A survey on the discrete-time differentiators in closed-loop control systems: experiments on electro-pneumatic system and rotary inverted pendulum

Mohammad Rasool Mojallizadeh[a,*], Bernard Brogliato[a], Andrey Polyakov[b], Subiksha Selvarajan[d], Loïc Michel[d], Franck Plestan[d], Malek Ghanes[d], Jean-Pierre Barbot[d,e], Yannick Aoustin[c]

[a] *Univ. Grenoble Alpes, INRIA, CNRS, Grenoble INP, LJK, 38000 Grenoble, France*
[b] *Inria, Univ. Lille, CNRS, Lille, France*
[c] *Université de Nantes—LS2N, UMR CNRS 6004, 44321, Nantes, France*
[d] *Ecole Centrale de Nantes—LS2N, UMR CNRS 6004, 44321, Nantes, France*
[e] *ENSEA, Cergy-pontoise, France*

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 including the exact differentiators and a linear filter. 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 based on the analytical analysis and numerical simulations, has been investigated experimentally, and several comments are provided to allow one to select an appropriate differentiation scheme in practical closed-loop control systems.

Keywords: discrete-time differentiator, implicit discretization, explicit discretization, electro-pneumatic system, rotary inverted pendulum, experimental data

20XX MSC: 00-00 00 00-00

1. Introduction

Online differentiation is an unavoidable subject in most 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 because 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 [1, 2].

While the differentiators have been widely studied using analytical calculations and numerical simulations [1, 2, 3, 4], their practical implementations are barely addressed in the literature. References [5, 6] are probably the only ones on this topic, according to our investigation, where the aim was to provide an experimental analysis among a sliding-mode based differentiator, kernel-based method, high-gain differentiator, ALIEN differentiator, extended Kalman filter, and homogeneous differentiator. Since the differentiators are nowadays implemented

*Corresponding author
Email addresses: mohammad-rasool.mojallizadeh@inria.fr (Mohammad Rasool Mojallizadeh), bernard.brogliato@inria.fr (Bernard Brogliato), andrey.polyakov@inria.fr (Andrey Polyakov), subiksha.selvarajan@eleves.ec-nantes.fr (Subiksha Selvarajan), loic.michel@ec-nantes.fr (Loïc Michel), franck.plestan@ec-nantes.fr (Franck Plestan), malek.ghanes@ec-nantes.fr (Malek Ghanes), barbot@ensea.fr (Jean-Pierre Barbot), Yannick.Aoustin@univ-nantes.fr (Yannick Aoustin)

on digital processors, their time-discretization should be addressed clearly.

The differentiators usually contain discontinuous terms to achieve the exactness. Such terms can be potentially a source of numerical chattering[1] if they are not implemented on digital processors correctly. According to the literature, time-discretization of the differentiators can be divided into two main categories, i.e., explicit and implicit Euler schemes. While the explicit time-discretization has long been used for sliding-mode control (SMC) as the unique discretization method, the implicit method has been developed recently for SMC [12, 13, 14, 15, 16, 17, 18] (see [19] for a survey) and for differentiators [1, 2, 3, 4]. In this context, few experimental validations of implicitly implemented SMC can also be found [20, 13, 12, 15]. However, the implicit discretization has not been experimentally studied for the differentiators yet.

Numerical chattering suppression under noise-free conditions is one of the main advantages of using the implicit discretization as reported by [12, 13, 14, 15, 19]. This is also specifically validated for the differentiators (see Lemma 3 in [1], the numerical simulations in [4], and the conclusions made in Section 6 of [3]). This property is called Theoretical Result 1 (TR1) throughout the manuscript. Note that this result was specifically obtained for the noise-free case. However, in a practical system, the chattering can also be caused by the measurement noise, in addition to the numerical chattering, which affects both explicit and implicit schemes. One of the objectives of this work is to study TR1 using practical experiments.

Another issue corresponding to the implicit discretization is practicability. This topic has been addressed for the differentiators in [1, 2] in case of causality and uniqueness of the solutions. Moreover, unlike the explicit schemes, the implicit discretization needs an iterative solver to compute the roots of a polynomial equation at each time step. The effect of the solver and its parameters on the performances of a specific type of implicit differentiator has been addressed before and it is concluded that the implicit differentiators can also be implemented if enough calculation resources are available (see Sec. 5.8.5 in [2] and [3]). However, the amount of the required calculation resources has not been reported. This is considered as the Theoretical Result 2 (TR2) in this manuscript and will be evaluated on two different laboratory setups.

As reported by [12, 13, 14, 15, 19], the implicit discretization of SMC can be insensitive to the gains under noise-free cases and during the sliding phase. This has been specifically validated for the implicit arbitrary-order super-twisting differentiator (I-AO-STD) (see Remark 7 in [1]). This is considered as Theoretical Result 3 (TR3) in this study, and the gain sensitivity of both implicit and explicit schemes will be studied in two applications where measurement noise always exists.

According to Corollary 1 in [1], increasing the order of the I-AO-STD leads to a longer transient time. More clearly, when the state variables of the I-AO-STD are in a specific region, defined by case 2 in [1], the differentiator converges to the exact differentiation after $n+1$ time step, where n is the order of the differentiator. Hence, increasing the order of the differentiator leads to a larger transient time. Note that this is proved when the derivative of order $n+1$ of the input signal vanishes. This result is named Theoretical Result 4 (TR4) in this study and it will be studied experimentally. Note that the transient response of a differentiator depends on several other factors, including the initial conditions and the parameters.

Following the literature, TR1 and TR3 have been also evaluated experimentally for the SMCs in [12, 20, 13, 15]. However, the validity of the mentioned results for the differentiators has not been experimentally addressed yet.

[1] Chattering refers to the high-frequency, finite-amplitude oscillations, affecting both the input and the output. While many factors including measurement noise and dynamical uncertainties can lead to the *chattering* [7], the *numerical chattering* refers to a specific kind of chattering that only appears due to the time-discretization of the discontinuous terms [8, 9, 10, 11].

The main contribution of this work is to investigate the above-mentioned theoretical results in the closed-loop systems, based on 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. To this end, two laboratory setups, an *electro-pneumatic setup* (EPS) and a *rotary inverted pendulum setup* (RIPS) have been used to implement the 25 differentiators which have been studied in [1, 2]. The main differences between these two setups are:

- **Measurement noise**: an analog sensor is mounted on the EPS which requires an analog-to-digital conversion. This imposes a significant amount of measurement noise. On the other hand, the RIPS uses two digital high-resolution shaft encoders, which makes the feedback path almost noise-free.
- **Dynamic response**: comparing the outputs of the EPS and the RIPS, one can see that while the EPS shows slow changes on the output, the RIPS exhibits sudden variations.

According to the characteristics of these setups, it can be concluded that the EPS mainly evaluates the robustness of the differentiators against the measurement noise while the RIPS allows to test the differentiators in the presence of fast dynamical variations. Since these setups behave differently in case of dynamic response and noise characteristics, it is expected that the general conclusions remain valid for a wide range of control applications.

The structure of this manuscript is as follows. The continuous-time differentiators and the corresponding discretization schemes are briefly reviewed in Secs. 2 and 3, respectively. The results drawn from the EPS as well as the RIPS are presented in Secs. 4 and 5, respectively. Finally, general conclusions and remarks are provided in Sec. 6. Note that the colors red black and blue are used in some tables to help showing the worst, moderate and the best results.

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 [1, 2]. Depending on the structure, differentiators can be categorized into several classes as indicated in Table 1. Some useful vocabulary is introduced:

- *SMB* (sliding-mode-based) differentiator refers to a kind of differentiator where discontinuous or *set-valued (SV)* terms are used to provide a sliding regime to achieve the exactness.
- Some differentiators can calculate *higher-order differentiations (HOD)* (second, third,...). Without such an ability, cascade configurations of first-order differentiators may be used to estimate higher-order derivatives.
- Differentiators may have *multiple outputs (MO)*, meaning that an MO differentiator with the order n can calculate derivatives of order $0, \dots, n-1$.
- Depending on the structure, differentiators may have one or more *tuning parameters (TP)* that should be tuned to ensure proper operation.
- Some differentiators use 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 sliding-mode algorithms. The Slotine-Hedrick-Misawa differentiator (SHMD) has been the first invented SMB differentiator [33]. The general form of the SHMD is as follows [33]:

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

Nomenclature	
AO-STD [21]	arbitrary-order super-twisting differentiator
E-AO-STD [21]	explicit arbitrary-order super-twisting differentiator
E-GHDD [22]	explicit generalized homogeneous discrete-time differentiator
E-HDD [23]	explicit homogeneous discrete-time differentiator
E-URED [24]	explicit uniform robust exact differentiator
E-QD [25]	explicit quadratic differentiator
E-STD [26]	explicit super-twisting differentiator
E-STDAC [27]	explicit super-twisting differentiator with adaptive coefficients
GE	generalized equation
GHDD [22]	generalized homogeneous discrete-time differentiator
HD [28, 29]	homogeneous differentiator
HDD [23]	homogeneous discrete-time differentiator
HGD [30]	high-gain differentiator
I-AO-FDFF [31]	implicit arbitrary-order differentiator with first-order sliding-mode filtering
I-AO-STD [3, 4]	implicit arbitrary-order super-twisting differentiator
I-FDFF [32]	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 [25]	implicit quadratic differentiator
I-STD	implicit super-twisting differentiator
LF	linear filter
ODE	ordinary differential equation
QD [25]	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 [33]	Slotine-Hedrick-Misawa differentiator
SMB	sliding-mode based
SNR	signal-to-noise ratio
STD [26]	super-twisting differentiator
STDAC [27]	super-twisting differentiator with adaptive coefficients
URED [24]	uniform robust exact differentiator
VGED [34]	variable gain exponent differentiator

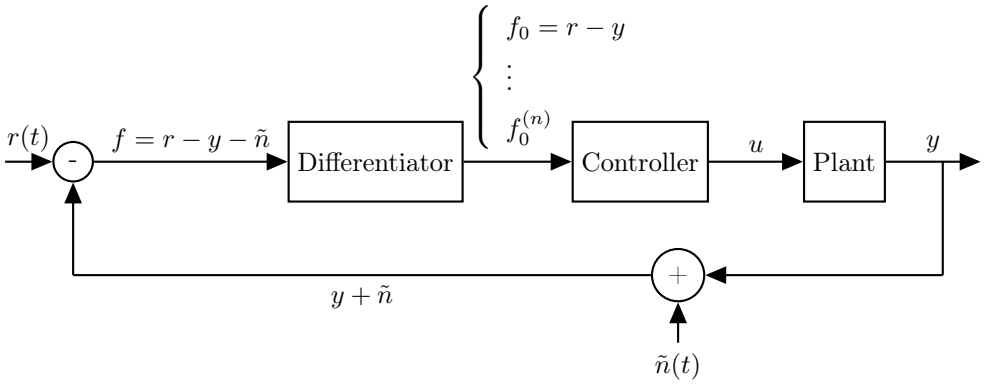


Figure 1: A typical control loop

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

Method	SMB	SV	HOD	MO	TP	AM	Source	Remarks
SHMD	✓	✓	✓	✓	$2(n+1)$	✗	[1] (S2.1)	existence of the sliding phase
STD	✓	✓	✗	✓	1	✗	[1] (S2.2)	finite-time convergence
AO-STD	✓	✓	✓	✓	1	✗	[1] (S2.3)	finite-time convergence
URED	✓	✓	✗	✓	2	✗	[1] (S2.4)	uniform convergence
QD	✓	✓	✗	✓	2	✗	[1] (S2.5)	improved transient
STDAC	✓	✓	✗	✓	2	✓	[1] (S2.7.1)	adaptive coefficients
VGED	✓	✗	✗	✓	4	✓	[1] (S2.7.2)	adaptive exponent
ALIEN	✗	✗	✓	✗	3	✗	[1] (S2.8)	algebraic formula
HGD	✗	✗	✓	✓	1	✗	[1] (S2.9)	linear structure

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

where the inclusion $\in$ is used instead of the equality, to indicate the use of set-valued signum function in the right-hand side. Moreover, $\sigma_0(t) = z_0(t) - f(t)$ 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, $i = 0, 1, \ldots, n$ are positive constants, n is the order of the differentiator, $\in$ is written instead of $=$ to indicate that the right-hand side is 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 [33].

Arbitrary-order super-twisting differentiator (AO-STD) is another SMB differentiator introduced in [35], that exhibits several useful properties including homogeneity and finite-time convergence [36, 37]. It is designed as follows:

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

where the notation is as before, L is a tuning parameter, the parameters $\lambda_i, i = 0, \ldots, n$ are provided in Table 2 [35, 21, 38, 39] (notice that the λ_i's may also be considered as tunable parameters, however this is out of the scope of this study), and $\lceil x \rfloor^n = |x|^n \operatorname{sgn}(x)$. For $n = 1$, AO-STD turns into the super-twisting differentiator (STD) [26]. Uniform robust exact differentiator (URED) [24] is a modification of the STD, where the aim is to ensure the uniform convergence of the differentiator by adding extra terms as follows:

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

The notation is similar to the above one, and $\mu \in \mathbb{R} > 0$. Note that the original form of the URED was introduced with $L = 1$ [24].

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

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

where $F > 0$ and $\alpha > 0$ are parameters to be tuned. The sliding variable $\sigma_0(t)$ of this differentiator has been further modified [40] 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 [27, 41, 42] dealt with the adaptive laws for the coefficients, while in other studies [43, 44, 34], the adaptation mechanisms are considered for the exponents.

One of the latest adaptation mechanisms for the coefficients has been developed in [27], where the following adaptive differentiator has been proposed. This differentiator is named super-twisting differentiator with adaptive coefficients (STDAC). It reads as:

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

It can be seen that for $\gamma(t) = \sqrt{L}$, (5) leads to the standard STD. The following adaptation law $\gamma(t)$ is proposed [27]:

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

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

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

been conducted in [34]. The continuous-time VGED reads as:

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

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 $\epsilon = \frac{1}{\mu}$ [34], 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 [30]. The design of this differentiator was further addressed in [45]. In this study, a third-order HGD will be considered as follows (see [5, 46]).

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

Obtaining the error band of the HGD in the presence of noise was the topic of [45].

Algebraic continuous-time differentiators are also proposed in the control literature. ALIEN [47] 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 [48]:

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

$z^{(n)}(t)$ 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 according to simulations [48].

3. Time-discretization of the differentiators

To implement the continuous-time differentiators introduced in Sec. 2 on a digital processor, 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

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

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 [1, 2, 3, 4] 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 for discrete-time sliding-mode controllers and differentiators, e.g., finite-time convergence, Lyapunov stability (see [1, 3, 14, 49, 50] and [19] for a survey).*

Throughout the manuscript, the notations E-X, SI-X, and I-X are for 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 using (10) on the continuous-time differentiators introduced in Sec. 2.

There are three other explicit discrete-time differentiators namely, homogeneous differentiator (HD) [28, 29], homogeneous discrete-time differentiator (HDD) [23], and generalized homogeneous discrete-time differentiator (GHDD) [22] 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 in the recursion.

It should also be noted that the explicit discretization of the SHMD is ignored because of too much numerical chattering [2]. Moreover, according to [32], 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 used instead of $z_i, i = 0, \ldots, n$ as the outputs [31]. 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 as their parameter-tuning procedures, are explained in [1, 2].

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 [21]:

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

where $z_k = [z_{0,k}, z_{1,k}, \ldots, z_{n,k}]^\top \in \mathbb{R}^{n+1}$ is the estimation vector. The parameters for a third-order Kalman differentiation are as follows (see [51] for the algorithm as well as the preliminary equations of the Kalman filter).

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

The Kalman filter has one parameter, denoted by R, which is obtained based on the power of the input noise [1, 2].

In the simplest case, the LF is a low-pass filter combined with the pure differentiator $\mathrm{d}/\mathrm{d}t$ 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 $c \to \infty$, LF leads to the Euler differentiator:

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

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)).

Since tuning parameters of differentiators in practical closed-loop systems is difficult and time consuming, the parameters are tuned using open-loop simulations as suggested in [1, 2]. In this case, the parameters have been tuned based on three different inputs. The first input $I_1 : f_1(t) = \sin(t) + \tilde{n}(t)$ ($h = 50$ms) with $\tilde{n}(t)$ being a white noise with SNR=30dB. The second and the third inputs are $I_2 : f_2(t) = 4 \times 10^4(t^2 + 2t + 3)$ and $I_3 : f_3(t) = 1000 \sin(200t + \frac{\pi}{4})$, respectively, which are polluted by the quantization noise (resolution=0.001) with $h = 2$ms. Subsequently, a randomized algorithm [52] 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.

While this procedure does not ensure the optimality of the parameters for the closed-loop system, it helps to tune all differentiators in a unified manner leading to fair comparisons. In Table 3, three values are given for each parameter in (P_1, P_2, P_3) format, where P_1, P_2 and P_3 denote the tuned parameter for the first (I_1), second (I_2) and the third (I_3) inputs.

Since, the EPS is a noisy and slow process, the low frequency input (I_1) contaminated by a significant amount of noise has been selected to tune the parameters. Such a low frequency signal leads to instability when dealing with the RIPS (except for the Euler differentiator which does not have any parameter). Hence high-frequency inputs (I_2 and I_2) are used to tune the parameters for the RIPS. These input signals are selected according to trial and error.

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.

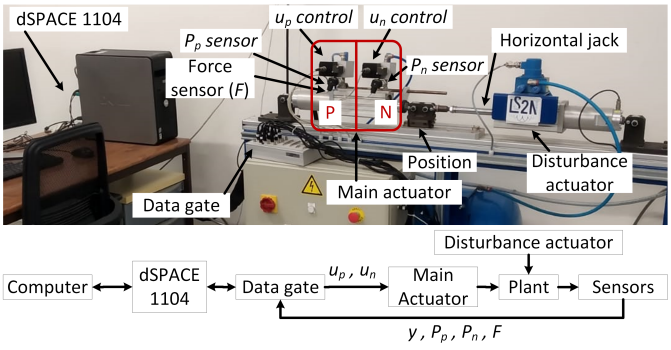


Figure 2: Electropneumatic setup

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. A DS1104 board has been employed to implement the controller and the differentiator for the EPS where a 64-bit MPC8240 processor working on 250 MHz and 32 MB RAM have been installed.

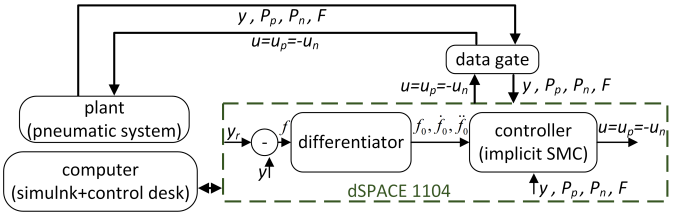


Figure 3: Closed-loop diagram of the EPS

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

The following model is used for the EPS, as detailed in [53, 54, 55]:

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

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, $T = 293.15$K denotes the temperature, $r = 287.0365$J/kg/K is the ideal gas constant, and $b_v = 50$ N 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 [56][1]. Eq. (15) can be rewritten as $\dot{x} = f(x) + g(x)u$ with the following uncertain vectors:

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

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

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

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, SMCs have been developed for the EPS [12, 13]. Following [12], since the control objective is the position tracking, the following sliding variable is defined:

$$\sigma(t) = e_2(t) + \lambda_1 e_1(t) + \lambda_0 e_0(t) \tag{18}$$

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_0 and λ_1 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, the reaching-phase, and the sliding phase. During the reaching phase, the control signal is designed such that $\sigma(t) \to 0$. At the end of the reaching-phase, $\sigma(t) = 0$ holds, and according to the stability of the sliding surface, $e(t) \to 0$ 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,

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

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

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

The functions $\bar{\Psi}$ and $\bar{\Phi}$ are not given here for the sake of briefness. They can be found in [53]. 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:

$$\dot{\sigma} = 0 \quad \to \quad \bar{u} \in -\frac{\bar{\Psi}}{\bar{\Phi}}. \tag{21}$$

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

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

where G is the gain of the control which is selected based on the following inequality [12]:

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

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 [12, 15], and it is concluded that the implicit discretization can provide several advantages compared to its explicit counterpart, including numerical chattering suppression, and insensitivity to the gains. The explicit discretization of (22) is avoided in this study since

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

it leads to chattering even in the presence of an implicit differentiator [12]. On the other hand, the implicit discretization of the set-valued part of (22) gives:

$$u_k \in -\frac{1}{\bar{\Phi}_k}[\bar{\Psi}_k - G\,\mathrm{sgn}(\sigma_{k+1})] \tag{24}$$

where u_k is applied on $[t_k, t_{k+1}]$. Substituting (24) into (20), and assuming that $\tilde{\Phi} = \tilde{\Psi} = 0$ gives:

$$\sigma_{k+1} \in -G\,\mathrm{sgn}(\sigma_{k+1}) + \sigma_k. \tag{25}$$

Eq. (25) can be rewritten as follows:

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

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

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

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

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

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

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

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

  The solution of the GE (26) satisfies $\sigma_{k+1} < 0$ which leads to $\mathrm{sgn}(\sigma_{k+1}) = -1 \rightarrow u_k \in -\frac{1}{\bar{\Phi}_k}[\bar{\Psi}_k + G]$. 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 [1, 2] are used to estimate the first and the second-order differentiations. The controller and differentiators are implemented on the EPS as follows:

- Some differentiators (STD, VGED, QD, STDAC, FDFF) can only estimate the first-order derivative. Cascade configurations of these differentiators are used to estimate both the velocity and the acceleration.

- Some differentiators have MOs (AO-STD, SHMD, HDD, GHDD, AO-FDFF, HD), and can estimate any differentiations up to order n $(0, \ldots, n)$ simultaneously, n being the order of the differentiation.

- 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 with $G = 10^5$.

- The sampling rates correspond to all subsystems (controller, differentiator, sensors, *etc*), and 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, the waveforms are recorded and the performances are calculated after a specific amount of time, to ensure zero initial conditions for

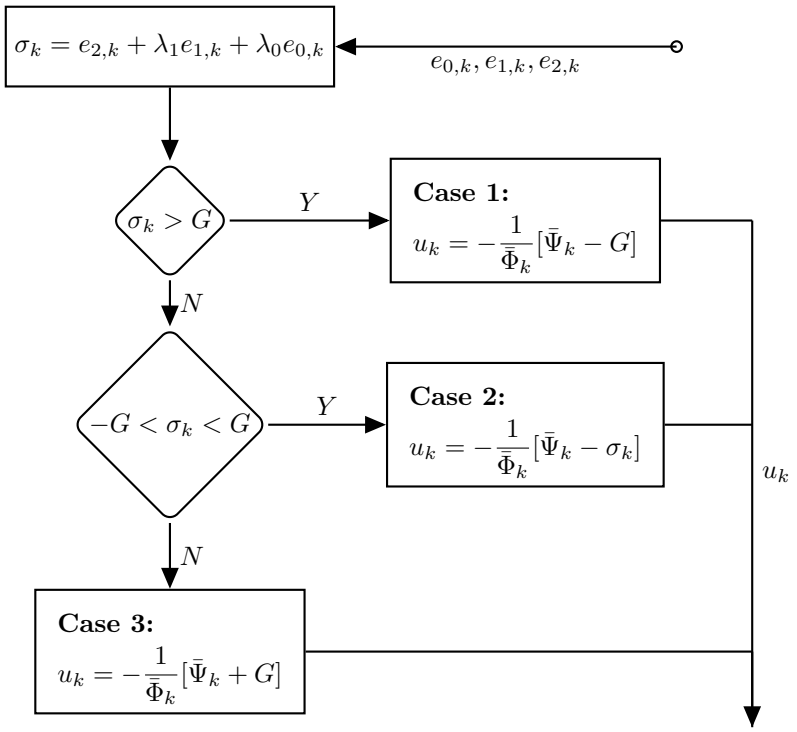


Figure 4: Flowchart of the first-order implicit SMC

all differentiators. Therefore, the reaching phase is neglected in all experiments.

- 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, I-HDD, and HD are implemented for $n = 2$, while for the I-GHDD, $n = 3$ is considered.

A differentiation toolbox (see Appendix E in [2])[1] has been used to tune the parameters. 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 used, but this is outside the scope of this work.

Unlike the numerical simulations, it is not possible to measure the performances of the differentiators directly since the exact values of the differentiations are not available in practical closed-loop systems. To solve this drawback, we have tried to evaluate the differentiators indirectly. To this end, we have studied the effect of each differentiator on the overall performances of the closed-loop control system such as output tracking or the chattering on the control input. More specifically, 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 value indicates better output tracking. This cost function is defined as follows:

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

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

[1] A 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 Mojallizadeh (email: mohammad-rasool.mojallizadeh@inria.fr)

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:

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

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

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

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, and the corresponding experiment is called "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 Fig. 13 in [1]), the E-QD shows too much chattering for large sampling times, 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

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

Method	Parameters
Euler	No parameter
LF	c=(7.1,184,109)
E-STD	L=(0.8,80949,1040)
I-STD	L=(0.7,80841,949)
SI-STD	L=(0.7,80859,1009)
E-URED	L=(0.1,2.7,100),μ=(20.9,0.002,0.01)
I-URED	L=(0.1,114,696),μ=(21.2,0.001,0.002)
E-QD	F=(4.4,46987,97),α= (0.4,3179,1218)
I-QD	F=(4.5,13 $\times 10^5$),α=(0.8,17 $\times 10^5$)
ALIEN	$\bar{T}$=(0.5,0.8,0.09),κ=(1,3,7),μ=(2,10,4)
HD	r=(2.6,3410,668)
E-AO-STD	L=(4.9,15662,4625)
I-AO-STD	L=(2.9,15412,4695)
SI-AO-STD	L=(2.8,15815,4788)
E-HDD	L=(4.9,15917,4605)
E-GHDD	L=(4.9,15899,4400)
I-HDD	L=(3.0,16116,4807)
I-GHDD	L=(3.0,16369,4690)
VGED	μ=(4.4,4,11),τ=(1.3,32,10) ω_c=(12.2,148,119) q=(0.3,8.5,012)
SI-URED	L=(0.1,134,461),μ=(93.6,0.003,0.01)
E-STDAC	α=(0.5,859,105),ϵ=(0.0,0.7,0.7)
I-FDFF	ω_s=(19.7,90,17),ω_f=(8.5,3.6,7) ρ=(8.7,473,1418),γ=(0.0,1042,714)
I-AO-FDFF	F=(37.8,17171508,190420) ϵ=(18.6,35712,35337),ω_s=(2.5, 22408,8814),ω_f=(63,8633,43917),α_1= (456,15621,19498),ρ=(88,40688,19353)
Kalman	($R = 8 \times 10^{-4}$,32,22)
HGD	L= (3.5,0.7,2.4)
Three values are provided for each parameter, corresponding to the inputs (P_1, P_2, P_3)	

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 if an implicit SMC is implemented, the use of explicit differentiators may lead to chattering on the control signal for large sampling times. In other words, to reduce the numerical chattering, specifically for large sampling times, 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 the cascade combinations of the explicit or semi-implicit differentiators presents 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 $h > 10$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 $h > 40$ms, it shows relatively large variations for smaller sampling times. The reason is that six parameters of this differentiator are tuned for $h = 50$ms. Decreasing the sampling time, deteriorates its variation, which indicates its sensitivity to the parameters. Another interesting observation is that Kalman's differentiator shows one of the best variations, comparable with the implicit methods. However, I-FDFF shows a smaller variation for $h = 1$ms, compared with that of the Kalman' differentiator.

The variations of the control signal corresponding to some of the SMB differentiators for large sampling times $h = 50$ms, $h = 40$ms, and $h = 30$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. This result confirms experimentally **TR1** on the chosen setup and control algorithm.

The $\bar{L}_2$ norms 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 differ-

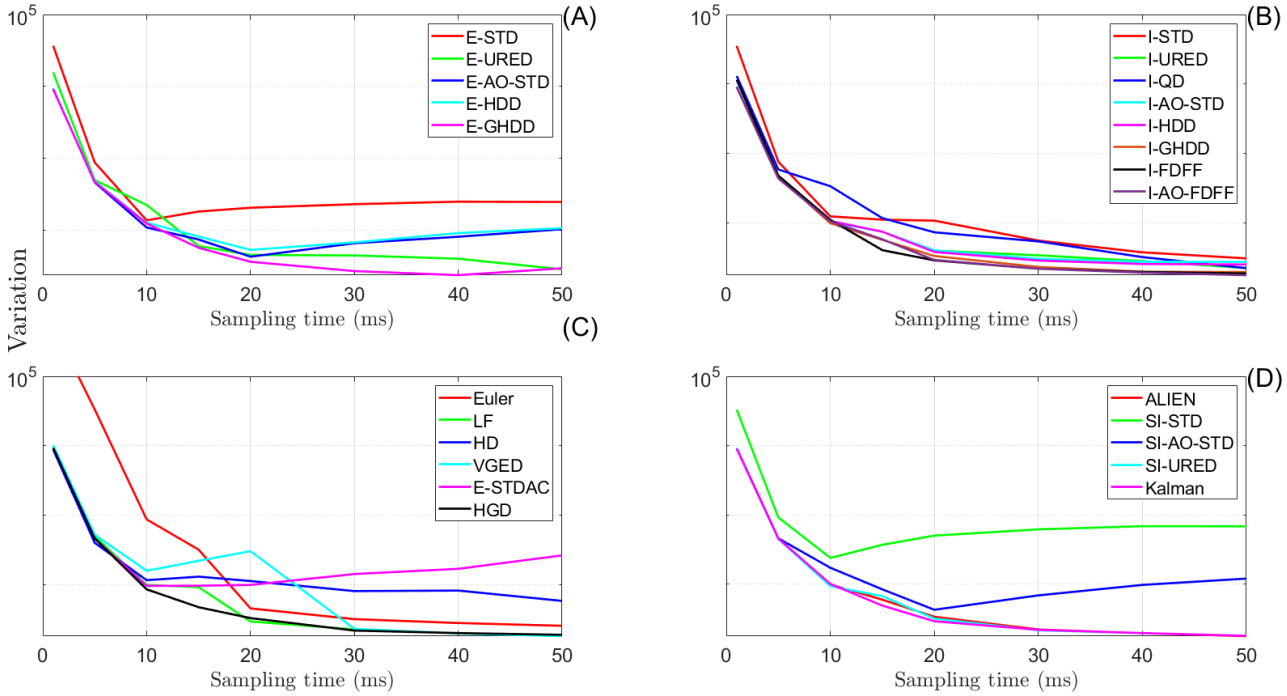


Figure 5: Variation of the control signal $u(t)$ under different sampling times

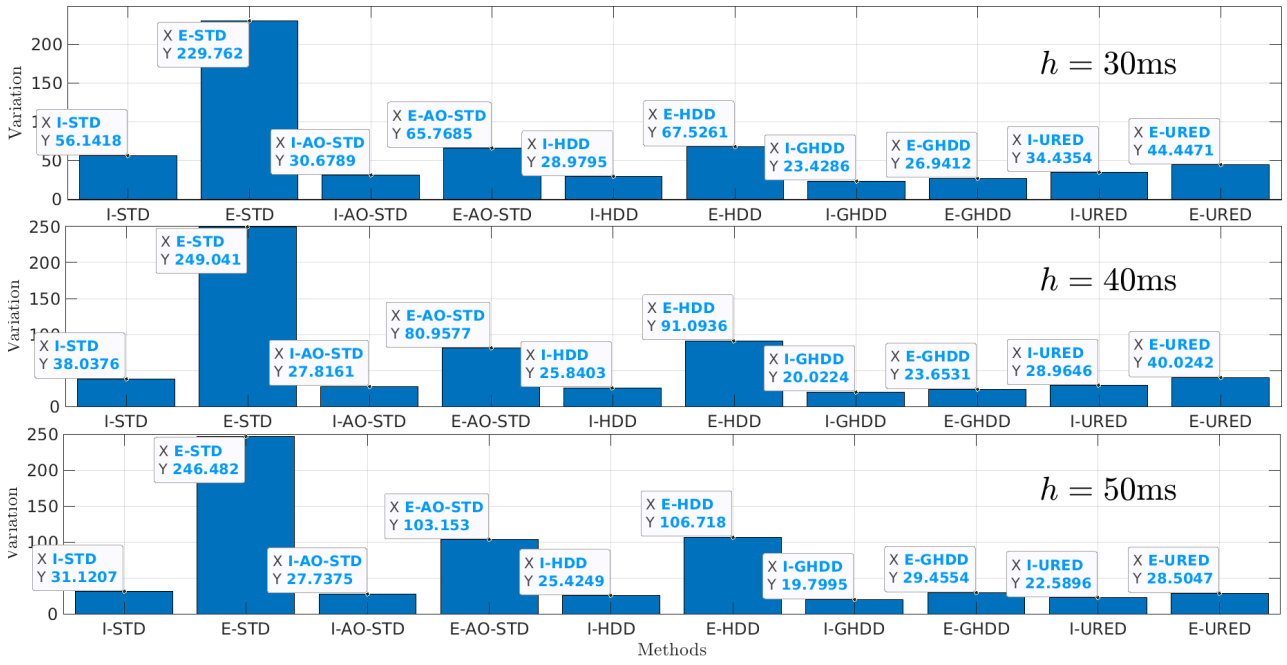


Figure 6: Comparison of the total variations for the implicit and explicit schemes

entiators 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 SMB differentiators can behave better than the LF.

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 predicted by **TR3**.

4.7. Solver's effect

As it was previously explained in [3, 4], some implicit differentiators I-AO-STD, I-HDD, and I-GHDD need an iterative solver to solve a polynomial equation at each time-step. The required calculation resources should be small enough to ensure the real-time operation of the system. A set of experiments has 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 [1, 2] (see also [3]), the required accuracy for the solver without affecting the performances is 10^{-5} when the input is $\sin(t)$. Hence, the Newton's solver with the same accuracy is used, and the maximum number of iterations for the real-time operation is obtained and given in Table 6.

As can be seen from Table 6, for $h < 1$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 $h > 1$ms, all the implicit methods can be implemented with the indicated number of iterations. *It has been noticed during these experiments that for a larger number of iterations, the implicit methods cannot be implemented in real-time.*

It is reported before (see [1] 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 $h > 1$ms is always higher than 7, one can conclude that the implicit methods can operate in real-time for $h > 1$ms. Normally, this should not be an obstacle to the implementations, since implicit SMC tolerates larger sampling periods without significant deterioration of the performance [13, 12].

4.8. Results obtained for the EPS: conclusions

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

- In a real system, the chattering is caused by several sources including the measurement noises, some dynamics uncertainties, and the numerical chattering. To validate TR1, several experiments have been made under different sampling times (as shown in Figs. 5 and 6). We noticed that while there is no significant difference between the explicit and implicit methods under small sampling times, the implicit methods show less chattering in large sampling times, compared to the explicit ones. It indicates that the implicit methods can reduce the numerical chattering compared to their explicit counterparts, because by increasing the sampling time, chattering amplitude corresponding to the numerical chat-

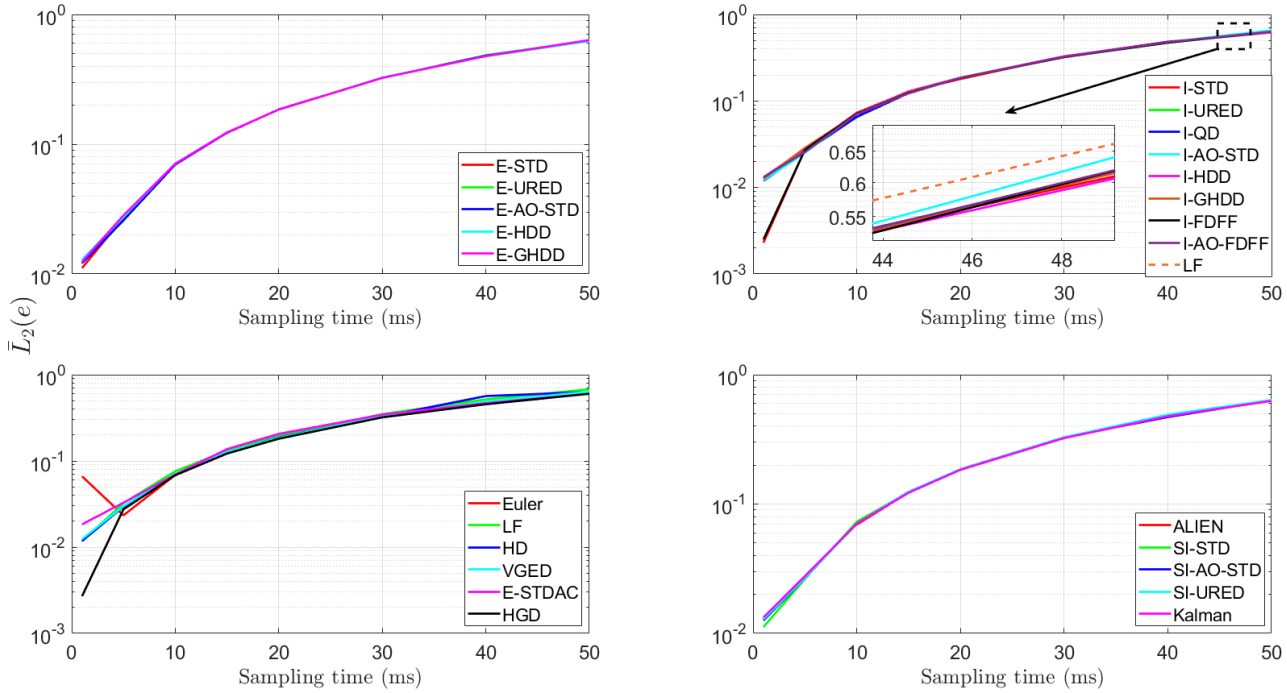


Figure 7: $\bar{L}_2$ for all methods

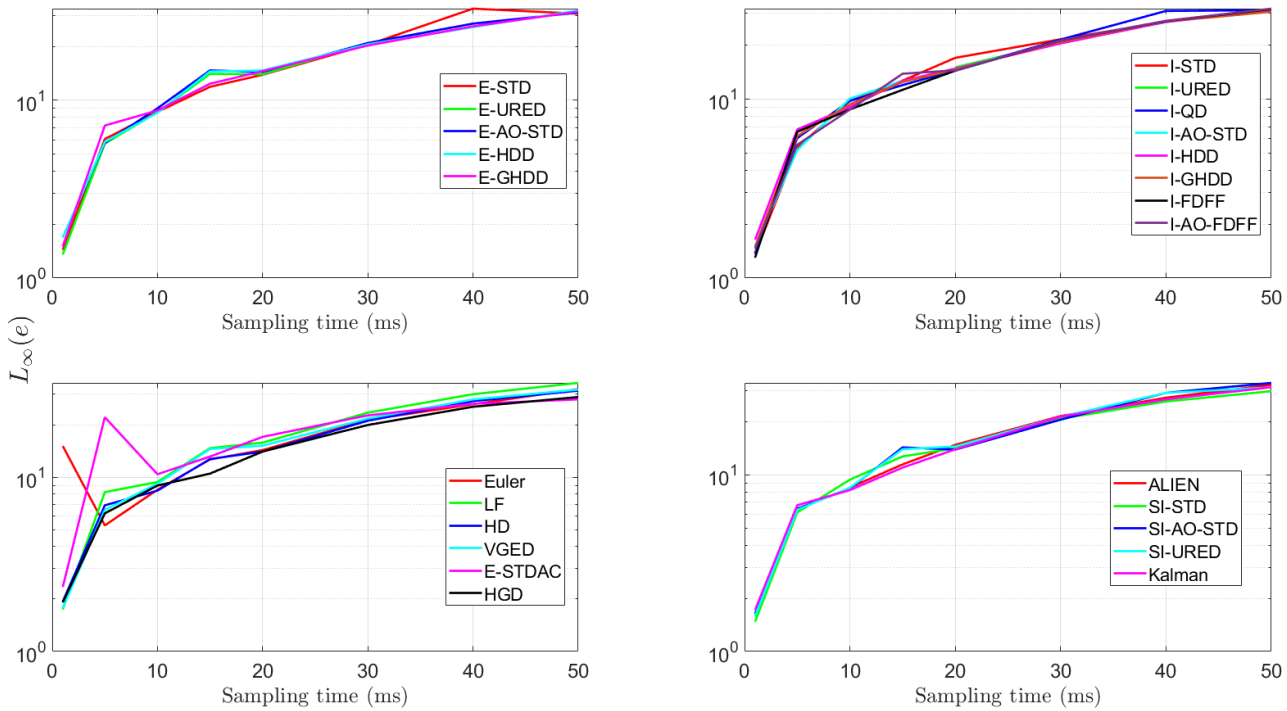


Figure 8: L_∞ for all methods

tering will be increased, and at the same time, the power of the noise will be decreased according to Nyquist–Shannon sampling theorem (since the high-frequency noise components cannot be recovered in the sampled measurements). In other words, by increasing the sampling time, the dominant chattering source would be numerical chattering. This validates **TR1** experimentally for the chosen setup and controller.

- According to Sec. 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 is in accordance with **TR2**.
- Based on the discussion presented in Sec. 4.6, the implicit methods show better insensitivity to the gains, which agrees with **TR3**.
- According to the results provided in Figs. 5 and 6, higher-order SMB differentiators, *e.g.*, AO-STD show better variation than that of the cascade configurations of the first-order counterparts, *e.g.*, STD.
- According to Fig. 7, in general, the implicit methods may provide better results than those of the LF.

5. Rotary inverted pendulum

A RIPS from QUANSER has been employed in this section as another case-study [1].

5.1. Mathematical modeling of the RIPS

The scheme of the RIPS is shown in Fig. 9. This mechanical system has two links: the rotary link and the pendulum link, which are indicated by the subscripts r and p, respectively, $m_r = 0.095$kg and $m_p = 0.024$kg are the weights of the rotary and pendulum links, $L_r = 0.085$m and $L_p = 0.129$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 upward position.

Equations of motion of the pendulum can be obtained using the Euler-Lagrange framework as explained in [57]. The state-space equations of the linearized system (about its unstable equilibrium point, i.e., $\theta = \dot{\theta} = \dot{\alpha} = 0, \alpha = 0$ and without considering the actuator dynamic) can be obtained as follows:

$$\dot{x} = Ax + Bu \tag{31a}$$

$$y = Cx + Du \tag{31b}$$

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:

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

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

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

[1] One may validate all the numerical and experimental results using the remote setup which is accessible via `http://valse-pendulum.lille.inria.fr:5000/`.

Table 5: Results for h=20ms 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
E-AO-STD	10^3	impracticability		
E-HDD	10^3	impracticability		
E-GHDD	10^3	impracticability		
h=20ms, with disturbance				

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

Method	$h = 0.2$ms	$h = 1$ms	$h = 2$ms	$h = 3$ms	$h = 4$ms	$h = 5$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						

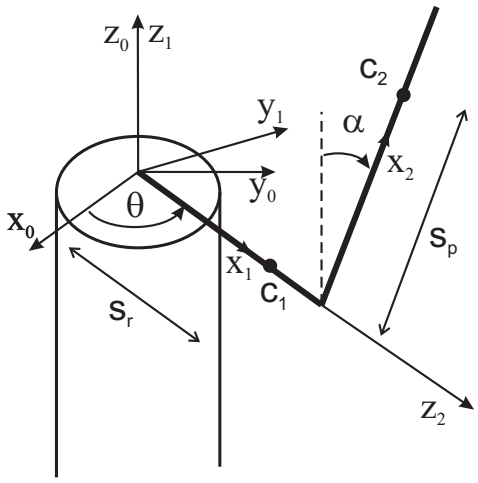


Figure 9: Scheme of the rotary inverted pendulum

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

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:

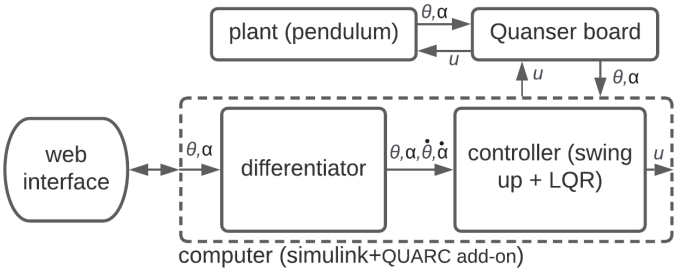


Figure 10: Closed-loop diagram of the RIPS

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

where u is the control signal which is automatically saturated to $[-10, +10]$ V by the laboratory setup to protect the motor, 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. The closed-loop diagram of the RIPS is shown in Fig. 10, and the controllers u_s and u_b are obtained in Secs. 5.2.1 and 5.2.2, respectively. A PC with an Intel Core i7 7700 processor working on 2.9GHz with 16 GB Memory has been used to implement the controller and the differentiators for the RIPS.

5.2.1. Swing-up control

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

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

where $\mu = 50\text{m/s/J}$ 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 $u_{sm} = 10$ V is the saturation factor. The term $\text{sgn}(\dot{\alpha}\cos(\alpha))$ is used to switch the sign of the control. Here, the sat function is defined as follows:

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

The energy of the pendulum is the sum of the kinetic (E_1) and potential (E_2) energies:

$$E = E_1 + E_2 \tag{37a}$$

$$E_1 = \frac{1}{2} J_p \dot{\alpha}^2 \tag{37b}$$

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

5.2.2. Balancing control

An LQR[1] controller ($u = G(x - x_r)$) is designed for (32) for the balancing as follows:

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

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

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.

5.4. Experiments

The remote setup which is used in this section is installed at INRIA Lille, France. To perform the experiments it is necessary to implement the controller and the observer in a compatible format and then to 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

The sampling time $h = 2$ ms is used by default on this setup. Tuning the parameters of the differentiators is one of the most challenging parts of this study (including previous works [1, 2]). For the EPS, the parameters obtained in the first part of the research [1] are directly used to implement the differentiators. Those parameters were assumed to be optimal for the input $I_1 : f(t) = \sin(t) + \tilde{n}(t)$ with $\tilde{n}(t)$ being a white noise with SNR=30dB. As it was seen in Sec. 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.*, $I_2 : f(t) = 4 \times 10^4(t^2 + 2t + 3)$ with quantization accuracy 0.001, is used to re-tune the parameters specifically for the RIPS (the parameters are listed in Table 3). 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 almost noise-free, 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 $t > 3$s is higher than $\pi/4$ rad, 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 make the tracking error converges into the interval $\alpha \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ rad for $t > 3$s. Otherwise, a differentiator leads to impracticability. This issue is further addressed in Sec. 5.4.2.

In summary, while EPS is a relatively slow system with significant noise, RIPS is a fast system with little 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

[1] Linear quadratic regulator is a well-defined controller (for instance see [58]) to minimize the objective function $J = \int (x^\top Q x + u^\top R u)\mathrm{d}t$, 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.

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 trajectories and stabilization precision. Moreover, 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 averaged results are provided in Table 7.

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 bandwidth 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 [1], 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. Consequently, 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 [1], 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 **TR4**.

5.4.2. Effect of the parameter tuning on the results

As it was mentioned before, the parameters are tuned based on the input $I_2 : f(t) = 4 \times 10^4(t^2 + 2t + 3)$ 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 $I_3 : f(t) = 1000 \sin(200t + \pi/4)$ with quantization (resolution=0.001) is considered for parameter tuning. The tuned parameters are provided in Table 3, and the corresponding results are summarized in Table 8.

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 7), are not practicable anymore. The reason is that according to **TR4**, 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 Table 3).

In summary, in spite of the fact that we do not propose in this article a systematic choice for the signal $f(t)$ to tune the parameters along the lines of [1, 2], we show that such a choice has to be done carefully depending on the closed-loop system dynamical behavior. This may pave the way towards the development of an engineer's guide for differentiator parameters tuning.

Generally speaking, comparing Tables 7 and 8, it is inferred that the differentiators behave as follows:

- **Euler, I-STD, I-URED, I-QD, I-AO-FDFF**: The results corresponding to the simulations and experiments are quite similar except for the values of $||u||_\infty$ where the experimental values are much higher than the simulation counterparts. Euler

Table 7: Average results based on 10 experiments when the parameters are tuned for I_2 (parameters are shown in Table 3)

	Simulations					Experiments				
Method	$\|\|e_\theta\|\|$	$\|\|e_\alpha\|\|$	$\|\|e_\theta\|\|_\infty$	$\|\|e_\alpha\|\|_\infty$	$\|\|u\|\|_\infty$	$\|\|e_\theta\|\|$	$\|\|e_\alpha\|\|$	$\|\|e_\theta\|\|_\infty$	$\|\|e_\alpha\|\|_\infty$	$\|\|u\|\|_\infty$
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-FDFF	0.1913	0.0099	0.3336	0.0086	0.0581	0.2565	0.0335	0.1136	0.0153	1.4766
I-AO-FDFF	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										

differentiator is, in theory, a pure differentiator with unlimited bandwidth. Other first-order differentiators, i.e., I-STD, I-URED, I-QD, as reported in [1, 2], just use a first filtration stage resulting in small time-delays and high-bandwidth responses. Hence, high-frequency components (including high-frequency noise) of the feedback can pass through these differentiators and enter the controller. This probably results in a large value of $||u||_\infty$ for the experiments as can be seen from Tables 7 and 8. I-AO-FDFF also behaves similarly to this group of differentiators.

- **LF**: This differentiator shows almost the same level of performance for both experiments and simulation as shown in Tables 7 and 8.

- **HD**: While this differentiator shows one of the best results for the simulations, it led to the worst experimental result. The reason for this observation is not clear. It is probably because of the large time-delay of this differentiator reported in [1, 2] which increases the time delay in the control loop.

- **I-AO-STD, SI-AO-STD, I-HDD, I-FDFF and ALIEN**: While these differentiators are practicable in Table 7, they lead to impracticability when their parameters are tuned based on the input I_3. Hence, the results corresponding to these differentiators are not provided in Table 8. Considering Table 7, the results for the simulations and experiments are quite consistent for this group of differentiators.

5.4.3. Results obtained for the pendulum system

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

- Unlike the implicit methods, the explicit SMB differentiators show too much chattering which led to

Table 8: Average results based on 10 experiments when the parameters are tuned for I_3 (parameters are shown in Table 3)

	Simulations					Experiments				
Method	$\|\|e_\theta\|\|$	$\|\|e_\alpha\|\|$	$\|\|e_\theta\|\|_\infty$	$\|\|e_\alpha\|\|_\infty$	$\|\|u\|\|_\infty$	$\|\|e_\theta\|\|$	$\|\|e_\alpha\|\|$	$\|\|e_\theta\|\|_\infty$	$\|\|e_\alpha\|\|_\infty$	$\|\|u\|\|_\infty$
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										
Performance: red<black<blue										

impracticability on the RIPS. This is in accordance with **TR1**.

- Solver-based implicit SMB differentiators (I-AO-STD, I-HDD, and I-URED) can also be implemented in real-time since they led to practicability in Table 7. It indicates that enough calculation resources, required by **TR2**, are available on the RIPS.

- As it can be seen from Table 8, 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 generally shows a larger transient time (**TR4**).

- According to Table 7, a higher-order differentiator may present better response (in case of $||e_\alpha||_\infty$ and $||u||_\infty$) than a first-order one (compare I-STD and I-AO-STD in Table 7).

6. General conclusions

Two laboratory setups, *i.e.*, an electro-pneumatic system and a rotary-inverted pendulum, are used in this study to implement 25 known differentiators on practical closed-loop control systems. These setups are representative enough and 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:

- Euler differentiator has the simplest structure without any tunable parameter. This is basically a pure differentiator that differentiates both the base signal as well as the measurement noise without any filtration. Surprisingly, this differentiator is practicable with comparable performances to other sophisticated methods for almost all conditions. However, this observation should not be misleading since, on any laboratory setup like the ones used in this work, there are some filtration stages implemented on the hardware or the software by the manufacturer (espe-

cially on analog to digital conversion blocks) which are not controlled by the user. Hence, the Euler differentiator should always be implemented with special attention.

- The LF shows one of the best results despite its simple structure and easy parameter tuning. The main difference between the LF and the sliding-mode-based (SMB) differentiators is that while the SMB differentiators can converge to the exact differentiation in noise-free case (see [1, 2]), the LF always shows a phase-lag for finite gains because of its linear structure. However, in practice, converging to the exact differentiation of the polluted signal should be avoided to vanish the noise differentiation. In such a condition, all methods including LF and the SMB differentiators exhibit phase-lag.

- The implicit differentiators can supersede the explicit ones in terms of chattering. Comparing the results, it can be concluded that the implicit methods always present smaller variations for large sampling times, 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 **TR1**. It should be noted that this conclusion has been drawn with the parameters listen in Table 3.

- The solver-based implicit differentiators (I-AO-STD, I-HDD, I-GHDD, I-URED) could 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 **TR2**.

- The implicit methods present better gain insensitivity, especially in low noise conditions. In other words, one may select a larger sampling gain for the implicit differentiators compared to the explicit counterparts. This follows **TR3**.

- While all differentiators are tuned in identical conditions, the higher-order differentiators (I-AO-STD, I-HDD, GHDD) may lead to impracticability for the RIPS since, compared to the first-order counterparts (I-STD, I-URED, I-QD), they present longer transient time. This is in accordance with **TR4**.

- 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.

- For the EPS, the implicit SMB differentiators provide a better output tracking than the LF.

In this work, 25 differentiators with different structures are implemented experimentally and their results have been compared. These differentiators have different structures containing nonlinear and discontinuous (set-valued) terms which make the parameter tuning difficult. Moreover, the uncertainties and measurement noise on the laboratory setup cannot be taken into account while tuning the parameters. Hence, it is not clear how to find the optimal parameters of all differentiators. An optimization algorithm (proposed in [1, 2]) has been employed to find the optimal parameters in specified intervals. In fact, all differentiators are uniformly optimized based on some selected input signals, *i.e.*, I_1, I_2, and I_3 in open-loop configuration. These inputs are selected by trial and error to observe satisfactory responses from all differentiators. *However, the way that the optimization algorithm has been used is still questionable since it is not clear whether or not this procedure will lead to the optimal parameters for the closed-loop systems. Hence, the selection of these inputs should be addressed more clearly in the future.*

In other words, an appropriate input used for the parameter tuning should be selected based on the noise characteristics and the closed-loop dynamics rather than trial

and error. However the design of a suitable optimization algorithm to achieve such a goal, is thought to be a non-trivial work which is left for the future. Nevertheless the results obtained in this paper are encouraging (since they show that modifying the base function can indeed strongly improve the results) and may pave the way towards an automatic and optimal selection of base functions. Moreover, the parameters of the controllers should also be tuned alongside the differentiators' parameters, and this increases the complexity of the problem. This can be considered as the topic of future works 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. As a future study, the differentiators may also be implemented on other laboratory setups such as robotic systems [59] and flexible manipulators [60]. Furthermore, more controllers can be implemented alongside the differentiators to make the conclusions more comprehensive.

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

References

[1] M. R. Mojallizadeh, B. Brogliato, V. Acary, Time-discretizations of differentiators: Design of implicit algorithms and comparative analysis, International Journal of Robust and Nonlinear Control 31 (16) (2021) 7679–7723.

[2] M. R. Mojallizadeh, B. Brogliato, V. Acary, Discrete-time differentiators: design and comparative analysis, Tech. rep., INRIA Grenoble-Alpes, https://hal.inria.fr/hal-02960923 (2020).

[3] J. E. Carvajal-Rubio, J. D. Sánchez-Torres, M. Defoort, M. Djemai, A. G. Loukianov, Implicit and explicit discrete-time realizations of homogeneous differentiators, International Journal of Robust and Nonlinear Control 31 (9) (2021) 3606–3630.

[4] J. E. Carvajal-Rubio, A. G. Loukianov, J. D. Sánchez-Torres, 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), Mexico City, 2019, pp. 1–6, DOI: 10.1109/ICEEE.2019.8884567.

[5] Y. Wang, G. Zheng, D. Efimov, W. Perruquetti, Differentiator application in altitude control for an indoor blimp robot, International Journal of Control 91 (9) (2018) 2121–2130.

[6] H. Ahmed, H. Ríos, B. Ayalew, Y. Wang, Second-order sliding-mode differentiators: an experimental comparative analysis using Van der Pol oscillator, International Journal of Control 91 (9) (2018) 2100–2112.

[7] A. Levant, Chattering analysis, IEEE Transactions on Automatic Control 55 (6) (2010) 1380–1389.

[8] Z. Galias, X. Yu, Discretization effect on equivalent control-based multi-input sliding-mode control systems, IEEE Transactions on Automatic Control 53 (6) (2008) 1563–1569.

[9] X. Yu, B. Wang, Z. Galias, G. Chen, Euler's discretization of single input sliding-mode control systems, IEEE Transactions on Automatic Control 52 (9) (2007) 1726–1730.

[10] Y. Yan, Z. Galias, X. Yu, C. Sun, Euler's discretization effect on a twisting algorithm based sliding mode control, Automatica 68 (2016) 203–208.

[11] Z. Galias, X. Yu, Complex discretization behaviors of a simple sliding-mode control system, IEEE Transactions on Circuits and Systems II: Express Briefs 53 (2006) 652–656.

[12] O. Huber, B. Brogliato, V. Acary, A. Boubakir, F. Plestan, B. Wang, Experimental results on implicit and explicit time-discretization of equivalent control-based sliding mode control, in: L. Fridman, J. Barbot, F. Plestan (Eds.), Recent Trends in Sliding Mode Control, IET, 2016, pp. 207–235.

[13] O. Huber, V. Acary, B. Brogliato, F. Plestan, Implicit discrete-time twisting controller without numerical chattering: Analysis and experimental results, Control Engineering Practice 46 (2016) 129–141.

[14] O. Huber, V. Acary, B. Brogliato, Lyapunov stability analysis of the implicit discrete-time twisting control algorithm, IEEE Transactions on Automatic Control 65 (6) (2020) 2619–2626.

[15] B. Wang, B. Brogliato, V. Acary, A. Boubakir, F. Plestan, Experimental comparisons between implicit and explicit implementations of discrete-time sliding mode controllers: Toward input and output chattering suppression, IEEE Transactions on Control Systems Technology 23 (5) (2015) 2071–2075.

[16] X. Xiong, R. Kikuuwe, M. Yamamoto, Backward-Euler discretization of second-order sliding mode control and super-twisting observer for accurate position control, in: ASME 2013 Dynamic Systems and Control Conference, Palo Alto, USA, 2013, pp. 1–8.

[17] X. Xiong, R. Kikuuwe, S. Kamal, S. Jin, Implicit-Euler implementation of super-twisting observer and twisting controller for second-order systems, IEEE Transactions on Circuits and Systems II: Express Briefs 67 (11) (2020) 2607–2611.

[18] R. Kikuuwe, S. Yasukouchi, H. Fujimoto, M. Yamamoto, Proxy-

based sliding mode control: A safer extension of PID position control, IEEE Transactions on Robotics 26 (4) (2010) 670–683.

[19] B. Brogliato, A. Polyakov, Digital implementation of sliding-mode control via the implicit method: A tutorial, International Journal of Robust and Nonlinear Control 31 (9) (2021) 3528–3586.

[20] O. Huber, V. Acary, B. Brogliato, Lyapunov stability and performance analysis of the implicit discrete sliding mode control, IEEE Transactions on Automatic Control 61 (10) (2016) 3016–3030.

[21] A. Levant, M. Livne, Robust exact filtering differentiators, European Journal of Control 55 (2020) 33–44.

[22] S. Koch, M. Reichhartinger, M. Horn, L. Fridman, Discrete-time implementation of homogeneous differentiators, IEEE Transactions on Automatic Control 65 (2) (2020) 757–762.

[23] M. Livne, A. Levant, Proper discretization of homogeneous differentiators, Automatica 50 (8) (2014) 2007 – 2014.

[24] E. Cruz-Zavala, J. A. Moreno, L. M. Fridman, Uniform robust exact differentiator, IEEE Transactions on Automatic Control 56 (11) (2011) 2727–2733.

[25] S. Jin, R. Kikuuwe, M. Yamamoto, Real-time quadratic sliding mode filter for removing noise, Advanced Robotics 26 (8-9) (2012) 877–896.

[26] A. Levant, Robust exact differentiation via sliding mode technique, Automatica 34 (3) (1998) 379–384.

[27] M. Reichhartinger, S. Spurgeon, An arbitrary-order differentiator design paradigm with adaptive gains, International Journal of Control 91 (9) (2018) 2028–2042.

[28] S. Koch, M. Reichhartinger, Discrete-time equivalent homogeneous differentiators, in: 2018 15th International Workshop on Variable Structure Systems (VSS), 2018, pp. 354–359.

[29] M. Reichhartinger, S. Koch, H. Niederwieser, 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.

[30] H. K. Khalil, Nonlinear systems, Pearson Education, 2015.

[31] G. Byun, 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. 18 (2020) 3001–3014.

[32] R. Kikuuwe, R. Pasaribu, G. Byun, A first-order differentiator with first-order sliding mode filtering, IFAC-PapersOnLine 52 (16) (2019) 771–776.

[33] J.-J. E. Slotine, J. K. Hedrick, E. A. Misawa, On Sliding Observers for Nonlinear Systems, Journal of Dynamic Systems, Measurement, and Control 109 (3) (1987) 245–252.

[34] M. Ghanes, J. P. Barbot, L. Fridman, A. Levant, R. Boisliveau, A new varying gain exponent based differentiator/observer: an efficient balance between linear and sliding-mode algorithms, IEEE Transactions on Automatic Control 65 (12) (2020) 5407–5414.

[35] A. Levant, Higher-order sliding modes, differentiation and output-feedback control, International Journal of Control 76 (9-10) (2003) 924–941.

[36] E. Cruz-Zavala, J. A. Moreno, Levant's arbitrary-order exact differentiator: A Lyapunov approach, IEEE Transactions on Automatic Control 64 (7) (2019) 3034–3039.

[37] E. Cruz-Zavala, J. A. Moreno, Lyapunov functions for continuous and discontinuous differentiators, IFAC-PapersOnLine 49 (18) (2016) 660–665.

[38] A. Levant, Filtering differentiators and observers, in: 2018 15th International Workshop on Variable Structure Systems (VSS), 2018, pp. 174–179.

[39] A. Levant, Homogeneous filtering and differentiation based on sliding modes, in: 2019 IEEE 58th Conference on Decision and Control (CDC), Nice, France, 2019, pp. 6013–6018.

[40] Z. Lv, S. Jin, X. Xiong, J. Yu, A new quick-response sliding mode tracking differentiator with its chattering-free discrete-time implementation, IEEE Access 7 (2019) 130236–130245.

[41] C. Vazquez, S. Aranovskiy, L. B. Freidovich, L. M. Fridman, Time-varying gain differentiator: A mobile hydraulic system case study, IEEE Transactions on Control Systems Technology 24 (5) (2016) 1740–1750.

[42] J. A. Moreno, Exact differentiator with varying gains, International Journal of Control 91 (9) (2018) 1983–1993.

[43] M. Ghanes, J. P. Barbot, L. Fridman, 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.

[44] M. Ghanes, J. P. Barbot, L. Fridman, A. Levant, A second order sliding mode differentiator with a variable exponent, in: 2017 American Control Conference (ACC), Seattle, USA, 2017, pp. 3300–3305.

[45] L. K. Vasiljevic, H. K. Khalil, Error bounds in differentiation of noisy signals by high-gain observers, Systems & Control Letters 57 (10) (2008) 856–862.

[46] W. Perruquetti, 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.

[47] M. Mboup, C. Join, M. Fliess, Numerical differentiation with annihilators in noisy environment, Numerical Algorithms 50 (4) (2009) 439–467.

[48] M. Mboup, S. Riachy, Frequency-domain analysis and tuning of the algebraic differentiators, International Journal of Control 91 (9) (2018) 2073–2081.

[49] O. Huber, V. Acary, B. Brogliato, Lyapunov stability analysis of the implicit discrete-time twisting control algorithm, IEEE Transactions on Automatic Control 65 (6) (2020) 2619–2626.

[50] B. Brogliato, A. Polyakov, D. Efimov, The implicit discretization of the super-twisting sliding-mode control algorithm, IEEE Transactions on Automatic Control 65 (8) (2020) 3707–3713.

[51] 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 2 (1994) 105.

[52] C. C. Y. Dorea, Expected number of steps of a random optimization method, Journal of Optimization Theory and Applications 39 (2) (1983) 165–171.

[53] M. Taleb, A. Levant, F. Plestan, Pneumatic actuator control: Solution based on adaptive twisting and experimentation, Control Engineering Practice 21 (5) (2013) 727–736.

[54] Y. Shtessel, M. Taleb, F. Plestan, A novel adaptive-gain supertwisting sliding mode controller: Methodology and application, Automatica 48 (5) (2012) 759–769.

[55] M. Taleb, A. Levant, F. Plestan, Twisting algorithm adaptation for control of electropneumatic actuators, in: 2012 12th International Workshop on Variable Structure Systems, 2012, pp. 178–183.

[56] M. Belgharbi, S. Sesmat, S. Scavarda, D. Thomasset, Analytical model of the flow stage of a pneumatic servo-distributor for simulation an nonlinear control, in: SICFP, Vol. 2, Tampere, Finland, 1999, pp. 847–860.

[57] X. Yang, X. Zheng, Swing-up and stabilization control design for an underactuated rotary inverted pendulum system: Theory and experiments, IEEE Transactions on Industrial Electronics 65 (9) (2018) 7229–7238.

[58] D. E. Kirk, Optimal control theory: an introduction, Courier Corporation, 2004.

[59] W. He, C. Xue, X. Yu, Z. Li, C. Yang, Admittance-based controller design for physical human–robot interaction in the constrained task space, IEEE Transactions on Automation Science and Engineering 17 (4) (2020) 1937–1949.

[60] W. He, H. Gao, C. Zhou, C. Yang, Z. Li, Reinforcement learning control of a flexible two-link manipulator: An experimental investigation, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2020) 1–11.