mathematics

Article

Sampling Rate Optimization and Execution Time Analysis for Two-Degrees-of-Freedom Control Systems [†]

Mircea Şuşcă [*,‡], Vlad Mihaly [‡], Dora Morar and Petru Dobra [*]

Department of Automation, Technical University of Cluj-Napoca, Str. G. Bariţiu nr. 26-28, 400027 Cluj-Napoca, Romania

[*] Correspondence: mircea.susca@aut.utcluj.ro (M.Ş.); petru.dobra@aut.utcluj.ro (P.D.)

[†] This paper is an extended version of our paper published in 2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 19–21 May 2022. https://doi.org/10.1109/AQTR55203.2022.9802027.

[‡] These authors contributed equally to this work.

Abstract: The current journal paper proposes an end-to-end analysis for the numerical implementation of a two-degrees-of-freedom (2DOF) control structure, starting from the sampling rate selection mechanism via a quasi-optimal manner, along with the estimation of the worst-case execution time (WCET) for the specified controller. For the sampling rate selection, the classical Shannon–Nyquist sampling theorem is replaced by an optimization problem that encompasses the trade-off between the fidelity of the controllers' representation, along with the fidelity of the resulting closed-loop systems, and the implementation difficulty of the controllers. Additionally, the WCET analysis can be seen as a verification step before automatic code generation, a computational model being provided. The proposed computational model encompasses infinite-impulse response (IIR) and finite-impulse response (FIR) filter models for the controller implementation, along with additional relevant phenomena being discussed, such as saturation, signal scaling and anti-windup techniques. All proposed results will be illustrated on a DC motor benchmark control problem.

Keywords: sampling rate; closed-loop cascade control; discrete-time systems; global optimization; rapid control prototyping; worst-case execution time; IIR filtering; FIR filtering

MSC: 93C05; 93C57; 93C62

Citation: Şuşcă, M.; Mihaly, V.; Morar, D.; Dobra, P. Sampling Rate Optimization and Execution Time Analysis for Two-Degrees-of-Freedom Control Systems. *Mathematics* **2022**, *10*, 3449. https://doi.org/10.3390/math10193449

Academic Editor: Dimplekumar N. Chalishajar

Received: 25 August 2022
Accepted: 16 September 2022
Published: 22 September 2022

1. Introduction

1.1. Literature Review

The problem of sampling rate choice to numerically implement a controller designed in the continuous-time domain has significant importance. A sub-optimal value of the sampling period represents a trade-off between the fidelity of the response, given by a shorter sampling rate, and implementability, obtained using a larger sampling rate. The most common manner to choose this sampling period is given by the Shannon sampling theorem [1]. However, this can only be a starting point in a control context, because, in practice, it can lead to unacceptable behavior of the discrete-time controller compared to its continuous-time equivalent, a significant justification being the presence of quantization errors unaccounted for in the signal and system models. In the available literature, the problem of choosing the sampling rate is specifically formulated for the purpose of the particular control system at hand. For example, for a structure with a P-type regulator proposed in [2], the key points used in sampling rate selection are the overshoot and the rise time. For the case of a PID-based control structure, an optimization criterion has been proposed in [3] to find the optimal and/or sub-optimal value of the sampling rate. The problem of sampling rate selection for parameter estimation of an induction machine has been solved via a metaheuristic procedure in [4]. A similar thematic is addressed in [5] in the case of permanent magnet synchronous motors using field programmable gate array devices, while the rapid control prototyping principle is illustrated for discrete-time closed-loop control of brushless DC motors in [6,7], respectively. Moreover, an important result regarding the dependence between the stability and the performance of the closed-loop system, and the choice of sampling rate is presented in [8].

After the first step of selecting the sampling period, an analysis regarding the practical implementation of the proposed control structure on a microprocessor-based system should be performed. This analysis needs to encompass several aspects which are not explicitly modeled in the control law, such as the register word lengths of the coefficients which directly influence the number of necessary assembly instructions, and saturation, overflow, underflow verifications on the computed command signals. In the specific context of rapid control prototyping (RCP), one key step before the proper code generation is to certify if its execution abides by the time span given by the sampling period. Moreover, other important verifications for phenomena such as underflow and overflow, or saturation and anti-windup techniques should be also performed before the code-generation step. As such, one important goal consists of approximating the number of software operations necessary to implement the designed control law, mandatory in the context of hard real-time systems. A survey of worst-case execution time computational methods is presented in the seminal paper [9]. A simplified study applied for control system-related interrupt service routines for state-space controller representations is presented in [10]. Another possibility for the computational error analysis is presented in [11].

In the Control Systems domain, the linear control system field plays an important role. Linear controller design techniques are available for both linear plants and nonlinear process models alike. Starting from the classical proportional–integral–derivative (PID) controller [12] towards the domain of robust controller [13–15], these techniques can be extended for nonlinear systems via the gain-scheduling method [16] or an additional passivity-based component [17]. Moreover, with great advantage in also coping with disturbance rejection problems, alongside reference tracking specifications are the so-called two-degrees-of-freedom (2DOF) control schemes [13]. In ref. [18], a 2DOF PID controller has been proposed, where the serial compensator is a classical PID controller, while the feedforward compensator is a PD controller, the integral effect being excluded due to the stability requirements. For the case of unstable plants with time delays, a discrete-time 2DOF control scheme has been proposed in [19]. As such, for reference tracking, the serial controller was designed using the $\mathcal{H}_2$ optimal control framework, while the feedforward controller has been tuned by imposing the desired closed-loop transfer function. Additionally, in terms of software toolbox implementations, the robust advanced PID (RaPID) toolbox described in [20] presents a set of possibilities to design a 2DOF PID control structure by minimizing certain error criteria, such as the integral of absolute error or integral of time multiplied by the absolute value of error.

1.2. Contributions and Paper Structure

The current journal paper represents an extension of the conference papers [10,21], and proposes a joint analysis on sampling time selection and execution time framing of the microcontroller operations into the previously established optimum for the case of a 2DOF control structure. The main contributions of the paper are:

(i) to extend the functionals initially proposed in [21] in order to encompass the fidelity of both controllers from the 2DOF structure and of the resulting closed-loop systems, i.e., ensuring the tracking and servo behavior of the initially designed continuous-time controllers, along with the implementation difficulty of the proposed controllers in terms of quantization and sampling rate span;

(ii) to formulate the optimization problem for sampling rate selection of a 2DOF control structure without considering any further normalizations of the final functional's terms and propose a metaheuristic-based solution to find an optimal or quasi-optimal value of the sampling rate which can offer a good trade-off between implementability and fidelity;

(iii) to extend the WCET analysis, initially proposed in [10] for the case of state-space regulator implementations, for the case of controllers implemented as infinite-impulse response (IIR) or finite-impulse response (FIR) filters, offering an upper bound estimation of the time span necessary to perform all the operations involved in the numerical implementation of the proposed 2DOF controller;

(iv) to offer an exhaustive analysis which can be further performed in an automatic manner and to be integrated in our open-source MATLAB-based CACSD toolbox [22];

(v) to present an end-to-end design procedure for the case of a DC motor control problem, starting from the controller design, followed by choice of the quasi-optimal sampling rate and the worst-case execution time analysis, finalizing with a detailed discussion.

The rest of the paper is organized as follows: Section 2 presents the 2DOF structure which will be studied, along with a mathematical background for the sampling rate choice and worst-case execution time analysis, and a set of theoretical results and optimization problems extended and adapted for the proposed control structure; in Section 3 a case study consisting of an end-to-end design procedure of a 2DOF PID controller is illustrated, while Section 4 deals with a set of conclusions and further research directions.

1.3. Notations

We denote by $\mathscr{C}(z,r)$ the circle with the center in $z \in \mathbb{C}$ and radius $r > 0$. Additionally, $\mathscr{P}(G)$ and $\mathfrak{X}(G)$ denote the pole and zero sets of an LTI system G, respectively. An arbitrary sampling period will be denoted $T_s \equiv T > 0$ throughout the paper. Continuous-time regulators will be denoted $K(s) \equiv K$, while their discrete counterparts, as a function of the sampling period will be used as $K_T(z) \equiv K_T$. The set of continuous-time systems will be denoted by $\mathscr{G}$, while the set of discrete-time systems will be denoted by $\mathscr{G}_D$. The standard growth functions and their notations according to [23]:

$$\begin{aligned}
\mathcal{O}(g(n)) &= \{f(n) | \exists\, c, n_0 > 0 \text{ such that } 0 \leq f(n) \leq cg(n),\ \forall\, n \geq n_0\}; \\
\Omega(g(n)) &= \{f(n) | \exists\, c, n_0 > 0 \text{ such that } 0 \leq cg(n) \leq f(n),\ \forall\, n \geq n_0\}; \\
\Theta(g(n)) &= \{f(n) | \exists\, c_1, c_2, n_0 > 0 \text{ such that } 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n),\ \forall\, n \geq n_0\}.
\end{aligned}$$

2. Mathematical Background and Proposed Extensions

2.1. Numerical Closed-Loop Control

Modern control systems make use of numerically implemented regulators to coordinate continuous-time processes. The classical configuration of a numerical regulator is presented in Figure 1. It uses sample and hold circuits as interfaces to the continuous-time adjacent components and, as such, it imposes the zero-order hold discretization method for the plant G. As such, define the plant discretization methodology as the mapping $(G(s), T) \mapsto G_T(z) \equiv G(z) \in \mathscr{G}_D$:

$$G_T(z) = \mathfrak{Z}\left\{\mathscr{L}^{-1}\{G_{zoh,T}(s) \cdot G(s)\}\right\} : \mathscr{G} \times \mathbb{R}_+ \to \mathscr{G}_D. \tag{1}$$

with $\mathscr{G}$ denoting the continuous-time model set, $\mathscr{G}_D$ being the discrete-time model set, and with the digital-to-analog converter, i.e., zero-order hold, model:

$$G_{zoh,T}(s) = \frac{1 - e^{-sT}}{s}. \tag{2}$$

For the purpose of this paper, we consider a two-degrees-of-freedom (2DOF) control structure as in Figure 2, which has an inner controller $K_{\text{in}}(s)$ usually designed for disturbance rejection, and a feedforward controller $K_{\text{ff}}(s)$ which provides the necessary compensation for the steady-state tracking behavior, along with the process model with disturbance, described by $G(s)$ and $G_d(s)$. As K_{ff} does not influence the signal path from the disturbance $d(t)$ to the output $y(t)$, the usual design workflow is to synthesize K_{in} and, then, to fine-tune the transient response from $r(t)$ to $y(t)$ through K_{ff}. Both continuous-time controllers $K_{\text{in}}(s)$ and $K_{\text{ff}}(s)$ are assumed to have the following linear and time-invariant (LTI) state-space representations:

$$(K_x(s)) : \begin{cases} \dot{\mathbf{x}}_c &= A_{K,x}\mathbf{x}_c + B_{K,x}\mathbf{u}_c \\ \mathbf{y}_c &= C_{K,x}\mathbf{x}_c + D_{K,x}\mathbf{u}_c \end{cases}, \quad x \in \{\text{in}, \text{ff}\}. \tag{3}$$

Given that the continuous-time controllers $K_{\text{in}}(s)$ and $K_{\text{ff}}(s)$ must be numerically implemented on a microcontroller, the problem of selecting an appropriate sampling period $T \in (0, \infty)$ becomes a critical step. For a fixed sampling rate T, the discrete-time form of a controller $K_x(s)$ can be written as:

$$(K_x(z)) : \begin{cases} \mathbf{x}_{k+1} &= \Phi_x\mathbf{x}_k + \Gamma_x\mathbf{u}_k \\ \mathbf{y}_k &= C_{K,x}\mathbf{x}_k + D_{K,x}\mathbf{u}_k \end{cases}, \quad x \in \{\text{in}, \text{ff}\}. \tag{4}$$

To compute the transition matrix Φ_x and the input matrix Γ_x, a wide selection of discretization methods, denoted by $\mathbf{D}$, can be considered, such as zero-order hold, trapezoidal (Tustin), forward or backward Euler, frequency-response regression and so on. Additionally, in most situations, the output matrix remains the same as in the continuous-time representation, but there exist cases when it may be different than its continuous-time counterpart. As such, from a tuple of a continuous-time transfer matrix $K_x(s) = (A_{K,x}, B_{K,x}, C_{K,x}, D_{K,x}) \in \mathcal{G}$ and a sampling rate $T \in \mathbb{R}_+$ results an equivalent discrete-time transfer matrix $(K(s), T) \mapsto K_T(z) \equiv K(z) \in \mathcal{G}_D$, where:

$$K_T(z) = \mathbf{D}\{K(s), T\} : \mathcal{G} \times \mathbb{R}_+ \to \mathcal{G}_D. \tag{5}$$

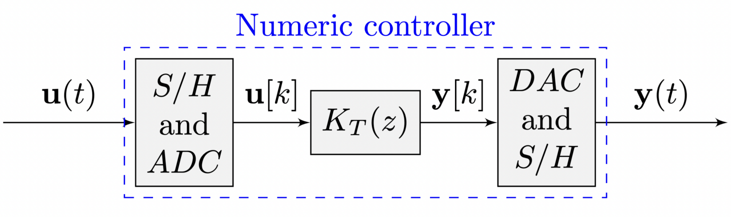

Figure 1. Numeric regulator $K_T(z)$ with a specified sampling rate $T > 0$ and its corresponding interface consisting of a sample and hold with the analog-to-digital converter, along with the digital-to-analog converter followed by a sample and hold circuit.

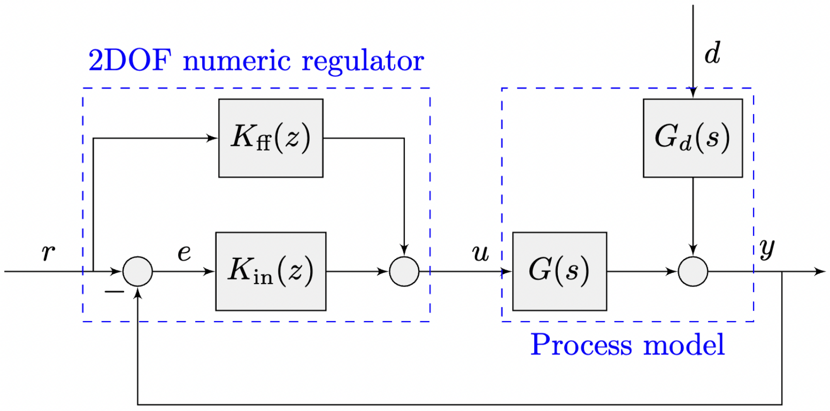

Figure 2. Two-degrees-of-freedom (2DOF) numerical control structure with components $K_{\text{in}}(z)$ and $K_{\text{ff}}(z)$, designed for the process models $G(s)$ with disturbance dynamics $G_d(s)$.

The open-loop process model has the input-output representation:

$$Y(s) = G_d(s) \cdot D(s) + G(s) \cdot U(s), \tag{6}$$

while the closed-loop system expression becomes:

$$Y(s) = \frac{G_d(s)}{1 + G(s)K_{\text{in}}(s)} \cdot D(s) + \frac{G(s)(K_{\text{in}}(s) + K_{\text{ff}}(s))}{1 + G(s)K_{\text{in}}(s)} \cdot R(s). \tag{7}$$

The discrete-time equivalent closed-loop model will be:

$$Y(z) = \frac{G_d(z)}{1 + G(z)K_{\text{in}}(z)} \cdot D(z) + \frac{G(z)(K_{\text{in}}(z)+K_{\text{ff}}(z))}{1 + G(z)K_{\text{in}}(z)} \cdot R(z) = H_{cl}^d(z)D(z) + H_{cl}^r(z)R(z), \tag{8}$$

which represents the starting point for various control design methodologies.

2.2. Sampling Rate Optimization

The current subsection presents an extension of the work from the paper [21].

2.2.1. Linear System Sampling Background

The well-known Shannon theorem [1] states that the necessary sampling rate to avoid information loss must be at most twice time smaller than the inverse of the continuous-time signal's maximum frequency component:

$$T_s < \frac{T_{\min}}{2} \quad \Leftrightarrow \quad f_s > 2 \cdot f_{\max}, \quad \text{where } f_{\max} = \frac{1}{T_{\min}}. \tag{9}$$

Moreover, the theoretical upper bound of the frequency range after which aliasing phenomena occur is called Nyquist frequency and is given by:

$$\omega_N = \frac{\omega_s}{2} = \pi \cdot f_s. \tag{10}$$

The above-mentioned version of Nyquist-Shannon theorem for signals can be extended to LTI systems by considering that the resulting discrete-time system must maintain all relevant dynamics of the continuous-time system. The dynamics of a system is given by its set of poles, one possible constraint for the sampling rate implying to be at most twice smaller than the inverse of the real part of the minimum value of poles. However, there are cases when the imaginary part of the poles is relevant, or even the values of the transmission zeros, for a good representation of the system's frequency response.

For the remainder of the paper, the sampling rate will be denoted by $T_s \equiv T$. The analytical relationship between the continuous-time s-plane and the discrete-time z-plane is illustrated by the equation:

$$z = e^{s \cdot T}. \tag{11}$$

Considering a point $s := \sigma + j\omega$, with $\sigma, \omega \in \mathbb{R}$, the resulting corresponding complex number is $z = e^{\sigma T}(\cos(\omega T) + j\sin(\omega T))$. As such, the left half of the complex s-plane can be split into an infinite number of disjoint strips of height $\frac{2\pi}{T}$, due to the periodical nature of the $\sin(\cdot)$ and $\cos(\cdot)$ functions. The resulting primary strip contains points which correspond to $\omega \in [-\omega_N, \omega_N]$. The s-plane to z-plane mapping causes the following topologies of the primary strip of the s-plane:

- the imaginary axis in is mapped to the unit circle $\mathscr{C}(0, 1)$;
- the upper and the lower edges are both mapped to the negative axis;
- the negative real axis is mapped to the positive axis inside the unit circle;
- the interior of the primary strip is mapped in the unit disk.

When sampling an LTI controller, several remarks can be considered then the behavior of the singularities is studied:

(i) unstable zeros and poles tend decreasingly to the unit circle's circumference, i.e., $|z| \searrow 1$ and $|p| \searrow 1$ as $T \searrow 0$; for poles, this aspect is relevant when sampling systems, as controllers generally do not employ unstable poles in their structure;

(ii) stable zeros and poles tend increasingly to the unit circle's circumference, i.e., $|z| \nearrow 1$ and $|p| \nearrow 1$ as $T \searrow 0$, requiring additional decimal or binary digits for an accurate representation.

(iii) poles and zeros from the imaginary axis in the s-domain are maintained on the unit circle in the z-domain irrespective of the sampling period $T > 0$. Additionally, integrator and derivative terms do not matter in deciding the practical sampling time.

(iv) the quantization error increases as $T \searrow 0$ in the case of stable closed-loop systems.

Remark 1. *The quantization error mentioned in (iv) increases as $T \searrow 0$ as proved in [24], due to the guaranteed steady-state error being proportional to $\frac{1}{1-\rho(\Phi)}$, while $\rho(\Phi)$ denotes the spectral radius of the closed-loop state matrix: $\Phi := \hat{A} - \hat{B}\hat{D}\hat{C}$, based on the series interconnection between the controller and the process model matrices.*

According to the previously mentioned aspects, a sub-optimal value of the sampling rate must ensure a good trade-off between the fidelity of the representation of the continuous-time controllers and their implementability difficulty. As such, we next introduce a set of functionals to quantify these two aspects.

2.2.2. Proposed Functionals

To measure the similarity between a continuous-time LTI system $H(s)$ and a discrete-time system $H_T(z)$ over the frequency range Ω, the following functional can be defined as $\mathcal{S}_H^{\Omega} : \mathscr{D} \to (0, \infty)$, with $\mathscr{D} = \mathscr{G} \times \mathbb{R}_+$:

$$\mathcal{S}_H^{\Omega}(T) = \int_{\Omega} \left(\overline{\sigma}(H(j\omega)) - \overline{\sigma}\left(H_T\left(e^{j\omega T}\right)\right)\right)^2 \mathrm{d}\omega, \tag{12}$$

where $\overline{\sigma}(\cdot)$ is the maximum singular value. Moreover, as mentioned before, the available frequency domain is $\Omega = (0, \omega_N)$, ω_N being the corresponding Nyquist-Shannon frequency to the sampling rate T. Therefore, the similarity functional $S_H : \mathscr{D} \to (0, \infty)$ becomes:

$$\mathcal{S}_H(T) = \int_{0^+}^{\omega_N^-} \left(\overline{\sigma}(H(j\omega)) - \overline{\sigma}\left(H_T\left(e^{j\omega T}\right)\right)\right)^2 \mathrm{d}\omega. \tag{13}$$

However, to compute the similarity integral term $\mathcal{S}_H$, a discrete set from the frequency range $(0, \omega_N) \to \overline{\Omega} = \{\underline{\omega} = \omega_1 < \omega_2 < \cdots < \omega_N - \omega_\varepsilon = \overline{\omega}\}$ can be considered, the performance index being approximated as follows:

$$\mathcal{S}_H^{\overline{\Omega}}(T) \simeq \sum_{\omega \in \overline{\Omega}} \left(\overline{\sigma}(H(j\omega)) - \overline{\sigma}\left(H_T\left(e^{j\omega T}\right)\right)\right)^2 \Delta\omega, \tag{14}$$

where $\omega_\varepsilon > 0$ is a predefined threshold used to avoid the prewarping phenomenon which ensues in the magnitude responses when $\omega \to \omega_N$.

Remark 2. *There are several alternatives in defining the similarity between two LTI systems, with particular interest for closed-loop connections, such as the normalized dissimilarity function metric, described in [25], the standard ν-gap metric, with its limitations exposed in [26], or the improved Vinnicombe ν-gap metric [27], described as:*

$$\delta(G_1, G_2) := \max\left(\overline{\delta}(G_1, G_2), \overline{\delta}(G_2, G_1)\right), \tag{15}$$

with:

$$\overline{\delta}(G_1, G_2) := \inf_{Q \in H_\infty} \left\| \begin{pmatrix} M_1 \\ N_1 \end{pmatrix} - \begin{pmatrix} M_2 \\ N_2 \end{pmatrix} Q \right\|_\infty, \tag{16}$$

where for the reference systems G_1 and G_2, their right normalized coprime factorizations will be considered: $G_1 := N_1 M_1^{-1}$ and $G_2 := N_2 M_2^{-1}$, which in context of the proposed sampling rate optimization problem, the differences should be computed with the subsystems applied in $G_1 \equiv K(s) \mapsto j\omega$ and $G_2 \equiv K_T(z) \mapsto e^{j\omega T}$, respectively.

To define the fixed-point implementability functional $\mathcal{I}_H(T) : \mathcal{D} \to (0, \infty)$ which measures the quantization implementation difficulty of the LTI system $H_T(z)$, the following expression can be considered:

$$\mathcal{I}_H(T) = \frac{1}{\min\{||\lambda| - 1|,\ \lambda \in \mathcal{P}(H_T) \cup \mathcal{Z}(H_T)\}}, \tag{17}$$

with $\mathcal{P}$ and $\mathcal{Z}$ denoting the pole and zero sets of H_T, respectively, excluding singularities from the set $\mathcal{C}(0,1)$.

Alongside the previous implementability functional, an execution time cost functional $\mathcal{T}_H(T) : \mathcal{D} \to (0, \infty)$ becomes necessary to limit the decrease of $T \searrow 0$ which would ideally lead the similarity functional costs to zero:

$$\mathcal{T}_H(T) = \frac{1}{T}. \tag{18}$$

A concluding functional with global effect, $\mathcal{J}_{\text{stab}(H)} : \mathcal{D} \to \{0, \infty\}$, will define the feasibility domain of the optimization problem, because it accepts or rejects a specific sampling period value. It induces an infinitely valued constant when the numeric system H_T becomes unstable, and otherwise, defaults to no extra penalization:

$$\mathcal{J}_{\text{stab}(H)}(T) = \begin{cases} +\infty, & \text{if } H_T \text{ has unstable poles;} \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

For the numerical implementation of the stability functional, a sufficiently large value α_∞ can be considered to mark the feasibility subdomain of $\mathcal{D}$ where the system H is stable, leading to an approximation of the original performance index:

$$\mathcal{J}^{\alpha_\infty}_{\text{stab}(H)}(T) = \begin{cases} +\alpha_\infty, & \text{if } H_T \text{ has unstable poles;} \\ 0, & \text{otherwise.} \end{cases} \tag{20}$$

2.2.3. Optimization Problem

For the two-degrees-of-freedom control structure proposed in Figure 2 the following functionals will be considered to formulate the optimization problem to determine the sampling rate:

- two terms $\mathcal{S}_{K_{\text{in}}}(T)$ and $\mathcal{S}_{K_{\text{ff}}}(T)$ representing the similarity between the continuous-time and the discrete-time representations of the controllers $K_{\text{in}}(s)$ and $K_{\text{ff}(s)}$ over the frequency range $(0, \omega_N)$;
- two terms $\mathcal{S}_{H^r_{cl}}(T)$ and $\mathcal{S}_{H^d_{cl}}(T)$ representing the similarity between the continuous-time and the discrete-time representations of the resulting closed-loop systems $H^r_{cl}(s)$ and $H^d_{cl}(s)$ over the same frequency range;
- the quantization implementation difficulty $\mathcal{I}_{K_{\text{in}}}(T)$ and $\mathcal{I}_{K_{\text{ff}}}(T)$ of each controller along with the execution time functional $\mathcal{T}(T)$;
- the stability functional $\mathcal{J}_{\text{stab}(H^r_{cl})}(T)$ of the resulting numerical closed-loop system $H^r_{cl,T}(z)$.

The first set of three terms are used to measure the fidelity between the components of the continuous-time designed system and the resulting components in the discrete-time domain, considering the transient and steady-state performance altering. The next set of three terms manages to encompass the implementation difficulty of the resulting controller, according to the already-mentioned remarks. The last term is nothing but a correction used to define the feasibility subdomain of $\mathscr{D}$. As such, a trade-off between the first two sets of performance indices must be established by taking into account the last feasibility term, resulting a non-convex optimization problem. However, in order to increase the flexibility of this optimization problem, a set of weights c_{1-6} can be considered as follows: the weights c_1, c_2, c_3, and c_4 are for the fidelity measurement indices, while weights c_5, c_6, and c_7 are for the implementability indices, the final functional $\mathscr{J} : \mathscr{D} \to \overline{\mathbb{R}}_+$ being:

$$\mathscr{J}(T) = c_1 \mathcal{S}_{K_{\text{in}}}(T) + c_2 \mathcal{S}_{K_{\text{ff}}}(T) + c_3 \mathcal{S}_{H_{cl}^r}(T) + c_4 \mathcal{S}_{H_{cl}^d}(T) c_5 \mathscr{I}_{K_{\text{in}}}(T) + c_6 \mathscr{I}_{K_{\text{ff}}}(T) + c_7 \mathscr{T}(T) + \mathscr{J}_{\text{stab}(H_0)}(T). \tag{21}$$

Moreover, considering the numerical approximations (14) and (20), the following numerically implementable non-convex optimization problem occurs:

Problem 1. *For an LTI plant model $G(s)$ included in a two-degrees-of-freedom control structure with the continuous-time inner controller $K_{in}(s)$ and with the continuous-time feedforward controller $K_{ff}(s)$ as in Figure 2, define an ordered set of pulsations, preferably in logarithmic scale:*

$$\overline{\Omega} = (\underline{\omega} = \omega_1 < \omega_2 < \ldots < \omega_N - \omega_\varepsilon = \overline{\omega}). \tag{22}$$

Then, the functional $\mathscr{J}^{\overline{\Omega}} : \mathscr{D} \to \mathbb{R}_+$:

$$\mathscr{J}^{\overline{\Omega}}(T) = c_1 \mathcal{S}^{\overline{\Omega}}_{K_{in}}(T) + c_2 \mathcal{S}^{\overline{\Omega}}_{K_{ff}}(T) c_3 \mathcal{S}^{\overline{\Omega}}_{H_{cl}^r}(T) + c_4 \mathcal{S}^{\overline{\Omega}}_{H_{cl}^d}(T) + c_5 \mathscr{I}_{K_{in}}(T) + c_6 \mathscr{I}_{K_{ff}}(T) + c_7 \mathscr{T}(T) + \mathscr{J}^{\alpha_\infty}_{stab(H_0)}(T), \tag{23}$$

with weighting terms c_{1-7}, leads to the following quasi-optimal sampling time optimization problem:

$$\min_{T \in \mathscr{D}} \mathscr{J}^{\overline{\Omega}}(T). \tag{24}$$

Remark 3. *The resulting optimization problem (24) is non-convex by nature. As such, the line-search procedure used to solve this problem must be able to explore the whole feasible domain. One possible solution is a metaheurisitc approach, such as particle swarm optimization (PSO) [28], which will be used for the purpose of this paper.*

2.3. Worst-Case Execution Time Analysis

The current subsection presents an extension of the work from the paper [10].

2.3.1. Execution Time Model

This subsection proposes the study of the implementation details for the case of linear control structures, where controllers will be modeled as infinite-impulse response (IIR) and finite-impulse response (FIR) filters. Starting from [10], a set of implementation aspects which are treated for state-space realizations will be considered in a unified manner for the case of transfer matrices in this paper. Additionally, this section also analyzes the 2DOF extension. The duration for each operation involved in the control structure can significantly impact the regulator implementability.

The necessary mathematical operations to fully implement an LTI-based control law must be formally defined. Besides the LTI-control law, the classical saturation and anti-windup nonlinearities will be considered, which are usually related to said LTI laws. The unary and binary mathematical operators defined in Table 1 and gathered in the operations alphabet $\mathcal{O} := \{\mathscr{n}, \mathscr{a}, \mathscr{m}, \mathscr{s}, \mathscr{w}, \mathscr{\ell}\}$, are considered with real operands and must be accounted for into a microprocessor-based environment.

Table 1. Formal operators necessary for the implementation of LTI-based control laws.

#	Operator	Domain	Definition	Observations
1	Addition	$\mathscr{a} : \mathbb{R}^2 \to \mathbb{R}$	$\mathscr{a}(x_1, x_2) = x_1 + x_2$	bilinear
2	Multiplication	$\mathscr{m} : \mathbb{R}^2 \to \mathbb{R}$	$\mathscr{m}(g, x) = g \cdot x$	homogenous
3	Saturation	$\mathscr{s}_{\underline{x},\overline{x}} : \mathbb{R} \to \mathbb{R}$	$\mathscr{s}_{\underline{x},\overline{x}}(x) = \begin{cases} \underline{x}, & \text{if } x < \underline{x}; \\ x, & \text{if } \underline{x} \leq x \leq \overline{x}; \\ \overline{x}, & \text{if } \overline{x} < x, \end{cases}$	nonlinear
4	Anti-windup	$\mathscr{w}_{\underline{x},\overline{x}} : \mathbb{R} \to \mathbb{R}$	various [12,29,30]	for integrators
5	Load/Store	$\ell(x) : \mathbb{R} \to \mathbb{R}$	$\ell(x) = x$	bilinear
6	Null	$\mathscr{n}(x) : \mathbb{R} \to \mathbb{R}$	$\mathscr{n}(x) = 0$	for delays

As such, the process of computing a command signal $\mathbf{y}[k]$ as in Figure 1 implies a finite and formal computational finite sequence $\mathscr{Sc} \in \mathcal{O}^N$, where all terms are mathematical operations as in Table 1, i.e., $\mathscr{Sc}[i] = p_i \in \mathcal{O}, i = \overline{1, N}$, where N depends on the structure of the controller $K(z)$. Moreover, in order to additionally specify a set of practical hardware specifications and constraints $\mathscr{H}$ and to uniformly describe the problem, the finite sequence can be now extended to a full sequence $\mathscr{Sc}^{K,\mathscr{H}} \in \mathcal{O}^\infty$:

$$\mathscr{Sc}^{K,\mathscr{H}} := (p_1, p_2, \ldots, p_N, \mathscr{n}, \mathscr{n}, \ldots),\ p_i \in \mathcal{O}. \tag{25}$$

Starting from an array $\mathscr{Sc}^{K,\mathscr{H}}$ of operations as (25), we follow with a general-purpose instruction set model. Assume a Random-Access Machine (RAM) computational model as in [23], with deterministic operations. RAM machines have practical counterparts, materialized through RISC machines. Reconfigurable RISC machines specialized on certain problems have been proposed in [31]. Additionally, there is the approach of multiply and accumulate (MAC) instructions supported in digital signal processors (DSP) [32]. Depending on the supported computer architectures of the RCP framework, relevant are also Single Instruction stream/Multiple instruction Pipelining (SIMP) [33] constructions with respect to single-processor architectures, or Single Instruction/Multiple Data (SIMD) features, which allow the practical parallelization of addition and multiplication operations for several sets of operands.

Definition 1. *The hardware constraints and specification set $\mathscr{H}$ encompasses metadata which imply extra operations or different approaches to the standard operations performed on the controller signals and implementation-specific information, with various outcomes on the total execution time, frequently found in practice being:*

- *reading reference signals $r[k]$ and plant measurements $y[k]$, all input signal reading steps may imply preprocessing constraints in terms of sensor delays, impulse counters or data type conversions—this equates to adding $p_i \in \{\mathscr{n}, \ell\}$ steps;*
- *scaling operations for the input and output signals imposed by the operating point used for plant linearization: $\Delta u[k] = u[k] - u_0$ and $y[k] = y_0 + \Delta y[k]$, which equates to augmenting the sequence set with $p_i \in \{\mathscr{a}, \mathscr{s}, \ell\}$ items;*
- *input and output signals scaling operations, i.e., $u_s[k] = a_u \cdot u[k] + b_u$ and $y_s[k] = a_y \cdot y[k] + b_y$, useful especially for sensor/adapter signals, and extend the operations with $p_i \in \{\mathscr{a}, \mathscr{m}, \mathscr{s}, \ell\}$;*
- *starting from the variable base word length L of the microcontroller arithmetic registers which allows operations to be executed in a single clock tick, each variable's type and size should be adequately adapted for 2 × L, 4 × L etc., which complicates the adding and multiplication routines with additional $p_i \in \{\mathscr{a}, \mathscr{m}, \ell\}$ steps;*
- *controller gain-scheduling verifications and updates based on the value of the input signal $u[k]$, leading to extra $p_i \in \{\ell\}$;*
- *underflow and overflow checks for involved signals, implying saturations $p_i \in \{\mathscr{s}\}$;*

- *availability of Direct Memory Access (DMA) modules, MAC instructions, circular buffers in opposition to linear buffering, or output bypassing, which has the advantage of discarding $p_i \in \{\ell\}$ operations.*

Such specifications will be quantified by scaling factors γ_j to the base duration of the RISC machine model operations. Depending on the significance of the alternative instruction, γ_j could be less than, equal to, or greater than one, respectively.

Therefore, a set of instructions $\mathcal{S}c^{K,\mathcal{H}}$ can be manually designed or automatically deduced. For manual modeling, the control engineer needs to find this set for each control law, while for automatic mode, an RCP tool can already deduce this set. Such an RCP tool generally deals with the code generation for different environments. Now, the sequence of operations generator procedures can be seen as a functional:

$$\Psi : \mathcal{G}_D \times \mathcal{H} \to \mathcal{O}^{\infty},\ \Psi(K,\mathcal{H}) = \mathcal{S}c^{K,\mathcal{H}}. \tag{26}$$

To implement the mathematical operations $p_i \in \mathcal{O}$ from Table 1, the atomic assembly instructions $a_i \in \mathcal{A} = \{\mathtt{NOP}, \mathtt{MF}, \mathtt{MS}, \mathtt{ADD}, \mathtt{MUL}, \mathtt{SH}, \mathtt{JMP}, \mathtt{CMP}\}$ from Table 2 will represent the starting point. It covers the basic arithmetical operations required in linear systems, with the additivity and homogeneity properties, conditional jumps for saturations and the anti-windup of integrator terms, forced and imposed delays through data acquisition hardware and access to memory devices. Each arbitrary atomic assembly operation will be denoted by a, as part of the formal set $\mathcal{A}$ all the equivalent assembly instructions supported by $\mathcal{H}$ for the implementation of $\mathcal{S}c$, with the RISC machine assumption that each instruction takes a fixed clock tick value $T_{clk} > 0$. The number of ways in obtaining the resulting assembly instructions is not unique and it further depends on the structure of $\mathcal{H}$. A straightforward example to illustrate this phenomenon is when the regulator coefficients are stored contiguously in the memory in comparison to arbitrary and uncorrelated memory registers. The execution time implications are obtained through different pointer operations. To conclude, a new mathematical operator analogous to (26), tasked with the generation of a computer-equivalent set of instructions given by the functional $\mathcal{S}p \in \mathcal{A}^{\infty}$ is:

$$\Xi : \mathcal{G}_D \times \mathcal{H} \to \mathcal{A}^{\infty},\ \Xi(K,\mathcal{H}) = \mathcal{S}p^{K,\mathcal{H}}, \tag{27}$$

which results in an infinite sequence of atomic software instructions, but with a finite number of them being different to NOP, located at the start of the sequence, which implement the linear controller formula:

$$\mathcal{S}p^{K,\mathcal{H}} := (a_1, \dots, a_M, \mathtt{NOP}, \mathtt{NOP}, \dots),\ a_i \in \mathcal{A}. \tag{28}$$

The difference between the sets $\mathcal{S}c$ and $\mathcal{S}p$ is that $\mathcal{S}c$ contains abstract mathematical operations $p_i \in \mathcal{O}$, and each such operation p_i will be practically implemented using equivalent $a_{i,j} \in \mathcal{A}$ steps, $j = \overline{1, N_i}$, as in the mapping:

$$p_i \mapsto \left(a_{i,1}, a_{i,2}, \cdots, a_{i,N_i}\right),\ \forall i = \overline{1, N}, \tag{29}$$

the number of atomic operations different from NOP being $M = N_1 + N_2 + \cdots + N_N$.

The importance of the previously defined sequences and functionals, i.e., $\mathcal{S}c^{K,\mathcal{H}}$, Ψ, $\mathcal{S}p^{K,\mathcal{H}}$ and Ξ, respectively, and the implications of estimating the number of assembly operations in a tight manner was insisted upon in the base paper [10]. Figure 3 gathers them and illustrates their connections in an RCP context. The mapping from $(K, \mathcal{H})$ to the set $\mathcal{S}c^{K,\mathcal{H}}$ is usually performed for Model-in-the-Loop (MiL) simulations through an application Ψ, while the mapping $(K, \mathcal{H}) \mapsto \mathcal{S}p^{K,\mathcal{H}}$ is made through Software-in-the-Loop (SiL) testing using an application Ξ. The master RCP program, with access to both $\mathcal{S}c^{K,\mathcal{H}}$ and $\mathcal{S}p^{K,\mathcal{H}}$, can subsequently perform a worst-case execution time analysis on the implementation of the digital regulator $K(z)$ in the production hardware context $\mathcal{H}$.

Table 2. Base assembly operations $a_i \in \mathscr{A}$ for a Random-Access Machine model used in an LTI-based control system context.

#	Operator	Abbreviation	Part of Operation	Observations
1	No operation	`NOP`	n	–
2	Memory fetch	`MF`	ℓ	SIMP
3	Memory store	`MS`	ℓ	SIMP
4	Add	`ADD`	a	SIMP, MAC, SIMD
5	Multiply	`MUL`	m	SIMP, MAC, SIMD
6	Binary shift	`SH`	m	SIMP, SIMD
7	Jump	`JMP`	s, w, ℓ	–
8	Compare	`CMP`	s, w, ℓ	SIMP

$$\begin{tikzcd}
K(z) \in \mathscr{G}_D, \mathscr{H} \arrow[r, "\Psi\ \texttt{MiL}"] \arrow[dr, "\Xi\ \texttt{SiL}"] & \mathcal{S}c^{K,\mathscr{H}} \arrow[d, leftrightarrow] \arrow[dr] & \\
 & \mathcal{S}p^{K,\mathscr{H}} \arrow[r] & \texttt{WCET}(\Psi, \Xi)
\end{tikzcd}$$

Figure 3. Rapid control prototyping relationship between the formal sets and functionals necessary to perform a worst-case execution time analysis for a regulator $K(z) \in \mathscr{G}_D$ in the production context $\mathscr{H}$.

Figure 4 presents the sequence diagram with the timing constraints of the discrete-time controller K with respect to other software threads from the microcontroller, with an illustration of the WCET of the controller interrupt service routine (ISR) thread. The main result of the section is gathered in the following theorem.

Theorem 1. *Given a numerical control law given by $K \in \mathscr{G}_D$, along with a microcontroller specification set $\mathscr{H}$ and a code-generation procedure given by a pair of MiL and SiL (Ψ, Ξ), the worst-case execution time estimation can be computed by the following formula:*

$$\texttt{WCET}(\Psi, \Xi) = \left(\left| \mathcal{S}p^{K,\mathscr{H}} \right| + \mathscr{O}(1) \right) \times T_{clk}, \tag{30}$$

where $\left|\mathcal{S}p^{K,\mathscr{H}}\right|$ represents the number of atomic assembly operations $a_i \in \mathscr{A}$ as in (28), and $\mathscr{O}(1)$ accounts the context switching operations for the other software threads. Additionally, the exact bounds from $\mathscr{O}(1)$ depend on all other software entities running on the same microprocessor and are not correlated with the input dimension m or the output dimension p of the numerical controller.

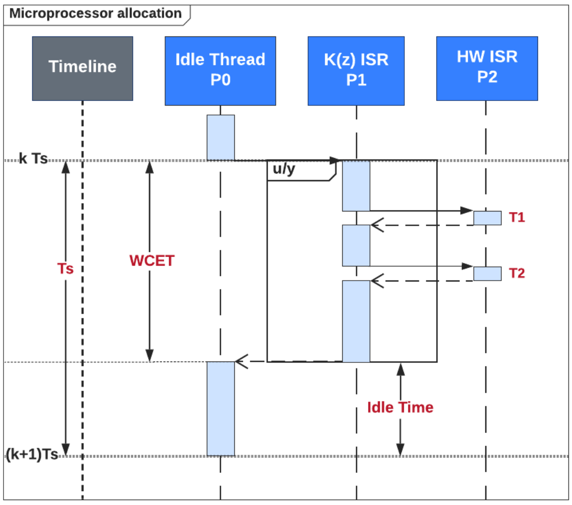

Figure 4. Sequence diagram illustrating the regulator $K(z)$ interrupt service routine execution among higher and lower priority threads for the duration of one sampling period T_s [10].

Proof. According to the schematic representation from Figure 3, starting from the control law $K \in \mathscr{G}_D$ and the specification set $\mathscr{H}$, the set $\mathcal{S}c^{K,\mathscr{H}}$ can be obtained via the MiL operator Ψ, resulting in $p_1, p_2, \ldots, p_N \in \mathcal{O}$. However, in order to measure each p_i, the set of atomic operations must be attached via the SiL operator Ξ: $p_i \mapsto \left(a_{i,1}, a_{i,2}, \cdots, a_{i,N_i}\right)$, each such atomic operation requiring exactly T_{clk}, leading to:

$$T_1 \equiv t(K) = \sum_{i=1}^{N} \sum_{l=1}^{N_i} t(a_{i,l}) = T_{clk} \cdot \sum_{i=1}^{N} \sum_{l=1}^{N_i} 1 = T_{clk} \times \left|\mathcal{S}p^{K,\mathscr{H}}\right|, \tag{31}$$

where $t(\cdot)$ is the time necessary to execute an operation or a set of operations.

Additionally, a second term $T_2 = \sum_{\text{ISR}} \lambda$ accumulates ISR switching and stack-handling operations handled by the scheduler, bounded by a processor-dependent constant $\lambda > 0$, which can be modeled as $\mathcal{O}(1) \times T_{clk}$. As such, the worst-case execution time can now be written as:

$$\texttt{WCET}(\Psi, \Xi) = T_1 + T_2 = \left(\left|\mathcal{S}p^{K,\mathscr{H}}\right| + \mathcal{O}(1)\right) \times T_{clk}, \tag{32}$$

which concludes the proof. □

Observation 1. *All possible delays caused by context switching to preemptive ISRs belonging to measurement data processing, with the cost of $\mathcal{O}(m)$, are included in the input processing step and do not remain unaccounted for in the execution time model of Theorem* (30).

Two additional performance qualifiers can be employed to globally assess the controller ISR implementation impact on the scheduling algorithm of the processor.

Definition 2. *The processor usage level qualifier relative to a fixed sampling period $T > 0$ of a discrete-time regulator $K(z) \in \mathscr{G}_D$, described in a relative manner, is defined by:*

$$\mathscr{U}(\Psi, \Xi, T) := \frac{\mathit{WCET}(\Psi, \Xi)}{T} \times 100\ [\%]. \tag{33}$$

Definition 3. *The processor idle time qualifier with respect to a fixed sampling period $T > 0$ of a discrete-time regulator $K(z) \in \mathscr{G}_D$, described in absolute units, is defined by:*

$$\mathscr{I}(\Psi, \Xi, T) := \max\{0, T - \mathit{WCET}(\Psi, \Xi)\}\ [s]. \tag{34}$$

2.3.2. Modeling Duration of Finite and Infinite-Impulse Response Topologies

Denote by $\mathbf{H} \in \mathscr{G}_D^{p \times m}$ a MIMO regulator with m inputs and p outputs, thus fully described by the expressions of $m \times p$ transfer functions $H_{ij} \in \mathscr{G}_D$:

$$(\mathbf{H}) : \begin{pmatrix} Y_1(z) \\ Y_2(z) \\ \vdots \\ Y_p(z) \end{pmatrix} = \begin{pmatrix} H_{11}(z) & H_{12}(z) & \cdots & H_{1m}(z) \\ H_{21}(z) & H_{22}(z) & \cdots & H_{2m}(z) \\ \vdots & \ddots & \cdots & \vdots \\ H_{p1}(z) & H_{p2}(z) & \cdots & H_{pm}(z) \end{pmatrix} \cdot \begin{pmatrix} U_1(z) \\ U_2(z) \\ \vdots \\ U_m(z) \end{pmatrix}. \tag{35}$$

Each element of the transfer matrix $\mathbf{H}$, i.e., H_{ij}, can be modeled as an infinite-impulse response (IIR) filter or as a finite-impulse response (FIR) filter. The case where $\mathbf{H}$ is modeled as a state-space representation is treated in the base conference paper [10], namely in Algorithm 1 and Table I, respectively.

For an arbitrary discrete-time IIR transfer function $H(z)$ of order n, define $\Omega(H)$ as the pair:

$$\Omega(H) := (n_2, n_1) = \begin{cases} \left(\lfloor \frac{n}{2} \rfloor, 1\right), & \text{if } n \text{ is odd;} \\ \left(\ \frac{n}{2},\ \ 0\right), & \text{if } n \text{ is even.} \end{cases} \tag{36}$$

Using this notation, the transfer function $H(z)$ can be written using a series of second-order sections and an additional first-order component, if necessary, as:

$$H_{IIR}(z) = \left(\frac{b_{0,1}z + b_{0,0}}{z + a_{0,0}}\right)^{n_1} \cdot \prod_{i=1}^{n_2}\left(\frac{b_{i,2}z^2 + b_{i,1}z + b_{i,0}}{z^2 + a_{i,1}z + a_{i,0}}\right) = ((H_1(z))^{n_1} \cdot \prod_{i=1}^{n_2} H_{2,i}(z), \quad (37)$$

with the terms from each triplet $(b_{i,2}, b_{i,1}, b_{i,0})$ not all null, H_1 known as a first-order section, and each $H_{2,i}$, with $i = \overline{1, n_2}$ being denoted in the literature as a second-order section (SOS).

Remark 4. *Second-order sections are usually described with an additional gain term multiplied separately to the output of the transfer function as:*

$$H_2(z) = g \cdot \frac{b_2 z^2 + b_1 z + b_0}{z^2 + a_1 z + a_0}, \; g \neq 0. \quad (38)$$

To not further complicate the notations, we will not explicitly write this gain term for every second-order section, but it will be implicitly considered in the implementation and execution time analysis.

There are multiple approaches of implementing digital biquadratic filters, a good example being the description from the monograph [34]. Table 3 exposes the four usual topologies, which principally implement the same input-output transfer function, but with important differences regarding numerical stability when selecting fixed-point or floating-point implementations. These configurations are referred to as the canonical forms: Direct Form I, Direct Form II, Transposed Form I, Transposed Form II, which differ in the numerical properties of their implementations and in the number of necessary delay elements. As observed in the third column of the table, all biquad topologies are based on four additions, five multiplications, and a different number of load/store operations, depending on the definition of the internal state variables. Such implementation details are relevant when studying particularized structures, such as in the situations treated in [35,36].

Table 3. Digital biquadratic topology implementations; all difference equations implement the same input-output second-order transfer function, but differ through the configurations of the state signals.

IIR SOS Topology	Difference Equation	min $\left|\mathcal{S}c^{K,\mathcal{H}}\right|$
Direct Form I (DFI)	$y[k] = g \cdot \left(\sum_0^2 b_i u[k-i] - \sum_1^2 a_i y[k-i]]\right).$	$4a, 6m, 12\ell$
Direct Form II (DFII)	$\begin{cases} y[k] = g \cdot \sum_0^2 (b_i \cdot x[k-i]); \\ x[k] = u[k] - \sum_1^2 (a_i \cdot x[k-i]). \end{cases}$	$4a, 6m, 14\ell$
Transposed Direct Form I (TDFI)	$\begin{cases} y[k] = g \cdot \sum_0^2 (b_i \cdot x)[k-i]; \\ x[k] = u[k] - \sum_1^2 (a_i \cdot x)[k-i]. \end{cases}$	$4a, 6m, 14\ell$
Transposed Direct Form II (TDFII)	$\begin{cases} y[k] = g \cdot (b_0 u[k] + x_1[k-1]); \\ x_1[k] = b_1 u[k] - a_1 y[k] + x_2[k-1]; \\ x_2[k] = b_2 u[k] - a_2 y[k]. \end{cases}$	$4a, 6m, 16\ell$

The ISR model for IIR controller structures is based on the pseudocode exposed in Algorithm 1. hlBased on the specifications for $\mathcal{H}$, each line of the pseudocode will have a set of mandatory mathematical operations, along with optional operations, which will be accounted for in the execution time analysis model through different constant weights.

Algorithm 1 Infinite-impulse response (IIR) filter interrupt service routine (ISR)

Input: $H_{ij}(z)$ as in (35), topology as in Table 3
Output: Execution time profiler for routine `iirFiltIsr`

```
Construct software structures $H_1$, $H_{2,1}$–$H_{2,n_2}$ according to (37) and Table 3
Initialize delays involved in second-order sections to zero
procedure IIRFILTISR($\Omega(H)$,$H_2$[ ] [,$H_1$])
    Read and scale $u[k]$ from input device
    $\widetilde{u} \leftarrow u[k]$
    for $i \leftarrow 1$ to $n_2$ do
        $\widetilde{u} \leftarrow H_2[i]$.call($\widetilde{u}$)        ▷ The output from $H_2[i-1]$ becomes input to $H_2[i]$
    end for
    [$\widetilde{u} \leftarrow H_1$.call($\widetilde{u}$)]
    $y[k] = \widetilde{u}$
    Scale $y[k]$ and write to output device
end procedure
function CALL($H$, topology)    ▷ First or second-order section call method for IIR filter
    Read input $u[k]$
    Shift input delays $u[k]$, $u[k-1]$ , [$u[k-2]$]
    Compute $y[k]$ according to input topology as in Table 3
    Shift output delays $y[k]$, $y[k-1]$ , [$y[k-2]$]
    Scale by second-order section gain $g$
    return $y[k]$
end function
```

In ref. [10], an analysis has been performed on the simplified case of a nth order IIR SISO transfer function in a series connection, described as:

$$H^s_{IIR}(z) := \frac{Y(z)}{U(z)} = \frac{b_m z^m + b_{m-1} z^{m-1} + \cdots + b_0}{z^n + a_{n-1} z^{n-1} + a_{n-2} z^{n-2} + \cdots + a_0}, \tag{39}$$

which, by design, it can implicitly include a delay z^{-n_d} by forcing the first n_d coefficients b_i to zero. Such a transfer function has its corresponding difference equation as:

$$y[k] = -\sum_{i=0}^{n-1} a_i \cdot y[k-i] + \sum_{j=0}^{m-1} b_j \cdot u[k-j]. \tag{40}$$

A further particularization on the structure of $H_{ij}(z)$ is to consider the expression of a FIR filter topology, which, by design, discards the previous output delays. The present command signal $\mathbf{y}[k]$ will, as such, depend only on an array of delays, i.e., delay tap of inputs $u[k-i]$, modeled as:

$$H_{FIR}(z) = g \cdot z^{-n_d} \cdot \sum_{i=0}^{m} b_i \cdot z^{-i} \equiv g \cdot \sum_{k=0}^{N-1} h_k \cdot z^{-k}, \; g \neq 0. \tag{41}$$

Four typical canonical forms are distinguished for FIR-type filters, namely the Direct Form (DF), Direct Form Transposed (DFT), Symmetric and Antisymmetric [34], respectively, with their definitions exposed in Table 4. The main difference between DF and DFT is that for the former, the delay word lengths are that of the input signals $u[k-i]$, while for the latter, the delays have the word length of the accumulator variable. The Symmetric and Antisymmetric cases make use of the linear phase of the filter through the regularity of the first $\left\lfloor \frac{N}{2} \right\rfloor$ coefficients as the symmetrical or antisymmetrical equivalents of the latter half of the coefficients. As mentioned in the third column of the table, this has a significant impact on the necessary multiplications and load operations involved in the implementation of

$y[k],\ k \geq 0$. Algorithm 2 emphasizes the corresponding SISO FIR filter ISR routine starting from the mathematical basis of Equation (41) and information from Table 4.

Table 4. Digital FIR filter topology implementations; all difference equations implement the same input-output Nth order transfer function, but vary through the configuration of the accumulator.

FIR Topology	Difference Equation	$\min\left\lvert \mathcal{Sc}^{K,\mathscr{H}} \right\rvert$
Direct Form	$y[k] = g \cdot \sum_0^{N-1}(h[i]u[k-i])$.	$N\mathcal{a},\ (N{+}1)\mathcal{m},\ (2N{+}2)\ell$
Direct Form Transposed	$y[k] = g \cdot \sum_0^{N-1}(h[i] \cdot u)[k-i]$.	$N\mathcal{a},\ (N{+}1)\mathcal{m},\ (2N{+}2)\ell$
Symmetric	$\begin{cases} y[k] = g\cdot \sum_0^{\lfloor N/2 \rfloor} h[i](u[i]{+}u[N{-}1{-}i]); \\ h[i] = h[N{-}1{-}i],\ i = \overline{0, \lfloor N/2 \rfloor}. \end{cases}$	$N\mathcal{a},\ \left(\left\lfloor \frac{N}{2} \right\rfloor{+}1\right)\mathcal{m},$ $\left(N{+}\left\lfloor \frac{N}{2} \right\rfloor{+}2\right)\ell$
Anti-symmetric	$\begin{cases} y[k] = g\cdot \sum_0^{\lfloor N/2 \rfloor} h[i](u[i]{-}u[N{-}1{-}i]); \\ h[i] = -h[N{-}1{-}i],\ i = \overline{0, \lfloor N/2 \rfloor}. \end{cases}$	$N\mathcal{a},\ \left(\left\lfloor \frac{N}{2} \right\rfloor{+}1\right)\mathcal{m},$ $\left(N{+}\left\lfloor \frac{N}{2} \right\rfloor{+}2\right)\ell$

Corollary 1. *Given a MIMO transfer matrix* $\mathbf{H} \in \mathscr{G}_D^{p\times m}$ *as in* (35)*, where each component* $H_{ij}(z)$*,* $i \in \overline{1,p}, j \in \overline{1,m}$ *can be described as an IIR filter of form* (37)*, with second-order sections as in Table* 3 *or a FIR filter of form* (41)*, with difference equations as in Table* 4*, the WCET can be computed as:*

$$\mathtt{WCET}(\Upsilon,\Xi) = \left(\mathcal{O}(1) + \sum_{i=1}^{p}\sum_{j=1}^{m}\gamma_{ij}\cdot\min\left\lvert\mathcal{Sp}^{H_{ij},\mathscr{H}}\right\rvert\right) \times T_{clk}, \tag{42}$$

with coefficients $\gamma_{ij} > 0$ *accounting for the hardware specifications set* $\mathscr{H}$ *from Definition* 1*, each assembly operation set* $\mathcal{Sp}^{H_{ij},\mathscr{H}}$ *determined individually by the RCP application, given the microprocessor tick* $T_{clk} > 0$*, and* $\mathcal{O}(1)$ *depends only on the other higher-priority software threads.*

Algorithm 2 Finite-impulse response (FIR) filter interrupt service routine (ISR)

Input: $H_{ij}(z)$ as in (41), topology as in Table 4
Output: Execution time profiler for routine `firFiltIsr`

```
Construct software structure $H_{FIR}$ with $N$ coefficients $h[i]$ and a $N$th order delay tap
Initialize $N$th-order input delay tap to zero
procedure FIRFILTISR($N$,$H$)
    Read and scale $u[k]$ from input device
    $\widetilde{u} \leftarrow u[k]$
    Shift input delay tap
    $\widetilde{y} \leftarrow H.\mathtt{call}(\widetilde{u})$                ▷ According to column 2 of Table 4
    $y[k] = \widetilde{y}$
    Scale $y[k]$ and write to output device
end procedure
```

The proof immediately follows based on Theorem 1 by replacing the general-purpose sequence $\mathcal{Sp}^{K,\mathscr{H}}$ with its corresponding sum of subsystems H_{ij} from the full MIMO regulator $K \equiv \mathbf{H}$ and their definitions.

3. Case Study

The Case Study section concerns with illustrating the proposed extensions on a motor servo control example, and will encompass the following key points: process description, control performance specifications, continuous-time controller design, regulator discretization methods, sampling time optimization, selection of different controller implementation topologies along with the worst-case execution time analysis and a detailed discussion of the obtained results. The reason for selecting this example is that the proposed theory can be illustrated in the same manner for a simple process model and a modest microcontroller device setup, compared to a more complex process and an adequate microcontroller setup, leading to similar processor workloads.

3.1. Process Model and Controller Synthesis

To illustrate the proposed theoretical techniques, we consider a numerical case study based on a brushed direct-current (DC) motor position control system. Figure 5 shows the closed-loop structure, emphasizing both the process model and the 2DOF regulator structure. The motor process has a control input represented by the source voltage V_a [V], along with the disturbance load torque T_d [Nm], and the output is considered to be the angular position θ [rad]. As noticeable from the transfer function blocks, the DC motor model has order three, and the nominal component values are listed in Table 5.

Table 5. DC motor physical parameters.

Parameter	Value	Parameter	Value
R	2 [Ω]	L	0.5 [H]
K_m	0.1 [Nm·A/V^2]	K_f	0.2 [Nm]
J	0.02 [kg · m^2/s^2]	K_b	0.1 [V·s/rad]

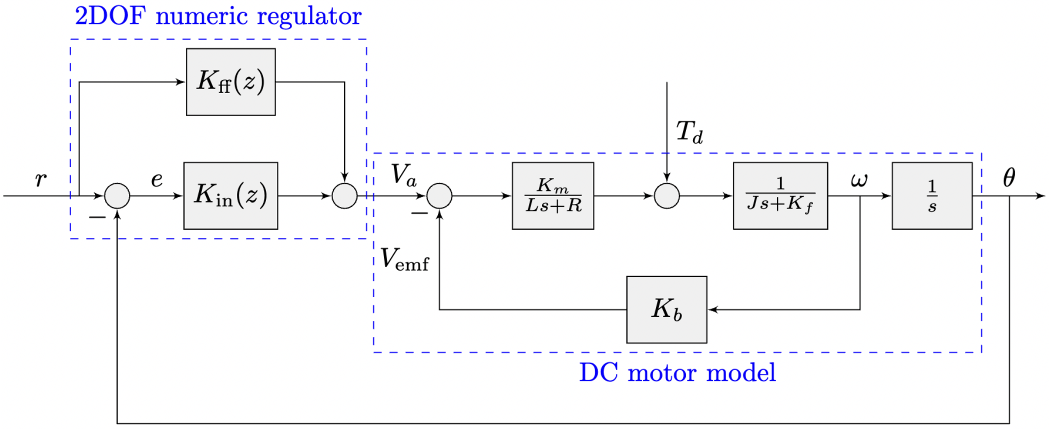

Figure 5. Closed-loop two-degree-of-freedom (2DOF) position control structure for the DC motor system with a control voltage input V_a and a disturbance load torque T_d. The 2DOF structure has an inner-loop component K_{in} designed for disturbance rejection and a feedforward component K_{ff} for good servo compensation.

The process model from its two inputs to the angular position output is described by:

$$\Theta(s) = \frac{1}{s} \cdot \frac{H_L(s)}{1 + H_r(s)H_L(s)H_a(s)} \cdot T_d(s) + \frac{1}{s} \cdot \frac{H_a(s)H_L(s)}{1 + H_r(s)H_L(s)H_a(s)} \cdot V_a(s), \tag{43}$$

with the armature transfer function H_a, along with the load component H_L and the reverse loop term H_r which denotes the back-electromotive voltage constant K_b:

$$H_a(s) = \frac{K_m}{Ls + R}, \quad H_L(s) = \frac{1}{Js + K_f}, \quad H_r(s) = K_b. \tag{44}$$

The 2DOF controller components have the proportional–integral–derivative plus filter (PIDF) structure for K_{in}, while the integral term is canceled for the feedforward controller K_{ff}. This structure allows straightforward implementation in many industrial contexts as such PIDF regulators can be directly acquired and there are multiple validated approaches in the literature for their parameter tuning [18,20]. As such, their mathematical expressions become:

$$K_{\text{in}}(s) = \frac{U_{\text{in}}(s)}{R(s)} = K_P + K_I\frac{1}{s} + K_D\frac{s}{T_f s + 1}, \tag{45}$$

$$K_{\text{ff}}(s) = \frac{U_{\text{ff}}(s)}{R(s)} = \widetilde{b} \cdot K_P + \widetilde{c} \cdot K_D\frac{s}{T_f s + 1}, \tag{46}$$

with the feedforward parameters usually specified as b and c, with $\widetilde{b} = -1 + b$ and $\widetilde{c} = -1 + c$. The command signal applicable to the motor input is comprised of two components derived from the 2DOF regulator as:

$$U(s) = U_{\text{in}}(s) + U_{\text{ff}}(s) = K_{\text{in}}(s)E(s) + K_{\text{ff}}(s)R(s). \tag{47}$$

The closed-loop control specifications were selected as follows: a reference tracking settling time of $t_s \leq 1.5$ [s] with an overshoot $M_p \leq 0.05 \equiv 5[\%]$, with the rise time as short as possible. Additionally, regarding the disturbance rejection specifications, a load torque of 1 [Nm] must be rejected in less than $t_s^d \leq 1$[s], i.e., its effect on the output measurement should become less than 0.1 [rad] in the specified t_s^d, with a maximum allowed disturbance of $y_{max}^d = 0.65$ [rad]. The recommended approach in such designs [12] is to tune the inner PIDF controller K_{in} to account for the disturbance rejection coefficients, as K_{ff} does not influence that control loop, as written in the discrete-time counterpart expressions (54). The PIDF parameter tuning has been done using global optimization methods by encompassing the desired specifications. The first iteration which covered all disturbance rejection performances was accepted and halted the optimization procedure. Additionally, a further optimization as been performed on the 2DOF parameters b and c of (46) with, again, halting the procedure after the reference tracking requirements have been fulfilled. The outcomes of the regulator tuning are illustrated in Table 6, where, alongside the PIDF and additional PD synthesis, a separate 1DOF PIDF regulator has been also added, to account for only the tracking response.

Table 6. Regulator designs leading to the proposed 2DOF control and their corresponding coefficients.

Regulator	K_p	K_i	K_d	T_f	b	c
PID (1DOF): tracking	21.0666	8.757	7.7497	0.0014717	1	1
PID (1DOF): disturbance rejection	52.6665	70.0560	7.7497	0.0014717	1	1
PID (2DOF)	52.6665	70.0560	7.7497	0.0014717	0.4	0.2

The closed-loop step responses for the reference tracking and disturbance rejection problems are portrayed in Figure 6, showing the effects of the three controller examples from Table 6, case in which the 2DOF structure is validated, as the obtained performance metrics are $t_s \approx 1.5$[s], $M_p \approx 0[\%]$, rise time $t_r \approx 0.75$ [s] and steady-state error $\varepsilon_{ss} = 0$. As illustrated in the figure, the 2DOF structure manages to ensure both the transient response performances and the disturbance rejection behavior, being the only regulator from the proposed triplet to cover both areas.

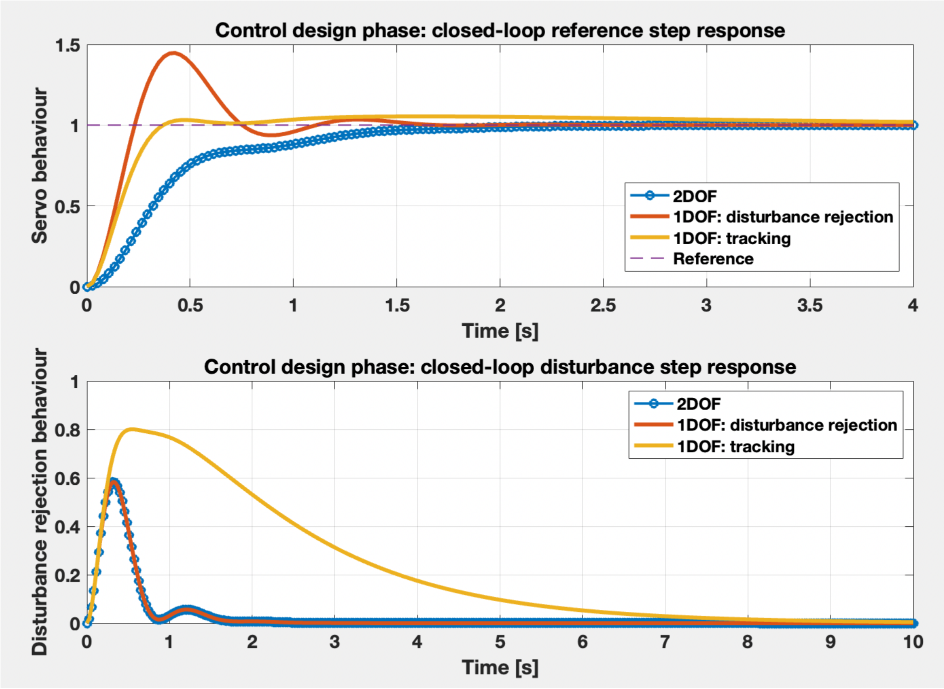

Figure 6. Closed-loop continuous-time domain simulations; step reference responses along with step disturbance rejections considering 1DOF regulators designed for servo tracking, disturbance rejection, and for performance in both cases, respectively.

3.2. Sampling Rate Selection

For the discretization of the PID regulators, the forward Euler method has been considered for the integrator term, with the approximation:

$$s \approx \frac{z-1}{T} = \frac{1-z^{-1}}{Tz^{-1}}, \tag{48}$$

while the derivative term is discretized using the backward Euler method as:

$$s \approx \frac{z-1}{Tz} = \frac{1-z^{-1}}{T}, \tag{49}$$

leading to the expression of K_{in}:

$$K_{\text{in}}(z) = K_P + \frac{K_I T}{z-1} + \frac{K_D}{T_f + \frac{Tz}{z-1}}, \tag{50}$$

with an expression of K_{ff} also as:

$$K_{\text{ff}}(z) = \widetilde{b} \cdot K_P + \widetilde{c} \cdot \frac{K_D}{T_f + \frac{Tz}{z-1}}, \tag{51}$$

which will be further used in the proceeding illustration of the sampling time analysis.

Starting from the continuous-time open-loop model from (43), the discrete-time equivalent using the zero-order hold method becomes:

$$\Theta(z) = \mathfrak{Z}\left\{\mathcal{L}^{-1}\{H_{zoh,T}(s)H_{aux}(s)\}\right\} \cdot T_d(z) + \mathfrak{Z}\left\{\mathcal{L}^{-1}\{H_{zoh,T}(s)H_a(s)H_{aux}(s)\}\right\} \cdot V_a(z), \tag{52}$$

with the auxiliary notations:

$$H_{aux}(s) = \frac{1}{s} \cdot \frac{H_L(s)}{1 + H_a(s)H_L(s)H_r(s)},\ \Theta(z) = H^{\text{op}}_{\text{dist}}(z) \cdot T_d(z) + H^{\text{op}}_{\text{servo}}(z) \cdot V_a(z). \tag{53}$$

The closed-loop system's expression thus becomes:

$$\Theta(z) = \frac{H^{\text{op}}_{\text{dist}}(z)}{1 + H^{\text{op}}_{\text{servo}}(z)K_{\text{in}}(z)} \cdot T_d(z) + \frac{H^{\text{op}}_{\text{servo}}(z)(K_{\text{ff}}(z) + K_{\text{in}}(z))}{1 + H^{\text{op}}_{\text{servo}}(z)K_{\text{in}}(z)} \cdot R(z). \tag{54}$$

By imposing the functionals $\mathcal{S}^{\overline{\Omega}}_{K_{\text{in}}}, \mathcal{S}^{\overline{\Omega}}_{K_{\text{ff}}}, \mathcal{I}_{K_{\text{in}}}, \mathcal{I}_{K_{\text{ff}}}, \mathcal{S}^{\overline{\Omega}}_{H^r_{cl}}, \mathcal{S}^{\overline{\Omega}}_{H^d_{cl}}, \mathcal{T}$, and $\mathcal{I}^{\alpha_\infty}_{\text{stab}(H_0)}$ from Equation (23), two weighting sets c_1–c_7 were considered as in Table 7, corresponding to two distinct experiments, the first numerical column designating emphasis on the difficulty of implementation functionals, while the second numerical column coefficients focus mainly on open-loop and closed-loop fidelity. With said coefficients, using a general-purpose PSO implementation as specified in Remark 3 [28], the optimal implementability sampling period becomes $T^{opt}_{s,1} = 2.866 \times 10^{-3}$ [s], while the fidelity sampling period is obtained at $T^{opt}_{s,2} = 1.260 \times 10^{-4}$ [s] $\ll T^{opt}_{s,1}$. To further extend the analysis, two extra sampling periods will be added in the discussion, the first representing the value obtained by applying the classical Shannon–Nyquist theorem (9), leading to a sampling period smaller than half the least time constant of the regulator, i.e., $T_{s,3} < \frac{T_f}{2} \Rightarrow T_{s,3} = 7.0081 \times 10^{-4}$ [s], while the latter relevant value is considered to be $T_{s,4} = T_{s,1} \cdot (1 + 0.03) = 2.9520 \times 10^{-3}$, which represents a 3[%] disturbance increase on the ideal implementability sampling rate, which causes instability in the closed-loop system. Figure 7 gathers all functionals and their weighted sums in its six subfigures, while also marking the positions of $T^{opt}_{s,1}$, $T^{opt}_{s,2}$ and $T_{s,3}$. The open and closed-loop similarity functionals J_1, J_2, J_5, J_6 are non-monotonic with respect to a variable T_s, while the implementability functionals J_3, J_4, J_7 are principally monotonically decreasing. The existing exceptions appear due to numerical errors.

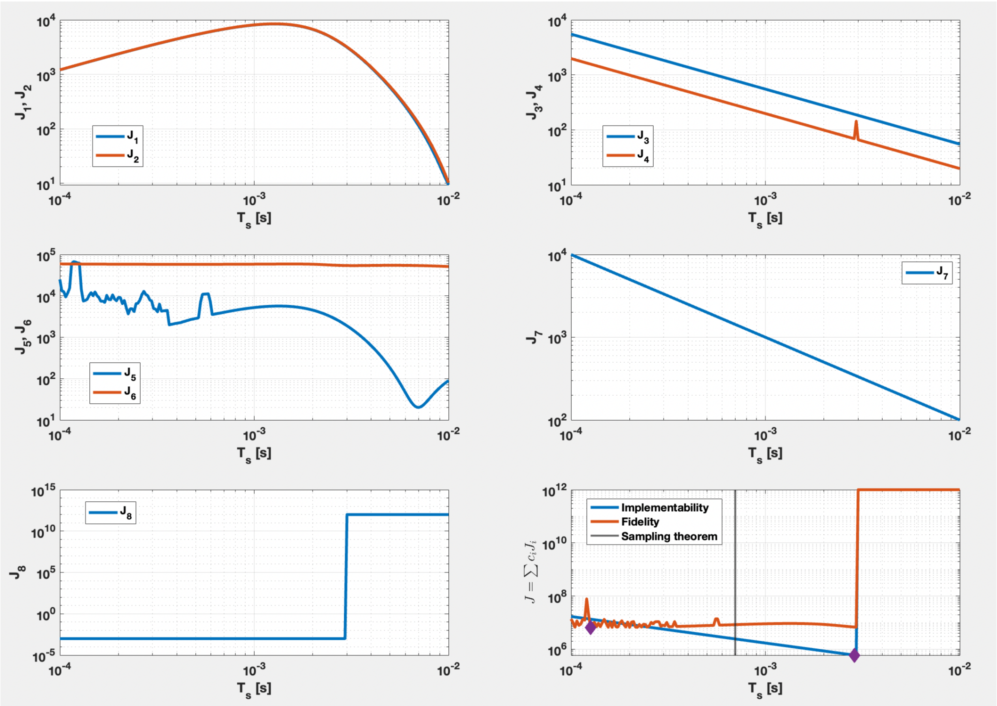

Figure 7. Functionals $J_1 - J_8$, corresponding to $\mathcal{S}^{\overline{\Omega}}_{K_{\text{in}}}, \mathcal{S}^{\overline{\Omega}}_{K_{\text{ff}}}, \mathcal{I}_{K_{\text{in}}}, \mathcal{I}_{K_{\text{ff}}}, \mathcal{S}^{\overline{\Omega}}_{H^r_{cl}}, \mathcal{S}^{\overline{\Omega}}_{H^d_{cl}}, \mathcal{T}, \mathcal{I}^{\alpha_\infty}_{\text{stab}(H_0)}$, respectively, and their weighted sum as in Equation (23) for both performed experiments. Besides the implementability and fidelity cases, a classical sampling-theorem approach is illustrated.

Table 7. Functional weighting coefficients for the two considered experiments of the DC motor case study: main focus on implementability of the resulting controller versus the focus on fidelity compared to the continuous-time control counterpart.

Parameter	Implementability Case Value	Fidelity Case Value
c_1	0.1	100
c_2	0.1	100
c_3	2000	1
c_4	2000	1
c_5	0.1	300
c_6	0.1	100
c_7	200	1
T_s^{opt}	$T_{s,1}^{opt} = 2.866 \times 10^{-3}$ [s]	$T_{s,2}^{opt} = 1.260 \times 10^{-4}$ [s]

There are multiple approaches for implementing the two PID regulators. Both $K_{\text{in}}(z)$ and $K_{\text{ff}}(z)$ can be fully encompassed into a biquadratic filter topology and, furthermore, a first-order structure for K_{ff}. The two main ones, given the simplicity of their structure, would be to implement it in parallel versus in series. For the inner regulator, the parallel topology, denoted with the superscript p can be split into three subsystems:

$$K_{\text{in}}^{p}(z) = H_{in,1}(z) + H_{in,2}(z) + H_{in,3}(z) = K_P + \frac{K_I T}{z-1} + \frac{K_D(z-1)}{(T_f+T)z - T_f}, \tag{55}$$

while the series topology, denoted with the superscript s, is:

$$K_{\text{in}}^{s}(z) = \frac{b_2 z^2 + b_1 z + b_0}{z^2 + a_1 z + a_0} = g \cdot \frac{\widetilde{b}_2 z^2 + \widetilde{b}_1 z + \widetilde{b}_0}{a_2 z^2 + a_1 z + a_0}, \;\; a_2 = 1. \tag{56}$$

where the equivalent normalized coefficients are obtained:

$$b_2 = \frac{K_P(T_f+T) + K_D}{T_f+T}; \;\; b_1 = \frac{-K_P(2T_f+T) + K_I T(T_f+T) - 2K_D}{T_f+T}; \tag{57}$$

$$b_0 = \frac{K_P T_f - K_I T T_f + K_D}{T_f+T}; \;\; a_1 = -\frac{2T_f+T}{T_f+T}; \;\; a_0 = \frac{T_f}{T_f+T}. \tag{58}$$

In the same manner, the parallel form of the feedforward regulator is adapted as:

$$K_{\text{ff}}^{p}(z) = \widetilde{b} H_{in,1}(z) + \widetilde{c} H_{in,3}(z) = \widetilde{b} \cdot K_P + \widetilde{c} \cdot \frac{K_D(z-1)}{(T_f+T)z - T_f}, \tag{59}$$

with a first-order series form of:

$$K_{\text{ff}}^{s}(z) = \frac{b_1 z + b_0}{z + a_0} = g \cdot \frac{\widetilde{b}_1 z + \widetilde{b}_0}{a_1 z + a_0}, \;\; a_1 = 1. \tag{60}$$

and equivalent normalized coefficients:

$$b_1 = \frac{\widetilde{b} K_P (T_f+T) + \widetilde{c} K_D}{T_f+T}; \;\; b_0 = \frac{-\widetilde{b} K_P T_f - \widetilde{c} K_D}{T_f+T}; \;\; a_0 = \frac{-T_f}{T_f+T}. \tag{61}$$

3.3. Execution Time Analysis

After the discretization procedure as specified by (48) and (49), using the right-hand side notation for the coefficients in (56) and (60), the numerical values of the coefficients become as in Table 8, using the for deduced sampling rates from Section 3.2, $T_{s,1}^{opt}$, $T_{s,2}^{opt}$, $T_{s,3}$, $T_{s,4}$. As observed, the gain coefficient $g \equiv b_2$ for K_{in} and $g \equiv b_1$ for K_{ff}, respectively, remains invariant in this set of experiments, while the remaining non-unitary coefficients vary with respect to $T_s > 0$ and necessitate increasingly more decimals as the sampling rate tends to zero.

Table 8. DC motor case study discrete-time ideal coefficients using the deduced sampling rates.

Coefficient	$T_{s,1}^{opt}$	$T_{s,2}^{opt}$	$T_{s,3}$	$T_{s,4}$
g: $K_{\text{in}}(z)$	5.3184815×10^3	5.3184815×10^3	5.3184815×10^3	5.3184815×10^3
$\widetilde{b}_2$: $K_{\text{in}}(z)$	1	1	1	1
$\widetilde{b}_1$: $K_{\text{in}}(z)$	−1.980677	−1.999150	−1.995275	−1.980097
$\widetilde{b}_0$: $K_{\text{in}}(z)$	0.980751	0.999150	0.995279	0.980175
a_1: $K_{\text{in}}(z)$	−0.052546	−1.914358	−1.523809	0.005877
a_0: $K_{\text{in}}(z)$	−0.947453	0.914358	0.523809	−1.005877
g: $K_{\text{ff}}(z)$	-4.244251×10^3	-4.244251×10^3	-4.244251×10^3	-4.244251×10^3
$\widetilde{b}_1$: $K_{\text{ff}}(z)$	1	1	1	1
$\widetilde{b}_0$: $K_{\text{ff}}(z)$	−0.985500	−0.999362	−0.996454	−0.985065
a_0: $K_{\text{ff}}(z)$	0.947453	−0.914358	−0.523809	1.005877

To account for the necessary word length analysis, two frequently used configurations have been considered: the first case is to store the operands, i.e., coefficients and inputs, states, outputs into 16-bit registers, considered the standard length for the RISC machine hosting the 2DOF controller, followed by a set of 32-bit length registers, which will increase the working precision, but with added execution time overhead, as modeled in continuation. Given the dynamic range of the final filter gains g, this final multiplication will be considered separately. Thus, the set $\mathscr{H}$ for this case study will encompass the properties:

- it has a clock tick period of $T_{clk} = \frac{1}{1\,[\text{MHz}]} = 1$ [μs], obtainable on standard microcontrollers by configurating the phase-locked loop circuit to a lower-power setting;
- implements the second-order sections using the Direct Form II for K_{in} and a series connection for the first-order term K_{ff}, denoted $y[k] = f(y[k-i], u[k-i])$;
- two configurations for the word lengths of the operands: 16-bit and 32-bit;
- applies saturation on output command signals $y[k]$;
- applies anti-windup on the integrator term of K_{in} using the back-calculation method:

$$w_{\underline{I},\overline{I}}(x[k+1]) = x[k] + \left[K_w \cdot \left(\mathfrak{s}_{\underline{Y},\overline{Y}}(y[k]) - y[k]\right)\right] \cdot (K_i T) \cdot u[k], \tag{62}$$

  with the additional parameters K_w, $\underline{I}$, $\overline{I}$ represented using a 16-bit word length;
- the output measurement Θ is gathered as a sum of impulses, with a maximum expected frequency of $f_{max} = 80{,}000$ impulses per 100 [ms] time unit, and each such impulse triggers a hardware interrupt with a 15 assembly operation stack commutation cost; the scaling is then performed using a multiplication with a 16-bit variable;
- K_{ff} accepts the reference signal $r[k]$ as input, while K_{in} accepts the error $e[k] = r[k] - \Theta[k]$.

The mathematical operations $p_i \in \mathfrak{O}$, along with their assembly instruction correspondents as in (29) are detailed in Table 9, with an emphasis on each type of operation based on its physical significance, as written in the last column. The hyperparameters γ_i in the case of standard 16-bit word lengths have been considered with the value 1, denoting that each base arithmetical operations costs only T_{clk}, while the values are scaled upwards for the case when the operands exceed the word length to 32 bits. The impulse counter assembly operation cost for $\Theta[k]$ has been computed as 80,000 $\times 0.1 \times 15 \times T$. Additionally, Table 10 totalizes the number of assembly operations based on the previous table's description, along with computing the worst-case execution times for the three stable sampling rate values: $T_{s,1}^{opt}$–$T_{s,3}$. As seen from the table, the optimal sampling rate solution featuring implementability emphasis $T_{s,1}^{opt}$ occupies the microprocessor for less than 25[%] of its capability in either 16 or 32-bit quantizations alike, with the Shannon theorem approach $T_{s,3}$ following with sufficient headroom in the scheduler algorithm, while the fidelity-based approach in this case occupies the scheduler with small margins in the case of 16-bit quantizations, and exceeds the allowed time frame in the case of the 32-bit configuration.

Table 9. Encountered operations in the implementation of the 2DOF structure for the DC motor case study in the hypothesis of a base word length of $L = 16$ bits and two considered quantization levels for the controller coefficients: 16-bit and 32-bit lengths, respectively.

$p_i = (\#r_i)$	p_i	K_{in}^s	K_{in}^p	K_{ff}^s	K_{ff}^p	γ_i ***16b***	γ_i ***32b***	***Observations for*** $K(z),\mathcal{H}$
1	ℓ	14	18	8	13	1	2	$y[k] = f(y[k-i], u[k-i])$
1	$\mathscr{a}$	4	5	2	3	1	2	$y[k] = f(y[k-i], u[k-i])$
1	$\mathscr{m}$	6	5	4	4	1	6	$y[k] = f(y[k-i], u[k-i])$
8	$\mathscr{s}$	1	1	1	1	1	2	$\mathscr{s}(y[k])$
20	$\mathscr{w}$	1	1	0	0	1	6	Integrator anti-windup
$120 \times 10^3 \times T_s$	$\mathscr{a}$	1	1	0	0	1	1	$\Theta[k]$ impulse counter
1	$\mathscr{m}$	1	1	0	0	1	6	$\Theta[k]$ scaling
1	$\mathscr{a}$	1	1	0	0	1	2	$e[k] = r[k] - \Theta[k]$
1	ℓ	3	3	0	0	1	2	$e[k] = r[k] - \Theta[k]$
1	ℓ	2	2	1	1	1	2	$y[k-i] \leftarrow y[k-i-1]$
1	ℓ	2	2	1	1	1	2	$u[k-i] \leftarrow u[k-i-1]$

Table 10. DC motor case study WCET analysis in the conditions from Table 9, by emphasizing the number of assembly operations $\left|\mathcal{S}\mathscr{p}^{K,\mathcal{H}}\right|$ necessary to implement the 2DOF structure comprised of K_{in} and K_{ff}, along with the processor usage level for the three stable sampling rate values.

	16b K_{in}^s	***16b*** K_{ff}^s	***16b*** K_{in}^p	***16b*** K_{ff}^p	***32b*** K_{in}^s	***32b*** K_{ff}^s	***32b*** K_{in}^p	***32b*** K_{ff}^p
$\left\|\mathcal{S}\mathscr{p}^{K,\mathcal{H}}\right\|$	77	24	81	30	246	64	250	76
$\left\|\mathcal{S}\mathscr{p}^{\{K_{\text{in}},K_{\text{ff}}\},\mathcal{H}}\right\|$	101		111		310		326	
`WCET` $T_{s,1}^{opt} = 2866[\mu\text{s}]$	429[μs]		439[μs]		638[μs]		654[μs]	
`WCET` $T_{s,2}^{opt} = 126[\mu\text{s}]$	101[μs]		111[μs]		310[μs]		326[μs]	
`WCET` $T_{s,3} = 700.8[\mu\text{s}]$	170[μs]		180[μs]		379[μs]		395[μs]	
$\mathcal{U}\left(\Psi, \Xi, T_{s,1}^{opt}\right)$	14.96[%]		15.31[%]		22.26[%]		22.81[%]	
$\mathcal{U}\left(\Psi, \Xi, T_{s,2}^{opt}\right)$	80.15[%]		88.09[%]		246.03[%]		258.73[%]	
$\mathcal{U}\left(\Psi, \Xi, T_{s,3}^{opt}\right)$	24.25[%]		25.68[%]		54.08[%]		56.36[%]	

Based exclusively on the previous WCET analysis, there are several feasible solutions. In order to decide between the three sampling rate and quantization pair configurations, further analysis is performed on the frequency response of the K_{in} and K_{ff} controllers, along with the closed-loop responses using said controllers. As such, Figures 8 and 9, respectively, gather the previously said behaviors. In addition to the three sampling rates, for completeness, the fourth, unstable sampling rate $T_{s,4}$ is shared, along with the ideal continuous-time equivalent controllers in the frequency-response plot. In both figures, columns one and two expose the 32-bit and 16-bit quantization configurations, respectively, while the lines distinguish between the K_{in}, K_{ff} controllers in the frequency-response figure and reference step response compared to the step disturbance responses in the time-domain simulation, respectively. The main conclusion drawn from this final pair of Figures is that the considered 2DOF control scheme is sensitive to the coefficient quantization levels, such that for $T_{s,2}$ and $T_{s,3}$, the only acceptable solution would be to use the precise 32-bit configuration, with the exception of the implementability solution which manages to follow the imposed closed-loop transient response specifications with a reasonable degradation of the performances. The 16-bit quantization frequency responses for $K_{\text{in}}(z)$ drastically alter the integrator effect, which, in effect, disturbs the ability of the closed-loop motor system to track reference signals and to reject step-like load torque disturbances.

To conclude the case study, the acceptable solutions are to consider the implementability-based sampling rate optimum $T_{s,1}^{opt}$, with practically ideal behavior if a 32-bit word length setup is acceptable, and with a small performance degradation if only the 16-bit standard word length is supported, depending also on other execution threads running in the microprocessor, not covered in this experiment.

Further extensions, as in using more complex controllers for both the tracking regulator and the feedforward component, considering cases such as fractional-order controllers or 2DOF robust control synthesis results can be treated in an analogous manner by splitting such control laws into their component second-order sections and applying the theory from Tables 3 and 4, along with Algorithms 1 and 2, respectively.

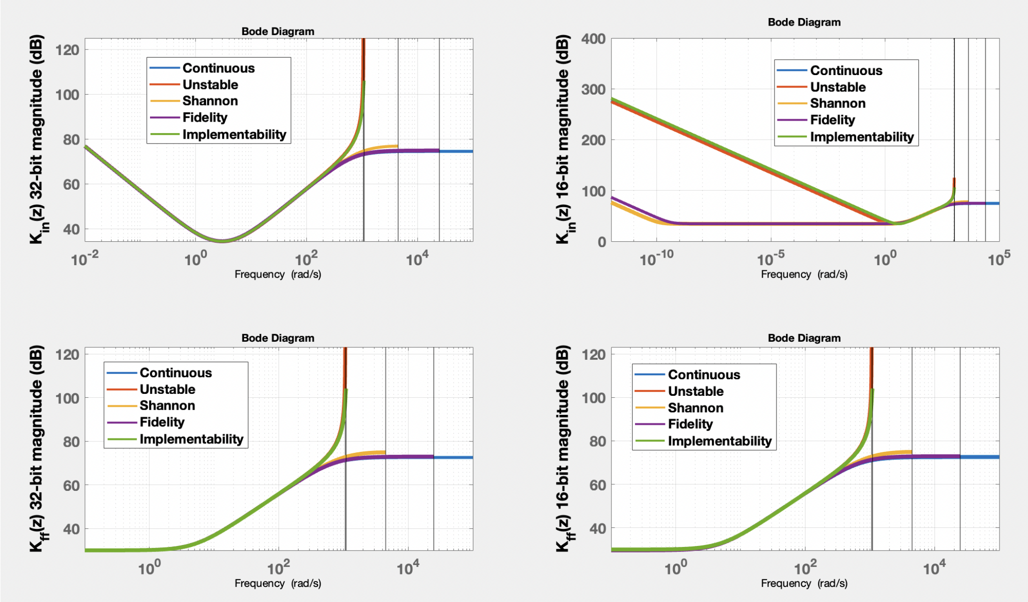

Figure 8. DC motor example regulator frequency magnitude responses; the first line gathers the inner regulator $K_{\text{in}}(z)$ with 32-bit and 16-bit quantized regulator coefficients, respectively, while the second line gathers the feedforward controller $K_{\text{ff}}(z)$ in the same 32-bit and 16-bit configurations. For completeness, the continuous-time ideal regulators $K_{\text{in}}(s)$ and $K_{\text{ff}}(s)$ were also added alongside their discrete-time counterparts.

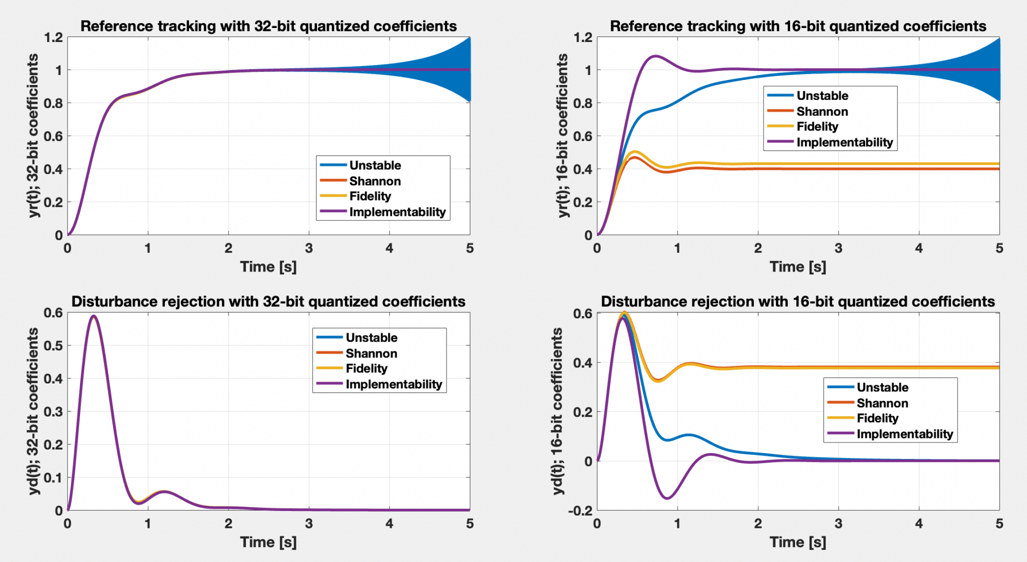

Figure 9. DC motor example closed-loop step responses using the proposed 2DOF control structure; the first line gathers the reference tracking behavior with 32-bit and 16-bit quantized regulator coefficients, respectively, while the second line gathers the disturbance rejection behavior in the same 32-bit and 16-bit configurations.

4. Conclusions

This paper gathered a set of analysis and design tools to determine the sampling rate of one and 2DOF control structures using an optimization-based approach, along with an approach of deducing a WCET for linear and time-invariant-based regulators through a formal language model which can be implemented in an RCP software tool. The execution time model is based on a deterministic, RISC architecture, where each operation is quantified with the same base clock tick duration. The end-to-end DC motor case study emphasizes the design of the controllers for the widely used benchmark system, by also focusing on the proposed framework.

Future work will concern on an online sampling rate optimization, as it is necessary in linear-parameter varying and linear-time variant models, along with the study and analysis of the continuity property of the regulator coefficient quantization effects as a function of the sampling rate. The problem of process controllability loss is also known in the literature and will be investigated for variable sampling rates in subsequent research. The mathematical framework proposed and extended in this paper will be included in the software toolbox initially proposed by the authors in [22], with the great advantage of obtaining an end-to-end solution for RCP, starting from the continuous-time controller design up to the optimal numerical implementation of said controller on a microprocessor-based system with a given set of specifications. A separate theoretical direction would be to investigate the link between sampling rate $T > 0$ and quantization step $Q > 0$ to guarantee that the minimum imposed performances are also fulfilled in the discrete domain.

Author Contributions: Conceptualization, M.Ş. and V.M.; methodology, M.Ş.; software, M.Ş. and V.M.; validation, M.Ş., V.M. and D.M.; formal analysis, P.D.; investigation, M.Ş.; resources, M.Ş. and V.M.; data curation, D.M.; writing—original draft preparation, M.Ş. and V.M.; writing—review and editing, V.M., D.M. and P.D.; visualization, M.Ş.; supervision, P.D.; project administration, P.D.; funding acquisition, P.D. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was financially supported by the Project "Entrepreneurial competences and excellence research in doctoral and postdoctoral programs—ANTREDOC", project co-funded by the European Social Fund financing agreement no. 56437/24.07.2019.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2DOF	Two-Degrees-of-Freedom
ADC	Analog-to-Digital Converter
DAC	Digital-to-Analog Converter
DC	Direct Current
DMA	Direct Memory Access
ISR	Interrupt Service Routine
LMI	Linear Matrix Inequality
LTI	Linear and Time-Invariant
MAC	Multiply-Accumulate
MiL	Model-in-the-Loop
MIMO	Multi-Input Multi-Output
NP	Non-deterministic Polynomial-time
PID	Proportional–Integral–Derivative
PIDF	Proportional–Integral–Derivative with Lowpass Filter
RAM	Random-Access Machine
RCP	Rapid Control Prototyping
SiL	Software-in-the-Loop
SOS	Second-Order Section
SIMD	Single Instruction/Multiple Data
SIMP	Single Instruction Stream/Multiple Instruction Pipelining
SISO	Single-Input Single-Output
WCET	Worst-Case Execution Time

References

1. Tan, L.; Jiang, J. *Digital Signal Processing: Fundamentals and Applications*, 3rd ed.; Elsevier: Amsterdam, The Netherlands , 2019.
2. Dohnal, F.; Rerucha, V. Practical Aspects of Digital Control Systems Design—Sampling Period Choice. In Proceedings of the IFAC Programmable Devices and Systems, Ostrava, Czech Republic, 8–9 February 2000.
3. Tomov, L.; Garipov, E. Choice of Sample Time in Digital PID Controllers. *RECENT* **2007**, *8*, 2.
4. Bensic, T.; Varga, T.; Barukcic, M.; Stil, V.J. Optimization Procedure for Computing Sampling Time for Induction Machine Parameter Estimation. *Appl. Sci.* **2020**, *10*, 3222. [CrossRef]
5. Mishra, I.; Tripathi, R.N.; Hanamoto, T. Synchronization and Sampling Time Analysis of Feedback Loop for FPGA-Based PMSM Drive System. *Electronics* **2020**, *9*, 1906.. [CrossRef]
6. Trusca, M.; Petreus, D.; Munteanu, R.A.; Sita, I.V.; Dobra, P. Wireless low cost embedded solution for electrical motors control. In Proceedings of the 5th European DSP Education and Research Conference (EDERC), Amsterdam, The Netherlands, 13–14 September 2012; pp. 144–148.. [CrossRef]
7. Duma, R.; Dobra, P.; Dobra, M.; Sita, I.V. Low cost embedded solution for BLDC motor control. In Proceedings of the 15th International Conference on System Theory, Control and Computing, Sinaia, Romania, 14–16 October 2011; pp. 1–6.
8. Janus, T.; Ulanicki, B. Effects of Sampling on Stability and Performance of Electronically Controlled Pressure-Reducing Valves. *J. Water Resour. Plan. Manag.* **2021**, *147*. . [CrossRef]
9. Wilhelm, R.; Engblom, J.; Ermedahl, A.; Holsti, N.; Thesing, S.; Whalley, D.; Bernat, G.; Ferdin, C.; Heckmann, R.; Mitra, T.; et al. The Worst-Case Execution Time Problem—Overview of Methods and Survey of Tools. *ACM Trans. Embed. Comput. Syst.* **2008**, *7*, 1–47. [CrossRef]
10. Șușcă, M.; Mihaly, V.; Morar, D.; Dobra, P. Worst-Case Execution Time Estimation for Numerical Controllers. In Proceedings of the 2022 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR), Cluj-Napoca, Romania, 19–21 May 2022; pp. 1–6. [CrossRef]
11. Nghiem, T.X.; Pappas, G.J.; Alur, R.; Girard, A. Time-triggered Implementations of Dynamic Controllers. *ACM Trans. Embed. Comput. Syst. (TECS)* **2012**, *11*, 58. [CrossRef]
12. Åström, K.J.; Hägglund, T. *Advanced PID Control*; ISA-The Instrumentation, Systems, and Automation Society: Raleigh, NC, USA, 2006.
13. Skogestad, S.; Postlethwaite, I. *Multivariable Feedback Control: Analysis and Design*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
14. Mihaly, V.; Şuşcă, M.; Dulf, E.H. μ-Synthesis FO-PID for Twin Rotor Aerodynamic System. *Mathematics* **2021**, *9*, 2504. [CrossRef]

15. Mihaly, V.; Şuşcă, M.; Morar, D.; Stănese, M.; Dobra, P. μ-Synthesis for Fractional-Order Robust Controllers. *Mathematics* **2021**, *9*, 911. [CrossRef]
16. Gahinet, P.; Apkarian, P. Automated Tuning of Gain-Scheduled Control Systems. In Proceedings of the IEEE Conference of Decision and Control, Firenze, Italy, 10–13 December 2013.
17. Mihaly, V.; Şuşcă, M.; Dobra, P. Krasovskii Passivity and μ-Synthesis Control Design for Quasi-Linear Affine Systems. *Energies* **2021**, *14*, 5571. [CrossRef]
18. Taguchi, H.; Araki, M. Two-degree-of-freedom PID Controllers—Their Functions and Optimal Tuning. In Proceedings of the IFAC Digital Control: Past. Present and Future of PID Control, Terrassa, Spain, 5–7 April 2000.
19. Wang, D.; Liu, T.; Sun, X.; Zhong, C. Discrete-time domain two-degree-of-freedom control design for integrating and unstable processes with time delay. *ISA Trans.* **2016**, *63*, 121–132. [CrossRef] [PubMed]
20. Oviedo, J.J.E.; Boelen, T.; van Overschee, P. Robust advanced PID control (RaPID): PID tuning based on engineering specifications. *IEEE Control Syst. Mag.* **2006**, *26*, 15–19. [CrossRef]
21. Șușcă, M.; Mihaly, V.; Morar, D.; Dobra, P. Quasi-Optimal Sampling Time Computation for LTI Controllers. In Proceedings of the 6th IFAC Conference on Intelligent Control and Automation Sciences, ICONS, Cluj-Napoca, Romania, 13–15 July 2022; Volume 55, pp. 87–92. [CrossRef]
22. Şuşcă, M.; Mihaly, V.; Stănese, M.; Morar, D.; Dobra, P. Unified CACSD Toolbox for Hybrid Simulation and Robust Controller Synthesis with Applications in DC-to-DC Power Converter Control. *Mathematics* **2021**, *9*, 731. [CrossRef]
23. Cormen, T.H.; Leiserson, C.E.; Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 3rd ed.; MIT Press: Cambridge, MA, USA , 2009.
24. Șușcă, M.; Mihaly, V.; Dobra, P. Fixed-Point Uniform Quantization Analysis for Numerical Controllers. In Proceedings of the 2022 IEEE Conference on Decision and Control (CDC), Cancún, Mexico, 6–9 December 2022.
25. Whorton, M.; Yang, L.; Hall, R. Similarity Metrics for Closed Loop Dynamic Systems. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, HI, USA, 18–21 August 2008. [CrossRef]
26. Hsieh, G.C.; Safonov, M.G. Conservatism of the gap metric. *IEEE Trans. Autom. Control* **1993**, *38*, 594–598. [CrossRef]
27. Vinnicombe, G. Frequency domain uncertainty and the graph topology. *IEEE Trans. Autom. Control* **1993**, *38*, 1371–1383. [CrossRef]
28. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
29. da Silva, L.R.; Flesch, R.C.C.; Normey-Rico, J.E. Analysis of Anti-windup Techniques in PID Control of Processes with Measurement Noise. In Proceedings of the 3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control PID, Ghent, Belgium, 9–11 May 2018; Volume 51, pp. 948–953. [CrossRef]
30. Tarbouriech, S.; Turner, M. Anti-windup design: An overview of some recent advances and open problems. *IET Control Theory Appl.* **2009**, *3*, 1–19. [CrossRef]
31. Singh, R.P.; Vashishtha, A.K.; Krishna, R. 32 Bit re-configurable RISC processor design and implementation for BETA ISA with inbuilt matrix multiplier. In Proceedings of the Sixth International Symposium on Embedded Computing and System Design (ISED), Patna, India, 15–17 December 2016; pp. 112–116. [CrossRef]
32. Salim, A.J.; Samsudin, N.R.; Salim, S.I.M.; Soo, Y. Multiply-accumulate instruction set extension in a soft-core RISC Processor. In Proceedings of the 10th IEEE International Conference on Semiconductor Electronics (ICSE), Kuala Lumpur, Malaysia, 19–21 September 2012; pp. 512–516. [CrossRef]
33. Murakami, K.; Irie, N.; Kuga, M.; Tomita, S. SIMP (single Instruction Stream/multiple Instruction Pipelining): A Novel High-speed Single-processor Architecture. In Proceedings of the 16th Annual International Symposium on Computer Architecture, Jerusalem, Israel, 28 May–1 June 1989; pp. 78–83. [CrossRef]
34. Proakis, J.G.; Manolakis, D.G. *Digital Signal Processing: Principles, Algorithms and Applications*, 5th ed.; Pearson, Prentice Hall: Upper Saddle River, NJ, USA, 2022; ISBN 9780137348657.
35. Essl, G. Topological IIR Filters Over Simplicial Topologies via Sheaves. *IEEE Signal Process. Lett.* **2020**, *27*, 1215–1219. [CrossRef]
36. Marjanovic, J. Deutsches Elektronen-Synchrotron (DESY), Low vs. High Level Programming for FPGA. In Proceedings of the 7th International Beam Instrumentation Conference, IBIC2018, Shanghai, China, 9–13 September 2018; JACoW Publishing: Geneva, Switzerland, 2018; ISBN 978-3-95450-201-1. [CrossRef]