Analysis of the quantization effects in the implementation of numerical filters

Sim Simona-Daiana
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
sim.simonadaiana@gmail.com

Zsófia Lendek
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
zsofia.lendek@aut.utcluj.ro

Petru Dobra
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
petru.dobra@aut.utcluj.ro

***Abstract*— The paper analyzes the efficiency of fixed-point implementation and the effect of quantization of coefficients and signals in the implementation of numerical filters on the STM32 Nucleo-64P development board. Discretization methods are also analyzed. By modelling the effects of quantization, it is possible to indicate how the system responds. Proper quantization can increase the performance. We model the quantization error as stochastic noise. The results show that the quantization of the coefficients and the fixed-point processing minimally change the response of the digital filter compared to the analog one, thus achieving very good results at high sampling frequencies. Also, through this analysis, the most efficient implementation can be chosen, taking into account the system and the characteristics of the development board.**

Keywords—IIR filter, discretization, quantization, STM32 Nucleo64P development board

I. INTRODUCTION

Analog filters are electrical circuits that eliminate undesired components or characteristics from a signal. They are common [1] in instrumentation, electronics, and communication systems, particularly in signal and image processing. A filter [2] helps to attenuate undesired components such as noise, interference, and distortion. The phase properties and relative amplitude of the various frequency components are altered by the ideal filter, whose gain is fully dependent on the signal's frequency. This paper compares two types of filters: analog and digital ones.

Processing signals requires methods that are relatively independent of the type of signal under consideration. The objectives of signal processing are to extract information, analyse the data, enhance, synthesise and compress the signal, transmit it and, finally, understand the information it contains. In an integrated information processing chain, these objectives are intertwined and in complex interaction. The Fourier transform is at the basis of the study of discrete time systems and constitutes the transition from discrete time to discrete frequency. The application of techniques has a certain degree of abstraction, and in concrete cases requires a body of theoretical knowledge. The goal of this paper is to bridge the gap between theory and practice through stochastic analysis of discretization and quantization of signals and coefficients.

Analog filters are commonly used in electronics and are considered a basic element of signal processing. They are used to [3] separate audio signals, combine multiple conversations on a single channel, to select a specific frequency etc. Digital filters [4] have been studied in the literature since the 1960s, and over time they have shown significant improvements. With the evolution of the digital filters, solutions appeared for the most common problems: memory, processing time, computational errors, etc. Digital filters have several advantages over analog ones. Once they are programmed, the programs in digital filters can be easily modified by rewriting their algorithms. This also allows digital filters to adapt [5]: filter parameters can be changed over time as the input signal changes. Digital filters are also capable of handling very low frequencies.

The challenge related to the implementation of filters is obtaining a digital filter that has the same characteristics as a classic analog one. This problem was noted in our past work [6], where a floating-point implementation at a high sampling frequency could not be obtained due to the limitations of the development board. Therefore, in this paper we analyze the effect of quantization. In order to optimize both the memory used and the processing time, the implementation uses coefficient quantization [7] and fixed-point computation. Quantization can have undesirable effects [5] such as reduced accuracy, impossibility to reconstruct the signal, characteristic noise, etc. This is why it is important to analyze the quantization. We want to achieve a result that is satisfactory in terms of performance, but respects a required accuracy. The aim is to obtain a digital filter with performance comparable to an analog one. In this respect the question of implementation and physical capabilities arises, which can be fulfilled by the chosen development board. The ideal implementation is to use a few resources that will achieve a good result. The classic implementation of digital filters involves floating point computation. However, this raises the standard to which the development board must be set, thus increasing the implementation cost. Working in fixed point greatly reduces processing time. Our goal is to combine an optimal implementation in terms of cost and result. In this sense, this article demonstrates how specifically quantization of coefficients and fixed-point work influence the implementation of numerical filters. An analysis of the quantization effect is also presented in [8], but for FIR (Finite Impulse Response) filters with a direct representation. Our article considers the effects on an IIR (Infinite Impulse Response) filter and presents its response in comparison with its analog counterpart. The problem of sensitivity and the choice of the optimal discretization method depending on the chosen system is also discussed. Through the results obtained from the calculations performed in the stochastic quantization analysis we arrive at an acceptable result whereby the output is influenced to the fourth decimal place. A punctual study is made depending on the capabilities of the development board. The sensitivity with respect to the coefficients is analyzed to highlight the stability of the process. Through these calculations and analyses we can formulate a certain expectation from the physical implementation and predict its response. Also, the analysis enables us to choose the appropriate fixed-point representation. Thus, we can achieve an effective implementation with lower costs.

We compare the analog filter with the quantized numerical filter. The steps taken are: modelling the analog filters, analysis of the discretization of the model. After discretization the coefficients are quantized. Finally, the quantization analysis of the signals and coefficients will be presented, then the algorithm with the quantized coefficients will be implemented on STM32 Nucleo-64P development board and the results will be compared to the analog filter. It is important to mention that the chosen filter and sampling frequency cannot be implemented on this board in floating-point computation.

The structure of the paper is as follows: Section II presents the analog filter and its modelling, Section III compares the discretized models, in Section IV notions of quantization are described and the quantized model is obtained. Section V presents the stochastic modelling of the quantization effect. Section VI presents the implementation results. Section VII concludes the paper.

II. Case Study and Problem Statement

In this paper, for the analysis we consider the low-pass filter [1] with amplifier in Fig. 1.

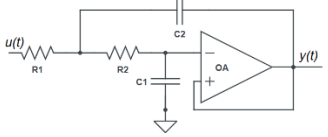

Fig. 1. Low-pass filter

The mathematical model is obtained from: $u(t) = u_{R1}(t) + u_{R2}(t) + u_{C1}(t)$, $u_{R2}(t) = u_{C2}(t)$, $i_{R1}(t) = i_{R2}(t) + i_{C2}(t)$, $i_{R2}(t) = i_{C1}(t), y = u_{C1}(t)$.

We denote: $x_1(t) = u_{C1}(t)$, $i_{C1}(t) = C_1\dot{x}_1(t)$, $x_2(t) = u_{C2}(t)$, $i_{C2}(t) = C_2\dot{x}_2(t)$, leading to the following differential equations:

$$\dot{x}_1(t) = \frac{1}{R_2C_1}\, x_2(t) \tag{1}$$

$$\dot{x}_2(t) = -\frac{1}{R_1C_2}\, x_1(t) - \frac{1}{C_2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right) x_2(t) + \frac{1}{R_1C_2} u(t) \tag{2}$$

The state-space model is:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{3}$$

with

$$A = \begin{pmatrix} 0 & \frac{1}{R_2C_1} \\ -\frac{1}{R_1C_2} & -\frac{1}{C_2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right) \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ \frac{1}{R_1C_2} \end{pmatrix} \tag{4}$$

$$C = (1\ 0) \qquad D = 0$$

The transfer function of the system is given in (5).

$$H(s) = \frac{\frac{1}{R_1R_2C_1C_2}}{s^2 + \frac{1}{C_2}\left(\frac{1}{R_1} + \frac{1}{R_2}\right)s + \frac{1}{R_1R_2C_1C_2}} \tag{5}$$

For our case study, the values of the parameters are: $R_1 = 3.9k\Omega$, $R_2 = 6.8k\Omega$, $C_1 = 10nF$, $C_2 = 22nF$, thus (5) becomes:

$$H(s) = \frac{1.714 \cdot 10^8}{s^2 + 1.834 \cdot 10^4 s + 1.714 \cdot 10^8} \tag{6}$$

In the following, we will present several discretization methods and analyze which of them produce results similar to the original. We will also study the problem of sensitivity of the system eigenvalues. Then we will analyze how the discretization and quantization will affect this filter.

III. Discrete-time models

A. General description

A signal [9] is a physical quantity that depends on one or more independent variables such as time, distance, temperature or pressure. A discrete-time signal [10], [11] is a sequence or a series of signal values defined in discrete points of time. Examples of discrete-time signals are logged measurements, the input signal, the simulated response of a dynamic system, etc. These discrete points of time can be denoted by t_k where k is an integer time index. The distance in time is the time-step or sampling period, denoted by h. Thus, $h = t_k - t_{k-1}$, and the time series can be written as: $x(t_k) = x(kh) = x(k)$.

An analog-to-digital converter (ADC) [12] converts an analog (continuous, infinitely variable) signal into a digital (discrete time, discrete amplitude) signal. ADCs perform this conversion through a form of quantization - mapping a contiguous set of values to a smaller (enumerable) set of values [13], usually by rounding. Therefore, the analog-to-digital conversion process will always result in some level of noise or error. The number of bits used to represent this analog voltage value depends on the resolution of an A/D converter. Converters generally have a resolution of 8 or 12 bits, which scale to 255 ($2^8 - 1$) or 4095 ($2^{12} - 1$) values, respectively. A digital-to-analogue converter (DAC) is [14] a data converter that generates an analog output from a digital input. The performance of a DAC [9] is determined by the number of samples it can process and the number of bits used in the conversion process. Overall, the quality and reproduction of a signal is influenced by the resolution of the ADC and DAC, and the processing time of the microcomputer. This computation time is strongly influenced by floating point (float, double) or fixed-point computation.

There are many methods for discretizing continuous-time systems among which: *first order hold* (foh), *Euler*, *Tustin*, modified *Tustin*, *zero order hold* (zoh), *least-squares*, etc. Discrete-time models are generally used in analysis and control of control systems.

B. Case Study

In order to make the implementation as stable as possible, it is important to analyse several types of discretization methods. The frequency response of system (6) using different discretization method (*foh*, *zoh*, *Tustin*) can be seen in Fig. 2: the closest response to the original one (ORG in the figure) is the one obtained by the *foh*, which will be used for implementation. In the case of *zoh* discretization it is observed that at high frequencies it no longer responds properly.

The sampling period is the first factor influencing signal reconstruction. This should be chosen taking into account Shannon's theorem. The sampling period chosen for discretizing our system is: 25 µs (40 kHz). Two discretization methods will be exemplified: *first order hold* and *Tustin*. The discrete-time approximation of (6) using *first order hold* and *Tustin*, respectively, are described in (7) and (8).

$$H_{foh}(z^{-1}) = \frac{0.0159 + 0.0566\, z^{-1} + 0.01264\, z^{-2}}{1 - 1.547\ z^{-1} + 0.6322\, z^{-2}} \tag{7}$$

$$H_T(z^{-1}) = \frac{0.02132 + 0.04264\, z^{-1} + 0.0213\, z^{-2}}{1 - 1.55\, z^{-1} + 0.635\, z^{-2}} \quad (8)$$

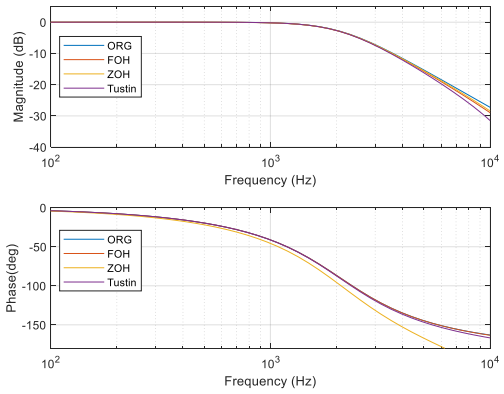

Fig. 2. Simulation for different discretization methods

The matrices of the state-space model obtained using *foh* is given in (9). They will be used in the following sections for the analysis of quantization.

$$A = \begin{pmatrix} 0.9542 & 0.2897 \\ -0.2296 & 0.5929 \end{pmatrix} \quad B = \begin{pmatrix} 0.812 \\ 0.1752 \end{pmatrix}$$
$$C = (1 \quad 0) \qquad D = (0.0159) \quad (9)$$

The transfer function (8) can be rewritten as (11). Next, the problem of the sensitivity [15] of the eigenvalues of the system to the quantization effect, which manifests itself in variations of the coefficients of the discrete-time transfer function, is analyzed by studying the root locus.

$$H(z) = \frac{b_2 z^2 + b_1\, z + b_0}{z^2 + a_1\, z + a_0} = \frac{N(z)}{D(z)} \quad (10)$$

$$H_{foh}(z) = \frac{0.0159 z^2 + 0.05662\, z + 0.01264}{z^2 - 1.547\, z + 0.6322} \quad (11)$$

The two coefficients a_1 and a_0 ensure stability of the system, being the coefficients of the denominator, further denoted by D(z). The system characterized by (10) is reconfigured in the form of a system with negative reaction, which on the direct path presents a transfer factor defined by the parameter (a_1 or a_0) in relation to which the sensitivity [16], [17] is studied. The reaction path has the transfer function $Hr(z) = \frac{R(z)}{S(z)}$ determined so that the denominator in (11) coincides with the denominator of the closed loop. Consider in general the transfer function $H(z) = \frac{N(z)}{D(z)}$, where $D(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_i z^i + \cdots + a_0$. To analyze the effect of the variation of the coefficient a_i on the stability of the $H(z)$, i.e., the roots of $D(z)$, we rewrite $D(z) = z^n + a_{n-1}z^{n-1} + \cdots + a_{i+1}z^{i+1} + a_{i-1}z^{i-1} + \cdots a_0 = 0$ or $Hr(z) = \frac{z^i}{z^n + a_{n-1}z^{n-1} + \cdots + a_{i+1}z^{i+1} + a_{i-1}z^{i-1} + \cdots a_0} = -1$ corresponding to the feedback system presented in Fig. 3. Thus, we analyze the loci of the roots of this system. For the specific case of (11), we will analyze how the variation of the coefficients in the denominator (a_1 or a_0) affects the stability of the system. For a_0 we have $Hr(z) = \frac{1}{z^2 - 1.547\, z}$ and for a_1, $Hr(z) = \frac{-z}{z^2 + 0.6322}$.

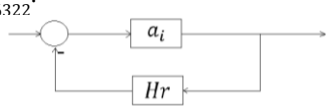

Fig. 3. Feedback structure

We will analyze this sensitivity for the function discretized by *Tustin* (equation (7)) and *foh* (equation (8)). A maximal variation of 30% was considered, meaning that $a_1 \epsilon [-2.0111; -1.0829]$ with nominal value -1.5470 and $a_0 \epsilon [0.4425; 0.8219]$ with nominal value 0.6320. The values of the parameters were varied with a constant step in the mentioned intervals.

Figures 4 and 5 show the effect of the variation of the parameters for the function obtained by *foh*. The circles represent the nominal values of the system, and the squares are the extremes at which the coefficient was varied.

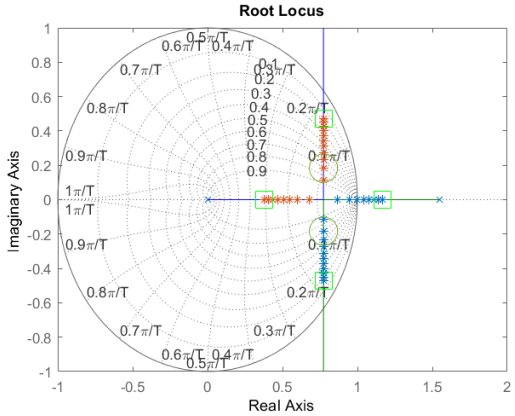

Fig. 4. Sensitivity of the root locus for a_0

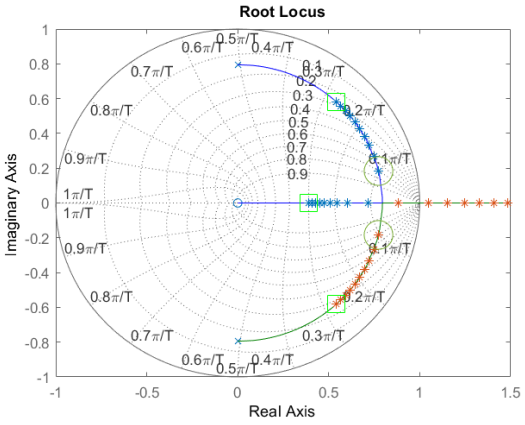

Fig. 5. Sensitivity of the root locus for a_1

As can be seen from Table I, the sensitivity of systems (7) and (8) with respect to the variation of parameters a_0 and a_1 is completely different. A higher sensitivity can be observed in the case of a_1. A change of a_0 with -15% makes the system unstable for both *foh* and *tustin* discretization. In the case of a_1, the system becomes unstable starting with a 15% increase in both cases. When both a_1 and a_0 vary at the same time, the stability domain is shown in Fig. 6. The white space represents the stability zone of the system, and the stars represent points where the system is unstable.

IV. Quantizantion

A. Quantizing signals and values

Quantization [5], [18] occurs in many places in DSP (Digital Signal Processing). Fixed-point programming leads to quantization, as the result of the calculations are truncated or rounded depending on the processor. Fixed-point quantization error analysis is usually based on

simplifying assumptions [19]. Another type of quantization analysis based on random variable modeling is presented in [19] for 8-bits. In our case, the analysis of the coefficients takes place for 16-bits as a calculation and using classical methods which are detailed in the following. "n" bit ADC converters [12] use uniform quantization to transform analog signals into digital ones by 2^n quanta (noted q); a quantum [13] is the difference between two adjacent steps. The disadvantages of quantizing signals are the addition of quantization noise on top of the signal obtained and the impossibility of reconstructing the original signal.

TABLE I. *VARIATION OF THE COEFFICIENTS A1 AND A0 FOR THE FUNCTIONS DISCRETIZED BY FOH AND TUSTIN*

a_0 , a_1	foh	Tustin
a_0 + 30%	$a_0 = 0.8219$ $\hat{s}_{01,2} = 0.774 \pm 0.473j$	$a_0 = 0.8255$ $\hat{s}_{01,2} = 0.775 \pm 0.476j$
a_0 + 15%	$a_0 = 0.7270$ $\hat{s}_{01,2} = 0.774 \pm 0.359j$	$a_0 = 0.73025$ $\hat{s}_{01,2} = 0.775 \pm 0.362j$
a_0 + 0%	$a_0 = 0.6320$ $\hat{s}_{01,2} = 0.774 \pm 0.189j$	$a_0 = 0.635$ $\hat{s}_{01,2} = 0.774 \pm 0.188j$
a_0 - 15%	$a_0 = 0.5374$ $s_{01} = 1.02$ $s_{01} = 0.527$	$a_0 = 0.53975$ $s_{01} = 1.02$ $s_{01} = 0.529$
a_0 - 30%	$a_0 = 0.4425$ $s_{01} = 1.17$ $s_{01} = 0.379$	$a_0 = 0.4445$ $s_{01} = 1.17$ $s_{01} = 0.38$
a_1 + 30%	$a_1 = -2.0111$ $s_{01} = 1.62$ $s_{01} = 0.389$	$a_1 = -2.015$ $s_{01} = 1.63$ $s_{01} = 0.390$
a_1 + 15%	$a_1 = -1.779$ $s_{01} = 1.29$ $s_{01} = 0.488$	$a_1 = -1.7825$ $s_{01} = 1.29$ $s_{01} = 0.496$
a_1 + 0%	$a_1 = -1.547$ $\hat{s}_{01,2} = 0.773 \pm 0.183j$	$a_1 = -1.55$ $\hat{s}_{01,2} = 0.774 \pm 0.186j$
a_1 - 15%	$a_1 = -1.315$ $\hat{s}_{01,2} = 0.675 \pm 0.417j$	$a_1 = -1.3175$ $\hat{s}_{01,2} = 0.659 \pm 0.448j$
a_1 - 30%	$a_1 = -1.0829$ $\hat{s}_{01,2} = 0.541 \pm 0.582j$	$a_1 = -1.085$ $\hat{s}_{01,2} = 0.542 \pm 0.584j$

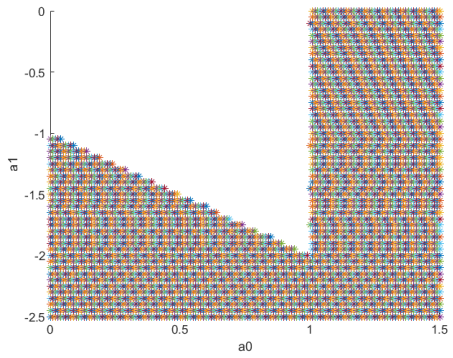

Fig. 6. Sensitivity in relation to both coefficients

An analog signal is quantized by discretizing the amplitude of the signal using a series of quantization levels. This requires [20] determining the least significant bit (LSB) when the analog input voltage is in the lowest subrange of the input voltage range. Both sampling and quantization lead to loss of information. The quality of the quantizer output [13] depends on the number of quantization levels used. The capacity of registers is limited, therefore parameters are also quantized. Quantizing the filter coefficients [21] also changes the values of the poles and zeros, leading to a deviation in the frequency response of the system.

B. *Case study*

We quantize the coefficients over 16 bits out of which 1 bit is reserved for the sign. The method is based on the magnitude truncation presented in [22]. Note that the largest value is *1.547* which can be approximated by 2^1. This leaves 14 bits and we multiply all the coefficients by 2^{14}(1 bit sign,1 bit integer, 14 bits fractional). The coefficients are rounded to the nearest quanta. Through this quantization, the accuracy of the coefficients is maintained up to the fourth decimal place as can be seen in the Table III. Having the discrete-time transfer function previously obtained by *foh* in (8), the floating point and the quantized values are given in the Tables II and III. For testing, the input signal was built as the sum of several sine signals of different frequencies. The input signal and the response of the quantized filter can be seen in Fig. 7. This is the result we expect even after the implementation on the development board. The difference between the response of the original and the discretized and quantized system is of the order 10^{-4} as can be seen in Fig.8.

TABLE II. *COEFFICIENTS IN FLOATING POINT REPRESENTATION*

k	b_k	a_k
0	0.015899494594913	1
1	0.056624683457761	-1.547075346446494
2	0.012639602532820	0.632239127031989

TABLE III. *COEFFICIENTS IN FIXED-POINT REPRESENTATION*

k	b_k	b_k *real*	a_k	a_k *real*
0	260	0.015869	16384	1
1	928	0.056640	-25347	-1.547058
2	207	0.012632	10359	0.632263

The quantized transfer function of the system is:

$$H_{foh_q}(z) = \frac{260 + 928\ z^{-1} + 207\ z^{-2}}{16384 - \ 25347\ z^{-1} + 10359\ z^{-2}} \qquad (12)$$

V. STOCHASTIC MODELLING OF QUANTIZATION EFFECTS

A. *Modelling of quantization effect*

For a discrete time system [23], [13] (obtained by sampling), we consider the errors due to discretization, quantization, fixed-point computation, etc., as noise affecting the system, modelled by random variables. The noise corresponding to the quantization errors in the input is modelled by the random variable $\varepsilon_u(k,w)$; output noise due to the DAC modelled by $\varepsilon_y(k,w)$ and the noise due to the computation errors by $\varepsilon_c(k,w)$. Under these conditions the state space model of the discrete-time system can be written as (13) and (14) and is shown in Fig. 9.

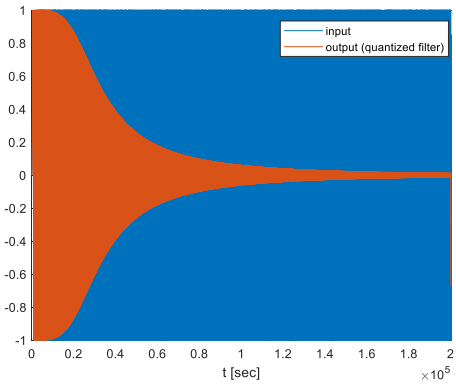

Fig. 7. Simulation of the response of the quantized filter

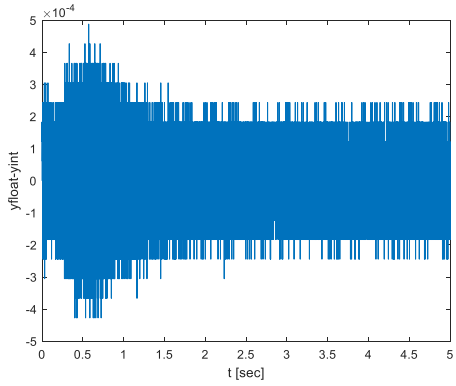

Fig. 8. Difference between floating and fixed-point output

$$x(k+1) = Ax(k) + B\big(u(k) + \varepsilon_u(k)\big) + \varepsilon_c(k) \quad (13)$$

$$y(k) = Cx(k) + D\big(u(k) + \varepsilon_u(k)\big) + \varepsilon_y(k) \quad (14)$$

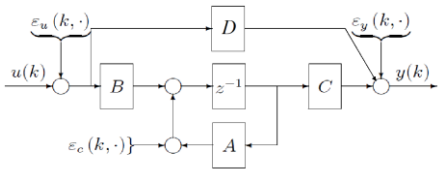

Fig. 9. Stochastic modeling of quantization effects

At each sampling time k, the noise model is considered to be of the form:

$$\varepsilon(k,\cdot) = \omega(k,\cdot) + \mu_\varepsilon \quad (15)$$

where $\varepsilon(k,\cdot)$ represents the realization of the stochastic process $\omega(k,\cdot)$ at sample k. In case of rounding, the random variable $\omega(k,\cdot)$ has zero mean $\mu_\varepsilon = 0$ and variance $\frac{q^2}{12}$, and in the case of the quantization by truncation $\mu_\varepsilon = -\frac{q}{2}$. In what follows we will use rounding for ADC (input) and truncation for output and calculations.

Under the realistic assumption that the output noise is uncorrelated with the state variables, it follows that the expected value of the noise at the output is zero and the covariance is:

$$\begin{aligned} &E\{y(k)y^T(k)\} = C \cdot E\{x(k)x^T(k)\} \cdot C^T \\ &+ D \cdot E\{\varepsilon_u(k,\cdot)\varepsilon_u^T(k,\cdot)\} \cdot D^T + E\{\varepsilon_y(k,\cdot)\varepsilon_y^T(k,\cdot)\} \\ &= CP_{xx}(k)C^T + DR_u(0)D^T + R_y(0) \end{aligned} \quad (16)$$

where $P_{xx}(k)$ represents the state covariance matrix and $R_u(0)$ and $R_y(0)$ are the autocorrelation functions of the input and of the output noise.

The computation of the state covariance matrix $P_{xx}(k)$ can be done recursively:

$$\begin{aligned} &P_{xx}(k+1) = E\{Ax(k)x^T(k)A^T + B\omega_u(k,\cdot)B^T + \\ &+\omega_c(k,\cdot){\omega_c}^T(k,\cdot)\} \\ &= AP_{xx}(k)A^T + BR_u(0)B^T + R_c(0) \end{aligned} \quad (17)$$

where $R_c(0)$ is the covariance of the noise due to computational errors.

For a stable discrete-time system, the state covariance matrix will converge to a constant value:

$$\lim_{k\to\infty} P_{xx}(k) \to P_\infty \quad (18)$$

B. Case study

Next, we will calculate the noise on the output for the discretized system (9). For the implementation, see Section VI, we will use the Nucleo64-P development board from STM32 which has in its composition an ADC and a DAC. On the input of the system there is a 12-bit ADC with analog input in the range 0-3.3 V. The computing system used is based on a 16-bit fixed-point digital signal processor. On the system output there is a 12-bit DAC with analog output in the range 0-3.3V.

The mean values μ_x and the variance σ_x^2 are:

- For the input noise $\varepsilon_u(k,\cdot)$:

$$\mu_u = 0 \quad (19)$$

$$\sigma_u^2 = \frac{\left(\frac{3.3}{2^{12}}\right)^2}{12} = 5.412 \cdot 10^{-8} \quad (20)$$

$$R_u = 5.412 \cdot 10^{-8} \quad (21)$$

- For noise due to computational errors $\varepsilon_c(k,\cdot)$:

$$\mu_{c_{1,2}} = \frac{\frac{1}{2^{15}}}{2} = 1.526 \cdot 10^{-5} \quad (22)$$

$$\sigma_{c_{1,2}}^2 = \frac{\left(\frac{1}{2^{15}}\right)^2}{12} = 7.761 \cdot 10^{-11} \quad (23)$$

$$R_{c_{1,2}} = 7.761 \cdot 10^{-11} \quad (24)$$

- For the output noise $\varepsilon_y(k,\cdot)$:

$$\mu_y = \frac{\frac{3.3}{2^{12}}}{2} = 4.023 \cdot 10^{-4} \quad (25)$$

$$\sigma_y^2 = \frac{\left(\frac{3.3}{2^{12}}\right)^2}{12} = 5.412 \cdot 10^{-8} \quad (26)$$

$$R_y = 5.412 \cdot 10^{-8} \quad (27)$$

The covariance matrix of the state can be computed as:

$$\begin{aligned} &P_{xx}(k+1) \\ &= \begin{pmatrix} 0.9542 & 0.2897 \\ -0.2296 & 0.5929 \end{pmatrix} P_{xx}(k) \begin{pmatrix} 0.9542 & 0.2897 \\ -0.2296 & 0.5929 \end{pmatrix}^T \\ &+ \begin{pmatrix} 0.812 \\ 0.1752 \end{pmatrix} 5.412 \cdot 10^{-8} \begin{pmatrix} 0.812 \\ 0.1752 \end{pmatrix}^T \\ &+ \begin{pmatrix} 7.761 \cdot 10^{-11} & 0 \\ 0 & 7.761 \cdot 10^{-11} \end{pmatrix} \end{aligned} \quad (28)$$

And converges to:

$$\lim_{k\to\infty} P_{xx}(k) = \begin{pmatrix} 6.608 \cdot 10^{-9} & -0.218 \cdot 10^{-9} \\ -0.218 \cdot 10^{-9} & 3.309 \cdot 10^{-9} \end{pmatrix} \quad (29)$$

as shown in Fig. 10.

The output variance is:

$$\begin{aligned} &E\{y(k)y^T(t)\} \\ &= (1 \quad 0) \begin{pmatrix} 6.608 \cdot 10^{-9} & -0.218 \cdot 10^{-9} \\ -0.218 \cdot 10^{-9} & 3.309 \cdot 10^{-9} \end{pmatrix} \\ &\cdot (1 \quad 0)^T + (0.0159) \cdot 5.412 \cdot 10^{-8} \cdot (0.0159)^T \\ &+ 5.412 \cdot 10^{-8} = 6.0736 \cdot 10^{-8} \end{aligned} \quad (30)$$

In case of no input, using the calculated mean values of input, state and output noise in the relation (13) we have:

$$\begin{aligned} &E\begin{Bmatrix} x_1(k+1) \\ x_2(k+1) \end{Bmatrix} = \begin{pmatrix} 0.9542 & 0.2897 \\ -0.2296 & 0.5929 \end{pmatrix} \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + \\ &\begin{pmatrix} 0.812 \\ 0.1752 \end{pmatrix}(0) + \begin{pmatrix} 0.812 \\ 0.1752 \end{pmatrix}(0) + \begin{pmatrix} 1.526 \cdot 10^{-5} \\ 1.526 \cdot 10^{-5} \end{pmatrix} \end{aligned} \quad (31)$$

$$\begin{aligned} E\{y(k)\} &= (1\ 0) \begin{pmatrix} x_1(k) \\ x_2(k) \end{pmatrix} + (0.0159)u(k) \\ &+ 5.412 \cdot 10^{-8} \end{aligned} \quad (32)$$

At steady state we obtain:

$$\lim_{k\to\infty} E\begin{Bmatrix} x_1(k) \\ x_2(k) \end{Bmatrix} \to \begin{pmatrix} 0.1526 \cdot 10^{-4} \\ 0.1526 \cdot 10^{-4} \end{pmatrix} \quad (33)$$

$$\lim_{k\to\infty} E\{y(k)\} \to 0.1531 \cdot 10^{-4} \quad (34)$$

As can be seen, the output is influenced to the fourth decimal by the effect of quantization, a value that is acceptable in most applications.

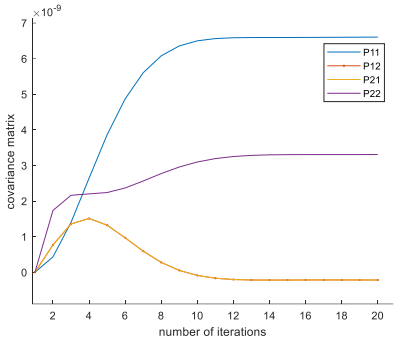

Fig. 10. The elements of the covariance matrix

VI. Experimental results

Finally, the implementation was performed on the STM32 Nucleo64-P development board using quantized coefficients and fixed-point processing [24]. The sampling frequency is 40kHz. The output is calculated as (35) and implemented using the direct form I.

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \qquad (35)$$

For the test we compared the response of an analog filter and the digital filter implemented on the development board. For the implementation the quantized coefficients obtained in the previous sections were used. The results can be seen in Fig. 11. The blue signal represents the input which has a variable frequency from 0 to 10kHz, the purple one represents the output of the digital filter, and the yellow signal represents the output of the analog filter. It can be seen that operating in fixed point and quantized coefficients does not affect the response of the digital filter for this particular case. Because of the required computational costs, the same performance in the speed of processing time could not be obtained using the floating-point calculation.

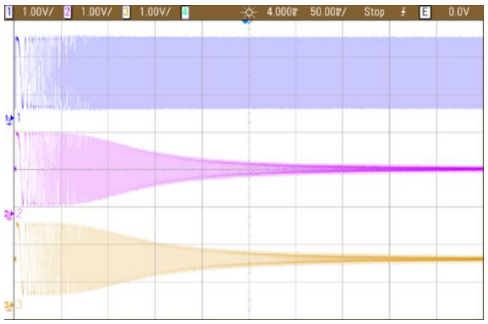

Fig. 11. The results obtained on the oscilloscope

VII. Conclusions

In this paper we analyzed the effect of sampling and quantization of a digital filter implementation. For the analysis of the sensitivity of the coefficients the root locus was used. For a sampled system, stochastic modelling of the quantization effects occurring in the ADC, the DAC and the calculation errors has been realized. It has been shown that quantization does not affect the numerical filter response until the fourth decimal place, which, depending on the application, may be negligible. Once the terms were quantized and working in fixed-point, the processing performance increased. As future research work, we will determine the maximum calculation frequency and a mathematical relation between quantization, accuracy, and computation costs.

References

[1] H. Zumbahlen, Linear Circuit Design Handbook, Newnes/Elsevier, 2008.

[2] A. De Cheveigné and I. Nelken, "Filters: When, Why, and How (Not) to Use Them," *Neuron*, pp. 280-293, 2019.

[3] S. Rolf, H. Xiao and V. Mac Van, Design of Analog Filters 2nd edition, Oxford University Press, 2009.

[4] M. Cherniakov, An Introduction to Parametric Digital Filters and Oscillators, Wiley, 2003.

[5] J. G. Proakis and D. G. Manolakis, Digital Signal Processing, Third Edition ed., Prentice-Hall International INC, 1996.

[6] S.-D. Sim, Zs. Lendek and P. Dobra, "Implementation and testing of digital filters on STM32 Nucleo-64P," *IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania*, pp. 1-6, 2022.

[7] T. Y., "Digital signal processing in continuous time: a possibility for avoiding aliasing and reducing quantization error," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[8] C. D. and R. L., "Analysis of quantization errors in the direct form for finite impulse response digital filters," in *IEEE Transactions on Audio and Electroacoustics*, 1973.

[9] L. R. Rabiner and B. Gold, Theory and application of digital signal processing, Prentice-Hall, 1975.

[10] C. L. Byrne, Signal Processing. A mathematical approach, CRC Press, 2015.

[11] D. Williamson, Discrete-time Signal Processing, London: Springer London Ltd, 1999.

[12] M. J. Pelgrom, Analog-to-Digital Conversion, Springer-Verlag New York, 2013.

[13] B. Widrow and I. Kollár , Quantization Noise: Roundoff Error in Digital Computation, Signal Processing, Control, and Communications, Cambridge University Press, 2008.

[14] C. Martin, High-Performance D/A-Converters: Application to Digital Transceivers, Springer Berlin, Heidelberg, 2013.

[15] P. Dobra, *Nonlinear and stochastic systems. Lecture notes*, Technical University of Cluj-Napoca, 2022.

[16] A. Saltelli, S. Tarantola, F. Campolongo and M. Ratto, Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models, John Wiley & Sons Inc, 2004.

[17] M. Susca and P. Dobra, "Notch Filter Sensitivity Analysis With Root Locus Considering Parameter," in *20th International Conference on System Theory, Control and Computing*, Sinaia, Romania, 2016.

[18] A. V. Oppenheim, R. W. Schafer and J. R. Buck, Discrete-Time Signal Processing, New Jersey: Prentice Hall, 1998.

[19] G. Dehner, I. Dehner, R. Rabenstein, M. Schäfer and C. Strobl, "Analysis of the quantization error in digital multipliers with small wordlength," in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016.

[20] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression, Springer New York, 1992.

[21] A. S. Rowell, "Digital filters with quantized coefficients: optimization and overflow analysis using extreme value theory," Stanford University, 2012.

[22] C. T., M. W. and P. J., "Quantization noise analysis for fixed-point digital filters using magnitude truncation for quantization," *IEEE Transactions on Circuits and Systems*, 1975.

[23] V. Madisetti, Digital Signal Processing Fundamentals, CRC Press, 2010.

[24] R. Oshana, DSP Software Development Tehniques for Embedded and Real-Time Systems, Elsevier, 2006.