HAL open science

A survey on the discrete-time differentiators in closed-loop control systems: experiments on an electro-pneumatic system

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

▶ **To cite this version:**

Mohammad Rasool Mojallizadeh, Bernard Brogliato, Andrey Polyakov, Sevarajan Subischa, Loïc Michel, et al.. A survey on the discrete-time differentiators in closed-loop control systems: experiments on an electro-pneumatic system. Control Engineering Practice, 2023, 136 (105546), pp.1-23. 10.1016/j.conengprac.2023.105546 . hal-04534678

HAL Id: hal-04534678
https://hal.science/hal-04534678

Submitted on 5 Apr 2024

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.

Distributed under a Creative Commons Attribution 4.0 International License

A survey on the discrete-time differentiators in closed-loop control systems: experiments on an electro-pneumatic system

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] *Nantes Université, École Centrale Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France*
[e] *ENSEA, QUARTZ, 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, an electro-pneumatic system has been employed as the case study to implement 25 different differentiators including the exact differentiators and linear filters. 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, 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 on digital processors, their time-discretization should be addressed clearly.

Some differentiators employ discontinuous (set-valued) terms to improve their exactness. Such terms can be po-

*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)

tentially 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, 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 implicit discretization as reported by [12, 13, 14, 15, 19, 20]. This is 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 Feature 1 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 Feature 1 using practical experiments.

Another issue corresponding to 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, 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 [2] Sec. 5.8.5, and [3]). However, the amount of the required calculation resources has not been reported. This is considered as Feature 2 in this manuscript and will be evaluated on the laboratory setup.

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 Feature 3 in this study, and the gain sensitivity of both implicit and explicit schemes will be studied on the setup 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. Remark that this is proved when the derivative of order $n+1$ of the input signal vanishes. This result is named Feature 4 in this study and it will be studied experimentally. In addition, that the transient response of a differentiator depends on several other factors, including the initial conditions and the parameters.

Following the literature, Feature 1 and Feature 3 have been evaluated experimentally for the SMCs before in [12, 20, 13, 15]. However, the validity of the mentioned results for the differentiators has not been experimentally addressed yet. 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,

[1]Chattering refers to finite-amplitude oscillations where the frequency depends on the sampling time. While many factors including measurement noise and dynamical uncertainties can lead to *chattering* [7], the *numerical chattering* refers to a specific kind of chattering that only appears due to the time-discretization of the set-valued terms [8, 9, 10, 11].

and several remarks will be drawn to allow one to select the most appropriate differentiator in practical control systems. To this end, a laboratory setup, *i.e.,* an *electro-pneumatic setup* (EPS) has been used to implement the 25 differentiators which have been studied in [1, 2].

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 experimental results are presented in Sec. 4. Finally, general conclusions and remarks are provided in Sec. 5. Remark that the colors red, black, and blue are used in some tables to help showing the worst, moderate and the best results, respectively.

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 $0, \dots, n-1$. Remark that zero-order differentiation means the estimation of the noise-free input. Some differentiators possess this capability.
- 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]:

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

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 before being 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 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:

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

where the notation is as before, L is a tuning parameter, the parameters $\lambda_i, i = 0, \dots, n$ 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)

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

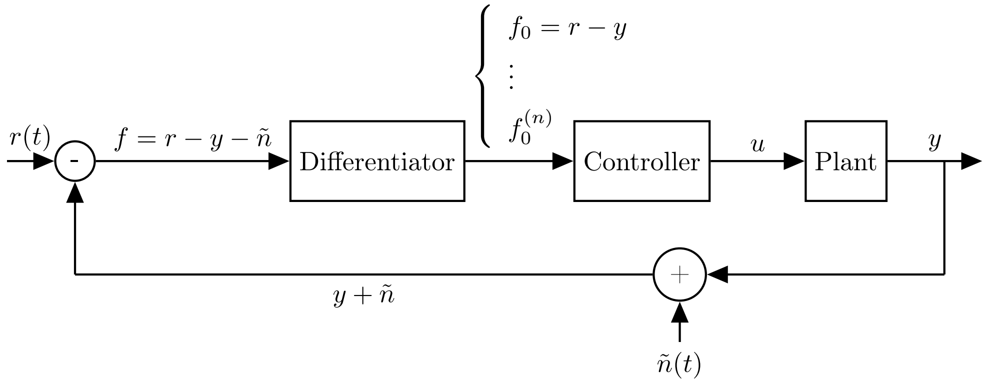

Figure 1: A typical control loop.

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

[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] deal 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

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

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

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

where $f_f(t)$ corresponds to high-frequency components of the input. 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 second-order HGD will be considered as follows (see [5, 46]).

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

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. The ALIEN differentiator is given as [48]:

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

where $z^{(n)}(t)$ 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 $h > 0$ 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, 20, 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 $z_i, i = 0, \dots, n$ 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 Kalman's differentiator. To design 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 second-order Kalman differentiation are as follows (see [51] for the algorithm as well as the preliminary equations of the Kalman filter).

$$A = \begin{bmatrix} 1 & h & \frac{h^2}{2} \\ 0 & 1 & h \\ 0 & 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}. \tag{12}$$

Remark that, unlike the standard Kalman filter, the Kalman differentiator has one parameter, denoted by R, which indicates the power of the input noise (see the complete algorithm in [1]). The other parameter that exists in the standard Kalman filter, *i.e.*, the process noise, does not exist in the Kalman differentiator since the process is composed of a chain of integrators, where the only source of perturbation in the process dynamics is caused by the truncation error corresponding to the Euler discretization. Hence, the process noise depends on the sampling time rather than being a tunable parameter [21, 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:

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

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

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

As it was mentioned, 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 the literature. In other words, the absence of a prefix indicates that the method is implemented using the explicit discretization scheme ($\alpha = 0$ in (10)).

In order to provide a fair comparison among all the mentioned differentiators, it is necessary to tune their parameters. While there is no standard or widely accepted way to tune the differentiators' parameters on laboratory setups, the parameters have been tuned based on the experiments made on the closed-loop system, for $h = 15$ms, according to the following steps as suggested in [52]:

1. The differentiators have been implemented in the closed-loop system on the laboratory setup under the conditions explained in Sec. 4.3.
2. The waveforms of the system, *e.g.*, the zero-order differentiation (z_0) (noise-free input signal) estimated by the differentiator, and the signal to be differentiated are recorded. Remark that some differentiators are able to estimate the zero-order differentiation by themselves. For the other differentiators, we integrated the first-order differentiation to obtain the zero-order differentiation. In addition, for the cascade configuration, where two first-order differentiation blocks are used, we calculated the double integration of the second-order differentiation to calculate the zero-order estimation used to tune the second block.
3. The zero-order differentiation has been compared with the input signal, and it is tried to tune the parameters in an empirical way such that the zero-order differentiation converges to the input signal without chattering. For most of the differentiators, increasing the gains, up to a limit, improves the convergence. However, with oversized gains, the zero-order differentiation usually shows chattering caused by the numerical discretization or input noise. Hence, several experiments have been done with different empirically selected parameters to observe the best convergence of the

zero-order differentiation to the input signal without chattering. The waveforms corresponding to the zero-order differentiation, and measured position for all differentiators with the tuned parameters listed in Table 3 are shown in Fig. 2.

According to Fig. 2, the convergence of all differentiators, except ALIEN, E-STDAC, and Kalman are visible in these figures. it is tried to change the parameters of these differentiators such that the zero-order differentiation converges to the input signal. However, it was not possible because of the following reasons:

- **ALIEN**: In this method, the differentiation of the input signal is calculated by a series of integrations rather than direct differentiation. While this method is known for its robustness to noise, it usually needs a very great sampling frequency compared to the frequency bandwidth of the input signal. Hence, for such a large sampling time $h = 15$ms, the zero-order differentiation does not converge to the input signal. Moreover, since the first and second-order ALIEN differentiators do not provide zero-order differentiation, we used an integration block to recover the zero-order differentiation. However, it seems that an initial value should be considered for the integration (which is unknown) to see the convergence since the zero-order differentiation follows the input signal pattern with a bias. It should be mentioned that the ALIEN differentiator is usually employed in applications where the sampling period can be considered as a tuning parameter. In fact, the sampling period should be large enough for the integrators to be able to filter the noise and small enough for the delay to be acceptable. Hence, the ALIEN differentiator does not show its full potential abilities in the considered scenario where the sampling time is not a design parameter.

- **Kalman**: We tried to tune the parameter of this differentiator for the convergence. However, perfect convergence did not happen. By decreasing the tuning parameter, it is possible to improve the convergence, However, after a certain value, the closed-loop system leads to instability.

- **E-STDAC**: This is an adaptive differentiator where the gains are calculated adaptively which adds an extra dynamic to the system. We did our best to tune the parameters of this differentiator, however, as can be seen in Fig. 2, sometimes there is a significant amount of difference between the input signal and the zero-order differentiation, which is probably caused by the adaptation mechanism that leading to small gains for some specific times.

Remark 2. *Parameter tuning is still an open problem and there is not a widely accepted way to do that. As it was mentioned, the parameters have been obtained in this work using an empirical procedure explained above. The parameters of the same differentiators have also been tuned based on some optimization algorithms using the numerical simulations in [53] and the corresponding parameters are provided there (see Table 3 in [53]). While the optimization algorithm used in [53] leads to different parameters, the conclusions drawn from the comparative analysis with the parameters in [53] are in agreement with the ones obtained in this paper. we have no clear idea about the reason why both sets of differentiators gains obtained from two different tuning processes, yield similar closed-loop results. It is therefore much too early to draw any kind of general conclusions about such an a priori surprising result. More analysis and more experiments are needed in the future to bring some elements of answer.*

Remark 3. *As it was seen, parameter tuning is not unique and may be done in several different manners. However, all of them exhibit different kinds of drawbacks. For instance, numerical simulations based on the closed-loop model may be used to tune the parameters. However, due to the stochastic noise behavior, such methods do not usually lead to the optimal parameters for the real system. Another approach could be measuring the position and using it as a reference to tune the parameters offline. However, this method also is not efficient since it is not possible to distinguish between the real position and the measurement noise.*

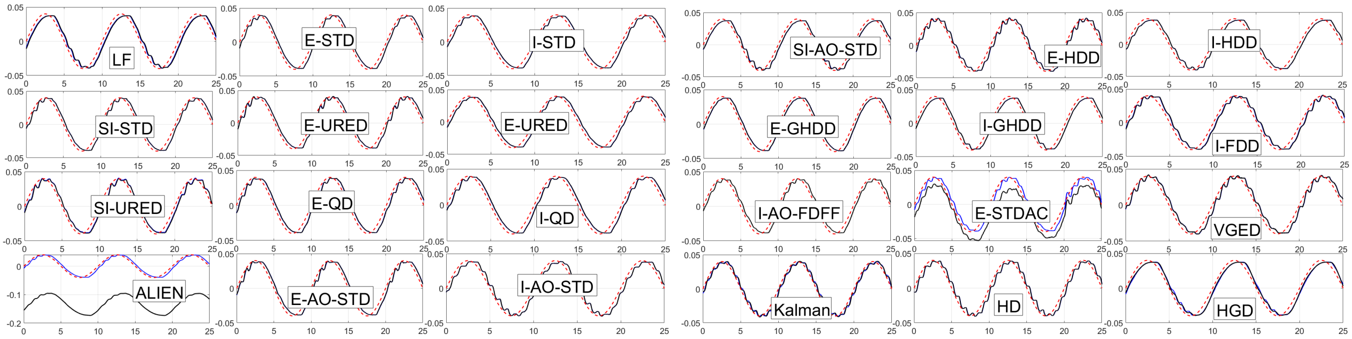

Figure 2: Waveforms corresponding to the zero-order differentiation (solid black), and measured position (solid blue) for all differentiators with the tuned parameters listed in Table 3. Note that the red dashed line corresponds to the reference position. Moreover, the x and y axes denote the time (s) and position (m), respectively.

4. The electropneumatic setup (EPS)

The EPS which is considered in this study is shown in Fig. 3. 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 necessary force to control the position of the jack, while the other one is to emulate disturbances that appear during real operation.

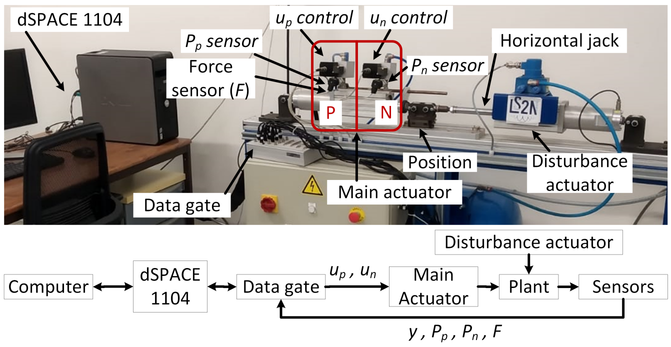

Figure 3: Electropneumatic setup.

The main actuator is double-acting, controlled by two servo-distributors with two chambers named P and N as indicated in Fig. 3. The block diagram corresponding to the closed-loop system of the EPS is shown in Fig. 4. A DS1104 board has been employed to implement the controller and the differentiator where a 64-bit MPC8240 processor working on 250 MHz and 32 MB RAM are available.

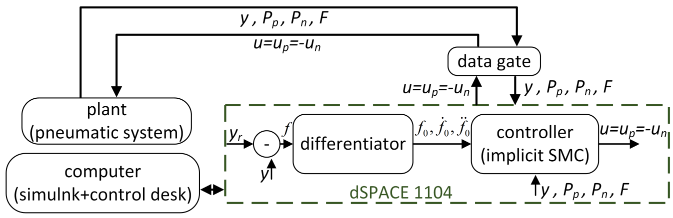

Figure 4: Closed-loop diagram of the EPS.

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 [54, 55, 56]:

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

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, $T = 293.15$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 [57]. 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 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 [54]. 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 digital hardware, a discretization method should be used. This topic has been studied in [12, 15], and it is concluded that 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 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

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

- **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. 5. 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 second-order differentiations. The differentiators are implemented on the EPS as follows:

- Some differentiators (Euler, LF, URED, 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 (Kalman, AO-STD, HGD, 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. 5 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.

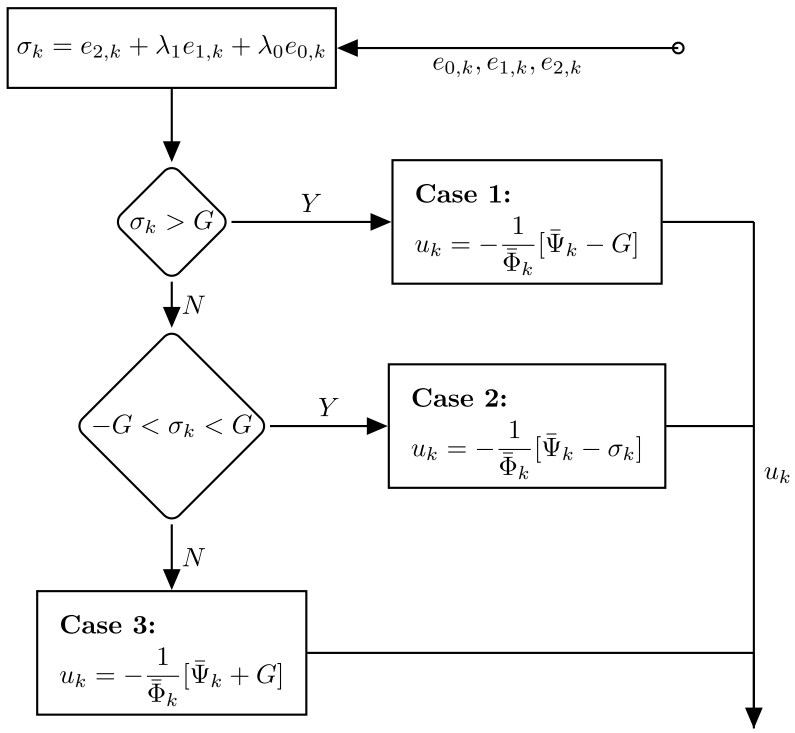

Figure 5: Flowchart of the first-order implicit SMC.

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 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.
- All higher-order differentiators are implemented in second-order configuration ($n = 2$).

Unlike 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. A smaller value indicates better output tracking. This cost function is defined as follows:

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

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

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

Smaller variation indicates less chattering.

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. The I-URED 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 $h = 15$ms. Hence, by decreasing the sampling time, this differentiator led to impracticability which indicates its sensitivity to the parameters. Note that by retuning the ALIEN's parameters for smaller sampling times, it may lead to practicability and good performances for smaller sampling times as well. However, such retuning has been avoided to show the robustness of the differentiators against their parameters. The performances for the impracticable cases are not provided in the next sections.

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

Method	Parameters
Euler	No parameter
LF	c=(20, 20)
E-STD	L=(0.5, 0.5)
I-STD	L=(1, 1)
SI-STD	L=(0.2, 0.2)
E-URED	L=(0.3, 0.3), μ=(21, 21)
I-URED	L=(1, 1), μ=(1000, 1000)
E-QD	F=(1, 1), α= (2, 2)
I-QD	F=(2, 2), α=(0.8, 0.8)
ALIEN	$\bar{T}$=(0.5, 0.9), κ=(4, 1),μ=(1, 1)
VGED	μ=(2, 2), τ=(1.33, 1.33), ω_c=(2π, 2π), q=(1, 1)
SI-URED	L=(100, 100), μ=(1000, 1000)
E-STDAC	α=(0.5, 0.5), ϵ=(10^{-6}, 10^{-6})
I-FDFF	ω_s=(10, 10), ω_f=(100, 100), ρ=(50, 50), γ=(2×10^{-4}, 2×10^{-4})
I-AO-FDFF	F=38, ϵ=19, ω_s=3, ω_f=63, α_1= 457, ρ=88
HD	r=3
E-AO-STD	L=3
I-AO-STD	L=3
SI-AO-STD	L=1
E-HDD	L=3
E-GHDD	L=8
I-HDD	L=3
I-GHDD	L=6
Kalman	$R = 1\times10^{-3}$
HGD	L= 2.5
The first and the second parameters inside each parenthesis correspond to the parameters of the first and the second blocks of cascade implementations.	

Table 4: Impracticable experiments.

Method	Impracticability condition
I-URED	$h < 5$ms
ALIEN	$h < 10$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. 3 and 5) under different sampling times is presented in Fig. 6. Comparing the explicit and the implicit methods in Fig. 6 (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.*

In Fig. 6 (D), ALIEN shows small variations (comparable to the implicit methods) for large sampling times. However, the values for small sampling times are not provided, since ALIEN is not practicable for $h < 10$ms. Considering the variations of the cascade configurations (STD, STDAC, URED) in Fig. 6, one can see that 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 $h > 10$ms, E-STD and SI-STD show the worst variations (Fig. 6 (A) and (D)).

For small sampling times, Euler is the worst differentiator. Moreover, while one may expect good performances from the VGED differentiator, because of its adaptation mechanism, it does not show an appropriate response in this specific application because of the cascade configuration. Notice that a higher-order version of this differentiator has been recently proposed in [58] that may improve its performance for such applications. It is worth to consider it in future comparisons. Another interesting observation is that Kalman's differentiator shows one of the best variations, comparable with the implicit methods.

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. 7. 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 Feature 1 on the chosen setup and control algorithm.

The $\bar{L}_2$ norms of the methods for different sampling times are provided in Fig. 8. It can be seen that, for all methods, increasing the sampling time highly affects the $\bar{L}_2$ norm. Concerning Fig. 8, all methods show almost the same evolution, except for the Euler method which shows the worst $\bar{L}_2$ for $h < 5$ms. Moreover, the ALIEN shows larger $\bar{L}_2$ for larger sampling times because its parameters need to be tuned for larger sampling times as well. Another observation is that the performances of the LF are comparable to the implicit SMB differentiators, in terms of variation, L_2 and L_∞, as can be seen in Figs. 6, 8 and 9.

While there is not a noticeable difference between the variation of the implicit differentiators and LF (compare Fig. 6 (B), (C)), according to Fig. 8 (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. 9. Apart from the Euler algorithm, all differentiators almost show the same behavior when increasing the sampling time.

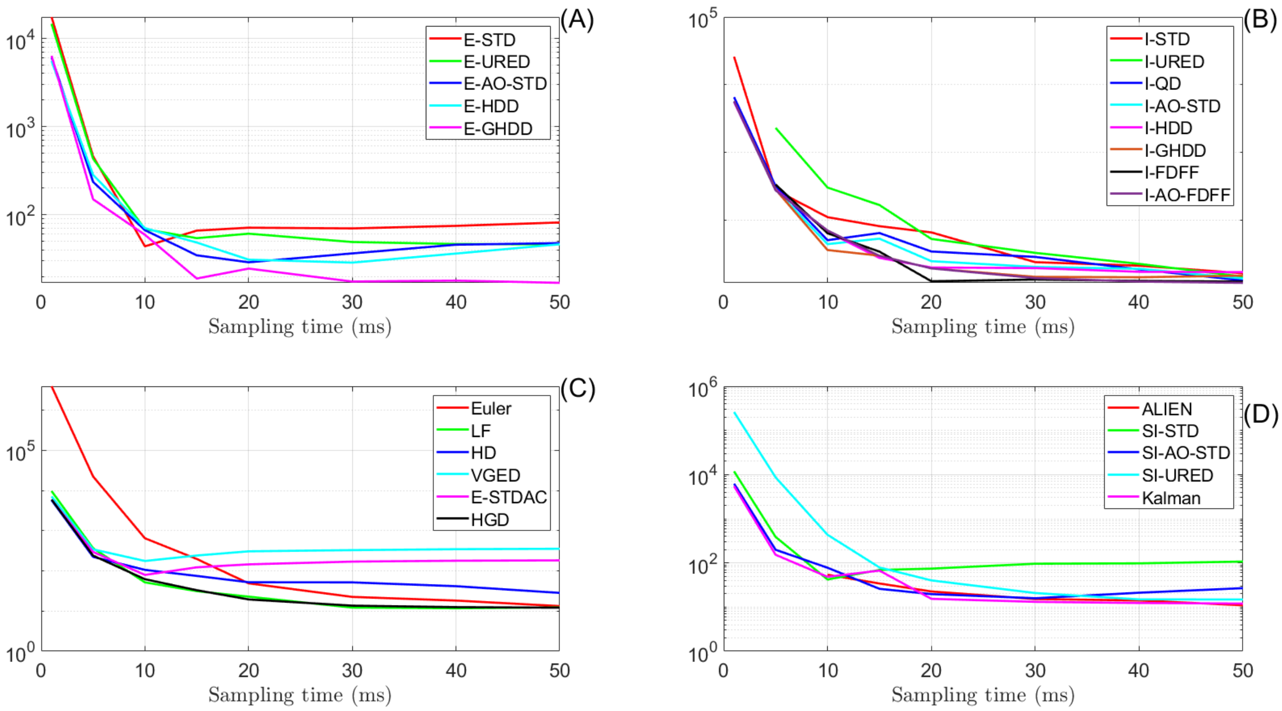

Figure 6: Variation of the control signal $u(t)$ under different sampling times.

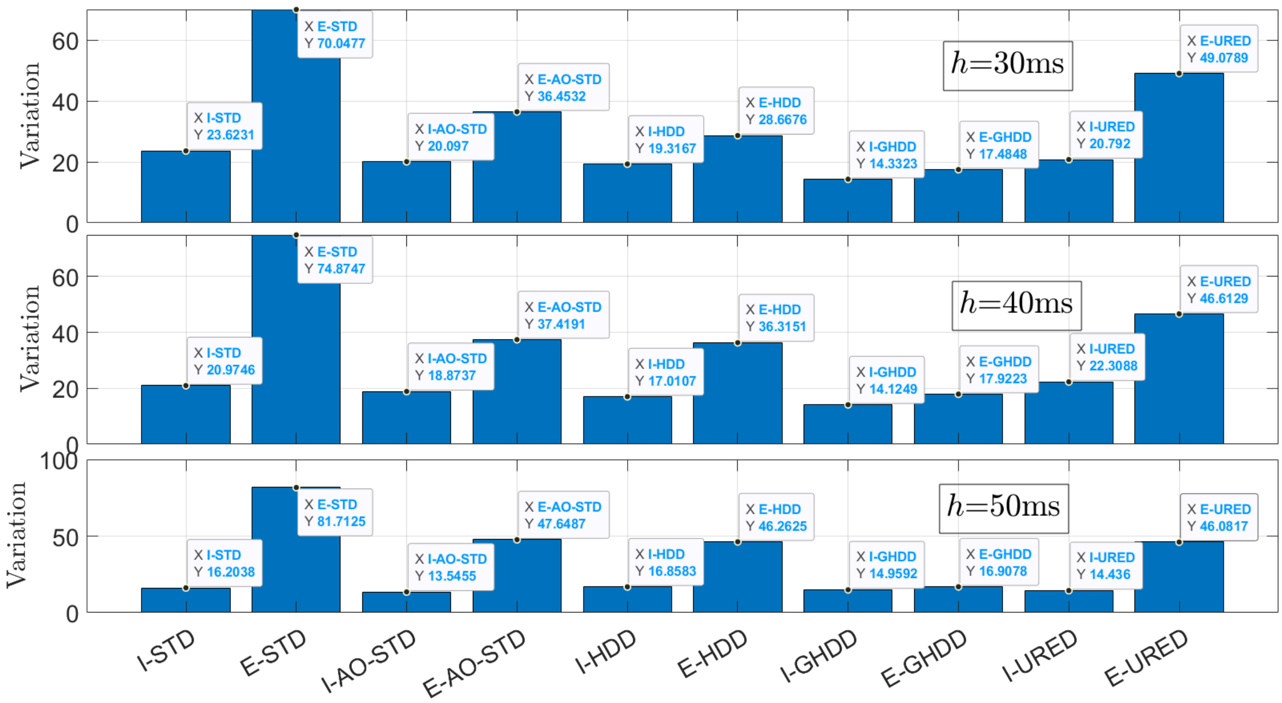

Figure 7: Comparison of the total variations for the implicit and explicit schemes.

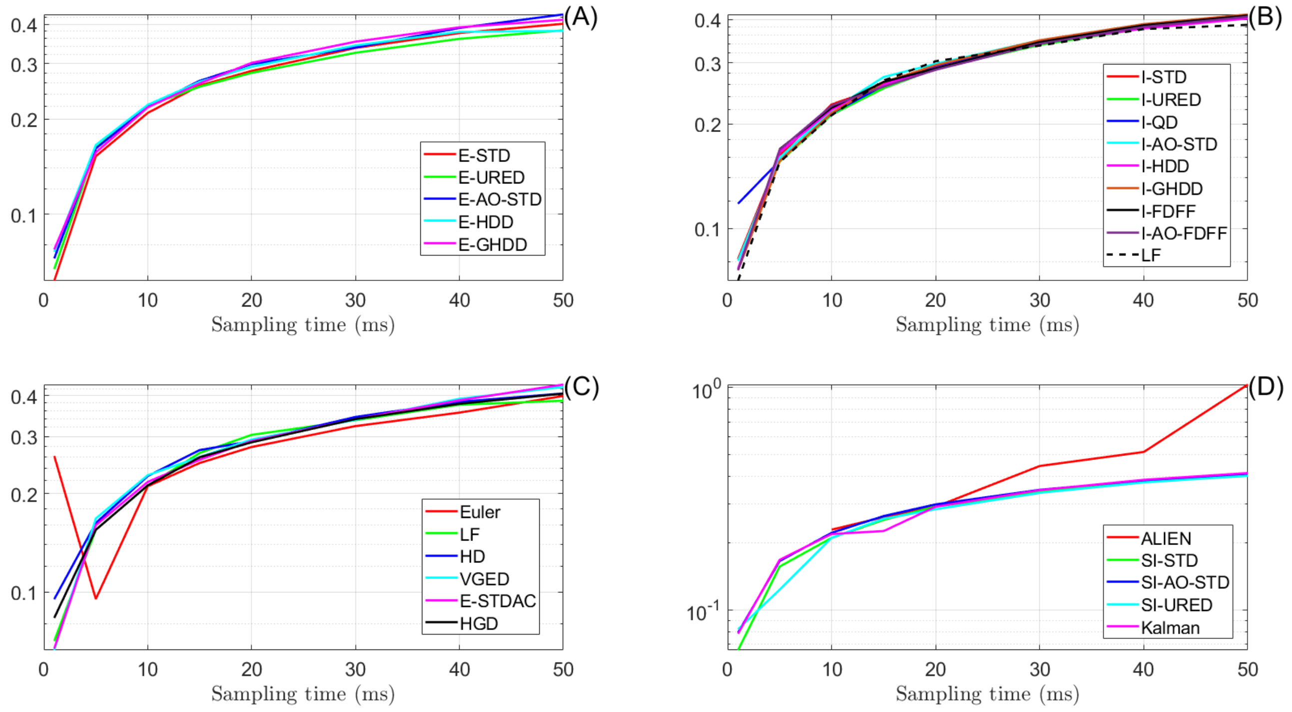

Figure 8: $\bar{L}_2$ for all methods.

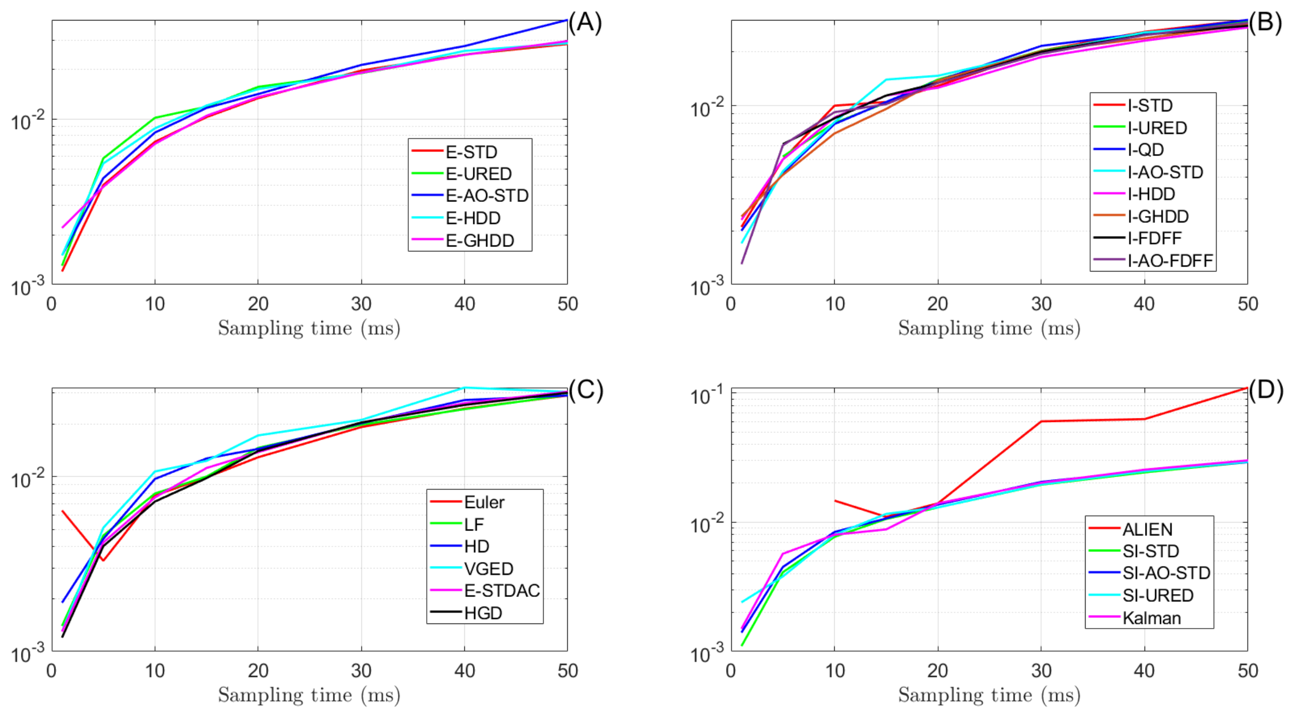

Figure 9: L_∞ for all methods.

4.6. Gain sensitivity

Some of the SMB differentiators are studied under oversized 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 in the implicit counterparts. For $L = 10^3$, all the selected differentiators are practicable except for the E-AO-STD. Note that the implicit counterpart of this differentiator, *i.e.*, remains practicable even for a larger gain $L = 10^4$. Comparing the explicit and implicit counterparts, it can be seen that the implicit ones show less variation and are less sensitive to oversized gains as predicted by Feature 3. Note that the E-GHDD shows almost an insensitive behavior to the oversized gains since the chattering caused by the discontinuous terms is avoided in its formula.

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 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, 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 Newton's solver can provide the accuracy 10^{-5} with the maximum number of 7 iterations for a specific case. According to Table 6, since the number of iterations for $h > 1$ms is always higher than 7, one may 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 experimental conclusions 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 Feature 1, several experiments have been made under different sampling times (as shown in Figs. 6 and 7). 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 chattering will be increased. In other words, by increasing the sampling time, the dominant chattering source would be numerical chattering. This validates Feature 1 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 Feature 2.
- Based on the discussion presented in Sec. 4.6, the implicit methods show better insensitivity to the gains compared to the explicit counterparts, which agrees with Feature 3.
- According to the results provided in Figs. 6 and 7, 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.

5. General conclusions

A laboratory setup, *i.e.*, an electro-pneumatic system is used in this study to implement 25 known differentiators in the closed-loop configuration. 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, in 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 (especially 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 experimental 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 Feature 1. It should be noted that this conclusion has been drawn with the parameters listed in Table 3, which have been tuned according to the method described in Sec. 4.3.
- The solver-based implicit differentiators (I-AO-STD, I-HDD, I-GHDD, I-URED) could be implemented on the selected laboratory setup in real-time. It indicates that calculation resources in typical laboratory setups can be sufficient enough to implement the solver-based methods. This validates Feature 2.
- The implicit methods present better gain insensitivity, especially in low noise conditions. In other words, one may select a larger sampling gain for

Table 5: Results for h=20ms under oversized gain.

Method	L	100 $\bar{L}_2(e)$	100 L_∞	var
E-AO-STD	20	30.09	1.62	95.08
E-HDD	20	29.15	1.51	76.39
E-GHDD	20	30.03	1.49	38.97
I-AO-STD	20	30.02	1.57	53.07
I-HDD	20	30.12	1.52	40.34
I-GHDD	20	29.39	1.46	27.3884
E-AO-STD	10^3	impracticability		
E-HDD	10^3	34.42	2.75	4.2636×10^3
E-GHDD	10^3	29.26	1.42	78.8276
I-AO-STD	10^3	29.46	1.46	53.52
I-HDD	10^3	28.57	1.58	327.54
I-GHDD	10^3	29.75	1.51	129.15
E-GHDD	10^4	30.54	2.69	315.21
I-AO-STD	10^4	29.73	1.44	52.32
I-HDD	10^4	impracticability		
E-HDD	10^4	impracticability		
I-GHDD	10^4	19.13	1.09	1.2438×10^3
h=20ms, with disturbance				

Table 6: Maximum number of iterations 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						

the implicit differentiators compared to the explicit counterparts. This follows Feature 3.

- Comparing the results, higher-order SMB differentiators, *e.g.*, AO-STD, I-HDD, and I-GHDD show smaller variation than the cascade configurations of the first-order counterparts, *e.g.*, STD.

In this study, the parameters have been tuned in an empirical way using practical experiments as explained in detail in the paper. However, the design of a suitable optimization algorithm is thought to be a non-trivial work that is left for the future. Moreover, the results in this study are obtained for the electro-pneumatic system under the selected parameter tuning, and may not be totally valid on other systems. Hence, future investigations are crucial to validate the results on other laboratory setups. Note that the comparative analysis has also been made on another laboratory setup, *i.e.*, rotary inverted pendulum system and the corresponding results are available in [53].

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] B. Andritsch, M. Horn, S. Koch, H. Niederwieser, M. Wetzlinger, M. Reichhartinger, The robust exact differentiator toolbox revisited: Filtering and discretization features, in: 2021 IEEE International Conference on Mechatronics (ICM), 2021, pp. 01–06.

[53] M. Rasool Mojallizadeh, B. Brogliato, A. Polyakov, S. Selvarajan, L. Michel, F. Plestan, M. Ghanes, J.-P. Barbot, Y. Aoustin, Discrete-time differentiators in closed-loop control systems: experiments on electro-pneumatic system and rotary inverted pendulum, Research report, INRIA Grenoble (Feb. 2022).
URL `https://hal.inria.fr/hal-03125960`

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

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

[56] 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.

[57] 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.

[58] M. Ghanes, J. A. Moreno, J.-P. Barbot, Arbitrary order differentiator with varying homogeneity degree, Automatica 138 (2022) 110111.