



HAL
open science

Discrete-time differentiators in closed-loop control systems: experiments on electro-pneumatic system and rotary inverted pendulum

Mohammad Rasool Mojallizadeh, Bernard Brogliato, Andrey Polyakov, Subiksha Selvarajan, Loïc Michel, Franck Plestan, Malek Ghanes, Jean-Pierre Barbot, Yannick Aoustin

► To cite this version:

Mohammad Rasool Mojallizadeh, Bernard Brogliato, Andrey Polyakov, Subiksha Selvarajan, Loïc Michel, et al.. Discrete-time differentiators in closed-loop control systems: experiments on electro-pneumatic system and rotary inverted pendulum. [Research Report] INRIA Grenoble. 2021. hal-03125960v1

HAL Id: hal-03125960

<https://inria.hal.science/hal-03125960v1>

Submitted on 30 Jan 2021 (v1), last revised 3 Feb 2022 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Discrete-time differentiators in closed-loop control systems: experiments on electro-pneumatic system and rotary inverted pendulum

Mohammad Rasool Mojallizadeh^{*1}, Bernard Brogliato^{†1}, Andrey Polyakov^{‡2}, Subiksha Selvarajan^{§3}, Loïc Michel^{¶3}, Franck Plestan^{||3}, Malek Ghanes^{**3}, Jean-Pierre Barbot^{††3},
and Yannick Aoustin^{‡‡3}

¹Univ. Grenoble Alpes, INRIA, CNRS, Grenoble INP, LJK, 38000 Grenoble, France

²Inria, Univ. Lille, CNRS, Lille, France

³Université de Nantes-LS2N, 44300 Nantes, France

January 29, 2021

Abstract

This paper is dedicated to the experimental analysis of discrete-time differentiators implemented in closed-loop control systems. To this end, two laboratory setups, namely an electro-pneumatic system and a rotary inverted pendulum have been used to implement 25 different differentiators. Since the selected laboratory setups behave differently in the case of dynamic response and noise characteristics, it is expected that the results remain valid for a wide range of control applications. The validity of several theoretical results, which have been already reported in the literature using mathematical analysis and numerical simulations, has been investigated, and several comments are provided to allow one to select an appropriate differentiation scheme in practical closed-loop control systems.

Index terms— discrete-time differentiator, implicit discretization, explicit discretization, electro-pneumatic system, rotary inverted pendulum, experimental data

*mohammad-rasool.mojallizadeh@inria.fr

†bernard.brogliato@inria.fr

‡andrey.polyakov@inria.fr

§subiksha.selvarajan@eleves.ec-nantes.fr

¶loic.michel@ec-nantes.fr

||franck.plestan@ec-nantes.fr

**malek.ghanes@ec-nantes.fr

††barbot@ensea.fr

‡‡Yannick.Aoustin@univ-nantes.fr

Contents

1	Introduction	4
2	Review of the continuous-time differentiators	6
3	Time-discretization of the differentiators	10
4	Electropneumatic setup	11
4.1	Model of the electropneumatic setup	13
4.2	Sliding-mode controller	13
4.3	Conditions of the experiments	16
4.4	Identifying the impracticable experiments	19
4.5	Experiments under different sampling times	19
4.6	Gain sensitivity	21
4.7	Solver's effect	23
4.8	Results obtained for the electropneumatic setup: conclusions	23
5	Rotary inverted pendulum	24
5.1	Mathematical modeling of the rotary inverted pendulum	25
5.2	Controller design for the rotary inverted pendulum	27
5.2.1	Swing-up control	27
5.2.2	Balancing control	28
5.3	Differentiator design for the rotary inverted pendulum	28
5.4	Experiments	29
5.4.1	Conditions of the experiments	29
5.5	Effect of the parameter tuning on the results	31
5.6	Results obtained for the pendulum system	34
6	General conclusions	35

List of Figures

1	A typical control loop	6
2	Electropneumatic setup	12
3	Closed-loop diagram of the EPS	12
4	Flowchart of the first-order implicit SMC	16
5	Variation of the control signal $u(t)$ under different sampling times	20
6	Comparison of the implicit and explicit schemes	21

7	\bar{L}_2 for all methods	22
8	L_∞ for all methods	22
9	Scheme of the rotary inverted pendulum [1]	26
10	Closed-loop diagram of the RIPS	27
11	Photograph of the rotary inverted pendulum [1]	28

List of Tables

1	Overview of the continuous-time differentiators [2, 3]	7
2	Constant parameters used for the SMB differentiators	9
3	Parameters of the differentiators obtained from the tuning procedure	18
4	Impracticable experiments	19
5	Results for $h=20\text{ms}$ under oversized gain	24
6	Maximum number of iteration for real-time operation	25
7	Parameters of the differentiators obtained from the tuning procedure	30
8	Average results based on 10 experiments (parameters are shown in Table 7)	32
9	Parameters of the differentiators obtained from the tuning procedure	33
10	Average results based on 10 experiments (parameters are shown in Table 9)	34

1 Introduction

Online differentiation is an unavoidable subject in closed-loop control systems. Considering a typical control loop shown in Fig. 1, controllers usually need the error signal $f_0 = r - y$ as well as its differentiations $f_0^{(i)}(t), i \in \mathbb{N}$. Since the error signal is always polluted by a stochastic noise $\tilde{n}(t)$, design of a differentiator can be challenging since the differentiator should filter out the noise. Design of such a differentiator under different conditions was the topic of the first part of this study [2, 3].

A complete review of the known continuous-time differentiators has been recently provided in [2, 3], and several time-discretization methods have been introduced for them. Subsequently, the properties of the continuous-time differentiators under several discretization schemes in the open-loop configuration have been studied using numerical simulations and analytical calculations in [2, 3].

This work aims to answer this question: *are the theoretical results which have been provided in [3] still valid in practical closed-loop control systems, or not?* To this end, two laboratory setups, an *electro-pneumatic setup* (EPS) and a *rotary inverted pendulum setup* (RIPS) have been utilized to implement the total number of 25 differentiators which have been studied in [2, 3]. Since these setups behave differently in case of dynamic response and noise characteristics, it is expected that the results remain valid for a wide range of control applications.

Key results (based on analytical calculations and numerical simulations in the open-loop configuration) drawn in [2, 3] are listed below:

- **Fact 1:** Compared to the explicit sliding-mode based (SMB) differentiators, the implicit ones suppress the numerical chattering corresponding to the time- discretization (see Remark 5 in [2]).
- **Fact 2:** Solver-based implicit differentiators can be implemented in real-time applications with acceptable accuracy (see Sec. 5.8 in [3])
- **Fact 3:** Unlike the explicit method, the implicit one is insensitive to the gains during the sliding-phase (see Remark 5 in [2]).
- **Fact 4:** A higher-order differentiator shows a longer transient time (see Remark 2 in [2]).
- **Fact 5:** A higher-order differentiator may potentially show better robustness to noise (see Sec. 6.3 in [2]).
- **Fact 6:** The implicit differentiators supersede linear filters (LF) (see Sec. 6.3 and 6.4 in [2]).

The main contribution of this work is to validate the above-mentioned facts in the closed-loop systems using practical experiments. Moreover, the behavior of the total number of 25 known differentiators will be analyzed in practice, and several remarks will be drawn to allow one to select the most appropriate differentiator in practical control systems.

Nomenclature	
AO-STD [4]	arbitrary-order super-twisting differentiator
E-AO-STD [4]	explicit arbitrary-order super-twisting differentiator
E-GHDD [5]	explicit generalized homogeneous discrete-time differentiator
E-HDD [6]	explicit homogeneous discrete-time differentiator
E-URED [7]	explicit uniform robust exact differentiator
E-QD [8]	explicit quadratic differentiator
E-STD [9]	explicit super-twisting differentiator
E-STDAC [10]	explicit super-twisting differentiator with adaptive coefficients
GE	generalized equation
GHDD [5]	generalized homogeneous discrete-time differentiator
HD [11, 12]	homogeneous differentiator
HDD [6]	homogeneous discrete-time differentiator
HGD [13]	high-gain differentiator
I-AO-FDFF [14]	implicit arbitrary-order differentiator with first-order sliding-mode filtering
I-AO-STD [15, 16]	implicit arbitrary-order super-twisting differentiator
I-FDFF [17]	implicit first-order differentiator with first-order sliding-mode filtering
I-GHDD	implicit generalized homogeneous discrete-time differentiator
I-HDD	implicit homogeneous discrete-time differentiator
I-URED	implicit uniform robust exact differentiator
I-QD [8]	implicit quadratic differentiator
I-STD	implicit super-twisting differentiator
LF	linear filter
ODE	ordinary differential equation
QD [8]	quadratic differentiator
SI-AO-STD	semi-implicit arbitrary-order super-twisting differentiator
SI-URED	semi-implicit uniform robust exact differentiator
SI-STD	semi-implicit super-twisting differentiator
SHMD [18]	Slotine-Hedrick-Misawa differentiator
SMB	sliding-mode based
SNR	signal-to-noise ratio
STD [9]	super-twisting differentiator
STDAC [10]	super-twisting differentiator with adaptive coefficients
URED [7]	uniform robust exact differentiator
VGED [19]	variable gain exponent differentiator

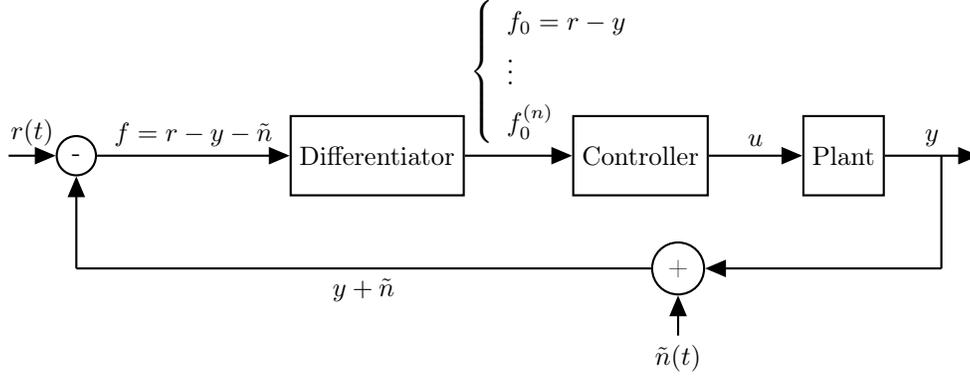


Figure 1: A typical control loop

The structure of this manuscript is as follows. The continuous-time differentiators and the corresponding discretization schemes are briefly reviewed in Sections 2 and 3, respectively, the results drawn from the EPS as well as the RIPS are provided in Sections 4 and 5, respectively. Finally, the general conclusions and remarks are provided in Section 6.

2 Review of the continuous-time differentiators

The purpose of this section is to briefly review the known continuous-time differentiators which have been recently analyzed in [2, 3]. Depending on the structure, differentiators can be categorized into several classes as indicate in Table 1. In practical applications, the input f_0 of a differentiator is always polluted by noise \tilde{n} . Hence, the differentiator should have the ability to filter-out the noise. Some basic terms of the differentiators are explained as follow:

- *SMB* differentiator refers to a kind of differentiator where discontinuous functions are used to provide a sliding regime to achieve an exact tracking.
- The numerical chattering caused by the discretization of set-valued terms can be drastically reduced by using an implicit discretization. This issue will be addressed in Section 3 (see also [2, 3]).
- Some differentiators can calculate *higher-order differentiations (HOD)* (second, third,...). Without such an ability, cascade configurations may be used to estimate higher-order differentiations using first-order differentiators [2, 3].
- Differentiators may have *multiple outputs (MO)*.
- Depending on the structure, differentiators may have one or more *tuning parameters (TP)* that should be tuned to ensure proper operation.

Table 1: Overview of the continuous-time differentiators [2, 3]

Method	SMB	SV	HOD	MO	TP	AM	Source	Remarks
SHMD	✓	✓	✓	✓	$2n$	✗	[2] (S2.1)	existence of the sliding-phase
STD	✓	✓	✗	✓	1	✗	[2] (S2.2)	finite-time convergence
AO-STD	✓	✓	✓	✓	1	✗	[2] (S2.3)	finite-time convergence
URED	✓	✓	✗	✓	2	✗	[2] (S2.4)	uniform convergence
QD	✓	✓	✗	✓	2	✗	[2] (S2.5)	improved transient
STDAC	✓	✓	✗	✓	2	✓	[2] (S2.7.1)	adaptive coefficients
VGED	✓	✗	✗	✓	4	✓	[2] (S2.7.2)	adaptive exponent
HGD	✗	✗	✓	✓	1	✗	[2] (S2.8)	linear structure
ALIEN	✗	✗	✓	✗	3	✗	[2] (S2.9)	algebraic formula
SMB: sliding-mode based				SV: set-valued				
AM: adaptation mechanism				HOD: higher-order differentiations				
TP: tunable parameters				MO: multiple outputs				
n : order of a differentiator								

- Some differentiators utilize a kind of *adaptation mechanism* (AM) to tune their parameters in an online manner.

From Table 1, it can be seen that the differentiators are mostly designed based on the sliding-mode algorithms. The Slotine-Hedrick-Misawa differentiator (SHMD) has been the first invented SMB differentiator [18]. The general form of the SHMD is as follows [18]:

$$\begin{cases} \dot{z}_i(t) \in z_{i+1}(t) - \alpha_i \operatorname{sgn}(\sigma_0(t)) - \kappa_i \sigma_0(t) \\ \dot{z}_n(t) \in -\alpha_n \operatorname{sgn}(\sigma_0(t)) - \kappa_n \sigma_0(t), \quad i = 0, \dots, n-1, \end{cases} \quad (1)$$

where $\sigma_0(t) = z_0(t) - f(t)$ is the sliding-variable, $f(t) = f_0(t) + \tilde{n}(t)$ is the input of the differentiator, $f_0(t)$ is the base signal which is polluted by noise $\tilde{n}(t)$, $z_i(t)$ is the estimation of $f_0^{(i)}(t)$, α_i and κ_i , $i = 0, 1, \dots, n$ are positive constants, n is the order of the differentiator, \in is written instead of $=$ to indicate that the right-hand side are set-valued ($\operatorname{sgn}(0) = [-1, 1]$). The existence of the sliding-phase and the behavior of the system in the reaching phase for the SHMD are studied in [18].

Arbitrary-order super-twisting differentiator (AO-STD) is another SMB differentiator introduced in [20], that exhibits several useful properties including homogeneity and finite-time convergence [21, 22] as follows

$$\begin{cases} \dot{z}_i(t) = -\lambda_i L^{\frac{i+1}{n+1}} [\sigma_0(t)]^{\frac{n-i}{n+1}} + z_{i+1}(t), \quad i = 0, \dots, n-1 \\ \dot{z}_n(t) \in -\lambda_n L \operatorname{sgn}(\sigma_0(t)), \end{cases} \quad (2)$$

where the notation is as before, L is a tuning parameter, the parameters λ_i , $i = 0, \dots, n$ are provided in table 2 [4, 20, 23, 24], and $[x]^n = |x|^n \operatorname{sgn}(x)$. For $n = 1$, AO-STD turns into the super-twisting differentiator

(STD) [9]. Uniform robust exact differentiator (URED) [7] is a modification of the STD, where the aim is to ensure the uniform convergence of the differentiator by adding extra terms as follows:

$$\begin{cases} \dot{z}_0(t) = -\lambda_0 L^{\frac{1}{2}} \left(\lceil \sigma_0(t) \rceil^{\frac{1}{2}} + \mu \lceil \sigma_0(t) \rceil^{\frac{3}{2}} \right) + z_1(t) \\ \dot{z}_1(t) \in -\lambda_1 L \left(\frac{1}{2} \operatorname{sgn}(\sigma_0(t)) + 2\mu\sigma_0(t) + \frac{3}{2} \lceil \mu\sigma_0(t) \rceil^2 \right). \end{cases} \quad (3)$$

The notation is similar to the above one, and $\mu \in \mathbb{R}_+^*$. Note that the original form of the URED was introduced with $L = 1$ [7].

Quadratic differentiator (QD) is another SMB differentiator where the aim is to improve the transient by modifying the sliding-surface as follows.

$$\begin{cases} \dot{z}_0(t) = z_1(t) \\ \dot{z}_1(t) \in \begin{cases} -\alpha F \operatorname{sgn}(\sigma_0(t)) & \text{if } \sigma_0(t)z_1(t) > 0 \\ -F \operatorname{sgn}(\sigma_0(t)) & \text{if } \sigma_0(t)z_1(t) < 0 \end{cases} \\ \sigma_0(t) = 2F(z_0(t) - f(t)) + |z_1(t)|z_1(t), \end{cases} \quad (4)$$

where $F > 0$ and $\alpha > 0$ are parameters to be tuned. The sliding variable $\sigma_0(t)$ of this differentiator has been further modified [25] to improve its convergence rate.

Some adaptation laws have been developed for the SMB differentiators to tune their parameters automatically. In this context, two different adaptation techniques, namely *adaptive coefficients* and *adaptive exponents*, have been introduced. Some studies [10, 26, 27] dealt with the adaptive laws for the coefficients, while in other studies [19, 28, 29], the adaptation mechanisms are considered for the exponents.

One of the latest adaptation mechanisms for the coefficients has been developed in [10], where the following adaptive differentiator has been proposed. Note that this differentiator is termed super-twisting differentiator with adaptive coefficients (STDAC). It reads as:

$$\begin{cases} \dot{z}_0(t) = -\lambda_0 \gamma(t) \lceil \sigma_0(t) \rceil^{\frac{1}{2}} + z_1(t) \\ \dot{z}_1(t) \in -\lambda_1 \gamma^2(t) \operatorname{sgn}(\sigma_0(t)). \end{cases} \quad (5a)$$

$$\quad (5b)$$

It can be seen that for $\gamma(t) = \sqrt{L}$, (5) leads to the standard STD. The following adaptation law $\gamma(t)$ is proposed [10]:

$$\dot{\gamma}(t) = \frac{\gamma(t)}{2} \alpha \begin{cases} |\sigma_0(t)|^{-\frac{1}{2}} & \text{for } |\sigma_0(t)| \geq 1 \\ |\sigma_0(t)| & \text{for } |\sigma_0(t)| < 1 \\ \frac{1}{\gamma(t)} - 1 & \text{for } |\sigma_0(t)| < 1.1\epsilon, \end{cases} \quad (6)$$

where $\gamma(0) = 1$, $0 < \alpha < \lambda_0$, and ϵ is a positive design constant which is selected based on the amplitudes of the chattering and noise.

The idea of adaptive exponent comes from the observation that by changing the exponent of an SMB differentiator, a trade-off can be made between the exactness and robustness to noise. The most recent study on the variable gain exponent differentiator (VGED) has been conducted in [19]. The continuous-time

Table 2: Constant parameters used for the SMB differentiators

Order	λ_0	λ_1	λ_2	λ_3	λ_4	λ_5
0	1.1					
1	1.5	1.1				
2	2	2.12	1.1			
3	3	4.16	3.06	1.1		
4	5	10.03	9.30	4.57	1.1	
5	7	23.72	32.24	20.26	6.75	1.1

VGED reads as:

$$\begin{cases} \dot{z}_0(t) = -\lambda_0 \mu |\sigma_0(t)|^{\alpha(t)} \operatorname{sgn}(\sigma_0(t)) + z_1(t) & (7a) \\ \dot{z}_1(t) = -\lambda_1 \alpha(t) \mu^2 |\sigma_0(t)|^{2\alpha(t)-1} \operatorname{sgn}(\sigma_0(t)) & (7b) \\ \dot{\gamma}(t) = -\tau \gamma(t) + \tau |f_f(t)| & (7c) \\ \alpha(t) = \frac{1}{2} \left(1 + \frac{\gamma^q}{\gamma^q + \epsilon} \right), & (7d) \end{cases}$$

where $f_f(t)$ corresponds to high-frequency components of the input. In other words, a fourth-order Butterworth high-pass filter with cutoff frequency ω_c is used to calculate $f_f(t)$ from the input $f(t)$. To decrease the number of parameters, it is assumed that $\epsilon = \frac{1}{\mu}$ [19], and λ_0 and λ_1 are presented in Table 2. In this case, the VGED only has four parameters to be tuned, i.e., μ , τ , ω_c , q .

The high-gain differentiator (HGD) is a special case of the VGED with $\alpha = 1$, introduced in [13]. The design of this differentiator was further addressed in [30]. In this study, a third-order HGD will be considered as follows (see [31, 32]).

$$\begin{cases} \dot{z}_0(t) = -L\lambda_0 [\sigma_0(t)] + z_1(t) \\ \dot{z}_1(t) = -L^2\lambda_1 [\sigma_0(t)] + z_2(t) \\ \dot{z}_2(t) = -L^3\lambda_2 [\sigma_0(t)] + z_3(t) \\ \dot{z}_3(t) = -L^4\lambda_3 [\sigma_0(t)]. \end{cases} \quad (8)$$

Obtaining the error band of the HGD in the presence of noise was the topic of [30].

There are also some algebraic continuous-time differentiators. ALIEN [33] is one of these differentiators, which calculates an arbitrary-order differentiation based on annihilators. In fact, it calculates the differentiation using integration to attenuate the noise effect. ALIEN differentiator is given as [34]:

$$z^{(n)}(t) = \frac{(-1)^n \gamma_{\kappa, \mu, n}}{T^n} \int_0^1 \frac{d^n}{d\tau^n} \{ \tau^{\kappa+n} (1-\tau)^{\mu+n} \} f(\tau \bar{T}) d\tau, \quad (9)$$

$z^{(n)}(t)$ is the n -th order differentiation of $f_0(t)$, $\gamma_{\kappa, \mu, n} = \frac{(\kappa+\mu+2n+1)!}{(\kappa+n)! (\mu+n)!}$, n is the differentiation order, \bar{T} is called the estimation window, κ and μ are two parameters which are designed from simulations [34].

3 Time-discretization of the differentiators

To implement the continuous-time differentiators introduced in Section 2, it is necessary to use a discretization method. Let $\dot{x} = g(x)$ be an ODE, where $g(\cdot)$ is a function. The Euler discretization of this ODE gives

$$x_{k+1} = h((1 - \alpha)g(x_k) + \alpha g(x_{k+1})) + x_k, \quad (10)$$

where h is the sampling time, $\alpha = 0$, $\alpha \in (0, 1)$, and $\alpha = 1$ lead to explicit, semi-implicit and (full) implicit discretizations, respectively. As can be seen, for $\alpha \in (0, 1]$, x_{k+1} appears in the input argument. To obtain this implicit argument (here x_{k+1}) at the time-step k , some extra manipulations are required (see [2, 3] for detailed explanation of time-discretization methods applied to the continuous-time differentiators).

Remark 1 *Assuming that $g(\cdot)$ in (10) is a discontinuous (set-valued) function, the implicit discretization allows to suppress the numerical chattering caused by the discretization effect based on the selection procedure. Moreover implicit discretization can provide several useful properties e.g., finite-time convergence (see [2, 3, 35]).*

Throughout the manuscript, the notations E-X, SI-X, and I-X denote the explicit ($\alpha = 0$), semi-implicit ($\alpha \in (0, 1)$), and implicit ($\alpha = 1$) discretizations of the continuous-time differentiator X, respectively. These discretizations can be obtained by utilizing (10) on the continuous time differentiators introduced in Section 2.

There are two other explicit discrete-time differentiators namely, homogeneous discrete-time differentiator (HDD) [6], and generalized homogeneous discrete-time differentiator (GHDD) [5] which can be obtained by special operations on the AO-STD. The purpose of the HDD is to keep the homogeneity of the AO-STD after discretization. Moreover, GHDD is proposed to attenuate the chattering by cancelling the discontinuous terms which appear in the recursion.

It should also be noted that the explicit discretization of the SHMD is ignored because of too much numerical chattering. Moreover, according to [17], for $n = 1$, the implicit SHMD is named I-FDFF. For $n > 1$, this differentiator is called I-AO-FDFF with this difference that an extra filtration is utilized instead of directly using $z_i, i = 0, \dots, n$ as outputs [14]. The I-FDFF has four parameters $\omega_s, \omega_f, \rho, \gamma$ that need to be tuned. On the other hand, the parameters of the I-AO-FDFF are $\omega_s, \omega_f, \rho, F, \epsilon, \alpha_1$. The implicit discretizations of the SHMD, i.e., I-FDFF and I-AO-FDFF, as well their parameter tuning are explained in [2, 3].

In addition to the introduced methods, there are still some other differentiators, e.g., Euler method, linear filters, and the Kalman's differentiator. To design the Kalman's differentiator, the differentiation problem is formulated as follows [4]:

$$\begin{cases} z_{k+1} = Az_k & (11a) \\ y_k = f_k - Cz_k, & (11b) \end{cases}$$

where $z_k = [z_{0,k}, z_{1,k}, \dots, z_{n,k}]^\top \in \mathbb{R}^{n+1}$ is the estimation vector. The parameters for a third-order Kalman differentiation are as follows (see [36] for the algorithm as well as the preliminary equations of the Kalman filter).

$$A = \begin{bmatrix} 1 & h & \frac{h^2}{2} & \frac{h^3}{3!} \\ 0 & 1 & h & \frac{h^2}{2} \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}. \quad (12)$$

The Kalman filter has one parameter, denoted by R , which is obtained based on the power of the input noise ([2, 3]).

The LF is a low-pass filter combined with the pure differentiator d/dt which can be implemented using the following formula:

$$z_{1,k} = \frac{z_{1,k-1} + c(f_k - f_{k-1})}{1 + hc}, \quad (13)$$

where the notation is as before, and c is a design parameter. For $c \rightarrow \infty$, LF leads to the Euler differentiator as follows:

$$z_{1,k} = \frac{f_k - f_{k-1}}{h}. \quad (14)$$

The prefixes E, I and SI indicate the explicit, implicit and semi-implicit discretizations. However, these prefixes are ignored for some methods (Euler, LF, ALIEN, HD, VGED and Kalman) since the implicit or semi-implicit discretizations of these differentiators have not been introduced yet. In other words, The absence of a prefix indicates that the method is implemented using the explicit discretization scheme ($\alpha = 0$ in (10)).

4 Electropneumatic setup

The EPS, which is considered in this study is shown in Fig. 2. The system is composed of two actuators, namely, the main and disturbance actuators. These actuators are coupled through a horizontal jack with the total mass M . The aim of the main actuator is to provide the required force to control the position of the jack, while the other one is to emulate disturbances which appear during real conditions. The main actuator is double acting controlled by two servo distributors with two chambers named P and N as indicated in Fig. 2. The block diagram corresponding to the closed-loop system of the EPS is shown in Fig. 3.

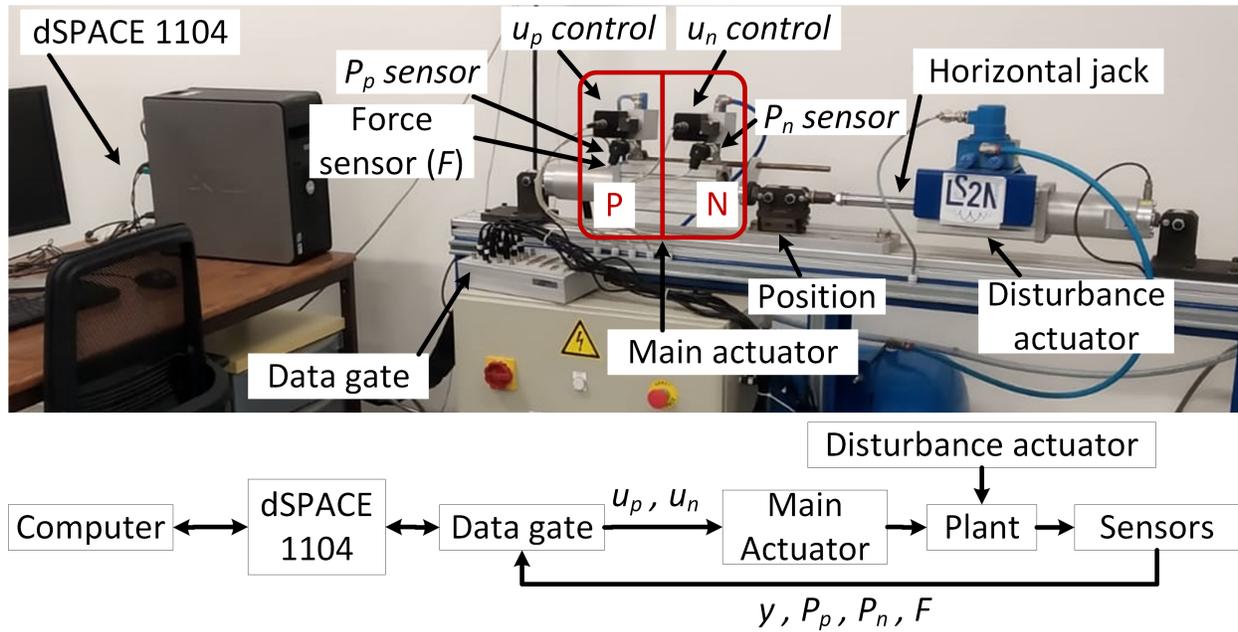


Figure 2: Electropneumatic setup

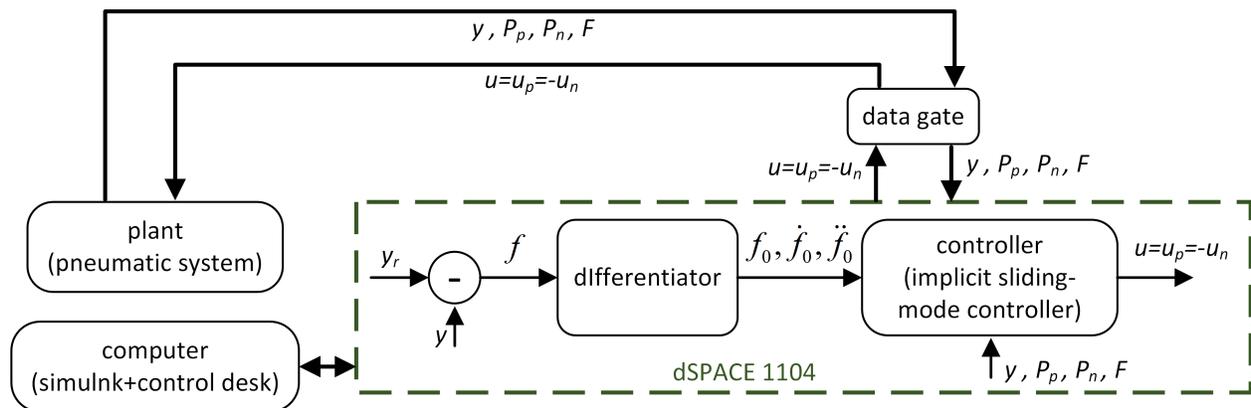


Figure 3: Closed-loop diagram of the EPS

The corresponding force of the disturbance actuator is controlled by a PID controller developed by the manufacturer, while the controller design for the main actuator will be considered in this section.

4.1 Model of the electropneumatic setup

We will use the following model for the EPS which has been explained in [37–39]:

$$\begin{cases} \dot{P}_p = \frac{Tkr}{V_P(y)} [\phi_P + \psi_p u_p - \frac{S}{T_r} P_p v] & (15a) \\ \dot{P}_n = \frac{Tkr}{V_N(y)} [\phi_N + \psi_n u_n + \frac{S}{T_r} P_n v] & (15b) \\ \dot{v} = \frac{1}{M} [S(P_p - P_n) - v b_v - F] & (15c) \\ \dot{y} = v, & (15d) \end{cases}$$

where P_p and P_n are the pressures in the P and N chambers respectively, y and v denote the position and the velocity of the jack. The force F is the disturbance that takes into account both the external disturbance and the friction, u_p and u_n are the control signals applied to the servo distributor corresponding to the chambers P and N , respectively, and it will be assumed that $u = u_p = -u_n$. Moreover, $k = 1.2$ is the polytropic constant, $T = 293.15\text{K}$ denotes the temperature, $r = 287.0365\text{J/kg/K}$ is the ideal gas constant, and $b_v = 50\text{N}\cdot\text{sec/m}$ is the viscous friction coefficient.

V_P and V_N are the volumes of the chambers P and N , respectively, which depend on the position of the jack y , $S = 0.0045\text{m}^2$ is the piston section, ϕ_x and ψ_x (x being N or P) are both fifth-order polynomials [40]¹. Eq. (15) can be rewritten as $\dot{x} = f(x) + g(x)u$ with the following uncertain vectors:

$$f(x) = \begin{bmatrix} \frac{Tkr}{V_P(y)} [\phi_P - \frac{S}{T_r} P_p v] \\ \frac{Tkr}{V_N(y)} [\phi_N + \frac{S}{T_r} P_n v] \\ \frac{S(P_p - P_n) - v b_v - F}{M} \\ v \end{bmatrix}, g(x) = \begin{bmatrix} \frac{Tkr}{V_P(y)} \psi_p \\ -\frac{Tkr}{V_N(y)} \psi_n \\ 0 \\ 0 \end{bmatrix}. \quad (16)$$

Because of the uncertainties, the vectors $f(x)$ and $g(x)$ are divided into nominal and uncertain parts as follows:

$$\dot{x} = (\bar{f} + \tilde{f})(x) + (\bar{g} + \tilde{g})(x)u, \quad (17)$$

where \bar{f} (resp. \bar{g}) and \tilde{f} (resp. \tilde{g}) denote the nominal and uncertain parts of the function f (resp. g).

4.2 Sliding-mode controller

As can be seen from (17), the EPS contains uncertain terms $\tilde{f}(\cdot)$ and $\tilde{g}(\cdot)$. To handle these uncertainties and disturbances, sliding-mode based controllers have been developed for the EPS [41, 42]. Following [41], since the control objective is the position tracking, the following sliding variable is defined.

$$\sigma(t) = e_2(t) + \lambda_1 e_1(t) + \lambda_0 e_0(t) \quad (18)$$

¹The simulation files, containing the exact values of the parameters, are available upon request to the first author: Mohammad Rasool Mojjallizadeh (email: mohammad-rasool.mojjallizadeh@inria.fr).

where $e_0(t) \triangleq y(t) - y_r(t)$, and y_r is the reference trajectory, $e_2(t) = \ddot{e}_0(t)$, and $e_1(t) = \dot{e}_0(t)$. The positive parameters λ_1 and λ_2 are designed such that the polynomial equation $z^2 + \lambda_1 z + \lambda_0 = 0$ is Hurwitz. This ensures the exponential convergence during the sliding-phase.

A typical SMC operates in two phases, namely, reaching-phase, and the sliding-phase. During the reaching phase, the control signal is designed such that $\sigma(t) \rightarrow 0$. At the end of the reaching-phase, $\sigma(t) = 0$ holds, and using the stability of the sliding-surface, $e(t) \rightarrow 0$ is achieved asymptotically. To ensure the persistency of the sliding-phase in the presence of uncertainties and disturbances, the control signal is designed such that $\dot{\sigma} = 0$ holds. Hence,

$$\begin{aligned} \dot{\sigma} = e^{(3)} + \lambda_1 \ddot{e} + \lambda_0 \dot{e} &= \frac{1}{M}[S(\dot{P}_p - \dot{P}_n) - b_v \dot{v} - \dot{F}] \\ &\quad - y_r^{(3)}(t) + \frac{\lambda_1}{M}[S(P_p - P_n) - b_v v - F] - \lambda_1 \ddot{y}_r(t) \\ &\quad + \lambda_0 (\dot{y} - \dot{y}_r(t)). \end{aligned} \quad (19)$$

From (19), and using (16), one can obtain two functions $\Psi(t)$ and $\Phi(t)$ such that

$$\dot{\sigma} = \Psi(t) + \Phi(t)u = \bar{\Psi}(t) + \tilde{\Psi}(t) + (\bar{\Phi}(t) + \tilde{\Phi}(t))u. \quad (20)$$

The functions $\bar{\Psi}$ and $\bar{\Phi}$ are not given here for the sake of space and can be found in [37]. Assuming that $\tilde{\Psi} = \tilde{\Phi} = 0$, the required control signal (\bar{u}), or so-called equivalent control, for keeping the sliding-phase ($\dot{\sigma} = 0$) can be obtained as follows:

$$\dot{\sigma} = 0 \quad \rightarrow \quad \bar{u} \in -\frac{\bar{\Psi}}{\bar{\Phi}}. \quad (21)$$

To ensure the presence of the reaching and the sliding phases for the perturbed system, the following discontinuous control signal is considered:

$$u(t) \in -\frac{1}{\Phi(t)}[\bar{\Psi}(t) - G \operatorname{sgn}(\sigma(t))] \quad (22)$$

Note that the inclusion \in is used instead of the equality, to indicate the use of set-valued signum function. Moreover, G is the gain of the control, and is selected based on the following inequality [41]:

$$G > \frac{\max \left| \tilde{\Psi} + \tilde{\Psi} \frac{\tilde{\Phi}}{\Phi} \right| + \eta}{\min \left(1 + \frac{\tilde{\Phi}}{\Phi} \right)} \quad (23)$$

with η being a positive constant.

To implement the controller (22) on a digital hardware, a discretization method should be utilized. This topic has been studied in [41], and it is concluded that implicit discretization can provide several advantages compared to the explicit counterpart, including digital chattering suppression, and insensitivity to the gains. The explicit discretization of (22) is avoided in this study since it leads to chattering even in the presence of an implicit differentiator [41]. On the other hand, the implicit discretization of the set-valued part of (22) gives:

$$u_k \in -\frac{1}{\Phi_k}[\bar{\Psi}_k - G \operatorname{sgn}(\sigma_{k+1})] \quad (24)$$

where u_k is applied on $[t_k, t_{k+1}]$.

To compute u_k and implicit equation has to be solved. To this end, we substitute (24) into (20), and assume that $\tilde{\Phi} = \tilde{\Psi} = 0$. Hence:

$$\sigma_{k+1} \in -G \operatorname{sgn}(\sigma_{k+1}) + \sigma_k. \quad (25)$$

Eq. (25) can be rewritten as follows:

$$\sigma_{k+1} - \sigma_k \in -G \operatorname{sgn}(\sigma_{k+1}), \quad (26)$$

The GE (26) can be solved based on the following conditions:

- **Case 1:** $\sigma_k > G$

In this case, the solution of the GE (26) satisfies $\sigma_{k+1} > 0$ which leads to $\operatorname{sgn}(\sigma_{k+1}) = 1 \rightarrow u_k \in -\frac{1}{\bar{\Phi}_k}[\bar{\Psi}_k - G]$. Hence, from (26) one has $\sigma_{k+1} = \sigma_k - G$.

- **Case 2:** $-G > \sigma_k > G$

For this case, the solution of the GE is $\sigma_{k+1} = 0$. Therefore, (26) gives

$$\begin{aligned} \sigma_k \in G \operatorname{sgn}(0) = G[-1, 1] & \quad \Leftrightarrow \\ \sigma_k = G\xi \quad \text{for some } \xi \in [-1, 1] & \quad \Rightarrow \\ \xi = \frac{\sigma_k}{G} \rightarrow u_k \in -\frac{1}{\bar{\Phi}_k}[\bar{\Psi}_k - \sigma_k] & \end{aligned} \quad (27)$$

- **Case 3:** $\sigma_k < -G$

In this case, the solution of the GE (26) satisfies $\sigma_{k+1} < 0$ which leads to $\operatorname{sgn}(\sigma_{k+1}) = -1 \rightarrow u_k \in -\frac{1}{\bar{\Phi}_k}[\bar{\Psi}_k + G]$. Hence, from (26) one has $\sigma_{k+1} = \sigma_k + G$.

The implicit discretization of the sliding-mode controller is depicted in Fig. 4. As can be seen, the controller needs the sliding variable $\sigma_k = e_{2,k} + \lambda_1 e_k + \lambda_0 e_{0,k}$ as well as vectors $\bar{\Phi}$ and $\bar{\Psi}$ at each time-step. To this end, it is necessary to build $x = [y, \dot{y}, \ddot{y}]^\top$. The EPS is equipped with a position sensor. However, the velocity (\dot{y}) and the acceleration (\ddot{y}), are not available, and need to be estimated. The differentiators which are analyzed in [3] are used to estimate the first and the second-order differentiations. The differentiators are implemented on a DS1104 board as follows:

- Some differentiators (STD, VGED, QD, HD, STDAC, FDFD) can only estimate the first-order derivative. Cascade configuration of these differentiators will be used to estimate both the velocity and the acceleration.
- Some differentiators can also estimate arbitrary-order derivatives (AO-STD, SHMD, HDD, GHDD, AO-FDFD). These differentiators can estimate any differentiations up to order n simultaneously, n being the order of the differentiation.

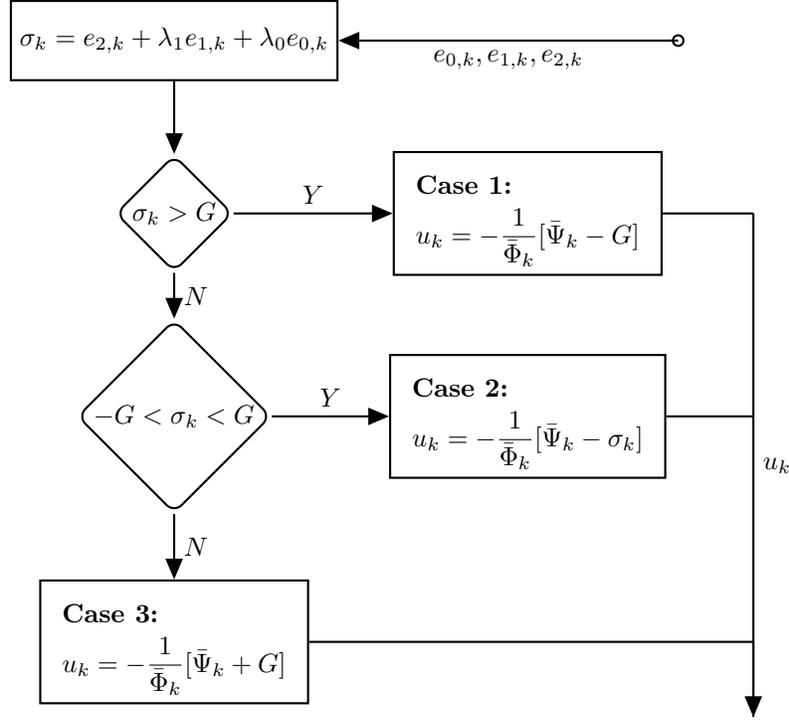


Figure 4: Flowchart of the first-order implicit SMC

- To implement the ALIEN differentiator, two ALIEN blocks are used to calculate both the velocity ($n = 1$) and the acceleration ($n = 2$) (see (9)) without using the cascade configuration.

4.3 Conditions of the experiments

The conditions of the experiments are listed below:

- The first-order SMC is implemented implicitly according to Fig. 4.
- The sampling rates correspond to all subsystems (controller, differentiator, sensors, etc) are always the same.
- The pneumatic system has some initial conditions (initial position of the horizontal jack, the initial value of the disturbance, ...) that take different values in each experiment. Hence, taking into account the transients will lead to unfair results since each differentiator faces different initial conditions. To solve this problem, we recorded the waveforms and calculated the performances after a specific amount of time to ensure zero initial conditions for all differentiators. Therefore, the reaching phase is neglected for all practicable systems.
- The disturbance actuator generates 400N square waveform force with 0.01Hz of frequency to emulate the disturbance.

- The control objective is to track the reference trajectory $y_r(t) = 40 \sin(0.2\pi t)$ in millimeters.
- I-AO-STD, and I-HDD are implemented for $n = 2$, while for the I-GHDD, $n = 3$ is considered.
- Since tuning the parameters of differentiators in practical closed-loop systems is difficult and time consuming, the parameters are tuned using open-loop simulations as suggested in [2, 3]. In this case, it is assumed that the input of the differentiators is $f(t) = \sin(t) + \tilde{n}(t)$ with $\tilde{n}(t)$ being a white noise with SNR=30dB. Subsequently, a randomized algorithm [2, 3] is used to find the optimal parameters for minimizing the estimation error corresponding to the first-order differentiation. The obtained parameters are listed in Table 3. This procedure helps to tune all differentiators in a unified manner which leads to fair comparisons.

The differentiation toolbox (see Appendix E in [3])¹ has been used to tune the parameters. The parameters of all the differentiators are tuned for a typical sinusoidal input $f(t) = \sin(t) + \tilde{n}(t)$ where $\sin(t)$ is the based signal and $\tilde{n}(t)$ is a white noise with SNR=30dB. The toolbox uses a randomized optimization algorithm [43] to find the optimal parameters in the predefined interval which has been obtained in [2, 3]. In order to improve the optimality of the parameters, the procedure has been conducted for 10^5 iterations which took around two hours on Intel Core i3-4030 CPU with two cores working at 1.9 GHz. Other optimization methods may also be utilized, but it goes out of the scope of this work.

To evaluate the performances, the following objective functions have been defined.

- **Average magnitude of the output error:** This cost function indicates the average energy of the output error. Smaller amount of this value indicates better output tracking. This cost function is defined as follows:

$$\bar{L}_2(e_k) = \frac{h}{t_f} \|e_k\| = \frac{h}{t_f} \sqrt{\sum_{k=0}^{t_f/h} e_k^2}, \quad (28)$$

where t_f is the final time, $e_k = y_k - y_{d,k}$ denotes the output error, y_k , and $y_{d,k}$ are the output and its corresponding reference, respectively, at time-step k (see Fig. 1).

- **L_∞ norm of the differentiation error:** This cost function indicates the maximum deviation of the output from its reference. Hence, it can be used to calculate the overshoots and the accuracy of the control system. This cost can be calculated as follows:

$$L_\infty(e_k) = \|e_k\|_\infty = \max_k |e_k|, \quad k = 0, \dots, t_f/h. \quad (29)$$

- **Total variation of the control signal:** This criterion can be used to qualify the chattering effect on the control signal. It is calculated as follows:

$$\text{VAR}(u_k) = \sum_{k=0}^{t_f/h} |u_k - u_{k-1}|. \quad (30)$$

¹A toolbox is developed in MATLAB during this research to conduct all the necessary numerical simulations. The toolbox is available upon request to the first author: Mohammad Rasool Mojjallizadeh (email: mohammad-rasool.mojjallizadeh@inria.fr)

Table 3: Parameters of the differentiators obtained from the tuning procedure

Method	Parameters
Euler	No parameter
LF	$c=7.1113$
E-STD	$L=0.7713$
I-STD	$L=0.7324$
SI-STD	$L=0.6985$
E-URED	$L=0.0770, \mu=20.8386$
I-URED	$L=0.1021, \mu=21.2075$
E-QD	$F=4.4026, \alpha=0.3780$
I-QD	$F=4.5323, \alpha=0.8123$
ALIEN	$\bar{T}=0.5020, \kappa=1, \mu=2$
HD	$r=2.5655$
E-AO-STD	$L=4.8973$
I-AO-STD	$L=2.9122$
SI-AO-STD	$L=2.8157$
E-HDD	$L=4.9392$
E-GHDD	$L=4.8970$
I-HDD	$L=2.9921$
I-GHDD	$L=2.9822$
VGED	$\mu=4.3694, \tau=1.3269, \omega_c=12.2205$ $q=0.2997$
SI-URED	$L=0.1434, \mu=93.5748$
E-STDAC	$\alpha=0.5318, \epsilon=0.0000$
I-FDFF	$\omega_s=19.6607, \omega_f=8.4727, \rho=8.6929$ $\gamma=0.0348$
I-AO-FDFF	$F=37.7845, \epsilon=18.6061, \omega_s=2.5068$ $\omega_f=62.6396, \alpha_1=456.7015, \rho=88.3003$
Kalman	$R = 8.4121 \times 10^{-4}$
Input signal: $\sin(t)$,	
Noise type: white, SNR=30dB, $h = 50\text{ms}$	

4.4 Identifying the impracticable experiments

Before providing the experimental results, it is crucial to identify the impracticable experiments. An impracticable experiment refers to a condition where it is necessary to stop the experiment to avoid any damage to the hardware. For the pneumatic setup, the horizontal jack can only move within $y \in [-70+70]$ mm. Beyond that, the jack will hit the barriers, which we call the corresponding experiment "impracticable" (note that the reference trajectory is $y_r(t) = 40 \sin(0.2\pi t)$ mm). Many factors, including instability of the closed-loop control system and overshoots, can potentially lead to an impracticable experiment.

The impracticable experiments with the corresponding conditions are listed in table 4. As it was reported before (see Table 4 in [3]), the E-QD shows too much chattering for large sampling time, which led to impracticability for this closed-loop experiment. The I-URED also leads to impracticability for small sampling times. Another observation is that the ALIEN leads to impracticability when decreasing the sampling time. Note that the parameters of the ALIEN (including the estimation window \bar{T}) are designed for $h = 50$ ms. Hence, by decreasing the sampling time, this differentiator led to impracticability which indicates its sensitivity to the parameters. The performances for the impracticable cases are not provided in the next sections.

Table 4: Impracticable experiments

Method	Impracticability condition
E-QD	$h > 5$ ms
I-URED	$h < 15$ ms
ALIEN	$h < 5$ ms

4.5 Experiments under different sampling times

The aim of this section is to study the behavior of the differentiators under different sampling times $h \in [1, 5, 10, 15, 20, 30, 40, 50]$ ms. The variation of the control signal $u(t)$ (see Figs. 2 and 4) under different sampling times is presented in Fig. 5. Comparing the explicit and the implicit methods in Fig. 5 (A) and (B), one can see that for large sampling times $h > 20$ ms, the implicit methods show smaller variations which indicate smaller chattering on the control signal. For smaller sampling times ($h < 20$ ms) there is not a significant difference between the explicit and the implicit methods. Hence, it can be concluded that even with the implicit SMC, the explicit differentiators may lead to chattering on the control signal for large sampling times. In other words, to suppress the numerical chattering at all sampling periods, it is necessary to implement both the controller and the differentiator implicitly.

From Fig. 5 (D), while ALIEN shows small variations (comparable to the implicit methods) for large sampling times, the values for small sampling times are not provided, since it is not practicable for $h < 5$ ms. Considering the variations of the cascade configurations (STD, STDAC, URED) in Fig. 5, one can see that

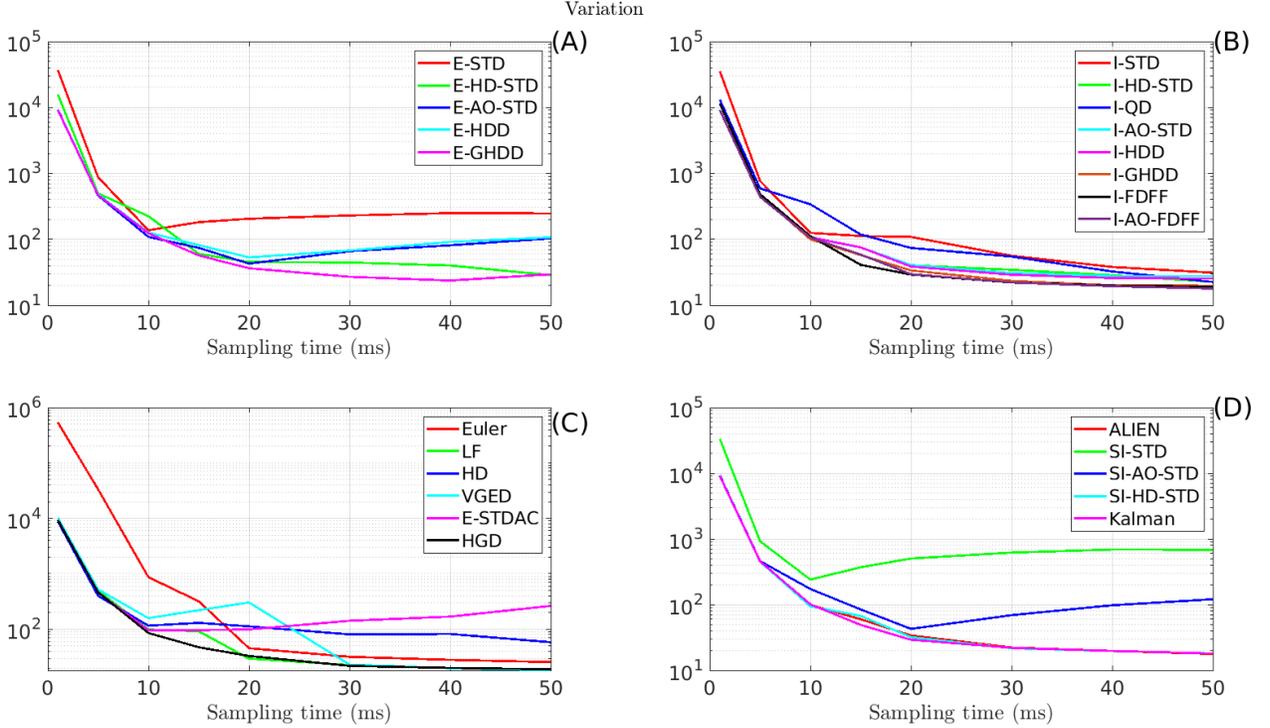


Figure 5: Variation of the control signal $u(t)$ under different sampling times

the cascade combinations of the explicit or semi-implicit differentiators present more chattering than the implicit counterparts, and should be avoided, since each stage amplifies the chattering of the previous one. As the result, for large enough sampling times $h > 10\text{ms}$, E-STD and SI-STD show the worst variations (Fig. 5 (A) and (D)).

For small sampling times, Euler is the worst differentiator, since according to the Nyquist theorem, by decreasing the sampling time, larger frequency components of the measurement noise can also be recovered (note that Euler is a pure differentiator with unlimited bandwidth). From Fig. 5 (C), while the VGED shows small variations for large sampling times $h > 40\text{ms}$, it shows relatively large variations for smaller sampling times. The reason is that six parameters of this differentiators are tuned for $h = 50\text{ms}$. Decreasing the sampling time, deteriorates its variation, which indicates its sensitivity to the parameters. Another interesting observation is that Kalman method shows one of the best variations, comparable with the implicit methods. However, I-FDFF shows a smaller variation for $h = 1\text{ms}$, compared with that of the Kalman method.

Also, note that, in general, the implicit methods show a smaller variation compared to that of the LF, which validates **Fact 6** (Section 1).

The variations of the control signal u corresponding to the SMB differentiators for large sampling time $h = 50\text{ms}$, $h = 40\text{ms}$, and $h = 30\text{ms}$ are provided in Fig. 6. Comparing the results, one can conclude that, in general, the implicit methods show smaller variations than those of their explicit counterparts.

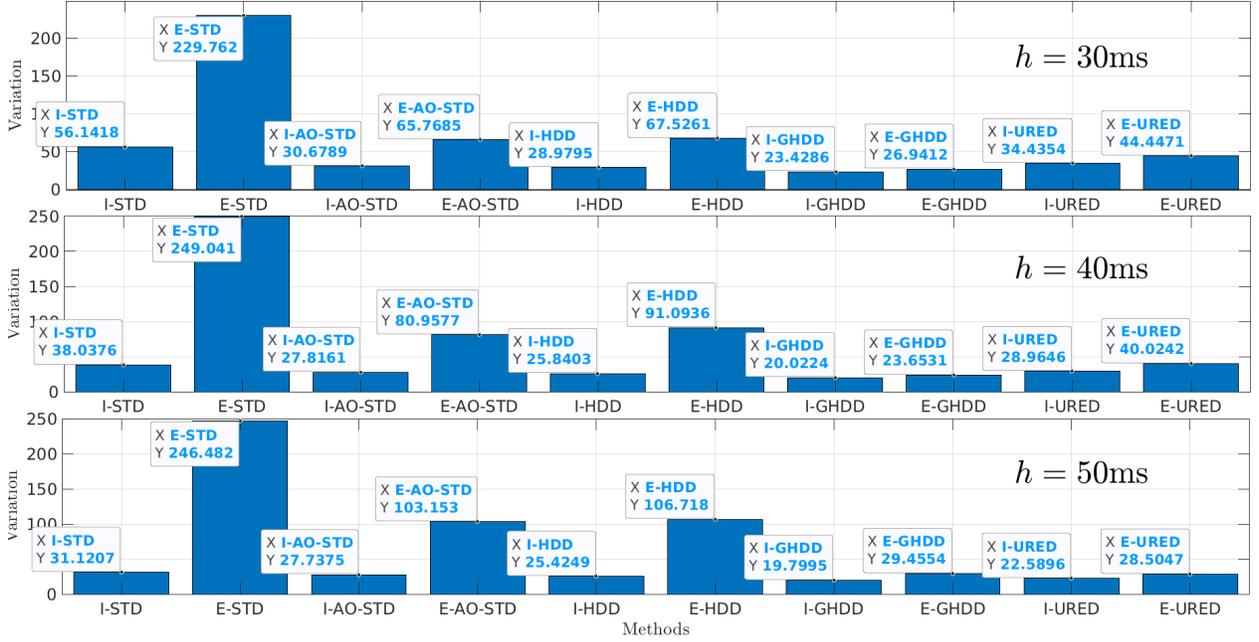


Figure 6: Comparison of the implicit and explicit schemes

The \bar{L}_2 norm of the methods for different sampling times are provided in Fig. 7. It can be seen that, for all methods, increasing the sampling time highly affects the \bar{L}_2 norm. Concerning Fig. 7, all methods show almost the same evolution, except for the Euler method which shows the worst \bar{L}_2 for $h < 5$ ms. E-STDAC also shows a large \bar{L}_2 for $h < 5$ ms, probably because it is too sensitive to the parameters.

Another observation is that while there is not a noticeable difference between the variation of the implicit differentiators and LF (compare Fig. 5 (B), (C)), according to Fig. 7 (B), the implicit methods show smaller \bar{L}_2 which indicates better output tracking. Hence, it can be concluded that the implicit methods behave better than the LF (Fact 6).

The L_∞ norms are presented in Fig. 8. Apart from the Euler and the E-STDAC, all differentiators almost show the same behavior when increasing the sampling time.

4.6 Gain sensitivity

The SMB differentiators are studied under over-sized gains and the results are provided in table 5. It can be seen that for $L = 20$ (compare with L in Table 3), all SMB differentiators can be implemented. However, comparing the explicit and the implicit methods, it can be seen that the variations related to the explicit methods are much more affected than the implicit counterparts. Another interesting observation is that while the implicit methods remain practicable for $L = 10^3$, the explicit ones lead to impracticability. Hence the implicit methods show much better gain-insensitivity than the explicit counterparts as was stated in **Fact 3**. Another observation is that the I-AO-STD remains practicable even for $L = 10^{100}$: it is completely insensitive to increase in L .

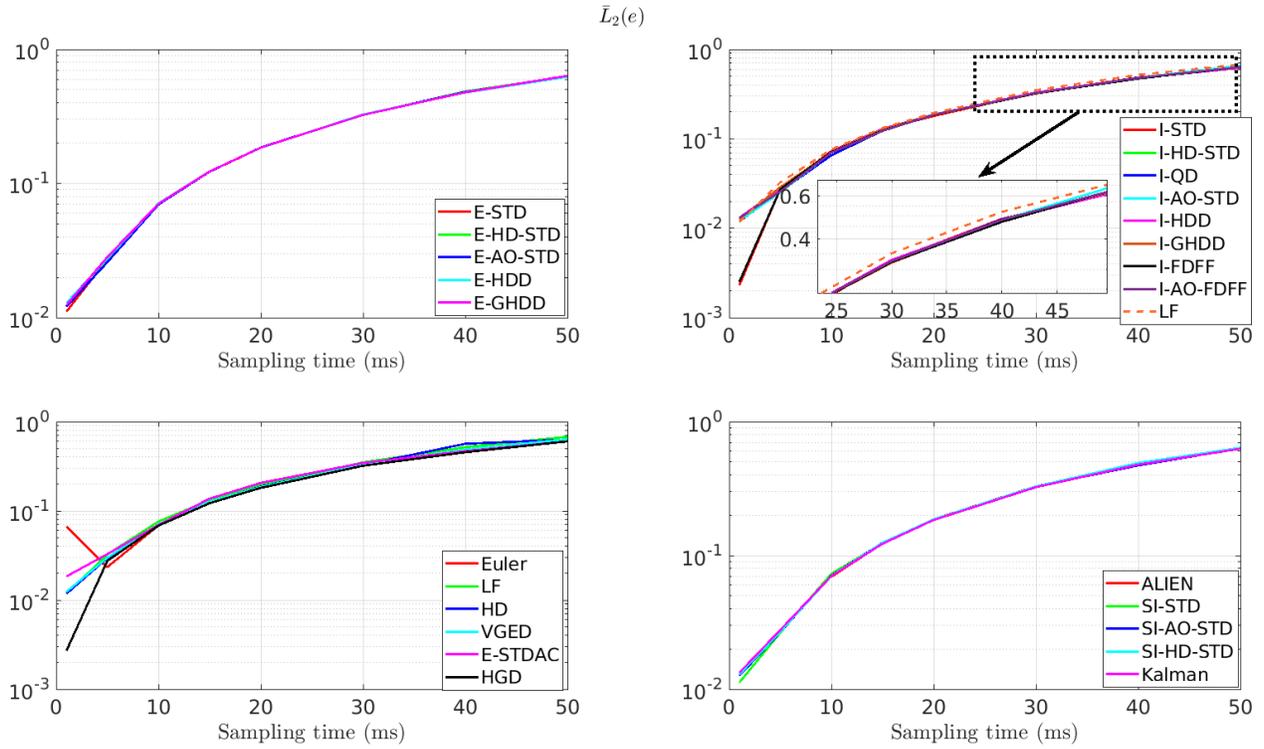


Figure 7: \bar{L}_2 for all methods

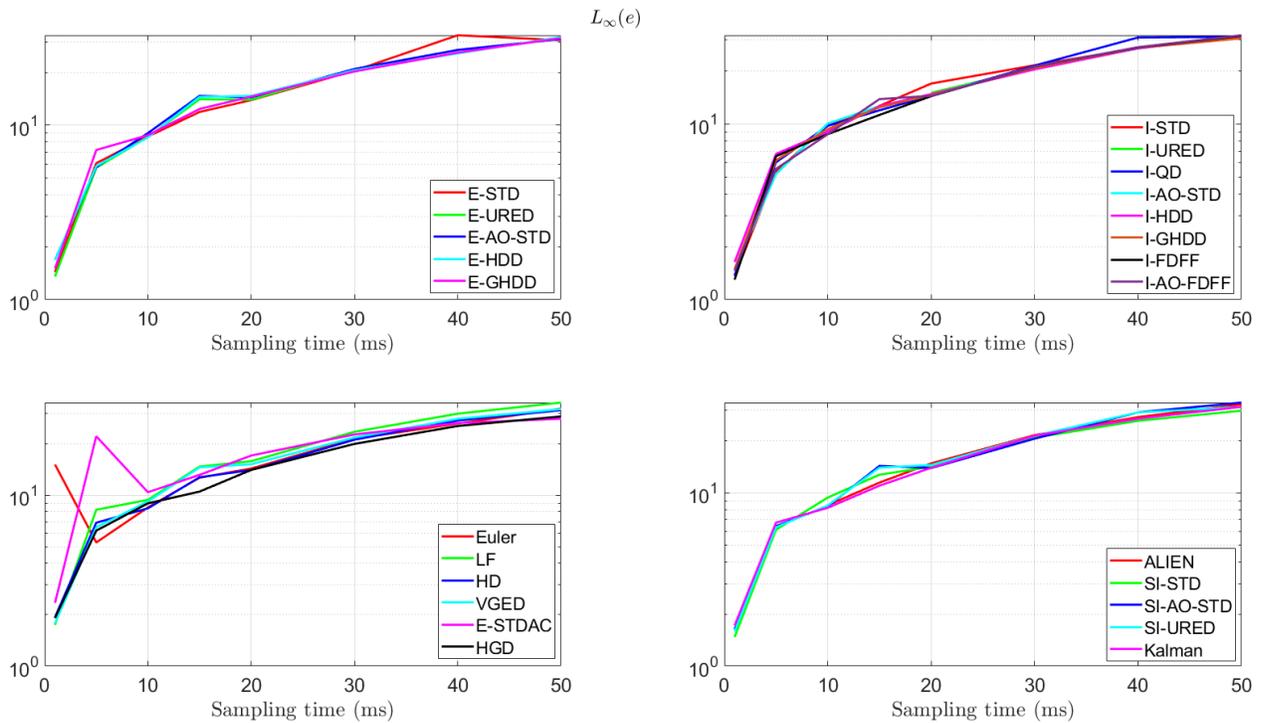


Figure 8: L_∞ for all methods

4.7 Solver's effect

As it was previously explained in [2, 3], some implicit differentiators I-AO-STD, I-HDD, and I-GHDD need an iterative solver to solve a polynomial equation at each time-step. Following [3], the required calculation resources should be small enough to ensure the real-time operation of the system. A set of experiments have been conducted to address this issue and obtain the maximum tolerable iterations for the real-time implementation. According to the open-loop simulations provided in [3], the minimum required accuracy for the solver without affecting the performances is 10^{-5} . Hence, the Newton's solver with the same accuracy is utilized, and the maximum number of iterations for the real-time operation is obtained and provided in table 6.

As can be seen from table 6, for $h < 1\text{ms}$, it is not possible to implement the solver-based implicit methods, i.e., I-AO-STD, I-HDD, I-GHDD, and I-URED. Our investigation shows that for such small sampling time, even non-solver-based methods, e.g., HD and ALIEN cannot be implemented in real-time operation. According to table 6, for $h > 1\text{ms}$, all the implicit methods can be implemented with the indicated number of iterations. We noticed that for a larger number of iterations, the implicit methods cannot be implemented in real-time.

It is reported before (see [3] Sec. 5.8) that the Newton's solver can provide the accuracy 10^{-5} with the maximum number of 7 iterations. According to table 6, since the number of iterations for $h > 1\text{ms}$ is always higher than 7, one can conclude that the implicit methods can operate in real-time for $h > 1\text{ms}$. Normally, this should not be an obstacle to the implementations, since implicit SMC tolerates larger sampling periods without significant deterioration of the performance [41, 42].

4.8 Results obtained for the electropneumatic setup: conclusions

The following results have been drawn based on the practical experiments on the EPS:

- Comparing the explicit, semi-implicit, and implicit discretization schemes in Figs. 5 and 6, one can conclude that the implicit differentiators presented smaller variations than those of other discretization schemes. Hence, the implicit methods show a smaller chattering which validates **Fact 1**.
- According to Section 4.7, the solver-based implicit methods (I-AO-STD, I-HDD, I-GHDD, and I-URED) can be implemented in real-time with the desired solver's accuracy on the EPS, which validates **Fact 2**.
- Based on the discussion presented in Section 4.6, the implicit methods show better in sensitivity to the gains which validates **Fact 3**.
- According to the results provided in Figs. 5 and 6, higher-order SMB differentiators, e.g., AO-STD show better variation than the cascade configurations of the first-order counterparts, e.g., STD. Apart from

Table 5: Results for $h=20\text{ms}$ under oversized gain

Method	L	$\bar{L}_2(e)$	L_∞	var
E-AO-STD	20	0.1990	27.8942	485.8769
E-HDD	20	0.2393	35.4567	455.1400
E-GHDD	20	0.3210	83.2670	58.6909
I-AO-STD	20	0.1952	15.8937	53.0042
I-HDD	20	0.3276	83.5963	126.9906
I-GHDD	20	0.2427	31.2091	37.7088
I-AO-STD	10^3	0.2277	44.2331	230.9670
I-HDD	10^3	0.1916	16.4728	710.1239
I-GHDD	10^3	0.1975	16.4151	188.9101
I-AO-STD	10^6	0.1942	15.1147	75.6753
I-AO-STD	10^7	0.1972	15.4931	71.3844
I-AO-STD	10^8	0.1962	15.1638	68.1815
I-AO-STD	10^9	0.1959	15.3237	69.8416
I-AO-STD	10^{10}	0.1958	15.0134	62.4789
I-AO-STD	10^{50}	0.1949	14.9311	48.5216
I-AO-STD	10^{100}	0.1947	14.9944	40.6776
I-HDD	10^6	impracticability		
I-GHDD	10^6	impracticability		
E-AO-STD	10^3	impracticability		
E-HDD	10^3	impracticability		
E-GHDD	10^3	impracticability		
$h=20\text{ms}$, with disturbance				

the numerical chattering, higher variation may also indicate higher noise sensitivity. Hence, higher-order differentiators may potentially present more robust behavior to noise which validates **Fact 5**.

- According to Fig. 5, in general, the implicit methods may provide better results than those of the LF, which validates **Fact 6**.

5 Rotary inverted pendulum

A RIPS from QUANSER (QUBE-Servo 2), shown in Fig. 11, is considered in this section as another case-study.

Table 6: Maximum number of iteration for real-time operation

Method	$h = 0.2\text{ms}$	$h = 1\text{ms}$	$h = 2\text{ms}$	$h = 3\text{ms}$	$h = 4\text{ms}$	$h = 5\text{ms}$
I-AO-STD	0	57	132	207	283	357
I-HDD	0	99	225	351	481	604
I-GHDD	0	57	132	207	283	357
I-URED*	0	28	65	101	138	175
Newton's solver, accuracy= 10^{-5} , no extra noise						
*Two blocks operating in cascade configuration						

5.1 Mathematical modeling of the rotary inverted pendulum

The scheme of the RIPS is shown in Fig. 9. As it can be seen, the setup has two links, namely, the rotary link and the pendulum link, which are indicated by the subscripts r and p , respectively, $m_r = 0.095\text{kg}$ and $m_p = 0.024\text{kg}$ are the weights of the rotary and pendulum links, $L_r = 0.085\text{m}$ and $L_p = 0.129\text{m}$ are the lengths of the rotary and the pendulum links, respectively, $J_r = m_r L_r^2/12$ and $J_p = m_p L_p^2/12$ denote the moment of inertia of the rotary and the pendulum links, respectively, and θ and α are the angles of the rotary and the pendulum links, respectively. The angle $\alpha = 0$ indicates that the pendulum is hanging in its downward position.

Equations of motion of the pendulum can be obtained using the Euler-Lagrange framework as explained in [44]. Following these references, the state-space equations of the linearized system (about its unstable equilibrium point, i.e., $\theta = \dot{\theta} = \dot{\alpha} = 0, \alpha = \pi$ and without considering the actuator dynamic) can be obtained as follows:

$$\dot{x} = Ax + Bu \quad (31a)$$

$$y = Cx + Du \quad (31b)$$

where $x = [\theta, \alpha, \dot{\theta}, \dot{\alpha}]^\top$ is the state vector, $u = \frac{k_m(V_m - k_m \dot{\theta})}{R_m}$ is the input (the torque), and $y = [\theta, \alpha]^\top$ is the output. Moreover, the notations are given by:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & m_p^2(L_p/2)^2 L_r g / J_T & -D_r(J_p + m_p(L_p/2)^2) / J_T & -m_p(L_p/2)L_r D_p / J_T \\ 0 & m_p g(L_p/2)(J_r + m_p L_r^2) / J_T & -m_p(L_p/2)L_r D_r / J_T & -D_p(J_r + M_p L_r^2) / J_T \end{bmatrix} \quad (32)$$

$$B = \begin{bmatrix} 0 & 0 & (J_p + m_p(L_p/2)^2) / J_T & m_p(L_p/2)L_r / J_T \end{bmatrix}^\top$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

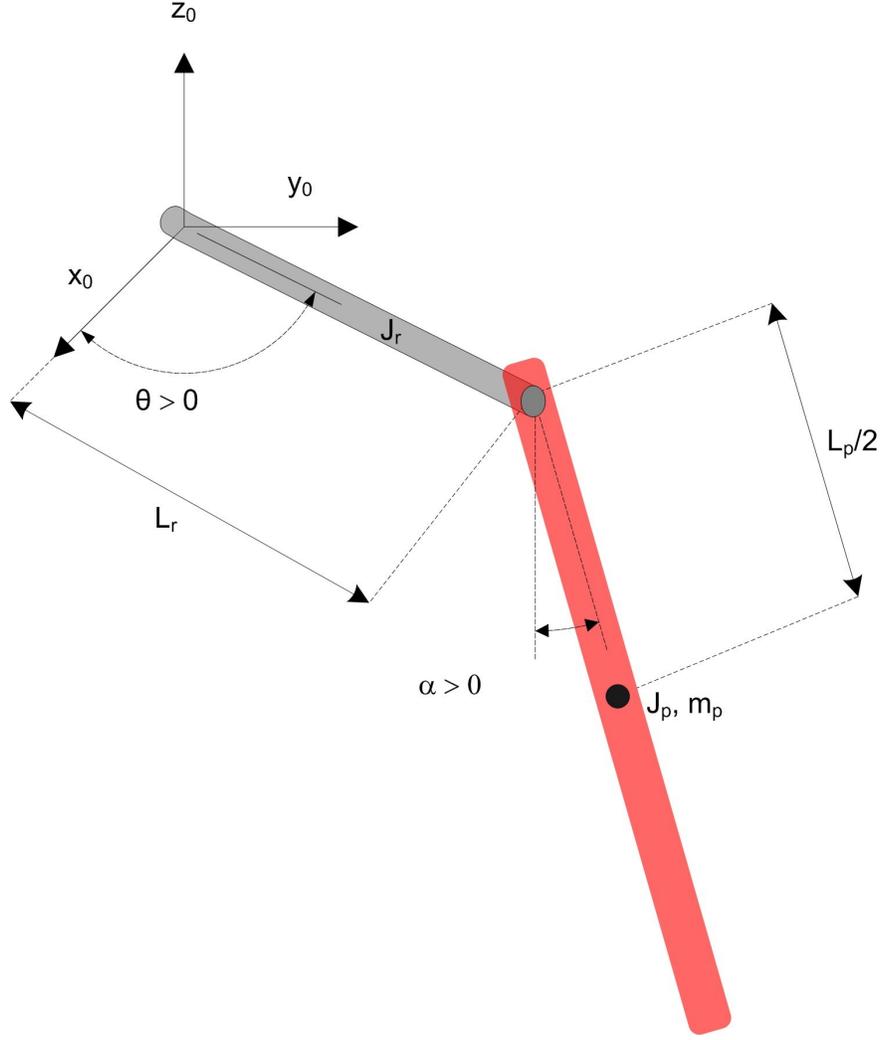


Figure 9: Scheme of the rotary inverted pendulum [1]

To take the dynamic of the actuator (DC motor) into account, (32) is modified as follows:

$$\begin{aligned} B &= \frac{k_t}{R_m} B, & A(3,3) &= A(3,3) - \frac{k_t^2}{R_m B(3)}, \\ A(4,3) &= A(4,3) - \frac{k_t^2}{R_m B(4)}, \end{aligned} \quad (33)$$

where $k_t = 0.042 \text{ N}\cdot\text{m}/\text{A}$ is the current-torque coefficient, $R_m = 8.4\Omega$ is the resistance of the motor armature circuit, $g = 9.81(\text{m}/\text{s}^2)$ is the acceleration of gravity, $D_r = 0\text{Nm}/(\text{rad}/\text{s})$, $D_p = 0\text{Nm}/(\text{rad}/\text{s})$ are the equivalent viscous damping coefficients, $k_m = 0.042\text{V}/(\text{rad}/\text{s})$ is the back-emf constant, and V_m is the control voltage applied to the motor.

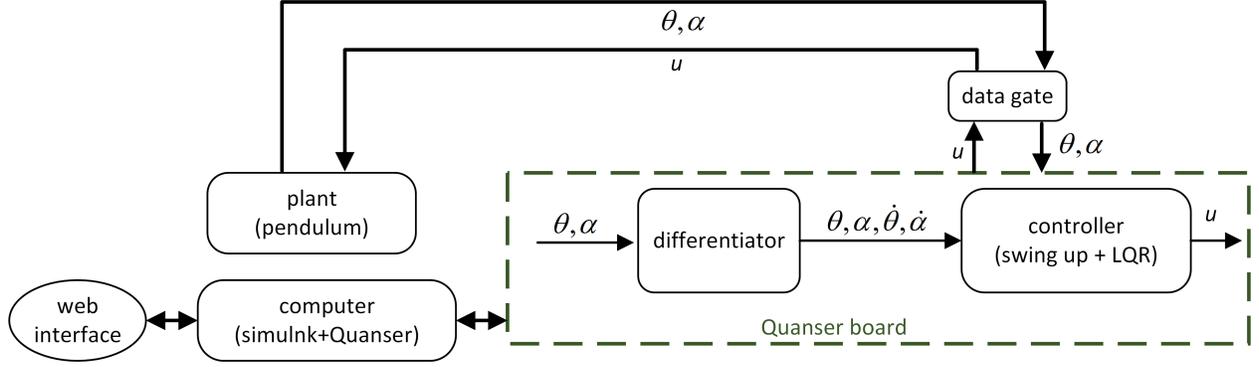


Figure 10: Closed-loop diagram of the RIPS

5.2 Controller design for the rotary inverted pendulum

A switching control scheme is used to regulate the pendulum on its unstable equilibrium point as follows:

$$u = \begin{cases} u_b & \text{if } |\alpha| - \pi \leq 20^\circ \\ u_s & \text{otherwise} \end{cases} \quad (34)$$

where u is the control signal, u_b is the balancing control and u_s is the swing-up control. It can be seen from (34) that the balancing control is only active within ± 20 degrees of the unstable equilibrium point. Closed-loop system of the RIPS is provided in Fig. 10, and the controllers u_s and u_b are obtained in sections 5.2.1 and 5.2.2, respectively.

5.2.1 Swing-up control

The following control law is designed to swing the pendulum around its upward position as follows:

$$u_s = \text{sat} \left(\mu(E_r - E) \text{sgn}(\dot{\alpha} \cos(\alpha)), u_{sm} \right) \quad (35)$$

where μ is the control gain, E is the energy of the system, $E_r = 40\text{mJ}$ is the potential energy of the pendulum on its upward position, and u_{sm} is the saturation factor. The term $\text{sgn}(\dot{\alpha} \cos(\alpha))$ is multiplied to switch the sign of the control. Here, the sat function is defined as follows:

$$\text{sat}(x, u_{sm}) = \begin{cases} u_{sm} & \text{if } x > u_{sm} \\ x & \text{if } |x| \leq u_{sm} \\ -u_{sm} & \text{if } x < -u_{sm}. \end{cases} \quad (36)$$

The energy of the pendulum is the summation of the kinetic (E_1) and potential (E_2) energies as follows:



Figure 11: Photograph of the rotary inverted pendulum [1]

$$E = E_1 + E_2 \quad (37a)$$

$$E_1 = \frac{1}{2} J_p \dot{\alpha}^2 \quad (37b)$$

$$E_2 = \frac{1}{2} M_p g L_p (1 - \cos(\alpha)) \quad (37c)$$

5.2.2 Balancing control

An LQR¹ controller ($u = G(x - x_r)$) is designed for (32) for the balancing as follows:

$$G = \begin{bmatrix} 2 & -35 & 1.5 & -3 \end{bmatrix}. \quad (38)$$

Note that the reference trajectory throughout the experiment is $x_r = \begin{bmatrix} 0.3 \sin(t) & 0 & 0.3 \cos(t) & 0 \end{bmatrix}^T$.

5.3 Differentiator design for the rotary inverted pendulum

Since the variables $\dot{\theta}$ and $\dot{\alpha}$ are not available, the differentiators which are introduced in the first part of the report, are employed to estimate $\dot{\theta}$ and $\dot{\alpha}$ by differentiating θ and α .

¹Linear quadratic regulator is a well-defined controller (for instance see [45]) to minimize the objective function $J = \int (x(t)^T Q x(t) + u^T(t) R u(t)) dt$, where Q and R are weighting matrices. This controller can be designed using the MATLAB function `G=lqr(sys, Q, R)`, where `sys` is the state-space representation of the plant and `G` is the calculated control gain.

5.4 Experiments

The remote setup which is used in this section is installed at INRIA Lille, France². To perform the experiments it is necessary to implement the controller and the observer in a compatible format and then upload them on the remote setup. A numerical simulation is conducted after uploading the codes to protect the experimental system by validating the codes. Afterward, the customized controller/observer is implemented on the real system, and both numerical and experimental results are available online within a few minutes.

5.4.1 Conditions of the experiments

Tuning the parameters of the differentiators is one of the most challenging parts of this study (including previous works [2, 3]). For the EPS, the parameters obtained in the first part of the research [3] are directly used to implement the differentiators. Those parameters are assumed to be optimal for the input $f(t) = \sin(t) + \tilde{n}(t)$ with $\tilde{n}(t)$ being a white noise with SNR=30dB. As it was seen in Section 4, almost all differentiators lead to practicability with the obtained parameters on the EPS, which demonstrate the validity of the approach. However, with these parameters, all differentiators (except for the Euler which does not depend on any parameter) lead to impracticability on the RIPS. Hence another input, i.e., $f(t) = 4 \times 10^4(t^2 + 2t + 3)$ with quantization accuracy 0.001, is used to re-tune the parameters specifically for the RIPS (the parameters are listed in Table 7). This input allows an optimization algorithm to select the parameters such that the bandwidths of the differentiators are large enough to track the high-frequency components of the plant's outputs. Unlike the EPS, such a high bandwidth is not problematic for the RIPS since its output is less affected by the measurement noise. In fact, the sensors used in the RIPS (shaft encoders) are less affected by noise, since the measurements are inherently digital without requiring any analog to digital conversion.

Another reason for selecting such large bandwidth is that during each simulation/experiment, if the tracking error for $t > 3s$ is higher than $\pi/4$, the experiment/simulation will be immediately stopped for the RIPS to protect it against unexpected behaviors. Hence, the transient response of the closed-loop system must be short enough to converge the tracking error in the interval $\alpha \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ for $t > 3s$. Otherwise, a differentiator leads to impracticability. This issue is further addressed in Section 5.5.

In summary, while EPS is a relatively slow system with significant noise, RIPS is a fast system with small noise. Their behaviors are then quite different, and this has a great influence on differentiators' gains tuning.

It was noticed that unlike the numerical simulations, repeating an experiment leads to different results since each experiment depends on some stochastic variables. For instance, a cable that connects the pendulum's encoder with the QUBE is a sort of spring, which may have different positions after each experiment. This spring may differently impact the pendulum in the upper positions as well. This modifies the system

²One may validate all the numerical and experimental results using the remote setup which is accessible via <http://valse-pendulum.lille.inria.fr:5000/>

Table 7: Parameters of the differentiators obtained from the tuning procedure

Method	Parameters
Euler	No parameter
LF	$c=184.1331$
E-STD	$L=80948.6599$
I-STD	$L=80841.0774$
SI-STD	$L=80858.8240$
E-URED	$L=2.7010, \mu=0.0020$
I-URED	$L=113.8287, \mu=0.0012$
E-QD	$F=46987.4413, \alpha=3179.5530$
I-QD	$F=1313071.0435, \alpha=16568.1128$
ALIEN	$\bar{T}=0.0802, \kappa=3, \mu=10$
HD	$r=3410.2030$
E-AO-STD	$L=15662.1872$
I-AO-STD	$L=15412.9100$
SI-AO-STD	$L=15815.3066$
E-HDD	$L=15916.9396$
E-GHDD	$L=15898.9603$
I-HDD	$L=16115.6386$
I-GHDD	$L=16368.7089$
VGED	$\mu=3.8759, \tau=32.2378, \omega_c=148.1756$ $q=8.5330$
SI-URED	$L=134.1932, \mu=0.0296$
E-STDAC	$\alpha=858.8981, \epsilon=0.6591$
I-FDFF	$\omega_s=90.4186, \omega_f=3.6192, \rho=473.7341$ $\gamma=1041.8622$
I-AO-FDFF	$F=17171508, \epsilon=35712, \omega_s=22408$ $\omega_f=8633, a_1=15621, \rho=40688$
Kalman	$R=31.8014$
HGD	$L=0.7439$
Input signal: $f(t) = 4 \times 10^4(t^2 + 2t + 3)$	

trajectories and stabilization precision. In addition, there exists a random noise in sensors (encoders), etc. On top of all this, the controller is hybrid (discontinuous), since it switches from swing-up to balance feedback law. In this case, a small perturbation of parameters may imply a large variation in trajectories. To tackle this problem, we have conducted each experiment ten times, and the average results are provided in Table 8.

The explicit SMB differentiators, i.e., E-STD, SI-STD, E-URED, E-AO-STD, E-QD, E-HDD, E-GHDD lead to impracticability. The reason is that for these large gains, explicit methods show a significant amount of chattering. So it is very difficult to find a good compromise between chattering reduction and speed increase for such algorithms. This is also the case for the SI-STD and SI-URED. Note that the SI-AO-STD formed a practicable system mainly because of two reasons. As it was reported in [3], while semi-implicit methods are still affected by numerical chattering, they can attenuate it significantly. The main difference between the SI-AO-STD and other semi-implicit methods, i.e., SI-STD and SI-URED, is that the SI-AO-STD is an arbitrary-order differentiator ($n = 2 > 1$) which also attenuates the chattering by extra integrations. As the result, the SI-AO-STD formed a practicable control system.

Another interesting observation is that while other implicit methods are practicable, the I-GHDD leads to impracticability in this case. The reason is that (unlike other arbitrary-order differentiators that are implemented with $n = 2$) the I-GHDD is a third-order arbitrary-order differentiator ($n = 3$). As it was concluded in [3], increasing n increases the transient time. Hence, for this case and due to the reason explained above, the I-GHDD led to impracticability as was mentioned in **Fact 4**.

5.5 Effect of the parameter tuning on the results

As it was mentioned before, the parameters are tuned based on the input $f(t) = 4 \times 10^4(t^2 + 2t + 3)$ to ensure the practicability of the methods. Here, the differentiators are tuned according to another input to see how the parameters affect the results. In this case, the input $f(t) = 100 \sin(200t)$ with quantization (resolution=0.001) is considered for parameter tuning. The tuned parameters are provided in table 9, and the corresponding results are summarized in Table 10.

The most important observation is that the arbitrary-order differentiators, i.e., I-AO-STD, I-HDD, and SI-AO-STD, which were practicable before (see Table 8), are not practicable anymore. The reason is that according to **Fact 4**, an arbitrary-order differentiator shows a longer transient response. Since smaller gains L are selected for this case, these differentiators show too long transient, which led to impracticability.

Another observation is that ALIEN and I-FDFF also led to impracticability which indicate their sensitivity to parameter tuning. Also, in this experiment, the LF shows one of the worst performances, since a smaller c is selected for this differentiator compared to the first case (compare Tables 7 and 9). This validates **Fact 6**.

In summary, in spite of the fact that we do not propose a systematic choice for the signal $f(t)$ to tune the parameters along the lines of [2, 3], we show that such a choice has to be done carefully depending on

Table 8: Average results based on 10 experiments (parameters are shown in Table 7)

Method	Simulations					Experiments				
	$\ e_\theta\ $	$\ e_\alpha\ $	$\ e_\theta\ _\infty$	$\ e_\alpha\ _\infty$	$\ u\ _\infty$	$\ e_\theta\ $	$\ e_\alpha\ $	$\ e_\theta\ _\infty$	$\ e_\alpha\ _\infty$	$\ u\ _\infty$
Euler	0.1761	0.0097	0.2975	0.0082	0.0584	0.2725	0.0325	0.1161	0.0221	13.4087
LF	0.1839	0.0098	0.3167	0.0085	0.0583	0.2665	0.0330	0.1182	0.0153	2.1529
I-STD	0.1761	0.0097	0.2975	0.0082	0.0584	0.2715	0.0315	0.1151	0.0184	12.0016
HD	0.1742	0.0097	0.2922	0.0083	0.0585	0.4910	0.1116	0.3099	0.0969	63.7493
I-URED	0.1761	0.0097	0.2975	0.0082	0.0584	0.2578	0.0315	0.1097	0.0184	12.1999
I-AO-STD	0.1737	0.0097	0.2920	0.0081	0.0584	0.3030	0.0340	0.1377	0.0156	3.4482
SI-AO-STD	0.1884	0.0098	0.3250	0.0086	2.4451	0.3036	0.0337	0.1304	0.0153	3.3816
I-HDD	0.1768	0.0097	0.2998	0.0082	0.2269	0.3310	0.0347	0.1422	0.0156	2.8290
I-QD	0.1761	0.0097	0.2975	0.0082	0.0584	0.2609	0.0319	0.1102	0.0187	12.2404
ALIEN	0.6756	0.3938	0.9455	0.1435	34.2894	1.0450	0.4065	1.5871	0.4893	47.0513
I-FDFD	0.1913	0.0099	0.3336	0.0086	0.0581	0.2565	0.0335	0.1136	0.0153	1.4766
I-AO-FDFD	0.1761	0.0097	0.2976	0.0082	0.0584	0.2571	0.0319	0.1084	0.0187	11.9735
The following differentiators lead to impracticable experiments: VGED, HGD, E-QD, E-HDD, E-GHDD, I-GHDD, E-URED, SI-URED, E-STD, SI-STD, E-AO-STD, Kalman										
Performance: red<black<blue										

Table 9: Parameters of the differentiators obtained from the tuning procedure

Method	Parameters
Euler	No parameter
LF	$c=109.4369$
E-STD	$L=1040.8758$
I-STD	$L=949.3691$
SI-STD	$L=1009.1908$
E-URED	$L=99.7870, \mu=0.0063$
I-URED	$L=696.0010, \mu=0.0026$
E-QD	$F=97.3600, \alpha=1218.0461$
I-QD	$F=870935.4030, \alpha=1686.2462$
ALIEN	$\bar{T}=0.0866, \kappa=7, \mu=4$
HD	$r=668.3708$
E-AO-STD	$L=4625.2663$
I-AO-STD	$L=4695.8255$
SI-AO-STD	$L=4787.8132$
E-HDD	$L=4605.2213$
E-GHDD	$L=4399.5507$
I-HDD	$L=4807.4546$
I-GHDD	$L=4689.6282$
VGED	$\mu=10.6610, \tau=9.6238, \omega_c=118.5971,$ $q=0.1210$
SI-URED	$L=460.5057, \mu=0.0086$
E-STDAC	$\alpha=105.1825, \epsilon=0.7299$
I-FDFF	$\omega_s=16.6953, \omega_f=7.4529, \rho=1417.6540,$ $\gamma=713.5953$
I-AO-FDFF	$F=190419.5690, \epsilon=35336.8212$ $\omega_s=8814.4365, \omega_f=43917.0106,$ $a_1=19498.1469, \rho=19352.9618$
Kalman	$R=22.2069$
HGD	$L=2.4053$
Input signal: $f(t) = 1000 \sin(200t + \frac{\pi}{4})$	
Initial conditions are set to zero for all differentiators	
Noise type: Quantization (resolution=0.001), $h = 2\text{ms}$	

Table 10: Average results based on 10 experiments (parameters are shown in Table 9)

Method	Simulations					Experiments				
	$\ e_\theta\ $	$\ e_\alpha\ $	$\ e_\theta\ _\infty$	$\ e_\alpha\ _\infty$	$\ u\ _\infty$	$\ e_\theta\ $	$\ e_\alpha\ $	$\ e_\theta\ _\infty$	$\ e_\alpha\ _\infty$	$\ u\ _\infty$
Euler	0.1761	0.0097	0.2975	0.0082	0.0584	0.2725	0.0325	0.1161	0.0221	13.4087
LF	0.1889	0.0098	0.3282	0.0086	0.0582	0.2652	0.0323	0.1177	0.0147	1.5955
I-STD	0.1761	0.0097	0.2975	0.0082	0.0584	0.5135	0.0384	0.2179	0.0276	10.2306
HD	0.1742	0.0097	0.2922	0.0083	0.0585	0.5100	0.1058	0.3301	0.1019	62.6479
I-URED	0.1761	0.0097	0.2975	0.0082	0.0584	0.2484	0.0312	0.1088	0.0184	9.1952
I-QD	0.1761	0.0097	0.2975	0.0082	0.0584	0.2607	0.0307	0.1109	0.0184	12.1380
I-AO-FDFF	0.1761	0.0097	0.2976	0.0082	0.0584	0.2542	0.0305	0.1077	0.0184	11.6567

The following differentiators lead to impracticable experiments:
 VGED, I-FDFF, HGD, Kalman, ALIEN, I-GHDD, E-GHDD, I-HDD, E-HDD, E-QD,
 SI-AO-STD, E-AO-STD, I-AO-STD, SI-URED, E-URED, SI-STD, E-STD

the closed-loop system dynamical behavior.

5.6 Results obtained for the pendulum system

The following results are drawn according to the experiments conducted in Section 5.

- Unlike the implicit methods, the explicit ones show too much chattering which led to impracticability for all of them. This validates **Fact 1**.
- Solver-based implicit SMB differentiators (I-AO-STD, I-HDD, and I-URED) can also be implemented in real-time. They led to practicability in Table 8. This validates **Fact 2**.
- As it can be seen from Table 10, while the implicit first-order differentiators (I-STD, I-URED, I-QD) led to appropriate responses, the arbitrary-order counterparts (I-AO-STD, I-HDD, and I-GHDD) sometimes lead to impracticability. The reason is that a higher-order differentiator shows a larger transient time. This validate **Fact 4**.
- According to Table 8, a higher-order differentiator may present better response (better robustness to noise) than a first-order one (compare I-STD and I-AO-STD in table 8). This validates **Fact 5**.
- Comparing the results corresponding to the LF and the implicit SMB differentiators (I-STD, I-URED, I-QD, and I-AO-FDFF) one can see that the implicit methods supersede the LF. This validates **Fact 6**.

6 General conclusions

Two laboratory setups, i.e., an electro-pneumatic system and a rotary-inverted pendulum, used in this study to implement 25 known differentiators on practical closed-loop control systems. These setups behave differently in the case of dynamic response and noise characteristics. Hence, it is expected that the results remain valid for a wide range of control applications.

The general conclusions are summarized as follows:

- The implicit differentiators can supersede the explicit ones in case of chattering. Comparing the results, it can be concluded that the implicit methods always present smaller variations which imply a better implementation in case of actuator degradation and efficiency. The explicit methods even may lead to impracticability mainly because of too much chattering. This validates **Fact 1**.
- The solver-based implicit differentiators (I-AO-STD, I-HDD, I-GHDD, I-URED) can be implemented on both selected laboratory setups in real-time. It indicates that calculation resources in typical laboratory setups are sufficient enough to implement the solver-based methods. This validates **Fact 2**.
- The implicit methods are insensitive to the gains. In other words, one may select an arbitrary large gain (oversized-gain) for the implicit differentiators without observing numerical chattering. This validates **Fact 3**.
- While all differentiators are tuned in identical conditions, the higher-order differentiators (I-AO-STD, I-HDD, GHDD) led to impracticability since, compared to the first-order counterparts (I-STD, I-URED, I-QD), they present longer transient time. This is in accordance with **Fact 4**.
- Comparing the results, higher-order SMB differentiators, e.g., AO-STD, I-HDD, and I-GHDD show better variation than the cascade configurations of the first-order counterparts, e.g., STD. Apart from the numerical chattering, chattering may also correspond to noise sensitivity. Hence, higher-order differentiators may potentially present more robust behavior to noise which validates **Fact 5**.
- For the EPS, the implicit differentiators show smaller variations than the LF which is in accordance with **Fact 6**. This is also the case for the RIPS since the implicit SMB ones supersede the LF.

According to the experimental data, parameter tuning can significantly affect the results, and may even lead to impracticability. Two important aspects, i.e., transient response and noise filtration, should always be taken into account in parameter tuning. This topic can be considered as a future work to develop more efficient tuning algorithms in closed-loop systems. To this end, multi-criteria optimization methods may be used to improve the optimality of the parameters.

Acknowledgements: This work was supported by the ANR project DigitSlid, ANR-18-CE40-0008-01.

References

- [1] “Qube – servo 2,” <https://www.quanser.com/products/qube-servo-2/>, accessed: 23-04-2020.
- [2] M. R. Mojjallizadeh, B. Brogliato, and V. Acary, “Discrete-time differentiators: design and comparative analysis,” INRIA Grenoble-Alpes, <https://hal.inria.fr/hal-02960923>, Tech. Rep., 2020.
- [3] M. Mojjallizadeh, B. Brogliato, and V. Acary, “Time-discretizations of differentiators: design of implicit algorithms, and comparative analysis,” *Submitted to the International Journal of Robust and Nonlinear Control*, vol. under submission, 2020.
- [4] A. Levant and M. Livne, “Robust exact filtering differentiators,” *European Journal of Control*, vol. 55, pp. 33–44, 2020.
- [5] S. Koch, M. Reichhartinger, M. Horn, and L. Fridman, “Discrete-time implementation of homogeneous differentiators,” *IEEE Transactions on Automatic Control*, vol. 65, no. 2, pp. 757–762, 2020.
- [6] M. Livne and A. Levant, “Proper discretization of homogeneous differentiators,” *Automatica*, vol. 50, no. 8, pp. 2007 – 2014, 2014.
- [7] E. Cruz-Zavala, J. A. Moreno, and L. M. Fridman, “Uniform robust exact differentiator,” *IEEE Transactions on Automatic Control*, vol. 56, no. 11, pp. 2727–2733, 2011.
- [8] S. Jin, R. Kikuuwe, and M. Yamamoto, “Real-time quadratic sliding mode filter for removing noise,” *Advanced Robotics*, vol. 26, no. 8-9, pp. 877–896, 2012.
- [9] A. Levant, “Robust exact differentiation via sliding mode technique,” *Automatica*, vol. 34, no. 3, pp. 379–384, 1998.
- [10] M. Reichhartinger and S. Spurgeon, “An arbitrary-order differentiator design paradigm with adaptive gains,” *International Journal of Control*, vol. 91, no. 9, pp. 2028–2042, 2018.
- [11] S. Koch and M. Reichhartinger, “Discrete-time equivalent homogeneous differentiators,” in *2018 15th International Workshop on Variable Structure Systems (VSS)*, 2018, pp. 354–359.
- [12] M. Reichhartinger, S. Koch, H. Niederwieser, and S. K. Spurgeon, “The robust exact differentiator toolbox: Improved discrete-time realization,” in *2018 15th International Workshop on Variable Structure Systems (VSS)*, 2018, pp. 1–6.
- [13] H. K. Khalil, *Nonlinear systems*. Pearson Education, 2015.
- [14] G. Byun and R. Kikuuwe, “An improved sliding mode differentiator combined with sliding mode filter for estimating first and second-order derivatives of noisy signals,” *Int. J. Control Autom. Syst.*, vol. 18, p. 3001–3014, 2020.

- [15] J. Carvajal-Rubio, J. Sánchez-Torres, M. Defoort, and A. Loukianov, “On the discretization of robust exact filtering differentiators,” *arXiv preprint arXiv:1911.09232*, 2019.
- [16] J. E. Carvajal-Rubio, A. G. Loukianov, J. D. Sánchez-Torres, and M. Defoort, “On the discretization of a class of homogeneous differentiators,” in *2019 16th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2019, pp. 1–6.
- [17] R. Kikuuwe, R. Pasaribu, and G. Byun, “A first-order differentiator with first-order sliding mode filtering,” *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 771–776, 2019.
- [18] J.-J. E. Slotine, J. K. Hedrick, and E. A. Misawa, “On Sliding Observers for Nonlinear Systems,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 109, no. 3, pp. 245–252, 1987.
- [19] M. Ghanes, J. P. Barbot, L. Fridman, A. Levant, and R. Boisliveau, “A new varying gain exponent based differentiator/observer: an efficient balance between linear and sliding-mode algorithms,” *IEEE Transactions on Automatic Control*, vol. 65, no. 12, pp. 5407–5414, 2020.
- [20] A. Levant, “Higher-order sliding modes, differentiation and output-feedback control,” *International Journal of Control*, vol. 76, no. 9-10, pp. 924–941, 2003.
- [21] E. Cruz-Zavala and J. A. Moreno, “Levant’s arbitrary-order exact differentiator: A Lyapunov approach,” *IEEE Transactions on Automatic Control*, vol. 64, no. 7, pp. 3034–3039, 2019.
- [22] E. Cruz-Zavala and J. A. Moreno, “Lyapunov functions for continuous and discontinuous differentiators,” *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 660–665, 2016.
- [23] A. Levant, “Filtering differentiators and observers,” in *2018 15th International Workshop on Variable Structure Systems (VSS)*, 2018, pp. 174–179.
- [24] —, “Homogeneous filtering and differentiation based on sliding modes,” in *2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France*, 2019, pp. 6013–6018.
- [25] Z. Lv, S. Jin, X. Xiong, and J. Yu, “A new quick-response sliding mode tracking differentiator with its chattering-free discrete-time implementation,” *IEEE Access*, vol. 7, pp. 130 236–130 245, 2019.
- [26] C. Vazquez, S. Aranovskiy, L. B. Freidovich, and L. M. Fridman, “Time-varying gain differentiator: A mobile hydraulic system case study,” *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1740–1750, 2016.
- [27] J. A. Moreno, “Exact differentiator with varying gains,” *International Journal of Control*, vol. 91, no. 9, pp. 1983–1993, 2018.
- [28] M. Ghanes, J. P. Barbot, L. Fridman, and A. Levant, “A novel differentiator: A compromise between super twisting and linear algorithms,” in *2017 IEEE 56th Annual Conference on Decision and Control (CDC), Melbourne, Australia*, 2017, pp. 5415–5419.

- [29] —, “A second order sliding mode differentiator with a variable exponent,” in *2017 American Control Conference (ACC), Seattle, USA, 2017*, pp. 3300–3305.
- [30] L. K. Vasiljevic and H. K. Khalil, “Error bounds in differentiation of noisy signals by high-gain observers,” *Systems & Control Letters*, vol. 57, no. 10, pp. 856–862, 2008.
- [31] Y. Wang, G. Zheng, D. Efimov, and W. Perruquetti, “Differentiator application in altitude control for an indoor blimp robot,” *International Journal of Control*, vol. 91, no. 9, pp. 2121–2130, 2018.
- [32] W. Perruquetti and T. Floquet, “Homogeneous finite time observer for nonlinear systems with linearizable error dynamics,” in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 390–395.
- [33] M. Mboup, C. Join, and M. Fliess, “Numerical differentiation with annihilators in noisy environment,” *Numerical Algorithms*, vol. 50, no. 4, pp. 439–467, 2009.
- [34] M. Mboup and S. Riachy, “Frequency-domain analysis and tuning of the algebraic differentiators,” *International Journal of Control*, vol. 91, no. 9, pp. 2073–2081, 2018.
- [35] B. Brogliato and A. Polyakov, “Digital implementation of sliding-mode control via the implicit method: A tutorial,” *Int. J. Robust and Nonlinear Control*, Special issue.
- [36] K. Alonzo, “A 3d state space formulation of a navigation Kalman filter for autonomous vehicles,” *Carnegie Mellon University. Technical Report CMU-RI-TR-94-19-REV*, vol. 2, p. 105, 1994.
- [37] M. Taleb, A. Levant, and F. Plestan, “Pneumatic actuator control: Solution based on adaptive twisting and experimentation,” *Control Engineering Practice*, vol. 21, no. 5, pp. 727–736, 2013.
- [38] Y. Shtessel, M. Taleb, and F. Plestan, “A novel adaptive-gain supertwisting sliding mode controller: Methodology and application,” *Automatica*, vol. 48, no. 5, pp. 759–769, 2012.
- [39] M. Taleb, A. Levant, and F. Plestan, “Twisting algorithm adaptation for control of electropneumatic actuators,” in *2012 12th International Workshop on Variable Structure Systems*, 2012, pp. 178–183.
- [40] M. Belgharbi, S. Sesmat, S. Scavarda, and D. Thomasset, “Analytical model of the flow stage of a pneumatic servo-distributor for simulation an nonlinear control,” in *SICFP*, vol. 2, Tampere, Finland, May 1999, pp. 847–860. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01119358>
- [41] O. Huber, B. Brogliato, V. Acary, A. Boubakir, F. Plestan, and B. Wang, “Experimental results on implicit and explicit time-discretization of equivalent control-based sliding mode control,” in *Recent Trends in Sliding Mode Control*, L. Fridman, J. Barbot, and F. Plestan, Eds. IET, 2016, pp. 207–235.
- [42] O. Huber, V. Acary, B. Brogliato, and F. Plestan, “Implicit discrete-time twisting controller without numerical chattering: Analysis and experimental results,” *Control Engineering Practice*, vol. 46, pp. 129–141, 2016.

- [43] C. C. Y. Dorea, “Expected number of steps of a random optimization method,” *Journal of Optimization Theory and Applications*, vol. 39, no. 2, pp. 165–171, 1983.
- [44] X. Yang and X. Zheng, “Swing-up and stabilization control design for an underactuated rotary inverted pendulum system: Theory and experiments,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 9, pp. 7229–7238, 2018.
- [45] D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2004.