequency: the greater the ω_n, the shorter the transient time. As it will be mentioned later, the theory behind the operation of parameter-varying filters was developed taking advantage of these ideas.

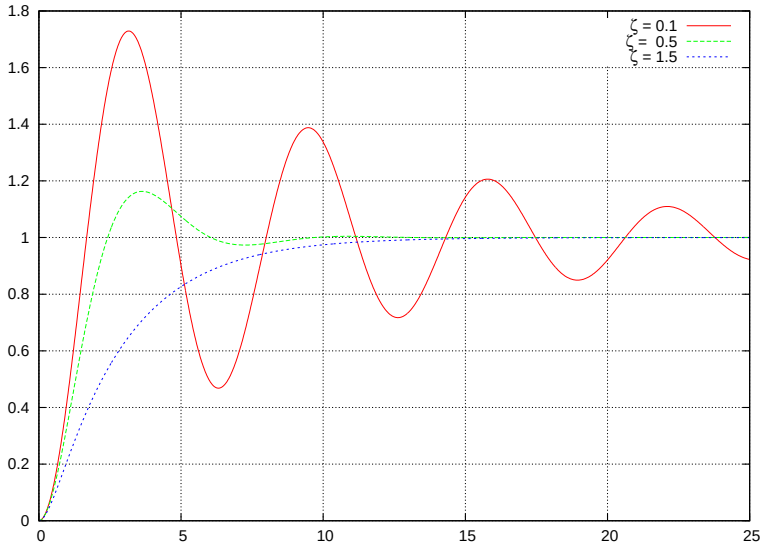


Figure 2.2: Responses of the second-order system described by the transfer function 2.3 for different values of ζ with $\omega_n = 1$ rad/s and $K = 1$.

Linear Time-Varying Systems

Linear time-varying (LTV, for short) systems as well as LTI systems may have more than one input/output. For these cases, it is preferable to model them by state-space equations. This topic will be covered in the next subsection. Therefore, it only will be stated some differences between the variant and the invariant case.

- Linear time-varying systems cannot be easily represented with a transfer function as in the LTI case. Therefore, it is more complicated to find the response of a LTV system.

- Albeit the Laplace transform may be applied in the analysis of relatively simple LTV systems, this tool cannot be used in the general case.

Finally, it is necessary to state that, in most cases, the response of an LTV system can only be found by means of a digital computer [19].

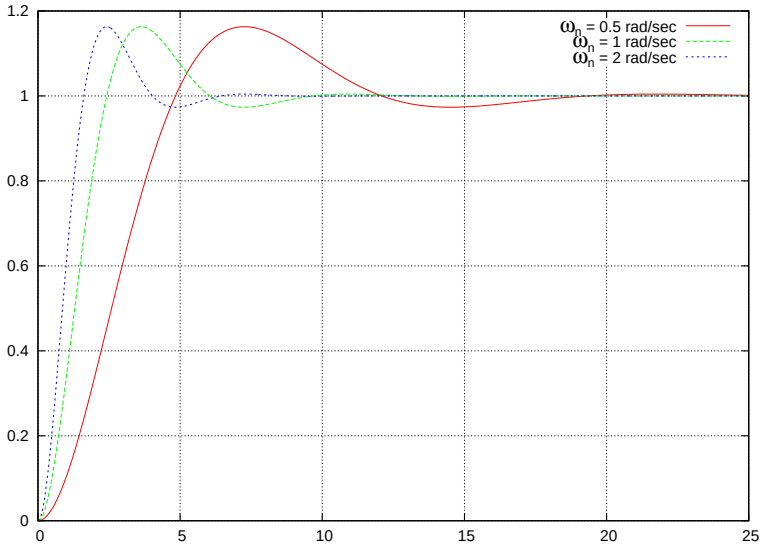


Figure 2.3: Responses of the second-order system described by the transfer function 2.3 for different values of ω_n when $\zeta = 0.5$ and $K = 1$.

2.4 The State-Space Representation

As it was previously mentioned, the mathematical modeling with scalar differential equations of a given system is not useful when its complexity increases, that is, when there is more than one input and more than one output. For this kind of systems, the state-space representation is a better approach to model them.

State-space representation of systems is a relatively new technique for their analysis and design. It is extensively used in the so-called modern control theory and it has been developed since the decade of the 60's [18]. This representation is based on the idea of state.

The concept of state is the core idea in state-space modeling. In informal terms, a state can be viewed as a picture of the system in a specific instant of time. It gives information about the system. In electronic systems, a state variable is one of the system's physical variables (most commonly charges or fluxes) which are representative of its dynamics. Therefore, the correct choice of variables is an important issue

in state-space modeling. It is necessary to emphasize that a formal representation of states is done in terms of vectors.

Assuming that for a particular system there is a judicious choice of state variables, it is possible to obtain a set of states from different time instants. Thus, it is achievable to analyze a system by studying its evolution of states through time instants starting from an initial state.

A mathematical model for the state-space representation of a LTV system with an arbitrary number of inputs and outputs is given below

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t) \tag{2.4a}$$

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t), \tag{2.4b}$$

where:

- $\mathbf{x}(t)$ is a n-dimensional state vector.
- $\mathbf{y}(t)$ is the output vector.
- $\mathbf{u}(t)$ is the input vector.
- $\mathbf{A}(t)$ is a $n \times n$ matrix. This matrix is called the state matrix.
- $\mathbf{B}(t)$ is the input matrix.
- $\mathbf{C}(t)$ is the output matrix.
- $\mathbf{D}(t)$ is called the coupling matrix between the inputs and outputs.

The state vector is n-dimensional. Each of the components of this vector contains a state variable. It means that this system has n variables. Of course, matrices and vectors in equation (2.4) have proper dimensions according to the number of states, inputs and outputs.

In the particular case that $\mathbf{u}(t) = 0$, the state equation becomes:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t). \tag{2.5}$$

This equation is known as an homogeneous state equation. As it will be shown in Subsection 2.5, equation (2.5) is important in order to analyze some stability conditions for system (2.4).

The Transition Matrix

It was previously stated that the homogeneous response of an equation (2.5) is important in order to analyze the stability properties of system (2.4). In the following, an important concept related to the latter will be developed: the transition matrix.

Definition 1 *[19] The transition matrix $\Phi(t, t_0)$ associated to equation (2.5) is a $n \times n$ matrix which is the solution of the equation*

$$\frac{d}{dt}\mathbf{\Phi}(t, t_0) = \mathbf{A}(t)\mathbf{\Phi}(t, t_0).$$

Then, the solution to equation (2.5) subject to the initial condition $\mathbf{x}(t_0) = \mathbf{x_0}$ is given by:

$$\mathbf{x}(t; \mathbf{x_0}, t_0) = \mathbf{\Phi}(t, t_0)\mathbf{x_0}.$$

The properties of transition matrix $\mathbf{\Phi}(t, t_0)$ are given below.

- $\mathbf{\Phi}(t_0, t_0) = \mathbf{I}$
- If $\mathbf{A}(t)$ is constant, *i.e.* $\mathbf{A}(t) = \mathbf{A_C}$, then $\mathbf{\Phi}(t, t_0) = \exp\left[\mathbf{A_C}(t - t_0)\right]$
- If for all t, $\int_{t_0}^{t} \mathbf{A}(\tau)d\tau$ and $\mathbf{A}(t)$ commute[1], then $\mathbf{\Phi}(t, t_0) = \exp\left[\int_{t_0}^{t} \mathbf{A}(\tau)d\tau\right]$
- $\det \mathbf{\Phi}(t, t_0) = \exp\left[\int_{t_0}^{t} \alpha(\tau)d\tau\right]$, where $\alpha(\tau) = \operatorname{tr} \mathbf{A}(\tau) = \sum_{i=1}^{n} a_{ii}(\tau)$

[1]This is a special property for transition matrices. This property is accomplished when $\mathbf{A}(t)\int_{t_0}^{t} \mathbf{A}(\tau)d\tau = \left(\int_{t_0}^{t} \mathbf{A}(\tau)d\tau\right)\mathbf{A}(t)$ for all t. Constant matrices and diagonal matrices with time-varying coefficients fulfill this condition.

- $\mathbf{\Phi}(t_3, t_2)\ \mathbf{\Phi}(t_2, t_1) = \mathbf{\Phi}(t_3, t_1)$

The last property is known as the composition property and because its importance int the context of this work, it will be explained with more detail in the following lines. This will be made with the aid of Figure 2.4.

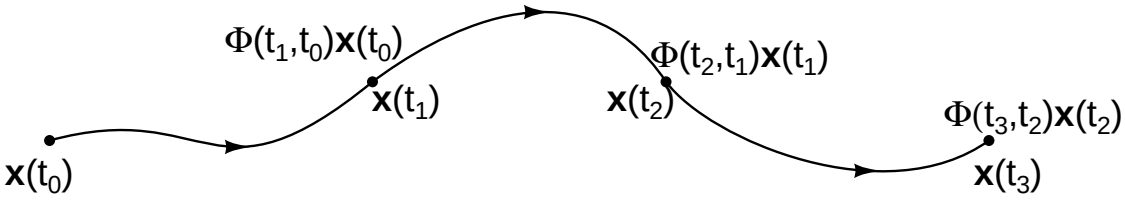


Figure 2.4: A graphic explanation of the evolution of a linear system response in terms of its transition matrix.

Figure 2.4 shows a curve which represents the evolution in time of a solution of equation (2.5) — whose transition matrix is $\mathbf{\Phi}(t, t_0)$— for the initial condition $\mathbf{x}(t_0)$, from t_0 to t_3, where $t_0 < t_1 < t_2 < t_3$. It can be observed that the solution in a given time instant can be expressed in terms of the transition matrix and previous states lying in a given trajectory. For instance $\mathbf{x}(t_i)$ may be expressed as

$$\mathbf{x}(t_i) = \mathbf{\Phi}(t_i, t_{i-1})\mathbf{x}(t_{i-1})$$

where

$$\mathbf{x}(t_{i-1}) = \mathbf{\Phi}(t_{i-1}, t_{i-2})\mathbf{x}(t_{i-2})$$

Because of the latter, in a more general form

$$\mathbf{x}(t_i) = \left(\prod_{j=0}^{i-1} \mathbf{\Phi}(t_{i-j}, t_{i-j-1}) \right) \mathbf{x}(t_0).$$

From another point of view, this property implies that the transition matrix maps a new state $\mathbf{x}(t_i)$ from a previous state $\mathbf{x}(t_{i-1})$ of a homogeneous system represented by Equation (2.5).

2.5 Stability Concepts

The study of the stability properties of a given system is an important topic in the field of systems theory. For this reason, different concepts and definitions of stability will be developed in the following lines.

Stability in the sense of Lyapunov

For a dynamical system — which can be represented by Equation (2.4a) — an equilibrium point, say $\mathbf{x_{eq}}$, can be qualified as a stable point if for *small* perturbations, the free motion around this point is *small* too. This concept is depicted in Figure 2.5. In that figure, it can be observed that $\mathbf{x}(t)$, which is the trajectory followed by the system after a small perturbation, remains into a defined circular region.

According to [19], the formal definition of stability in the sense of Lyapunov is as follows.

Definition 2 *[19] A system is said to be zero-input stable in the sense of Lyapunov, if for any t_0 and any $R > 0$, there is a $r > 0$ depending on R and t_0 such that*

$$\|\mathbf{x_0}\| < r \Rightarrow \|\mathbf{x}(t; \mathbf{x_0}, t_0)\| < R \qquad \forall t \geq t_0.$$

Asymptotic Stability

Definition 3 *[19] The zero state $\mathbf{x} = \mathbf{0}$ is said to be asymptotically stable if:*

- *It is stable in the sense of Lyapunov.*
- *For any t_0 and for any $\mathbf{x_0}$ sufficiently close to $\mathbf{0}, \mathbf{x}(t; \mathbf{x_0}, t_0) \to \mathbf{0}$ as $t \to \infty$.*

Exponential Stability

Definition 4 *[19]*

A system is called exponentially stable if there exist finite constants γ and λ such that for any t_0 and x_0 the corresponding solution satisfies the following relation:

$$\|\mathbf{x}(t)\| \leq \gamma \exp^{-\lambda(t-t_0)} \|\mathbf{x_0}\|, t \geq t_0.$$

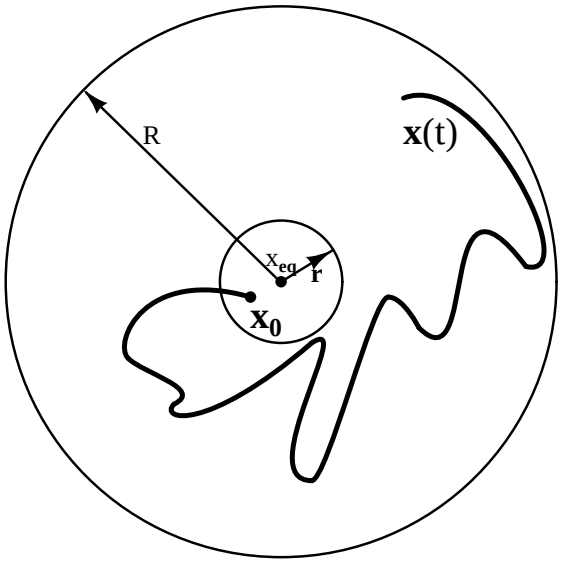


Figure 2.5: Stability concept in the sense of Lyapunov

The latter definition leads to another one which can be expressed in terms of the transition matrix of the system under consideration as indicated below

$$\|\mathbf{\Phi}(t,\tau)\| \leq \gamma \exp^{-\lambda(t-\tau)} .$$

Bounded-Input Bounded-Output (BIBO) Stability

According to [19], the stability of a system from the point of view of its inputs and outputs and taking into account equation (2.4a) is expressed as follows.

Definition 5 *[19] A system whose output is* $\mathbf{x}$ *and whose input is* $\mathbf{u}$ *is said to be* BIBO stable *if for all* t_0 *for all initial states* $\mathbf{x_0}$ *and for all bounded inputs* $\mathbf{u}_{[t,\infty)}$*, the output* $\mathbf{x}(t; \mathbf{x_0}, t_0; \mathbf{u}_{[t,\infty)})$ *is bounded on* $[t, \infty)$*.*

In the case of LTI systems, the BIBO stability conditions is accomplished by making sure that their poles are in the left half plane of the complex plane. However, in the case of LTV systems of the form given in Equation (2.4), other conditions must be fulfilled. These conditions are:

- The system represented in Equation (2.5) must have exponential stability.
- The entries of matrices $\mathbf{A}(t)$, $\mathbf{B}(t)$, $\mathbf{C}(t)$, and $\mathbf{D}(t)$ must be bounded.

The proof of the validity of the last assertions can be found in [3].

Chapter 3

Switched-Capacitor Circuits

3.1 Introduction

In VLSI circuit design there are some techniques for the implementation of a given analog/digital circuit which are aimed to achieve significant improvements in a particular design constraint such as area, power consumption, etc. The switched-capacitor circuit technique is one of them.

Switched-capacitor networks were originally conceived to emulate the behavior of a resistor. The most simple method which may be used by a circuit designer for the implementation of a resistor in an integrated circuit is to use polysilicon. However, the resulting device may require an excessive amount of die area. Moreover, given the fact that the manufacture parameters of a CMOS process usually exhibit a great degree of variability (not only between process runs but also in the same wafer in which identical circuits are fabricated), it is not possible to control with precision the resistance of a polysilicon resistor. As it will be seen in this chapter, the switched-capacitor technique only needs a capacitor and a set of switches for the implementation of a resistor with a precise resistance with much less area used from the silicon die unlike the resistors implemented with polysilicon.

The switched-capacitor technique brings other advantages than the above mentioned. These advantages will be later discussed in this chapter which is organized as follows. In Section 3.2 essentials of switched-capacitor circuits are presented whereas

the necessary theory of filters designed with switched-capacitor circuits is dealt with in Section 3.3.

3.2 Fundamentals of Switched-Capacitor Circuits

3.2.1 Switched-Capacitor Resistor Simulation

As it was said earlier, the switched-capacitor circuit technique can be viewed as a way to emulate a resistor through the usage of a capacitor and a set of switches whose state is periodically changed in time. Some switched-capacitor networks used for the emulation of resistors are shown in Figure 3.1. As it can be seen, a resistor implemented with switched-capacitor networks may take different forms. Each of these switched networks possesses different characteristics: from smaller number of switches or capacitors to greater noise immunity at the expense of more circuit elements. In the same figure, clock signals ψ_1 and ψ_2 are represented as well. These clock signals play an important role on emulating the resistor and the frequency of the signal that switched-capacitor circuits can manage. These clock signals define the alternate switching instants of the switches present in each of the two-port networks shown before. Most references use the Greek letter Φ to denote a clock signal. However, the symbol ψ will be used in this document instead because the former is employed to designate the transition matrix already considered in Section 2.4.

Without loss of generality, the principle of operation of switched-capacitor circuits will be briefly examined with the aid of the circuit shown in Figure 3.1d. This two-port network is the most common implementation used switched-capacitor filter design for a resistor because of its parasitic-insensitive characteristic.

In the time domain, the analysis begins with all switches open and the capacitor C discharged at time instant t_0. The state of switches 1 and 4 (open or close) is controlled by clock signal ψ_1 whereas the state of switches 2 and 3 is controlled by clock signal ψ_2. As shown in Figure 3.2, ψ_1 and ψ_2 are non- overlapping. Consequently, switches controlled by different clocks will not be closed at the same time.

When switches are opening and closing according to the clock signals previously mentioned, capacitor C charges and discharges. This leads to a charge transfer during

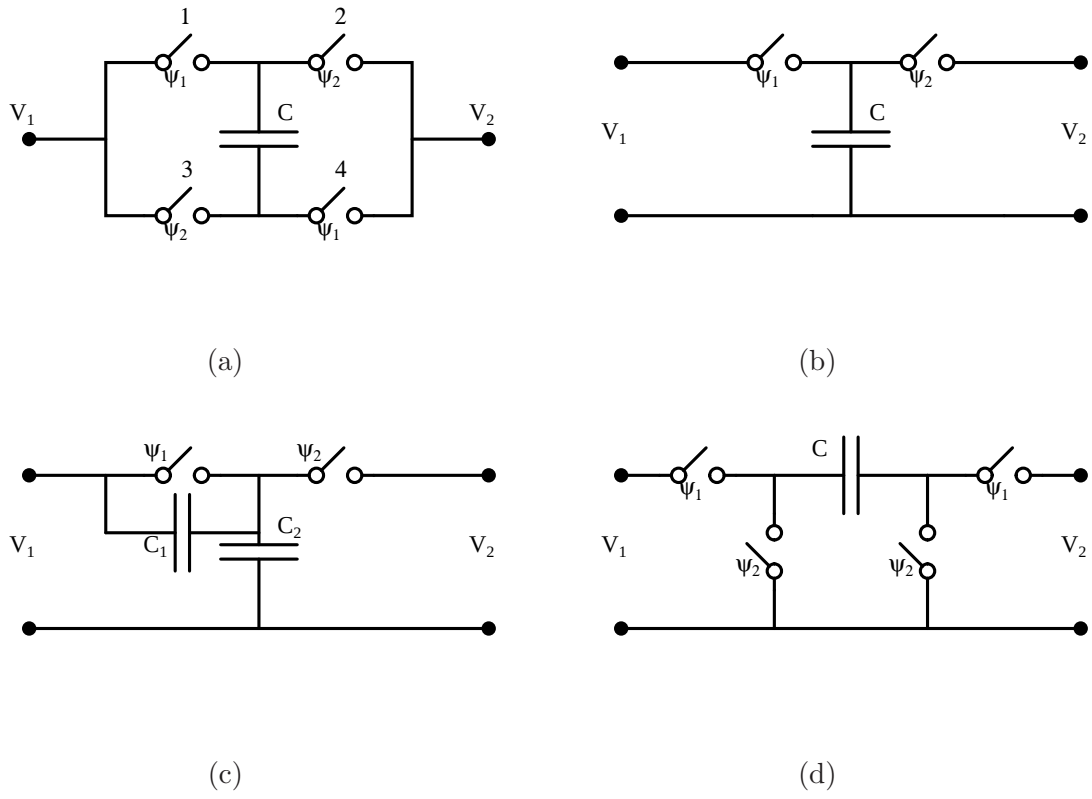


Figure 3.1: Two-port networks used for the emulation of resistors based on arrays of switches and capacitors.

a finite time interval from the input port to the output port. The electrical current is defined as the rate of charge change Δq in a given time interval Δt as indicated below

$$I = \frac{\Delta q}{\Delta t}.$$

In the circuit shown in Figure 3.1d, a charge equal to Δq will be transferred from the input port to the output port in a finite time T. On the other hand, Δq can be expressed as $C(V_1 - V_2)$. Therefore, the current flowing from the input port to the output port may be written as

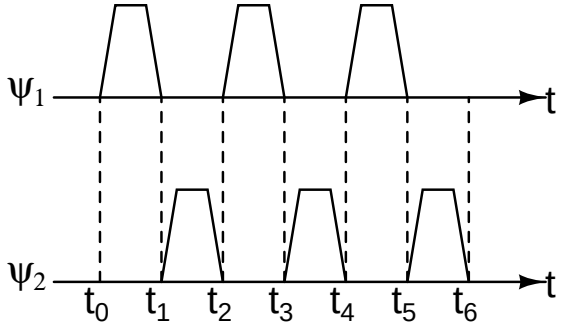


Figure 3.2: Clock signal.

$$I = \frac{C(V_1 - V_2)}{T}.$$

From this expression, an equivalent resistance placed between the input and the output ports of the switched network R_{eq} is readily identified as given below

$$R_{eq} = \frac{T}{C} = \frac{1}{Cf_{clk}}. \tag{3.1}$$

The equivalent resistance given in equation (3.1) can vary from one network to other. In order to find the equivalent resistance for these networks, the method applied above can be used.

3.2.2 Advantages of SC Circuits

It was mentioned above that the reduction in the area consumption in a silicon wafer is one of the improvements achieved by the switched-capacitor resistor emulation. However, this is not the only advantage attained with this technique. The other advantage of SC circuits is the capability to control their frequency response by simply adjusting capacitance ratios.

The fact that the frequency response may be formulated for switched-capacitor filters in terms of capacitance ratios is important and can be clarified if a simple first-order RC circuit is considered. In such a circuit, the time constant (usually denoted by τ) is an important parameter and is calculated by the product of R and

C. As in VLSI circuits the precision of the value of a given linear resistor is generally dependent of temperature, τ is also dependent on temperature because is directly related to the resistor's value. Therefore,

$$\frac{d\tau}{\tau} = \frac{dR}{R} + \frac{dC}{C}$$

since R and C are uncorrelated.

If the resistor is now substituted by a switched-capacitor equivalent of the form T/C_R, for instance, it is possible to conclude that the variation of τ is given by:

$$\frac{d\tau}{\tau} = \frac{d\left(\frac{C}{C_R}\right)}{\left(\frac{C}{C_R}\right)}$$

provided that T is considered as invariant or with high precision as occurs in the practice.

The latter equation shows that the precision of τ is dependent of a ratio of capacitances. In an integrated circuit, the accuracy expected for the absolute value of a resistor is usually inferior to the accuracy of the ratio of two capacitances (provided that different layout techniques are considered in order to improve the matching between C and C_R). Thus, the ratio of capacitances is preferred over resistances.

One final note is about the clock signal used in switched-capacitor circuits. According to [21], in switched-capacitors circuits, the clock frequency f_{clk} should satisfy the relation $f_{max} \leq f_{clk}/100$, where f_{max} is the highest signal frequency.

3.2.3 MOSFET Switches

In switched-capacitor circuits, switches are implemented with CMOS transistors. Switches in transmission gate configuration with both PMOS and NMOS are preferred. MOS transistor switches present some drawbacks which introduce errors in the signal processing. In order to complete the discussion about switched-capacitor fundamentals, the three main mechanisms that introduce errors when switches are implemented with MOS transistor are briefly presented in this section. These mech-

anisms are: channel charge injection, clock feedthrough and kT/C noise. The reader interested to study in deep into this issues should read references [22] and [21].

Charge injection is present when the MOS transistor switch is turned off. In this case, the charge present in the channel formed when the switch was turned on is discharged through both source and drain terminals. Since the signal output is connected to one of them, the charge injected to that terminal introduce an error at the output signal.

The clock feedthrough error introduced with MOS transistor switches refers to the error due to the clock signal which pass through the gate-source or gate-drain overlap capacitance to the output. Due to this phenomenon, the output port is formed by two parallel capacitors which modify the output signal.

Finally, as a MOS switch in on state can be seen as an RC circuit, thermal noise — also known as kT/C noise — is present. This noise introduces a voltage error at the output which value is approximately $\sqrt{kT/C}$.

3.3 Switched-Capacitor Filters

One of the most widely used application of switched-capacitor circuits is in the area of filters because of some interesting features such as those mentioned in Subsection 3.2.2 and Section 3.1. Different approaches can be taken in order to implement a switched-capacitor filters as indicated below:

- Replacement of resistors in active RC filters by equivalent switched-capacitor equivalents.
- Derivation of the switched-capacitor filter from analog continuous-time models via an appropriate transformation from s plane to the z plane.
- Simulation of digital filters in a cascade configuration.

Each of these approaches will be now explained.

The replacement of all the resistors in a RC active filter with switched-capacitor equivalents is the most simple method for obtaining a switched-capacitor filter. The

chief disadvantage of this approach is that not all the RC-filter realizations will be amenable to it because they cannot emulate all the characteristics of the original filter.

Another possibility to synthesize a switched-capacitor filter is to obtain a discrete-time transfer function $H(z)$ deriving it from a continuous-time transfer function $H(s)$. This derivation is done using either a bilinear or a LDI s to z transformation. Once having the transfer function in the z domain, the circuitry to be implemented comes from a signal flow graph derived from $H(z)$. The signal flow graph consists in integrators, adder circuits and delays.

Digital filter simulation by means of switched-capacitor is similar to the second approach. However, in this case, when the $H(z)$ has been obtained, the synthesis of the filter is made using techniques commonly applied in digital filter design [23].

Some filter topologies have been studied in several articles using different approaches. Examples of them are reported in [24, 25] and [26].

A block diagram of a filter based on switched-capacitor circuits is shown in Figure 3.3. From this figure, it can be seen that such a filter needs other companion blocks in order to do its task in comparison with its continuous-time counterpart. At the input, before the signal is processed by the SC filter, the continuous-time signal is filtered to frequencies lower than twice the clock sampling frequency. Then, the signal is sampled by the sample and hold circuit in order to be processed by the switched-capacitor filter. The output of the switched-capacitor filter is still sampled by a sample-and-hold circuit. Finally, a reconstruction filter is used to generate a continuous-time signal. This filter is usually a low-pas filter.

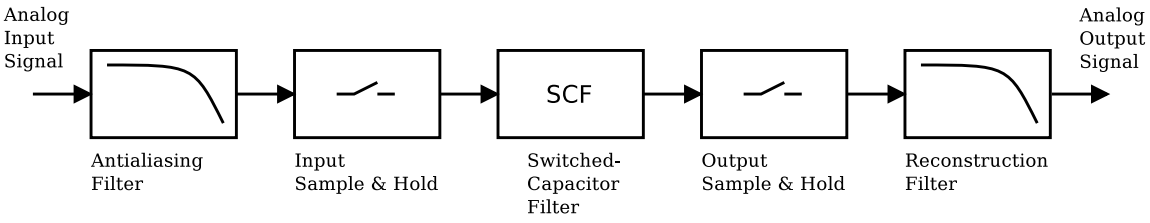


Figure 3.3: Switched-Capacitor filter block diagram.

A final remark must be made here. It was mentioned that for first-order switched-capacitor filters, their cut-off frequency is adjusted by the ratios of capacitors present on it. Therefore, it is clear that the transfer function implemented by switched-capacitor filters can be easily configured by changing the value of the appropriate capacitors. Due to this fact, the way of changing the characteristics of a switched-capacitor filters is fully exploited in this work in order to vary the parameters in a parameter-varying filter, as it will be seen later on.

Chapter 4

Parameter-Varying Filters

4.1 Introduction

In Chapter 1 a brief discussion was held on classical filters and how they are designed by means of their specifications in the frequency domain. These design methods of filters have different advantages and disadvantages. An evident advantage is their simplicity and that they consist in a very well-known steps. Additionally, they have been studied and used for a long time. Nevertheless, their main disadvantage is that the designer can not have a full control over the time-domain specifications of the filter. Therefore, in some applications where these specifications are important, the classical filter design approach is not convenient. That is where parameter-varying filters come into play.

In outline, a parameter-varying filter is a filter whose one or more of its descriptive parameter are changing with the time. This change must accomplish some requirements in order to maintain the filter characteristics in the steady state, as it will be shortly explained. The chief advantage of a parameter-varying filter is that they present a reduced transient response in comparison with an invariant-time filter.

In this chapter, the theory related to the core concepts of parameter-varying filters will be treated. The main characteristics of parameter-varying filters are also shown.

4.2 Typical Characteristics and Applications

At this point, it has been clarified that the more visible drawback of the classical approach of filter design is the inability to control the time-domain characteristics in applications where they are relevant, in particular the settling time. A possible solution of the problem of improving the settling time of a filter designed in the frequency domain is to increase its order. Another solution could be using another filter approximation. In any case, the behavior is improved, but only slightly.

Additionally, in Chapter 2 was mentioned that the larger the bandwidth of a low-pass filter is, the shorter its transient time is. Therefore, in order to modify the filter's transient response to make it shorter, it is necessary to modify the bandwidth of the filter making it larger. It will be shortly mentioned that one strategy used in the parameter-varying filter is based on this idea.

Parameter-varying filters is a relatively new topic. The first works on this field were made in the nineties. They arose from the necessity to improve the transient time of an invariant-time filter. Since then, several works have been done in this field, for example [2,4–6]. The main niche of possible use of parameter-varying filters is in instrumentation systems where long transient responses are seen as undesirable, thus a reduced transient time is preferred. The works mentioned above have been focused on the study of the stability of parameter-varying filters.

Other works are about the design of these classes of filters and their possible use in a system where large transients should be corrected. For example, a typical case occurs when a linear phase is needed from a filter, then a well-known technique to correct the phase is to cascade an all-pass filter with the first one. By doing this, the filter improves its phase characteristic, but, the settling time of the transient response increases. In [8–11] a parameter-varying filter is used in order to compensate an increment of this transient response.

Despite, the promising applications of parameter-varying filters, a bottle-neck in the field is regarded with the way these filters can be implemented. In fact, there are few implementations on parameter-varying filters. Some examples are [17], [14] and [15]. In this sense, this work focuses in the implementation of these kinds of filters.

A parameter-varying filter can be represented by means of a block diagram as in Figure 4.1. In this figure, $u(t)$ and $y(t)$ are the input and the output of the filter, respectively, whereas $p(t)$ represents the function that achieves the parameter variation.

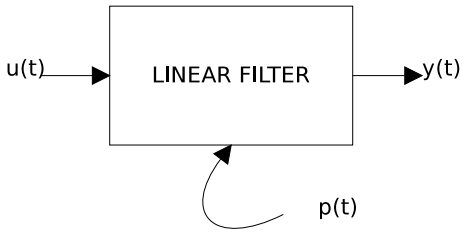


Figure 4.1: Block diagram of a parameter-varying filter.

The variation of the parameter should be applied in a significant change at the input of the filter in order to reduce its transient response. With this aim, a scheme of control has been proposed [3] in order to detect this sudden variation of the amplitude in the input $u(t)$, as shown in Figure 4.2.

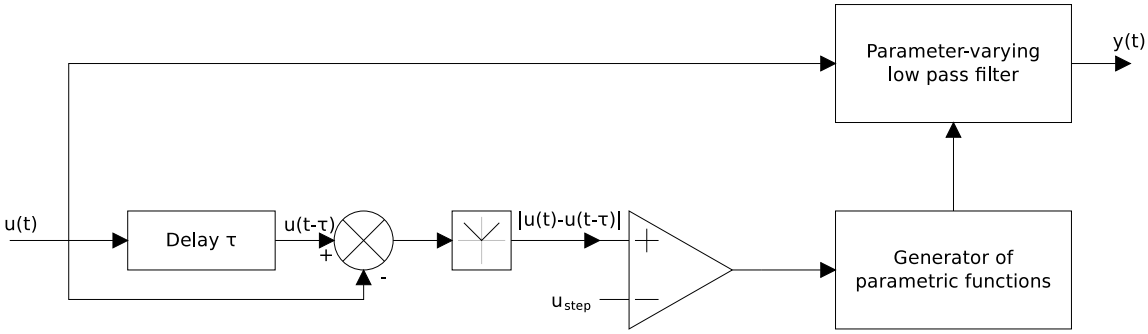


Figure 4.2: Amplitude based control scheme for a parameter-varying low pass filter .

It is necessary to note that with the introduction of the control scheme, the whole system becomes a nonlinear system. However, the analysis done in this work assumes that the parameter-varying filter is linear. This assumption is valid because the scheme control can be treated as a static nonlinear system which generates a control signal when a sudden variation is detected at the input signal [27].

4.3 Stability Conditions

In the case of invariant-time filters, the stability is ensured when their poles are located in the left half of the complex plane. However, for the parameter-varying filters the aforementioned condition can not be applied. However, the stability of parameter-varying filters is important in order to implement them as electronic systems [3].

The stability conditions are established with the goal of guaranteeing that: a) The filter must be asymptotically stable and b) The filter must be BIBO stable. Therefore, in the following, assertions related to the stability of parameter-varying filters are mentioned. Firstly, the stability conditions for a first-order low pass filter are shown. Furthermore, for the case of a second-order filter, these conditions are also presented by using two different approximations.

First-Order Filter

The first-order low pass parameter-varying filter can be modeled as

$$\dot{y}(t) + p(t)y(t) = p(t)u(t) \tag{4.1}$$

where $u(t)$ and $y(t)$ are the input and the output of the filter, respectively.

As Equation (4.1) is a linear first-order differential equation, the response is given by

$$y(t) = Ke^{-\int p(t)dt} + e^{-\int p(t)dt} \int e^{-\int p(t)dt} p(t)u(t)dt \tag{4.2}$$

From (4.2), it is possible to notice that the filter is exponentially stable if function $p(t)$ is positive. This will result clearer if it is noted that the homogeneous part of the response is an exponential function. Likewise, in order to assure that the filter is BIBO stable, $p(t)$ must be bounded.

In [6], these stability conditions are demonstrated, as well as the reduction of the transient time of a first-order low pass filter.

Second-Order Filter

In the following, the case of the second-order low-pass filter is analyzed. The first approximation proposed in [2] for a second-order parameter-varying filter is:

$$\ddot{y}(t) + 2\zeta(t)\omega_n(t)\dot{y}(t) + \omega_n^2(t)y(t) = \omega_n^2(t)u(t) \tag{4.3}$$

whose corresponding homogeneous equation is:

$$\ddot{y}(t) + 2\zeta(t)\omega_n(t)\dot{y}(t) + \omega_n^2(t)y(t) = 0 \tag{4.4}$$

In both equations (4.3) and (4.4), $y(t)$ and $u(t)$ are the output and the input of the filter.

Reference [4] shows that in order to demonstrate the stability of systems modeled as in Equation (4.3) is sufficient to prove the stability of its corresponding homogeneous representation as modeled in Equation (4.4). In order to find the stability conditions of the system with varying parameters is necessary to use the second method of Lyapunov [4]. This method is not explained here because it is not the purpose of this document. The reader interested in this topic can review references [4] and [19].

Thus, the stability conditions for the homogeneous second-order filter shown in Equation (4.4) are:

$$\omega_n(t) > 0 \tag{4.5a}$$

$$\zeta(t) > 0 \tag{4.5b}$$

$$|\dot{\omega}_n(t)| < |2\zeta(t)\omega_n(t)| \tag{4.5c}$$

The latter conditions assure that the response of the filter is asymptotically stable but these conditions does not guarantee that the response of the filter is BIBO stable.

The parameter variation of a parameter-varying filter must stabilize after passing the transient state so that its characteristic in the frequency domain remains at steady state. It was determined heuristically in [2] and [9] that functions $\zeta(t)$ and $\omega_n(t)$ should be exponentially varied. These ideas are depicted in Figure 4.3. In that figure, $p(t)$ represents the parameter which is varying.

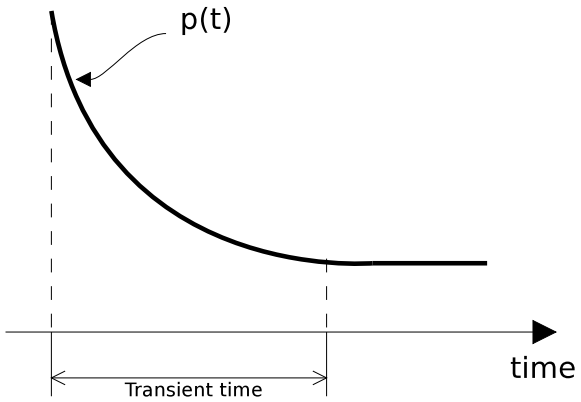


Figure 4.3: Variation of a parameter for a determined filter.

Another model for a second-order parameter-varying low pass filter is proposed in [3], that is obtained as the cascade of two first-order filter stages. In this form the problem of determining the stability conditions for the second-order filter can be reduced to determine the same conditions for two first-order filters. The resulting conditions are simpler than the conditions for the model proposed in Equation (4.3). Besides, BIBO stability is also fulfilled.

Chapter 5

Analysis of Switched-Capacitor Parameter-Varying Filters

5.1 Introduction

It was mentioned in Chapter 4 that the chief characteristic of parameter-varying filters is their short transient response compared with their linear time-invariant counterpart. In addition, the main conditions needed to obtain an asymptotically stable homogeneous response was established.

In this chapter the parameter-varying low pass filter implemented with switched-capacitor circuits is shown. The stability of this filter is analyzed. Additionally, its homogeneous response is obtained, too. It will be demonstrated that when some of its parameters are varied, the homogeneous response of this filter is shorter than the homogeneous response of a low pass filter with switched-capacitor circuits whose describing parameters remain constants. In order to develop this discussion, concepts previously considered are now used.

It is necessary to say that even though the analysis is done over a particular filter, the conclusions arrived to can be extrapolated to a general parameter-varying filter implemented with switched-capacitors circuits and whose parameter variation is done discretely as in the present case. This assertion obeys to the fact that, as

will be shortly seen, the method followed here can be applied to any other filter with the aforementioned characteristics.

This chapter is organized as follows. In Section 5.2, a parameter-varying filter based on switched capacitor circuits is presented. In Section 5.3, the stability analysis of the aforementioned filter is carried out and, finally, in Section 5.4 the analysis of the response of the associated homogeneous system is carried out.

5.2 A Parameter-varying low pass filter based on switched-capacitor circuits

In Section 3.3, it was stated that there are three types of synthesis for switched-capacitor filters from continuous-time filters. In this section the reverse process — from switched-capacitor filter to continuous-time filter — is used in order to analyze the switched-capacitor filter from a continuous-time point of view. The main reason of doing the aforementioned technique is that it is easier to analyze a continuous circuit than its discrete-time counterpart.

The Martin-Sedra biquadratic filter, which is shown in Figure 5.1 is used in this work for implementing a low pass parameter-varying filter. This filter easily implemented with the resources provided by the FPAA AN231. A detailed discussion on this device may be found at Appendix A. It can be noticed that although the filter implemented with the AN231 is a fully-differential one, the filter to be analyzed in this chapter is a single-ended one. Despite the latter, the results obtained in this analysis are also valid for a fully-differential filter. It must be noticed that for the Martin-Sedra biquadratic filter, parasitic-insensitive switched-capacitor networks are used to emulate the resistors required by the circuit.

The transfer function in the Laplace domain for the low pass filter shown in Figure 5.1 is given by

$$H(s) = -\frac{4\pi^2 f_0^2 G}{s^2 + \frac{2\pi f_0}{Q} s + 4\pi^2 f_0^2}$$

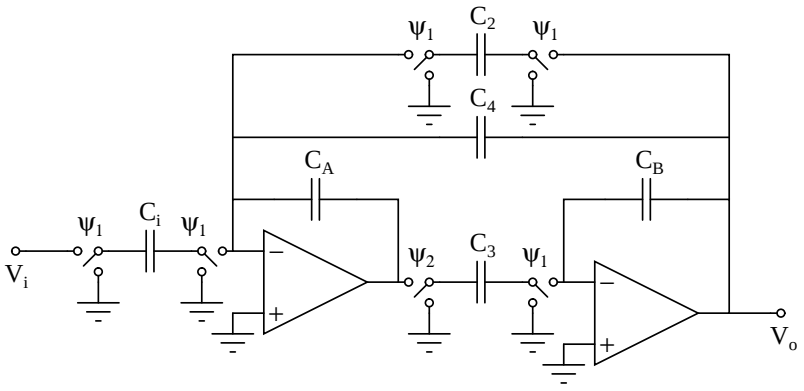


Figure 5.1: Low Pass filter with switched-capacitor circuits.

with

$$G = -\frac{C_1}{C_2} \tag{5.1a}$$

$$f_0 = \frac{f_C}{2\pi}\sqrt{\frac{C_2 C_3}{C_A C_B}} \tag{5.1b}$$

$$Q = \frac{1}{C_4}\sqrt{\frac{C_A C_B C_2}{C_3}} \tag{5.1c}$$

In the previous expression,

- G is the DC gain of the low pass filter.
- f_0 is the corner frequency.
- Q is the quality factor.
- f_C is the clock frequency of the switched-capacitor circuits.
- C_A, C_B, C_1, C_2 and C_4 are the capacitor values of the switched capacitor circuits as shown in Figure 5.1.

The switched-capacitor filter may be derived from a continuous-time equivalent if the resistors present on the continuous-time circuit are substituted by switched-capacitor networks which emulate their behavior. The continuous-time circuit for the Martin-Sedra biquadratic filter is depicted in Figure 5.2. For reasons of simplicity, a

negative resistor with resistance equal to $-R_3$ is presented. This resistor is readily implemented using switched-capacitor networks by exchanging the clock signals in the circuit presented in Figure 3.1d.

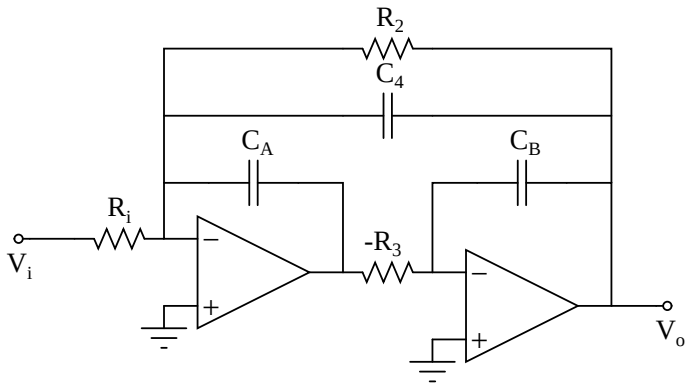


Figure 5.2: Filter used in the continuous-time domain.

The state equations of the circuit shown in Figure 5.2 are indicated below

$$\begin{bmatrix} \dot{V}_{CA}(t) \\ \\ \dot{V}_{CB}(t) \end{bmatrix} = \begin{bmatrix} -\dfrac{C_4}{C_A C_B R_3} & -\dfrac{1}{C_A R_2} \\ \\ \dfrac{1}{C_B R_3} & 0 \end{bmatrix} \begin{bmatrix} V_{CA}(t) \\ \\ V_{CB}(t) \end{bmatrix} + \begin{bmatrix} \dfrac{1}{C_A R_1} \\ \\ 0 \end{bmatrix} V_i(t). \tag{5.2}$$

As it was stated in Chapter 4, the variation of parameters in parameter-varying filter should be done in an exponential manner. The variation strategy can be applied to a given parameter or a group of parameters. In this case, the variation should be done of corner frequency f_0 and the quality factor Q. In relations 5.1 can be observed that both parameters can be varied by varying capacitor C_3. In the case of its continuous-time equivalent circuit, R_3 must be varied. Given that R_3 is inversely proportional to C_3, the variation in the value of R_3 should be carried out as indicated below.

$$R(t) = R_{3C} - R_x e^{-t/t_d}, \tag{5.3}$$

where,

- R_{3_C} is the value of resistor in steady state.
- R_x is the minimum value of $R(t)$ over R_{3_C}.
- t_d is the time-constant which depends of transient time of the circuit.

It is necessary to mention that the variation in the value of R_3 is carried out when the filter exhibits its transient state behavior in order to ensure that the values of the filter parameters in the stationary regime are maintained. Given that it is only possible to induce discrete changes in the value of any capacitor belonging the switched-capacitor filter depicted in Figure 5.1 when it is implemented on the AN231, the variation of the value of R_3 is induced by means of a piecewise function $R_p(t)$ in discrete-time intervals as it is shown in Figure 5.3.

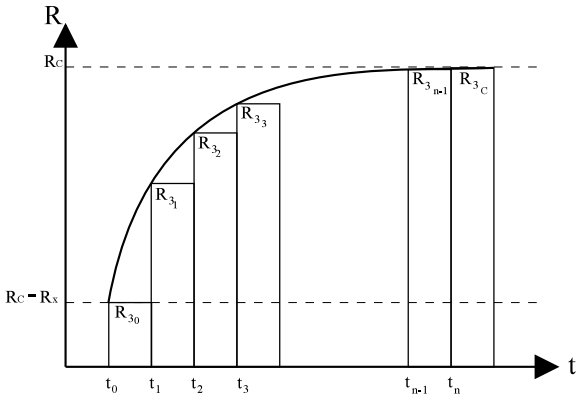


Figure 5.3: Variation of R(t).

Function $R_p(t)$ is given by

$$R_p(t) = \begin{cases} R_{3_0} & t_0 \leq t < t_1 \\ R_{3_1} & t_1 \leq t < t_2 \\ \vdots & \vdots \\ R_{3_i} & t_i \leq t < t_{i+1} \\ \vdots & \vdots \\ R_{3_{n-1}} & t_{n-1} \leq t < t_n \\ R_{3_C} & t \geq t_n \end{cases} \tag{5.4}$$

Therefore, the state-space representation of the parameter-varying filter is indicated below

$$\begin{bmatrix} \dot{V}_{CA}(t) \\ \\ \dot{V}_{CB}(t) \end{bmatrix} = \begin{bmatrix} -\dfrac{C_4}{C_A C_B R_p(t)} & -\dfrac{1}{C_A R_2} \\ \\ \dfrac{1}{C_B R_p(t)} & 0 \end{bmatrix} \begin{bmatrix} V_{CA}(t) \\ \\ V_{CB}(t) \end{bmatrix} + \begin{bmatrix} \dfrac{1}{C_A R_1} \\ \\ 0 \end{bmatrix} V_i(t). \tag{5.5}$$

The stability analysis of the system modeled by Equation (5.5) will be now carried out.

5.3 Stability Analysis of Switched-Capacitor PVF

Consider the homogeneous system

$$\begin{bmatrix} \dot{V}_{CA}(t) \\ \\ \dot{V}_{CB}(t) \end{bmatrix} = \begin{bmatrix} -\dfrac{C_4}{C_A C_B R_p(t)} & -\dfrac{1}{C_A R_2} \\ \\ \dfrac{1}{C_B R_p(t)} & 0 \end{bmatrix} \begin{bmatrix} V_{CA}(t) \\ \\ V_{CB}(t) \end{bmatrix} \tag{5.6}$$

Given that $R_p(t)$ is a piecewise function, the system matrix of the previous system in the interval $t_{i-1} \leq t < t_i$ becomes a constant matrix $\mathbf{A_i}$ of the form indicated below

$$\mathbf{A_i} = \begin{bmatrix} -\dfrac{C_4}{C_A C_B R_{3_i}} & -\dfrac{1}{C_A R_2} \\ \\ \dfrac{1}{C_B R_{3_i}} & 0 \end{bmatrix} \tag{5.7}$$

The transition matrix $\mathbf{\Phi_i}(t, t_0)$ associated to system

$$\dot{\mathbf{x}}(t) = \mathbf{A_i}(t)\mathbf{x}(t) \tag{5.8}$$

is given by

$$\mathbf{\Phi_i}(t, t_0) = e^{\mathbf{A_i}(t,t_0)} \tag{5.9}$$

$$\tag{5.10}$$

$$\mathbf{\Phi_i}(t,t_0) = \begin{bmatrix} \frac{\alpha_i-\gamma}{2\alpha_i}g_1(t-t_0)+\frac{\alpha_i+\gamma}{2\alpha_i}g_2(t-t_0) & -\frac{C_BR_{3_i}}{\alpha_i}\left(g_1(t-t_0)-g_2(t-t_0)\right) \\ \frac{C_AR_2}{\alpha_i}\left(g_1(t-t_0)-g_2(t-t_0)\right) & \frac{\alpha_i+\gamma}{2\alpha_i}g_1(t-t_0)+\frac{\alpha_i-\gamma}{2\alpha_i}g_2(t-t_0) \end{bmatrix} \tag{5.11}$$

where,

$$\begin{aligned} g_1(t) &= \exp\left(-\frac{\gamma-\alpha_i}{\beta_i}t\right) \\ g_2(t) &= \exp\left(-\frac{\gamma+\alpha_i}{\beta_i}t\right) \\ \alpha_i &= \sqrt{R_2^2C_4^2-4C_AC_BR_2R_{3_i}} \\ \beta_i &= 2C_AC_BR_2R_{3_i} \\ \gamma &= C_4R_2 \end{aligned} \tag{5.12}$$

The response of the system defined by expression (5.6) in the interval $t_{i-1} \leq t < t_i$ is:

$$\mathbf{x_i}(t) = \left(\mathbf{\Phi_i}(t,t_{i-1})\prod_{j=1}^{i-1}\mathbf{\Phi_{i-j}}(t_{i-j},t_{i-j-1})\right)\mathbf{x}(t_0). \tag{5.13}$$

The norm of the response given in Equation (5.13) can be expressed as

$$\|\mathbf{x_i}(t)\| = \left\|\left(\mathbf{\Phi_i}(t,t_{i-1})\prod_{j=1}^{i-1}\mathbf{\Phi_{i-j}}(t_{i-j},t_{i-j-1})\right)\mathbf{x}(t_0)\right\|. \tag{5.14}$$

If the following relations are considered for the norm of the product of a matrix with a vector and the norm of the product of two matrices [28]:

$$\begin{aligned} \|\mathbf{Av}\| &\leq \|\mathbf{A}\|\|\mathbf{v}\|, \\ \|\mathbf{AB}\| &\leq \|\mathbf{A}\|\|\mathbf{B}\|, \end{aligned}$$

Equation (5.14) may be rewritten as indicated below

$$\|\mathbf{x_i}(t)\| \leq \left(\|\mathbf{\Phi_i}(t, t_{i-1})\| \prod_{j=1}^{i-1} \|\mathbf{\Phi_{i-j}}(t_{i-j}, t_{i-j-1})\| \right) \|\mathbf{x}(t_0)\|. \tag{5.15}$$

From Definition 1 of a transition matrix in Chapter 2, it is possible to state that the norm of the general transition matrix in the entire interval $[t_0, t)$ in Equation (5.15) can be expressed as:

$$\|\mathbf{\Phi_i}(t, t_0)\| = \|\mathbf{\Phi_i}(t, t_{i-1})\| \prod_{j=1}^{i-1} \|\mathbf{\Phi_{i-j}}(t_{i-j}, t_{i-j-1})\|. \tag{5.16}$$

Equation (5.16) indicates that the norm of the general transition matrix is the product of the norms of all transition matrices defined in each interval. In the remainder of this discussion, the norm which will be used for all the matrices is the known as the infinity-norm:

$$\|\mathbf{A}\|_\infty = \max_i \sum_i |a_{ij}|$$

The norm of the transition matrix in the interval $t_{i-1} \leq t < t_i$ is as follows.

$$\|\mathbf{\Phi_i}(t, t_{i-1})\| = \max \left\{ \begin{array}{l} \left| \dfrac{\alpha_i - \gamma}{2\alpha_i} + \dfrac{C_B R_{3_i}}{\alpha_i} \right| g_1(t - t_{i-1}) + \left| \dfrac{\alpha_i + \gamma}{2\alpha_i} + \dfrac{C_B R_{3_i}}{\alpha_i} \right| g_2(t - t_{i-1}), \\ \left| \dfrac{\alpha_i + \gamma}{2\alpha_i} + \dfrac{C_A R_2}{\alpha_i} \right| g_1(t - t_{i-1}) + \left| \dfrac{\alpha_i - \gamma}{2\alpha_i} + \dfrac{C_A R_2}{\alpha_i} \right| g_2(t - t_{i-1}) \end{array} \right\}$$

where the operator $|\cdot|$ stands for the absolute value.

In the last equation it is quite difficult to choose the maximum value, and consequently $\|\mathbf{\Phi_i}(t, t_{i-1})\|$, if values of α_i, γ, C_B and R_{3_i} are not known in advance. However, it is clear that whatever the maximum value is, $\|\mathbf{\Phi_i}(t, t_{i-1})\|$ will exponentially decrease in value towards zero as $t \to \infty$. For this reason, the system represented in Equation (5.6) is exponentially stable.

5.4 Analysis of Homogenous Response of Switched-Capacitor PVF

In this section the evolution of the response of the parameter-varying low pass filter with switched-capacitors will be analyzed. A geometric approach will be used. In [3] the same approach was utilized. In order to use this approach the following background is necessary.

As it was already stated, the columns of the transition matrix is associated to the fundamental set of solutions — a linearly independent set of solutions — of the system from which it was derived. It follows that the columns of the transition matrix constitutes a basis for the space of solutions and, consequently, it can represent all the solutions of the system. That is:

$$\mathbf{x}(t) = \Phi(t, t_0)\mathbf{x_0}. \tag{5.17}$$

Graphically, in the state plane, a particular state in time of a second-order system can be represented by a 2-dimensional vector. This solution corresponds to the linear combination of two linearly independent solutions contained in the columns of the system's transition matrix. This concept is depicted in Figure 5.4:

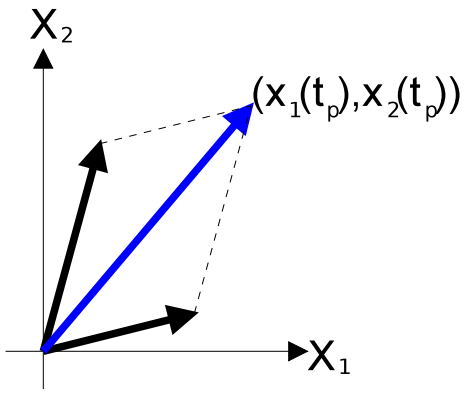


Figure 5.4: Graphic representation of a solution in a particular instant of time.

In that sense, the analysis of the homogeneous response of the system represented in Equation (5.6) will be done in terms of the area of the parallelogram formed by

its solution vectors. It is essential to recall that this area can be calculated through the determinant of the transition matrix of the system. Fortunately, as it was seen in Chapter 2, this determinant can be easily calculated using the trace of the system matrix. For the sake of clarity, this property is reproduced in the following equation:

$$\det \mathbf{\Phi}(t, t_0) = \exp\left(\int_{t_0}^{t} \mathrm{tr}\mathbf{A}(t)dt\right).$$

The previous expression is also known as the Liouville Formula.
This analysis will be done under the next considerations:

- The state equation of the homogeneous system to analyze is:

$$\begin{bmatrix} \frac{d}{dt}V_{CA} \\ \frac{d}{dt}V_{CB} \end{bmatrix} = \begin{bmatrix} -\frac{C_4}{C_A C_B R_p(t)} & -\frac{1}{C_A R_2} \\ \frac{1}{C_B R_p(t)} & 0 \end{bmatrix} \begin{bmatrix} V_{CA} \\ V_{CB} \end{bmatrix}$$

- The value of function $R_p(t)$ is increased discretely in well-defined time instants. Thus, it can be ensured that for every interval of variation, the value of resistor R_3 remains constant as shown in Figure 5.3.

- The variation in values of function $R_p(t)$ ocurrs in the interval $t_0 \leq t \leq t_c$. Constant t_c is a constant which depends on the number of steps involved in the variation of the value of resistor R_3 as well as of the time step considered for inducing a change in the value of the aforementioned element.

- The evolution of the transient response will be analyzed geometrically, according to the procedure shown in Reference [3], that is, observing the evolution of the area of a parallelogram formed by the column vectors of the associated transition matrix to the system.

Analysis in each interval of variation

Given that the variations of the value of resistor R_3 are done in discrete intervals, in every interval, the system may be considered as an invariant-time system in a finite time interval. Therefore, the filter can be modeled by the following state equation:

$$\dot{\mathbf{x}}_\mathbf{i} = \mathbf{A_{Ci}}\mathbf{x}$$

where the matrix $\mathbf{A_{Ci}}$ is constant. in the interval $t_i \leq t \leq t_{i+1}$.

Owing to the transition matrix properties, the filter response in the aforementioned time interval is given by:

$$\mathbf{x}(t) = \Phi_i(t_{i+1}, t_i)\mathbf{x}(t_i).$$

This expression may be rewritten as

$$\mathbf{x}(t) = \left(\mathbf{\Phi_n}(t, t_{n-1}) \prod_{j=1}^{n-1} \mathbf{\Phi_{n-j}}(t_{n-j}, t_{n-j-1}) \right) \mathbf{x}(t_0).$$

Therefore, the transition matrix of the interval $t_0 \leq t_c$ will be:

$$\Phi(t, t_0) = \left(\mathbf{\Phi_n}(t, t_{n-1}) \prod_{j=1}^{n-1} \mathbf{\Phi_{n-j}}(t_{n-j}, t_{n-j-1}) \right).$$

A method to verify the evolution of the transient response of the system in the interval $t_0 \leq t \leq t_C$ is calculating the area of the parallelogram whose edges are composed by the column vectors of the transition matrix.

Due the Liouville formula:

$$\det(\Phi(t, t_0)) = |\Phi(t, t_0)| = \exp\left(\int_{t_0}^{t} \mathrm{tr}\mathbf{A} dt \right).$$

The determinant of the general transition matrix may be rewritten as indicated below

$$|\Phi(t, t_0)| = |\mathbf{\Phi_n}(t, t_{n-1})| \prod_{j=1}^{n-1} |\mathbf{\Phi_{n-j}}(t_{n-j}, t_{n-j-1})|. \tag{5.18}$$

For each determinant, we have:

$$|\Phi_i(t_i, t_{i-1})| = \exp\left(\int_{t_{i-1}}^{t_i} \mathrm{tr}\mathbf{A_i} dt \right).$$

The trace of $\mathbf{A_i}$ is $\frac{C_4}{C_A C_B R_{3i}}$. Therefore:

$$|\Phi_i(t_i, t_{i-1})| = \exp\left(-\frac{C_4 t}{C_A C_B R_{3i}}\Bigg|_{t_{i-1}}^{t_i}\right) = \exp\left(-\frac{C_4}{C_A C_B R_{3i}}(t_i - t_{i-1})\right). \tag{5.19}$$

Substituting Equation (5.19) in the Equation (5.18):

$$|\Phi(t, t_0)| = \left(-\frac{C_4}{C_A C_B}\frac{(t - t_{n-1})}{R_{3_n}}\right)\prod_{i=1}^{n-1}\exp\left(-\frac{C_4}{C_A C_B}\frac{(t_i - t_{i-1})}{R_{3_i}}\right). \tag{5.20}$$

Given that $R_{3n} = R_{3_C}$, Equation (5.20) can be rewritten as:

$$|\Phi(t, t_0)| = \exp\left[-\frac{C_4}{C_A C_B}\left(\frac{t - t_{n-1}}{R_{3_C}} + \sum_{i=1}^{n-1}\frac{t_i - t_{i-1}}{R_{3_i}}\right)\right]. \tag{5.21}$$

If the difference $t_i - t_{i-1}$ is taken as constant, Equation (5.21) can be rewritten as:

$$|\Phi(t, t_0)| = \exp\left[-\frac{C_4}{C_A C_B}\left(\frac{t - t_{n-1}}{R_{3_C}} + \sum_{i=1}^{n-1}\frac{\Delta T}{R_{3_i}}\right)\right] \tag{5.22}$$

where

$$\Delta T = t_i - t_{i-1}.$$

On the other hand, for a linear time-invariant filter which has the same topology than the shown in Figure 5.2, the evolution in the area of the parallelogram composed by the column vectors of its constant transition matrix in the interval $t_0 \leq t \leq t_C$, is:

$$|\Phi(t, t_0)| = \exp\left(-\frac{C_4(t - t_0)}{C_A C_B R_{3C}}\right). \tag{5.23}$$

Equation 5.23 can be expressed as:

$$|\Phi(t, t_0)| = \exp\left[-\frac{C_4}{C_A C_B}\left(\frac{t - t_{n-1}}{R_{3_C}} + \sum_{i=1}^{n-1}\frac{\Delta T}{R_{3_C}}\right)\right]. \tag{5.24}$$

By observing equations (5.22) and (5.24) is possible to say that the absolute value of the argument of the exponential for the parameter-varying is greater than the

corresponding for the linear time-invariant filter. The justification for this assertion is that $R_{3i} \leq R_{3_C}$. Put it in other words, by observing Equations (5.22) and (5.24) is possible to assert that the homogeneous response for a switched-capacitor parameter-varying filter is faster than its invariant-time filter counterpart.

Finally, after analyzing Equation (5.22), it is readily concluded that a reduction of the duration of the transient response of the parameter-varying switched-capacitor low pass filter may arise under the following conditions:

- If the number of steps involved in the variation of the values of function $R_p(t)$ is increased, the homogeneous response of the switched-capacitor filter will be faster.
- In the same manner, increasing values of parameter Δt will lead to a faster homogeneous response of the parameter-varying filter.

Chapter 6

Implementation and Tests of a PVF with Switched-Capacitor Circuits

6.1 Introduction

In Chapter 4 was mentioned that there are few implementations of parameter-varying filters. For instance, an analogue implementation is cited in [13].

In this work, a parameter-varying low pass filter based on switched-capacitor techniques is presented. This filter is implemented with a Field-Programmable Analog Array (FPAA)[1].

This chapter is organized as follows: in Section 6.3 the filter implementation in FPAA AN231 is discussed and in Section 6.4 measurement results are given.

6.2 Time-Invariant Second-Order Low Pass Filter

The filter that is embedded in the FPAA is a low pass biquad filter. The realization of the low pass transfer function is achieved by the switched-capacitor circuit shown in Figure 6.1.

[1]See Appendix A for a full description of this device.

As known from the theory of switched-capacitor networks [23], arrays of capacitor and switches are used to emulate resistors. In this filter the switched-capacitors are C_1, C_2, and C_3, i.e. they emulate resistors. On the other hand, the capacitors with fixed values are: C_A, C_B, and C_4.

The design equations for this filter are repeated here for sake of clarity:

$$f_0 = \frac{f_c}{2\pi}\sqrt{\frac{C_2 C_3}{C_A C_B}} \tag{6.1a}$$

$$G = -\frac{C_1}{C_2} \tag{6.1b}$$

$$Q = \frac{1}{C_4}\sqrt{\frac{C_A C_B C_2}{C_3}} \tag{6.1c}$$

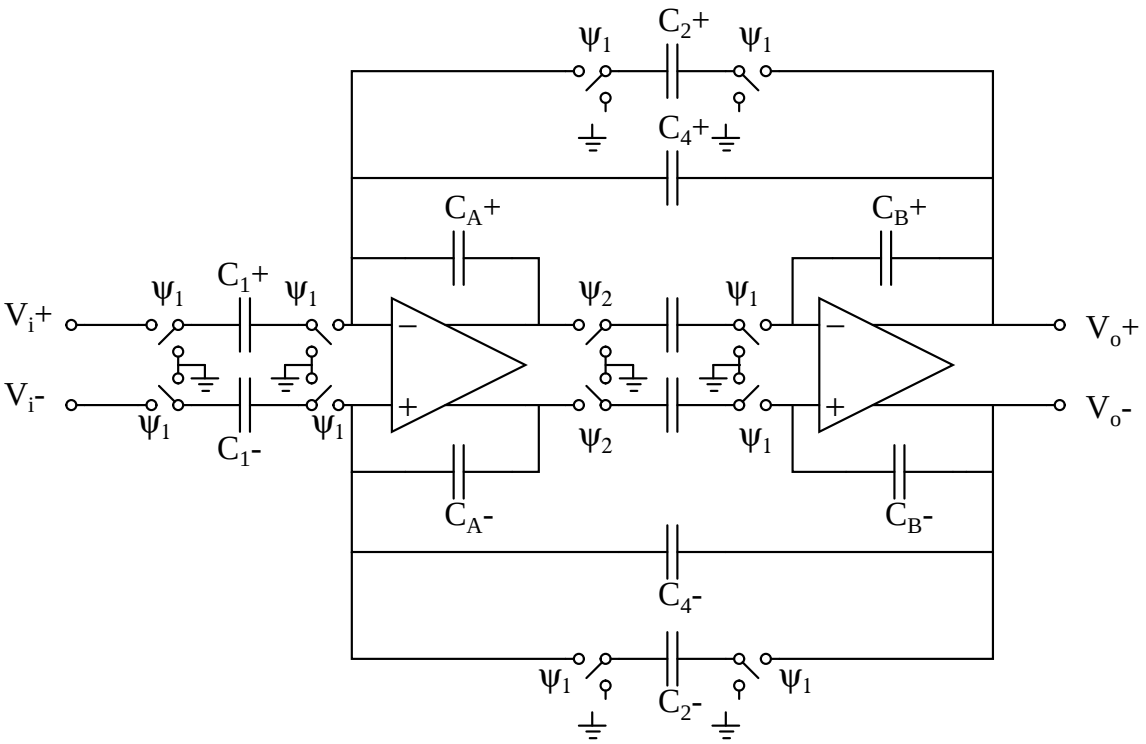


Figure 6.1: Biquadratic type II Filter implemented in the FPAA AN231.

A Case of Design

A design of the time-invariant filter is carried out with the next set of specifications:

- Corner frequency = 1 kHz.
- Quality factor = 20.
- Gain = 1
- Clock frequency = 250 kHz.

The step response of the time-invariant filter described so far is shown in Figure 6.2.

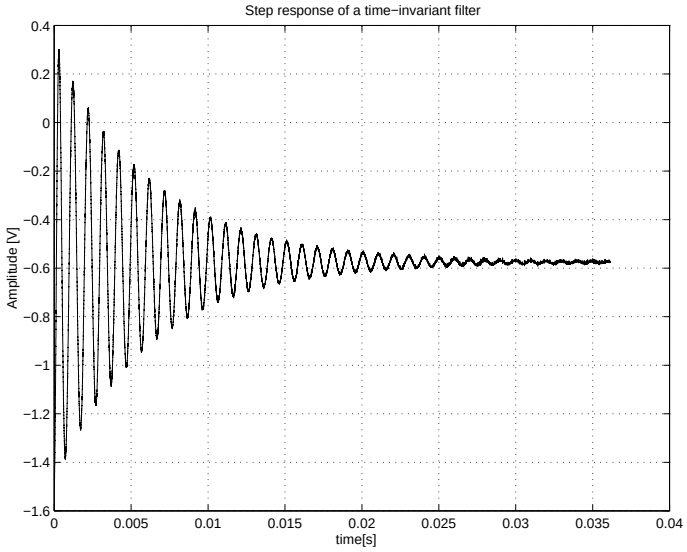


Figure 6.2: Step response of the time-invariant filter implemented in the FPAA AN231.

The design procedure of the time-invariant implementation has been achieved with the AnadigmDesigner2 program. In this program a CAM named FilterBiquad version 1.0.1 was chosen with the specifications already mentioned. It is important to notice that this implementation is remarkably simple because it only requires one Configurable Analogue Block (CAB).

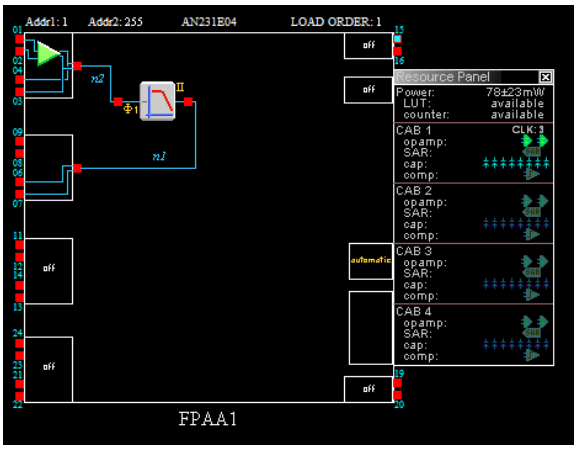


Figure 6.3: Second-order filter implemented in the FPAA AN231.

This design represents the starting point when parameter variations are included in the final implementation.

In order to complete the description of this implementation, it is worthy to mention that this Configurable Analogue Module (CAM) uses all capacitors and both operational amplifiers included in this Configurable Analogue Block (CAB). Besides, the data transfer from the shadow RAM to the configuration RAM is automatically done inside the FPAA. Special attention is devoted to the configuration of the IO Cells of the FPAA. The IO Cell 1 is configured in input mode and functions as an operational amplifier with the idea of conditioning the voltage signal arriving to the filter. The IO Cell 2 is configured in output mode and works as a bypass. Both cells are located in the top left corner on the snapshot shown in Figure 6.1.

6.3 Parameter-varying Filter Implementation

From Equation (6.1), it can be observed that both parameters[2], the corner frequency f_0 and the quality factor Q depend on C_3. It clearly results that if C_3 is programmed then we can achieve a filter with varying parameters, while keeping the DC gain unchanged. This fact is used in order to set up a parameter-varying filter based on switched-capacitors. A general block diagram of this setting up is shown in Figure 6.4.

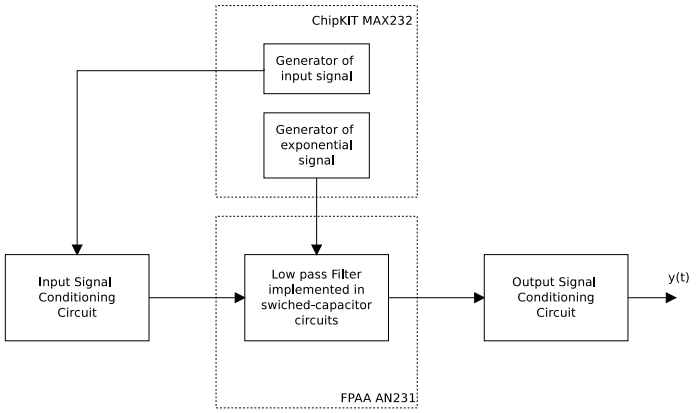


Figure 6.4: Block diagram of the implemented parameter-varying filter.

A general description of the set up is given hereafter. The FPAA AN231 block runs the time-invariant second-order filter described in Section 6.2. The ChipKit[3] block is a development board that synchronizes the input signal generator and the exponential signal that programs the capacitors that determines the variations in the parameters of the filter.

[2]The FPAA datasheet describes two types of filter topologies. The type I topology has design equations where the parameters of the filter depend on more than one capacitor which involves a more difficult design procedure. On the other hand, type II topology results in the design formulas given in Equation (6.1), i.e. it is possible to simultaneously change two parameters by only changing one capacitor.

[3]Related information to this development board can be found in Appendix B.

Besides, two signal conditioning blocks are used one for the input signal and another for the output signal. More details about each block are given in the following subsections.

The ChipKit Development Board

In this block two signals are generated. A first signal conveys the exponential function that is used to achieve the programming of the parameters of the filter. The second signal is an input signal that is synchronized with the first one in order to be able of comparing the response of the filter with fixed parameters with the response of the parameter-varying filter.

The exponential function generator allows us to program the values of those capacitors involved in the design formulas that determine the cut-off frequency and the quality factor. Besides, the programming of the parameter-varying filter can be done on-the-fly due to the features of the FPAA AN231. Therefore, the idea behind this implementation of parameter-varying filter with SC circuits is to reconfigure a filter with switched-capacitors on-the-fly.

The controlling signal is in fact a set of data that contains all the information related to the type of CAM used and how their parameters are related to the capacitor values. This data is transmitted to the FPAA through an SPI interface. Further information can be found in Appendix A.

Another task of the ChipKit development board is to send the input signal to the filter. By doing the latter, it is possible to synchronize both the input signal and the beginning of the transmission of the data to the FPAA, that is, the reconfiguration of the filter. In addition, this allows us to achieve a correct comparison between the responses of the time-invariant filter and the parameter-varying counterpart.

The programming of this block is reported in in Appendix D, Section D.1.

Signal Conditioning Circuits

In order to couple the different signals that appear in the circuit, two signal conditioning circuits are implemented.

The first signal conditioning block consists of a pulse generator and a Rauch filter. This block is used to couple the synchronizing signal, coming from the dsPIC, to the IO Cell of the FPAA.

The pulse generator consists of a DC voltage generator and an analogue summing amplifier. The DC signal is added to the pulse train generated by the dsPIC in order to form a pulse train signal with a DC offset as seen in Figure 6.5. This pulse is filtered by a Rauch topology in which the active part is emulated by the IO cell 1 port of the FPAA (see Figure 6.6). The main tasks of the Rauch filter are: 1) to convert a single-ended signal into a differential signal, and 2) to serve as an antialiasing filter in order to reduce the noise.

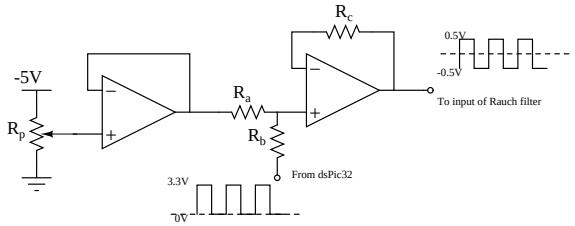


Figure 6.5: Conditioning signal circuit from ChipKit to Rauch filter.

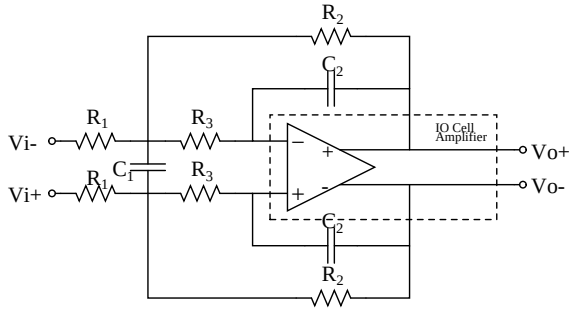


Figure 6.6: Rauch Filter used.

The second signal conditioning block consists of an amplifier that converts a differential signal into a single-ended signal with ground reference. This block is implemented by the circuit shown in Figure 6.7.

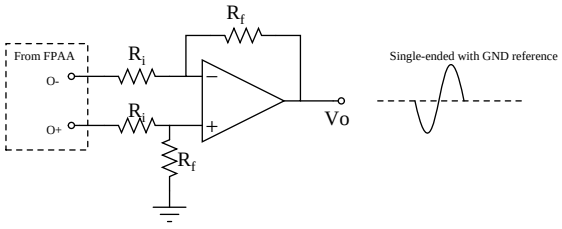


Figure 6.7: Signal conditioning circuit at the output of FPAA.

6.4 Experiments and Results

In order to demonstrate the possibility of reconfiguring a switched-capacitor based parameter-varying filter on-the-fly, setups were studied. Two main experiments were carried out in order to demonstrate the feasibility of the implementation with FPAA, namely:

Experiment 1 The cut off frequency of the filter was varied.

Experiment 2 The cut off frequency and the quality factor were simultaneously varied.

As established in Chapter 4, in order to guarantee an stable behavior of the filter, the variation of the parameters must be done exponentially. In this work two methods to generate this kind of variation are used in the experiments carried out. In the following these two experiments are explained.

Experiment 1

In this experiment, the data used to achieve the reconfiguration of the FPAA is generated by the AnadigmDesigner2 program which is explained in Appendix A. The data generated corresponds to the primary configuration and to the update configuration. Both sets of data are allocated in the dsPIC memory to be sent to the FPAA AN231.

The cut off frequency is varied according to the next mathematical rule:

$$\omega(t) = A\exp(-t/\tau) \tag{6.2}$$

where τ is the constant time of the control signal and A is the difference between the maximum frequency and the frequency at the steady state (the cut off frequency of the invariant-time filter). In fact, τ is the time interval between two successive transmissions of data for update configuration.

For this experiment, A takes the values 1 to 9 kHz in steps of 1 kHz. For every A, τ takes the following values: 0, 100, 200, 300, 400, 500, 600, 700, 800, 1000, 1500 and 2000 milliseconds. As already mentioned, one of the reasons to design parameter-varying filters is the decrease of the settling time of the transient response, that is to say faster filters. The settling time in all tests was calculated as the time required for the response curve to reach and stay within a range of 1% of the final value (the step amplitude).

In Figure 6.8, the settling time as function of A and τ are plotted — different perspectives of the plot are shown.

It can be observed that the greater A is, the shorter the settling time is. In other words, faster filters are obtained with larger A's.

With respect to the dependence on τ, it can be observed that the greater τ is, the shorter the settling time is. Here again, faster filters are obtained with larger τ's.

The slower filters correspond to the cases where τ and A are very small. This can be seen in the other perspectives. In fact, these cases tend to have a limit behavior as the time-invariant filter.

Experiment 2

As it was previously stated, in this experiment both parameters the cut off frequency and the quality factor are simultaneously varied. From the design Equations (6.1), it can be observed that the capacitor C_3 is related with the cut off frequency and the quality factor in a very convenient way. In order to achieve the reconfiguration of the filter, a 2-base exponential rule is applied, which can be obtained by executing a simple right-shift in the register that contains the binary value of C_3. See Appendix C for further information.

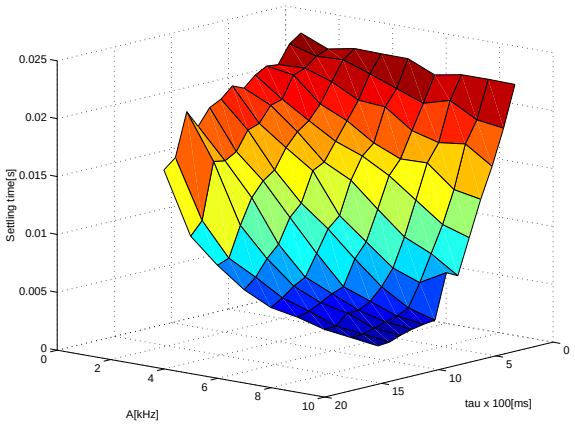
The exponential sweep of the values of C_3 yields the set of filter parameters recast in Table 6.1. The first row of the table corresponds to the filter configuration

associated to the maximum value of C_3, while the last row corresponds to the configuration of the time-invariant low pass filter. It can be noticed that the Gain remains unchanged in this reconfiguration procedure.

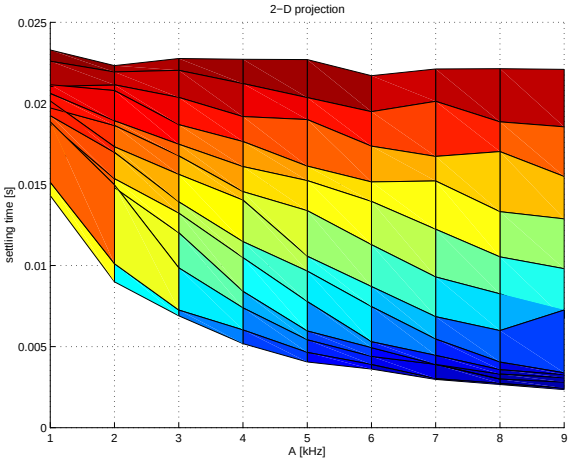
C_3 Register	f_0	G	Q
0xb0	3987.937070	1.000000	4.988623
0x58	2819.897345	1.000000	7.054979
0x2c	1993.968535	1.000000	9.977247
0x16	1409.948673	1.000000	14.109958
0x0b	996.984268	1.000000	19.954545

Table 6.1: Filter parameters for C_3 variation.

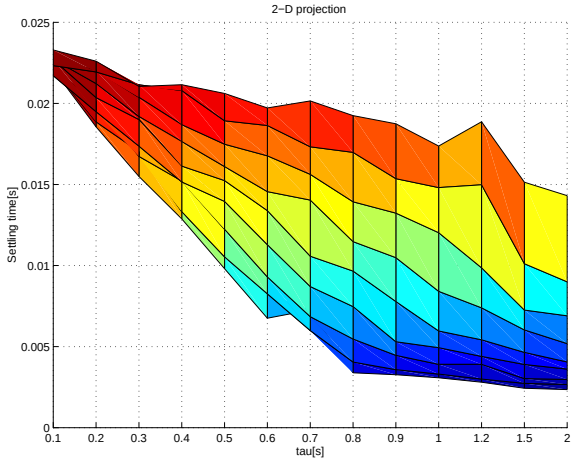
The step-response of the parameter-varying filter is shown in Figure 6.9 and it is displayed next to the step-response of the time-invariant counterpart. It results evident that the steady-state is reached faster in the parameter-varying implementation.



(a) 3D View.

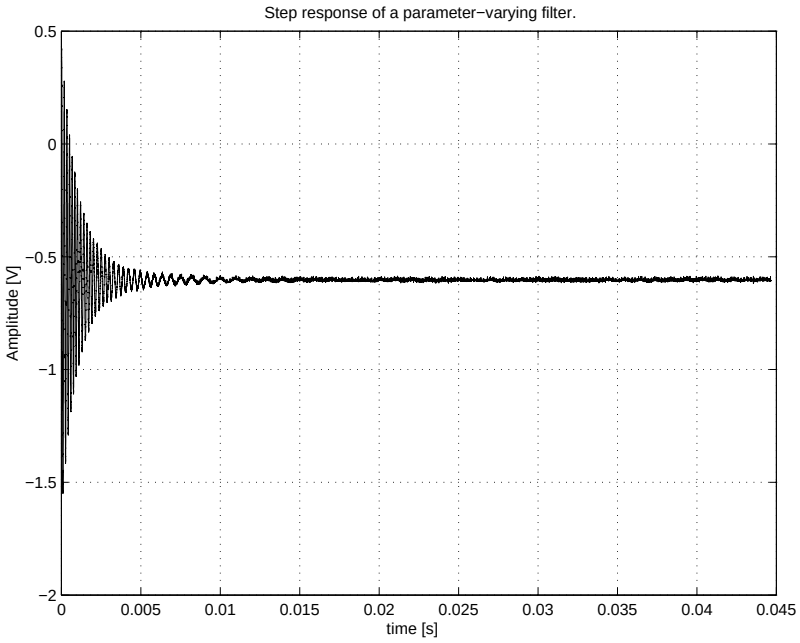


(b) t_s vs A.

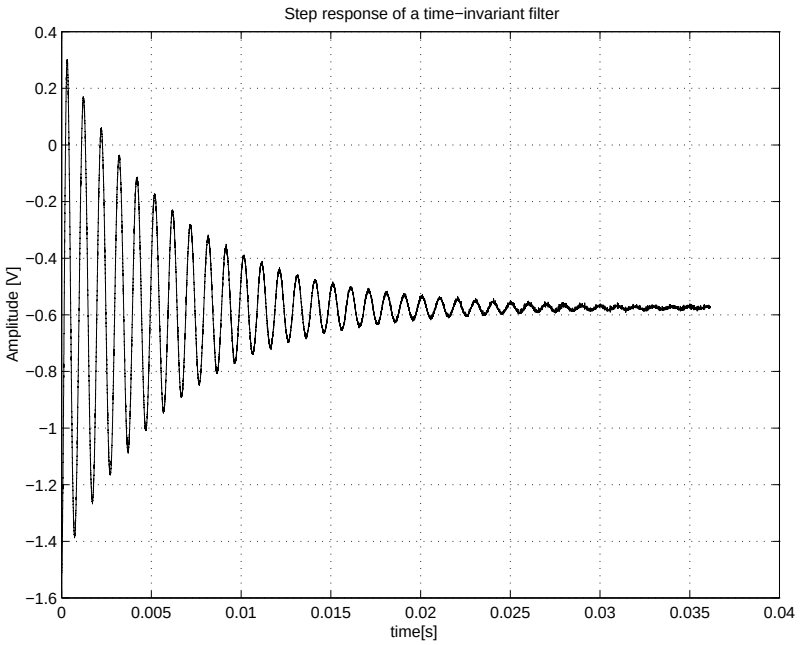


(c) t_s vs τ.

Figure 6.8: Settling times of Experiment 1.



(a) Step-response of the parameter-varying filter.



(b) Step-response of the time-invariant filter.

Figure 6.9: Comparison of the step responses of Experiment 2.

Conclusions

Parameter-varying filters are characterized by their shorter transient time response than their time-invariant filter counterpart. This speed in their transient response is a direct consequence of the variation achieved on their descriptive parameters, which is obtained by means of a predefined control rule, usually recast in a time-depending function. Despite this theoretical advantage, there is still a bottle-neck in the practical implementation of these kinds of filters.

In this thesis, an implementation of a low pass parameter-varying filter with switched-capacitors circuits has been proposed. The theoretical aspects of this implementation were extensively studied by analyzing its homogeneous response. As an important result, two aspects that are relevant to this kind of filters were corroborated: the exponential stability and the reduction of the transient response. The implementation was tackled by using a Field-Programmable Analog Array that constitutes the design framework. In addition, in order to generate the control rules that govern the reconfiguration of the filter parameters, it was necessary to program a digital development board. Regarding the implementation of this work, it is worthy to mention that signal conditioning circuitry has also been designed.

The implementation of the parameter-varying low pass filter was tested with two experiments. Not only do the experimental results demonstrate the feasibility of this specific implementation but, also agree with the theory of this kind of filters.

The conclusions of this work can be summarized as follows:

- The feasibility of implementing parameter-varying filters with switched-capacitor techniques within a FPAA framework has been demonstrated.

- The reconfiguration of the SC parameter-varying filter results quite straightforward due to the fact that the parameter that have been varied depend on a reduced set of components.
- The resulting parameter-varying filter exhibits the expected behavior of having shorter settling times while the frequency-domain response is identical to the time-invariant filter.

Further research can be oriented in two main directions. On one side, the switched-capacitor implementation can be realized with an integrated circuit in a CMOS technology. On the other side, the methodology used in this thesis could also be applied to other types of filters such as high pass, band pass, all pass and notch.

Appendix A

Description of the FPAA AN231

A.1 General Overview

From a general point of view, a Field-Programmable Analog Array (FPAA, from hence) can be considered as the analog counterpart of a Field-Programmable Gate Array (or FPGA). In this regard, it can be programmed to make a particular function in the analog domain. The latter constitutes the chief difference between a FPAA and a FPGA. In this appendix, the FPAA characteristics, its advantages and its disadvantages will be treated.

Usually, FPAAs are divided into two groups: continuous-time FPAAs and discrete-time FPAAs. The former, generally, uses operational transconductance amplifiers [29] (OTAs) or other types of amplifiers with passive components while the latter utilizes switched-capacitor circuits. As it will be seen shortly, the FPAA described in this appendix belongs to the last group.

Architecturally, it can be said that a FPAA is composed by two functional elements: routing and computing. Both elements are present in all the designs of FPAAs regardless of whether they belong either to the discrete-time domain or to the continuous-time domain. The routing elements are mostly arrays of switches which serve to connect the computing elements. The computing elements may be operational amplifiers, capacitors, etc. Usually, these components are grouped together inside the FPAA, which constitutes an analog block. Every block is known

as a Configurable Analog Block or CAB, and they constitute the core of an FPAA because its configurability characteristic —in other words, its programmability – lies on them.

FPAAs are commonly used for prototyping some classes of analog circuits. The design-fabrication-testing cycle in an analog integrated circuit is usually expensive. However, by using an FPAA in this process leads to a reduction in the cost of the prototyping process by eliminating the fabrication step in the design. That is, the testing can be done on the FPAA circuit. Nevertheless, there are some limitations associated to the use of the FPAA: the characteristic of the circuit implemented on this design platform such as bandwidth, gain, etc. are subordinated to the characteristics of the FPAA under consideration. Therefore, the choice of the FPAA to be used is an important step in the prototyping circuits with this kind of devices.

In this thesis, the FPAA AN23104 fabricated by Anadigm® was used. This device is commercially available and has a number of characteristics that will be shortly mentioned. Because the main objective of this thesis is to demonstrate the reliability of parameter-varying filter implemented with switched-capacitor circuits, the main reason to choose this device obeys to the switched-capacitor approach used in the design of configurable circuit blocks with it. Other circuit parameters such as bandwidth, SNR, etc. were seen as secondary aspects on the election of the device in the context of this work.

A.2 Description and Architecture of the FPAA AN231

The AN231 is a discrete-time FPAA fabricated by Anadigm®. It presents the following main characteristics:

- Fully-differential architecture.
- It has 4 Configurable Analog Blocks (CABs). Every CAB contains:
  - 2 Operational Amplifiers.

  - 1 Comparator.
  - A bank of 8 capacitors.
  - A successive approximation register (SAR).

- 7 Input/Output Cells, which serve either as inputs or outputs to the CABs.
- 2 Dedicated output cells.
- Its structure (CABs and Input/Output cells) is based on switched-capacitor circuits as well as on conventional circuits.
- Dynamic configuration capabilities (the behavior of a given circuit block can be changed on the fly during its operation).
- Typical bandwidth = 0 - 2[MHz] (CAM dependent).
- Signal to Noise Ratio: Broadband =90 dB and Narrowband = 120 dB.
- Total Harmonic Distortion = 100 dB.

It is notorious from this device that the reconfiguration is done in one clock cycle. More details about this characteristic are explained in the following subsection.

The architecture of the FPAA AN231 by Anadigm® is depicted in Figure A.1. For the sake of simplicity, in that figure, the power and ground references and the block associated to the generation of clock signals are not presented. However, these blocks and their working are explained in the corresponding subsections.

A.2.1 The input/output Cells

The input/output cells are circuit blocks which lead in or out of the FPAA the input and output voltage signals of one or more CAMs[1]. These blocks may also carry out elementary signal processing tasks as it will be seen in this section.

[1]CAM stands for Configurable Analog Module. This term is used for the circuit blocks with a more complex function (i.e., integrators, filters, oscillators, multipliers, peak detectors). These blocks are implemented with the resources of one or more CABs.

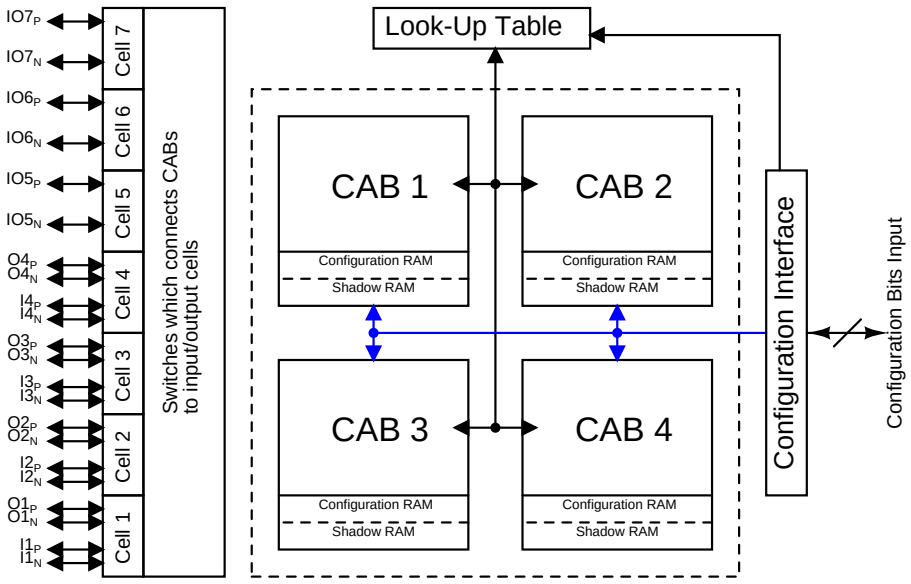


Figure A.1: Block diagram of the Anadigm® FPAA AN231.

The FPAA AN231 has 7 input/output cells. These cells are the path through the analog signals are routed in and out of the FPAA from or to the external world. There are 4 types of cells in this device: type 1, type 1a, type2 and type 2a. The type corresponding to the each cell are shown in Table A.1. In the following the different cell types will be explained.

Type 1 and 1a cells

Type 1 and 1a cells share the same characteristics. The only difference between them is that the type 1a cell has also a chopper amplifier with a low input offset. This allows small input signals to be accurately amplified so that they will be less affected by higher input offsets in the switched capacitor core of the FPAA. If the type 1a cell is configured in chopper mode, then it is necessary to indicate the gain. This gain should be from 0 to 60 dB in steps of 10 dB.

Type 1 and 1a cells can be configured in off mode, input mode, output mode and independent amplifier mode.

I/O Cell	Type
1	1
2	1
3	1a
4	1a
5	2a
6	2
7	2

Table A.1: Input/output cell types.

In off mode, the cell does not allow signals to pass, either from or to any of the elements present in any CAB.

In input mode, the internal circuitry of the cell is arranged in such a way to function in bypass mode, as a sample-and-hold circuit or as an amplifier. In bypass mode, the signal pass directly to any of the elements present in the CAB with no intervention of the cell. If the input cell is configured as a sample-and-hold circuit, it is necessary to indicate the input sample phase (1 or 2) and if it is differential, inverted differential, single-ended positive or single-ended negative. Moreover, if the cell is configured as an amplifier, it works as an independent amplifier with the difference that its output is not only sent to the output pin but also to any of the CAMs.

In output mode, the cell can be configured in bypass mode, as a sample-and-hold circuit, as a VMR block and as a digital block. In bypass mode, the signal is directly sent from a CAB to the output pin. All the circuitry of the cell is bypassed. If the cell is configured in VMR (Voltage Mid Rail) mode, the output voltage through the cell in the pin will be 1.5 V. Lastly, if the cell is configured as a digital block, the signal is sent to a digital buffer.

Finally, this cell can be used as an independent amplifier, that is, with no intervention of any CAM to do amplifier task. In this mode this cell can be used as a single full-differential operational amplifier. Electrical components as resistors and capacitors need to be outside of the chip in order to exploit its capabilities.

In Table A.2, a summary of the different configurations of cell Type 1 and 1a is presented.

Off	**Input** Bypass Sample and Hold Amplifier (Filter)	**Output** Bypass Sample and Hold VMR Digital buffer	**Independent Amplifier**	**Low Offset Chopper** (Only for Type 1a)

Table A.2: Different configuration of Type 1 and Type 1a input/ouput cells.

Type 2 and 2a Cells

Type 2 and 2a cells share the same characteristics except that the Type 2a cell has a low-offset chopper amplifier which may be used to amplify an input signal. The characterisitcs of the chopper amplifier were already described for the type 1 and 1a subcells previously introduced.

Type2 and 2a cells can be configured in off mode, input mode and output mode.

In off mode, the signal cannot pass through the cell.

In input mode the cell can be either in bypass mode, or control signal mode. The bypass mode permits the cell to route the signal with any intervention of the cell circuitry from the input pin into the FPAA. The control signal mode permits to connect input signals through digital buffers into the FPAA.

Finally, the type 2 and 2a cells can be configured as an output. The output cell can be configured in bypass mode, as a VMR block and as a digital block. Bypass and VMR modes work in the same way as for Type 1 and 1a cells when they are set as output. However, the single-ended digital block works as follows. Two independent digital signals are directed to the output by means of digital buffers. The origins of these digital signals can be the digital resources of the FPAA such as clock signals, RAM Transfer Pulse and Comparators. The digital block is used when these signals need to be routed out from the FPAA. Besides, these digital signals are used by the

FPAA to control its internal operations. More information about this is found in [30] and [31].

Dedicated Cells

Additionally to Type 1, 1a, 2 and 2a cells mentioned above, the AN231 possesses two dedicated cells: the digital cell and the RAM transfer cell.

The digital cell has only single-ended outputs. As it occurred with type 2 and 2a cells configured as single-ended digital outputs, the sources of these signals come from the digital configuration of the FPAA. The digital cell can be configured in either off mode, comparator, clock A, clock B, RAM transfer pulse or oscillator status. Because of the objective of this appendix is to serve as a reference, no more explanations about each of these signals will not be made here. The manual of the FPAA contains full explanations of these and other concepts; thus, the interested reader should review references [30] and [32].

The RAM transfer cell is commonly used for advanced purposes. For instance, it can be used to connect a cascade of AN231 devices and synchronize them. This cell deals with internal digital signals to configure or reconfigure the device. For the sake of understanding its operation it is necessary to briefly state for now, how the device configuration and reconfiguration are done by means of shadow RAM and configuration RAM even though this issue will be treated more deeply in the Section A.2.2.

Firstly, the configuration data is received into the shadow RAM and then transferred into the configuration RAM. Usually, this transference is done automatically but this behavior can be manipulated by means of the RAM transfer cell. Thus, the operation mode of this cell can be set up as Automatic, Event Driven or clock synchronization. These operation modes will be described in Section A.2.2.

A.2.2 The SRAM

The Static Random Access Memory (SRAM) contained in the FPAA AN231 accomplishes different functions, which will be explained in this subsection. There are three regions of SRAM: the Shadow RAM, the configuration RAM and Look-Up table.

The shadow RAM receives the configuration data of the corresponding CAB during the configuration or reconfiguration process. It is considered as a temporary memory before the data is put into the configuration RAM.

The configuration SRAM receives the data from the shadow memory and holds it. This memory is the responsible to maintain the data configuration for the behavior of the corresponding CAB. The configuration data transference occurs in one clock cycle. This configuration does not take effect until the FPAA receives the necessary indication through an internal or external signal. Usually, this indication is internal and automatic as will be indicated shortly.

The third region of SRAM is known as Look-Up table or LUT. It is a bank of memory which may contain replacement values for configuration SRAM of a determined CAB. With LUT memory and CAB's it is possible to construct arbitrary waveforms or any other non-linear functions.

In order to complete the information of the SRAM memory of the FPAA AN231 it is necessary to add the next words. It was stated previously that there is a dedicated cell (called RAM transfer cell) which can be configured to set up the time in which takes place the data transfers from shadow RAM to the configuration RAM. Additionally, it was mentioned that the operation of the RAM transfer cell can be configured in automatic, event-driven or clock sinchronization modes. These modes will be now explained.

In the automatic mode, the data is transferred to the configuration memory as soon as it has been completely received by the shadow memory.

In the event driven mode the transference of data will begin after a signal in high state is present. This signal can be either external or comes from an output of a comparator located inside the FPAA. Besides, it is possible to indicate the beginning of transference with both external and internal signals. Of course, in both cases, the external signal will come in by means of any input/output cell previously configured appropriately.

Finally, in the clock synchronization mode the transference of data will begin after all the clock presents in the FPAA have a simultaneous rising edge.

A.2.3 The Configurable Analog Blocks

The AN231 contains 4 configurable analog blocks. These blocks have switched-capacitor circuitry which can be configured according to the definition stored in the corresponding configuration memory.

Each CAB in the device contains the following elements:

- Two operational amplifiers.
- One comparator.
- Eight programmable capacitors.
- A successive approximation register.

Because of the aforementioned circuitry is based on switched-capacitor circuit blocks, the absolute values of the capacitors are not important but the ratios of them are. Thus, the response of the designed circuit is determined by these ratios and the value of the clock frequency that is used for the corresponding CAM. This characteristic allows that the circuit response implemented by a given CAM can be programmed.

In the following, the interface used to program this device will be described.

A.2.4 The Configuration Interface

Before using the AN231 FPAA, it must be configured first. This process is carried out through the configuration interface which contains a set of inputs and outputs.

The configuration interface is formed by the following inputs/outputs pins:

- Mode
- SCLK (Serial Clock).
- CS1b (Chip Select 1).
- CS2b (Chip Select 2).
- SI (Serial In).

- CFGFLGb (Configuration Flag).
- ACTIVATE.
- ERRb (Error).
- RESETb.
- SO (Serial Ouput).
- MEMCLK (Memory Clock).
- ACLK (Analog Clock).
- LCCb (Local Configuration Complete).
- RESETb.

The function of each of these pins will be now explained.

Mode. This input indicates if the FPAA will work as a master or a slave serial data device. When the Mode pin is connected to a high (V_{DD}) the FPAA will work as a master device. On the contrary, if this pin is tied to low (V_{SS}), the FPAA will work as a slave device.

SCLK. This input will be used as a serial data clock input if the device is working as a serial data slave. However, if it is working as a serial data master this input will be ignored. For the AN231, the maximum clock frequency should be 40 MHz.

MEMCLK. If the device is configured as master serial, then the output clock is taken from this pin.

ACLK. When the device is configured as master, then SCLK is ignored and all the necessary clock resources are taken from this input. The maximum value for ACLK in the AN231 should be 40 MHz.

SI. The serial input serves as the configuration data pin.

SO. When the device is configured as master serial, the data is sent through this pin.

LCCb. This pin is an indicator if the configuration of the device has been completed. During power-on-reset (on configuration time) its status is high. When the configuration is complete its status goes to low.

CS1b and **CS2b**. Both pins serve as chip select inputs and are active when they are in low status. Sometimes, the use of both pins is necessary above all when there are more than one FPAA device connected in cascade. In those cases, LCCb pin of one FPAA should be connected to the CS1b pin of the next device. In this manner, an FPAA only will be selected when the configuration of the previous device has been completed.

CFGFLGb. When a device is selected for configuration or reconfiguration, the CFGFLGb pin is driven to low. When the data transfer is complete this pin is driven to high. When a device senses that CFGFLGb is pulled to low then it will ignore the data on the SI pin. This characteristic is useful when there are more than one device connected in cascade. In those cases all the CFGFLGb pin should be connected to a common node.

ERRb. This pin is driven to low when an configuration error is detected. When a device senses that the pin ERRb is pulled to low, it will enter in reset state.

ACTIVATE. This pin is driven to low during the primary configuration. After that, this pin is sensed continuously. If this pin is driven to a high state after the FPAA has been configured, all the CAMs are activated.

RESET. This pin serves as a complete reset to the device. The FPAA will be reset when this pin is driven low.

A.2.5 The Clock Generation Sources and the Voltage References

It was mentioned previously that the block diagram shown in Figure A.1 does not include clock generation sources and voltage references, which are necessary in switched-capacitor circuits as it occurs with the circuit blocks present in the FPAA AN231.

Because of the ubiquitous nature of clock signals in a switched-capacitor circuitry, there are different treatments in the device AN231.

All the clock signals are taken from the clock signal obtained from the ACLK pin, called f_c. Firstly, this signal is divided into two system clocks: $sys1$ and $sys2$, forming two independent signal clocks. The period of either $sys1$ or $sys2$ is an integer

multiple of the period of the clock signal obtained from ACLK from 1 to 510. Next, both $sys1$ and $sys2$ are divided into 6 clocks. Each of these 6 clocks can use either $sys1$ or $sys2$ as their base. Again, the period of these 6 clock signals are multiple of either $sys1$ or $sys2$ from 0 to 510. Additionally, two of these six clocks can have an arbitrary phase delay from 0^o to 360^o. Given that there are two independent base clocks, it is possible to create two unrelated analog circuits with this device.

It is necessary to mention that the clock signal used for digital parts such as pins SO and SI are taken from ACLK too (divided by 16) when MODE is tied to high, that is, when device is working as serial master. In the other case, when the device is functioning as serial slave, the digital clock is taken from SCLK as was mentioned previously.

In relation to the voltages managed by the FPAA AN231, it is appropriate to mention the following. There are different notations for the different voltages inside the FPAA. All these voltages are derived from the bias voltage $V_{DD} = 3.3V$. Examples of these voltages are: VREF+, VREF- y VMR, which are used as reference voltages to CABs.

VREF+ stands for $V_{REFERENCE}$ positive and has a typical value of 2492 [mV].

VREF- stands for $V_{REFERENCE}$ negative and has a typical value of 501 [mV].

VMR stands for Voltage Mid Rail and has a typical value of 1500 [mV].

The voltages deviation output, the voltage temperature coefficient and the PSRR value can be found in the datasheet of the AN231 [30].

A.3 Analog Circuit Design with the FPAA AN231

The circuit design with the AN231 is done through a PC software interfaced with a development board called ADN231K04-DVLP3 AnadigmApex Development Board. The name of the software that runs in the PC is "AnadigmDesigner2". Through this program, in conjunction with the development board, the user can easily design, prototype and test his/her circuit. In this section, the procedure followed using this approach is shown.

The development board includes the chip FPAA-AN23104. It is shown in figure A.2. All its characteristic are mentioned in [31]. This board is connected to the PC by means of a USB or a RS-232 port.

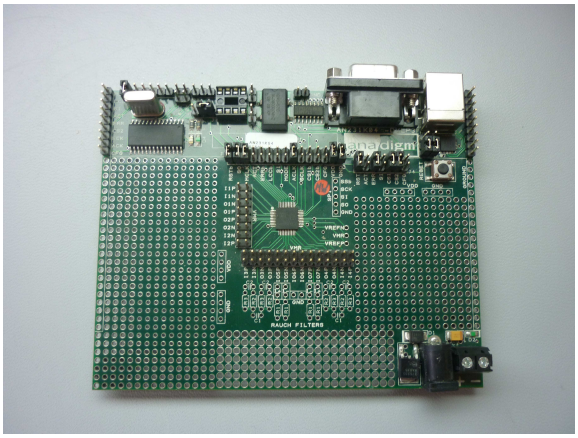


Figure A.2: AnadigmApex Development Board AN231K04-DVLP3.

Once the physical connection between the board and the PC is set up, the user should start the program AnadigmDesigner2. This program is able to recognize the type of device connected to the PC but permits the user to choose any other device if necessary. At this point, it is convenient to remark that the program AnadigDesigner2 can be only executed in Windows® environments and, thus, the controller of the aforementioned development board is for this kind of operating systems.

At a glance, the work flow of the circuit design can be summarized as follows. All the circuit design is done graphically by dragging and dropping the graphical equivalents of CAMs and connecting them. These CAMs will use the resources of one or more CABs. Once the desired connections of CAB's has been done, the software generates the configuration data. The configuration data can be used to program a master SPI device, which will be the responsible to program the AN231 through the SPI port. Even though the PC can be used to program directly the FPAA, the use of the master SPI device is preferred because it allows the use of the stand-alone operation of the device.

The main window of AnadigmDesigner2 is presented in the Figure A.3.

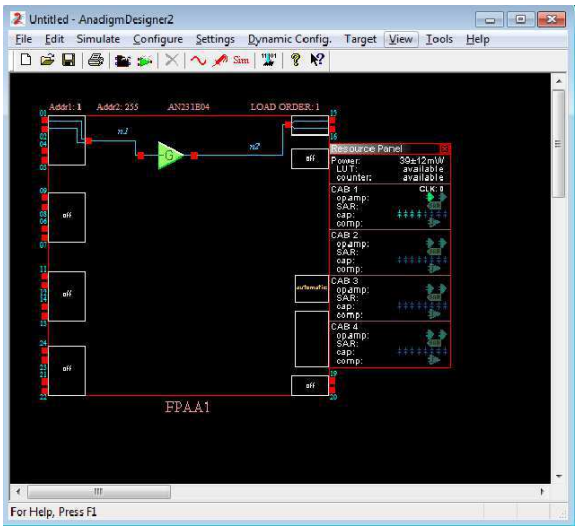


Figure A.3: Main window of AnadigmDesigner2 software

In the work area, the rectangle which represents the FPAA AN231 can be observed. Inside the rectangle the different input/output cells can be appreciated. Alongside the FPAA AN231 it is possible look the four CABs with its components.

The CAMs are dragged and dropped inside the rectangle to design any circuit. In figure A.3 it is depicted a CAM associated to an inverting amplifier.

Given that the design in AnadigmDesigner2 is done through a graphical interface with CAMs, the work flow turns out to be quite easy. This process is outlined below.

- Before beginning the design process it is necessary to configure the chip. This can be done in menu "Settings" $\rightarrow$ "Active chip settings..."
- Setting out all the clock signals to be used is the next step. This is done in the "Settings" menu, too.
- Once configured both the FPAA development board and the clock signals, the ports to be used need to be configured. This configuration is done in the work area.
- In this step, the designer can drag and drop CAMs to the work area and connect them in order to obtain the desired circuit.

- Finally, in order to obtain the desired circuit in the development board the data must be downloaded to the chip. This is done in the "Simulate" menu.

Appendix B

Description of the ChipKIT Max232 System Board

B.1 Main features of the ChipKIT System Board

The ChipKIT Max232, is a microcontroller-based development board with the following features:

- It contains the dsPIC Microchip® PIC32MX795F512.
- 80 Mhz 32-bit MIPS-based architecture.
- 512K Flash memory, 128K RAM memory.
- Compatible with many existing Arduino™ shields, code, and other resources.

Figure B.1 shows a picture of the development board.

B.2 Programming Language of the dsPic 32

As it was mentioned above, the chipKIT is compatible with Arduino™. In this sense, it can be programmed in the same way as it is done with the Arduino.

The programming language used in this development board is based in the C++ language.

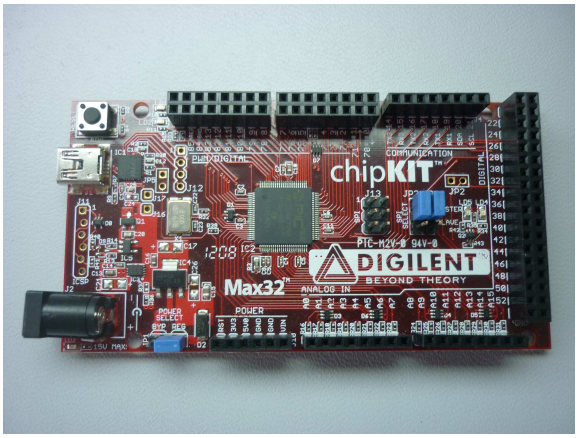


Figure B.1: The ChipKIT Max232 development board.

In the following, a brief introduction of this language is explained.

The structure of a program is shown in Figure B.2. In general terms a program consists by two main blocks: the setup block and the loop block.

Below, each block is briefly described.

The `setup()` Block

The `setup()` block consists in a set of functions which set the chipKIT up. Inside this block the input/output ports and all the peripherals to be used in the program should be configured.

The `loop()` Block

The `loop()` block can be considered as the main program. After the instructions in this routine have been executed according to the program flow, it will be called an indefinite number of times to carry out the same procedure.

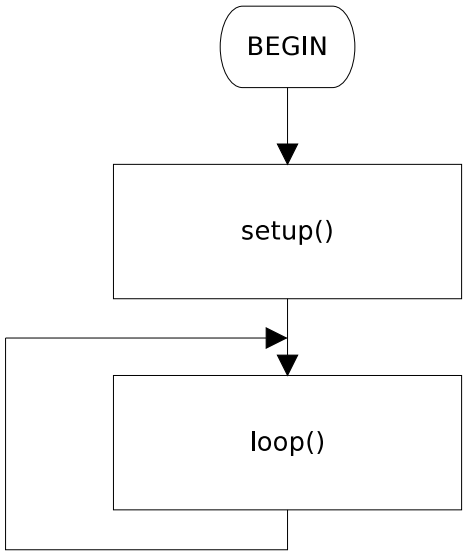


Figure B.2: Flow chart of the structure of a program using the ChipKIT development board.

Appendix C

Generation of Dynamic Data Sequences

In this appendix, the way to generate the dynamic data sequence used to implement a parameter-varying filter is shown.

As it was mentioned in Appendix A, one of the most important features of the FPAA AN231 is that it can be reconfigured dynamically, that is, a configuration sequence can be sent to the FPAA for modifying the operation of one or more of its CAMs on the fly. In other words, the operation of the CAMs which are subject to this process is not interrupted.

In the following, the design of a circuit in AnadigmDesigner2® and the manner of changing a particular circuit parameter of a given CAM are shown. Moreover, the traditional method is presented and it is compared with an alternative method used in this work.

C.1 Design with AnadigmDesigner2

As mentioned in Appendix A, designing with AnadigmDesigner2 is done through a graphical interface. In Figure C.1 an example of a circuit implemented with the aforementioned software is shown. It is possible to notice that the circuit contains a CAM and that two ports of the FPAA are used.

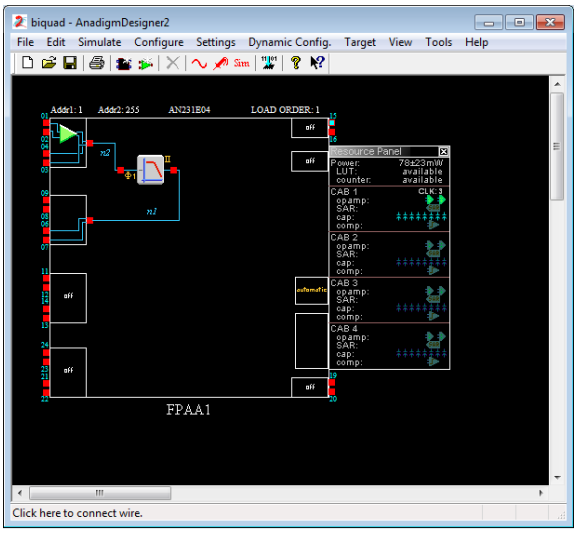


Figure C.1: Graphic Design with AnadigmDesigner2.

Configuration of a CAM

After the configuration of the development board is done, a CAM should be configured in order to implement a given circuit block. Each CAM has specific design parameters which must be indicated during the configuration process. For example, if a biquad filter is being configured, it is necessary to set up the following parameters:

- Filter Type (Lowpass, Highpass, etc.)
- Filter Topology (type II or type II)
- Input Sampling Phase
- Polarity (Inverting or not inverting)
- Corner Frequency
- Gain
- Quality Factor

The dialog window for this configuration process is shown in Figure C.2.

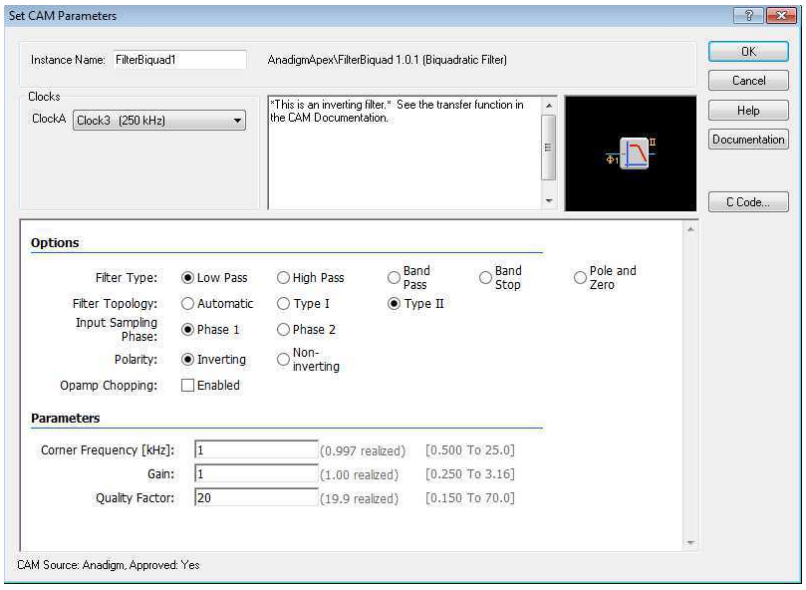


Figure C.2: Configuration of CAM biquad.

C.2 Obtaining the reconfiguration strings

It is assumed that the development board is connected correctly and a host system such as a PC or a microcontroller is sending the update configuration. In [31] other connections are shown. In the following, it will be explained how the data used by the host system to configure the development board may be obtained.

Once the primary configuration data has been obtained, there are two methods to get the update configuration data using the AnadigmDesigner2®. The first method is identified as the standard method and the second is the method used in this work.

In the standard method, the string of bytes containing the updated configuration for a given circuit is obtained through the option **State-Driven method** in the **Dynamic Config** Menu. In that case, by means of a series of options, the user should indicate the parameter to be varied, the range of variations and if this variation is linear or logarithmic. The program returns a file containing a set of strings of bytes. This data can be used to update the configuration of the FPAA when it is already operating.

The method used in this work is explained below. This method allows to modify the value of a circuit parameter of a given CAM, i.e., the value of a single capacitor. As it will be seen soon, the derivation of this method implies the use of inverse engineering techniques.

First of all it is necessary to find out the initial hexadecimal value of the capacitor related to the parameter to be varied. In order to do this, it is mandatory to know the set of relationships of the parameters with the capacitors in the CAB. This information is found in the documentation of each CAM.

Once identified the capacitor or capacitors related to the parameter of interest, the next step is to find out the value of this capacitor, that is the value of the capacitor according to the primary configuration. This value is obtained by studying the source code given by AnadigmDesigner2. This source code is generated by means of the option "Algorithm Method..." from the "Dynamic Config." Menu.

The two important source code files generated by AnadigmDesigner2 are named (by default): `ApiCode.c` and `ApiCode.h`.

By reviewing the `ApiCode.h` listing, it is possible to realize that there is a definition of a structure in C language, named (by default) `an_apiCapacitor`. This structure contains two byte variables called `bank` and `byteNum`.

Besides, in `ApiCode.h` is defined a C Array which contains 8 elements. Each element of this array is a structure of the type `an_apiCapacitor`. Thus, each element of this array contains the bank address and the byte number inside that bank which has the hexadecimal value of the capacitors used in a determined CAM. The name of this array is formed as the combination of the name given to the FPAA, the name given to the CAM and the word `Capacitor`. For example if the name of the FPAA is FPAA1 and the analyzed CAM's name is FilterBiquad1, the array will be denoted as `an_FPAA1_FilterBiquad1_Capacitor`.

On the other hand, in `ApiCode.c` each element of this array is defined. In that file is possible to find each capacitor value from the CAB. From this manner it is possible to find a determined hexadecimal value of a capacitor inside the primary configuration string.

From here, it is possible to change any capacitor value in order to do the reconfiguration of the FPAA. It is necessary to annotate that the user should take care of

generating a complete sequence of bytes which are to be sent to the FPAA. In other words, the user should include the control bytes, including the value of the capacitor to be modified, in order to send a complete update configuration.

An important advantage of this method is that the length of the update configuration string is shorter than the obtained by the standard method because in that string the information concerning the modification of a single capacitor is sent. Moreover, it is possible to take more control over the values of the capacitors inside the FPAA.

This method was used to implement a parameter-varying low pass filter which was presented in Chapter 6.

Appendix D

Source Codes Used in this Thesis

D.1 Source Code for the ChipKit

```
/*************************************************************
A parameter-varying filter implementation with AN231
*************************************************************/

// include the SPI library:
#if defined(__AVR__)
#include <avr/io.h>
#endif

#if defined(__PIC32MX__)
#include <p32xxxx.h>
#include <plib.h>
#endif

#include <DSPI.h>

// Some useful definitions
#define ACTIVATE 34
#define NO_ERROR 32
#define CS2B     36
#define PinSIG   40

```

```
23 #define LCCB      25
24 #define MODE      26
25 #define CS1B      27
26 #define CFG  28
27 #define test      29
28
29 #define SPI_PORT 1
30
31 // Instancing DSPI1 object called spi in SPI port1
32
33 DSPI1 spi;
34
35 /* SPI Port1 pines
36 DSPI1:
37 20  SDI1
38 3   SDO1
39 38  SCK1
40 74  SS1
41 */
42
43 #define PrimaryConfigSize 110
44
45 uint8_t PrimaryData[] =
46 {
47 0x00, 0x00, 0x00, 0x00, 0x00, 0xD5, 0xB7, 0x20,
48 0x01, 0x00, 0x01, 0xC1, 0xC4, 0x00, 0x0E, 0x20,
49 0x04, 0x00, 0x02, 0x21, 0x00, 0x00, 0x40, 0x00,
50 0x00, 0x51, 0xFF, 0x0F, 0xF1, 0x2A, 0xC2, 0x01,
51 0x01, 0x40, 0x2A, 0xDE, 0x01, 0x02, 0x02, 0xFF,
52 0x2A, 0xC7, 0x02, 0x01, 0x18, 0x2A, 0xD0, 0x02,
53 0x04, 0x10, 0xF0, 0x82, 0x05, 0x2A, 0xC0, 0x03,
54 0x08, 0xDC, 0xDB, 0x0B, 0xDC, 0xDB, 0x16, 0x0B,
55 0x0B, 0x2A, 0xD1, 0x03, 0x10, 0x20, 0x00, 0x20,
56 0x01, 0x31, 0x01, 0x82, 0x00, 0x30, 0x00, 0x10,
57 0x01, 0x12, 0x01, 0x81, 0x0F, 0x2A, 0x8D, 0x04,
58 0x13, 0x20, 0x00, 0x20, 0x00, 0x30, 0x00, 0x10,
59 0x00, 0x05, 0x00, 0x20, 0x00, 0x10, 0x00, 0x05,
60 0x01, 0x19, 0x01, 0x81, 0x2A, 0x00
```

```
61 };
62
63 int EstuvoBien;
64
65 #define Update_Size 9
66 #define StepSize    5
67 //unsigned char *Update[StepSize];
68 //uint8_t *Update[StepSize];
69 /* Cadenas de reconfiguacion a enviar */
70
71 /* C3 = 0xb0
72  * fc = 3987.937070
73  * Q = 4.988623
74  */
75 uint8_t UpdAux0[] = {
76 0xD5, 0x01, 0xC1, 0x82, 0x03, 0x01, 0xB0, 0x2A,
77 0x00
78 };
79 //Update[0] = UpdAux0;
80
81
82 /* C3 = 0x58
83  * fc = 3987.937070
84  * Q = 4.988623
85  */
86
87 uint8_t UpdAux1[] = {
88 0xD5, 0x01, 0xC1, 0x82, 0x03, 0x01, 0x58, 0x2A,
89 0x00
90 };
91 //Update[1] = UpdAux1;
92
93 /* C3 = 0x2c
94  * fc = 3987.937070
95  * Q = 4.988623
96  */
97
98 uint8_t UpdAux2[] = {
```

```
99  0xD5, 0x01, 0xC1, 0x82, 0x03, 0x01, 0x2C, 0x2A,
100 0x00
101 };
102 //Update[2] = UpdAux2;
103
104
105 /* C3 = 0x16
106  * fc = 3987.937070
107  * Q = 4.988623
108  */
109
110 uint8_t UpdAux3[] = {
111 0xD5, 0x01, 0xC1, 0x82, 0x03, 0x01, 0x16, 0x2A,
112 0x00
113 };
114 //Update[3] = UpdAux3;
115
116 /* C3 = 0x0b
117  * fc = 3987.937070
118  * Q = 4.988623
119  */
120
121 uint8_t UpdAux4[] = {
122 0xD5, 0x01, 0xC1, 0x82, 0x03, 0x01, 0x0B, 0x2A,
123 0x00
124 };
125 //Update[4] = UpdAux4;
126
127 /*************************************************************
128 *                        Setup Function
129 *************************************************************/
130
131 void setup()
132 {
133   // Setting pin to input/ouput
134       pinMode(ACTIVATE, INPUT);
135       pinMode(NO_ERROR, INPUT);
136       pinMode(CS2B, OUTPUT);
```

```
137         pinMode(PinSIG, OUTPUT);
138 
139         // Initializing SPI clock = 8 MHz
140         spi.begin();
141         spi.setSpeed(25000000);
142         // Loading Primary Configuration
143         EstuvoBien = CargarConfPrimaria();
144 }
145 
146 
147 /*************************************************************
148 *                       Loop Function
149 *************************************************************/
150 
151 void loop()
152 {
153         // Sending updates: Changing capacitor value C3 tau=500 [us].
154         digitalWrite(PinSIG, HIGH);
155         spi.transfer(Update_Size, UpdAux0);
156         delayMicroseconds(500);
157         spi.transfer(Update_Size, UpdAux1);
158         delayMicroseconds(500);
159         spi.transfer(Update_Size, UpdAux2);
160         delayMicroseconds(500);
161         spi.transfer(Update_Size, UpdAux3);
162         delayMicroseconds(500);
163         spi.transfer(Update_Size, UpdAux4);
164         delayMicroseconds(500);
165 
166         delay(30);
167         digitalWrite(PinSIG, LOW);
168         delay(30);
169 }
170 
171 int CargarConfPrimaria(void)
172 {
173         // Activating FPAA
174         digitalWrite(CS2B, LOW);
```

```
175         // Waiting for 2 ms in order to complete POR
176         delay(2);
177         // Sending Primary Configuration.
178         spi.transfer(PrimaryConfigSize, PrimaryData);
179 
180         // If all was fine configured: return 1, otherwise return 0.
181         if(digitalRead(ACTIVATE) && digitalRead(NO_ERROR))
182                 return(1);
183         else
184                 return(0);
185 }
```

D.2 Dynamically Generation of Data

```
1  #include <stdio.h>
2  #include <math.h>
3 
4  #define PI 3.141592654
5 
6  int main(void)
7  {
8          // Declaring used variables
9          int C1, C2, C3, C4, CA, CB, CA1, CA2, CB1, CB2, i;
10         double f0, G, Q;
11         long fc = 250000;
12 
13         // Initial capacitor values
14         CB2 = 0xDC;  // CB2
15         CB1 = 0xDB;  // CB1
16         C3 = 0x0B;  // C3
17         CA2 = 0xDC;  // CA2
18         CA1 = 0xDB;  // CA1
19         C4 = 0x16;  // C4
20         C2 = 0x0B;  // C2
21         C1 = 0x0B;  // C1
22 
23         CB = CB1 + CB2;
24         CA = CA1 + CA2;
```

```
25
26          /* Original value of parameters */
27          printf("\n\n**** Valores originales **** \n");
28          printf("%4s %15s %15s %15s\n", "C3", "Frecuencia", "Ganancia", "Q");
29          f0 = (fc/(2.0*PI))*sqrt(1.0*C2*C3/(CA*CB));
30          G = C1/C2;
31          Q = (1.0/C4)*sqrt(CA*CB*C2/C3);
32          printf("%#04x %15lf %15lf %15lf\n", C3, f0, G, Q);
33
34          /* Varying C3 to a high value shifting 4 bit to the left.  */
35          printf("\n\n*** Incrementando el valor de C3 hasta un valor maximo ***\n");
36          printf("%4s %15s %15s %15s\n", "C3", "Frecuencia", "Ganancia", "Q");
37          C3 = C3 << 4;
38          f0 = (fc/(2.0*PI))*sqrt(1.0*C2*C3/(CA*CB));
39          G = C1/C2;
40          Q = (1.0/C4)*sqrt(CA*CB*C2/C3);
41          printf("%#04x %15lf %15lf %15lf\n", C3, f0, G, Q);
42
43          /* Exponentially decreasing the value of C3.  */
44          printf("\n\n*** Decrementando el valor de C3 exponencialmente ***\n");
45          printf("%4s %15s %15s %15s\n", "C3", "Frecuencia", "Ganancia", "Q");
46          for (i=1; i<=4; i++){
47                  C3 = C3 >> 1;
48                  //printf("Nuevo valor C3 = %#x\n", C3);
49                  f0 = (fc/(2.0*PI))*sqrt(1.0*C2*C3/(CA*CB));
50              G = C1/C2;
51              Q = (1.0/C4)*sqrt(CA*CB*C2/C3);
52                  printf("%#04x %15lf %15lf %15lf\n", C3, f0, G, Q);
53          }
54
55          return 0;
56  }
```

Bibliography

[1] G. Daryanani, *Principles of Active Network. Synthesis and Design.* United States of America: John Wiley & Sons, 1976.

[2] R. Kaszyński, "A proposal of non-stationary low-pass Chebyshev's filters," in *Emerging Technologies and Factory Automation, 1996. EFTA'96. Proceedings., 1996 IEEE Conference on*, vol. 2. IEEE, 1996, pp. 759–762.

[3] M. Á. Gutiérrez de Anda and I. Meza Dector, "A second-order low pass parameter-varying filter based on the interconnection of first-order stages," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 58, no. 8, pp. 1840–1853, August 2011.

[4] R. Kaszyński, "Stability of parametric, analog low-pass filters," in *Emerging Technologies and Factory Automation, 1999. Proceedings. ETFA'99. 1999 7th IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 579–582.

[5] ——, "The problem of non-zero initial conditions in filters of the constant component," in *Circuits and Systems, 2002. APCCAS'02. 2002 Asia-Pacific Conference on*, vol. 2. IEEE, 2002, pp. 379–384.

[6] ——, "Properties of analog systems with varying parameters [averaging/low-pass filters]," in *Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on*, vol. 1. IEEE, 2003, pp. I–509.

[7] ——, "The parametric filter of signal constant component," in *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 1. IEEE, 2001, pp. 200–203.

[8] R. Kaszyński and J. Piskorowski, "Bessel filters with varying parameters," in *TENCON 2004. 2004 IEEE Region 10 Conference*. IEEE, 2004, pp. 641–644.

[9] ——, "Dynamic compensation of phase distortion in time-varying Butterworth filters," in *Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE*. IEEE, 2006, pp. 1421–1424.

[10] ——, "Selected structures of filters with time-varying parameters," *Instrumentation and Measurement, IEEE Transactions on*, vol. 56, no. 6, pp. 2338–2345, 2007.

[11] J. Piskorowski, "Some aspects of dynamic reduction of transient duration in delay-equalized Chebyshev filters," *Instrumentation and Measurement, IEEE Transactions on*, vol. 57, no. 8, pp. 1718–1724, 2008.

[12] P. Pietrzak, "Fast filtration method for static automatic catchweighing instruments using a nonstationary filter," *Metrology and Measurement Systems*, vol. XVI, no. 4, pp. 669–676, 2009.

[13] M. A. Gutiérrez de Anda, I. Meza Dector, J. C. Sánchez García, R. Kaszyński, and J. Piskorowski, "A first-order parameter-varying filter using dynamic translinear techniques," in *Methods and Models in Automation and Robotics*, vol. 14, no. 1, 2009, pp. 336–341.

[14] A. L. Mota Rodríguez, "Diseño de filtros pasa bajas con parámetros variantes en el tiempo basado en OTAs y capacitores," Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, February 2012.

[15] J. R. Rodríguez Fernández, "Filtro analógico pasa bajas de segundo orden con parámetros variantes en el tiempo," Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, February 2012.

[16] M. Hernández Sandoval, "Diseño de bloques de circuito para la implementación de filtros rechazabanda con parámetros variantes en el tiempo," Master's thesis, Instituto Nacional de Astrofísica, Óptica y Electrónica, December 2010.

[17] A. Piwowar, J. Walczak, and P. Baron, "Analogue realisation of first and second order parametric filters," in *International Conference on Fundamentals of Electrotechnics and Circuit Theory*, 2010.

[18] K. Ogata, *Modern Control Engineering*, 5th ed. Prentice-Hall, 2010.

[19] L. A. Zadeh and C. A. Desoer, *Linear System Theory. The State Space Approach*, ser. System Science. McGraw-Hill, 1963.

[20] J. C. Willems, *Dynamics Reported*. John Wiley & Sons Ltd and B.G. Teubner, 1989, vol. 2, ch. 5. Models for Dynamics, pp. 171–269.

[21] M. Liu, *Demystifying Switched-Capacitor Circuits*. Elsevier Inc., 2006.

[22] B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2001.

[23] P. E. Allen and E. Sánchez-Sinencio, *Switched-Capacitor Circuits*. Nostrand Reinhold Company Inc., 1984.

[24] D. J. Allstot, R. W. Brodersen, and P. R. Gray, "An electrically-programmable switched capacitor filter," *Solid-State Circuits, IEEE Journal of*, vol. 14, no. 6, pp. 1034–1041, 1979.

[25] F. P. E. and K. R. Laker, "A family of active switched-capacitor biquad building blocks," *The Bell System Technical Journal*, vol. 58, no. 10, pp. 2235–2269, December 1979.

[26] K. Martin, "Improved circuits for the realization of switched-capacitor filters," *Circuits and Systems, IEEE Transactions on*, vol. 27, no. 4, pp. 237–244, 1980.

[27] I. Meza Dector, "Diseño de filtros continuos pasabajos con parámetros variantes en el tiempo," Master's thesis, Instituto Politécnico Nacional, 2010.

[28] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2000.

[29] F. A. S. Ltd., "Totally reconfigurable analog circuits," March 1999.

[30] *AN231E04 Datasheet*, Anadigm, Inc., 2008.

[31] *AN231K04-DVLP3-Anadigm Development Board*, Anadigm, Inc., 2010.

[32] *AnadigmApex dpASP Family User Manual*, Anadigm, Inc., 2006.