An Optimization Perspective on Discrete-Time Sliding Modes

WHEN PROXIMAL-POINT ALGORITHMS MEET SET-VALUED SYSTEMS

FÉLIX MIRANDA-VILLATORO, FERNANDO CASTAÑOS, and BERNARD BROGLIATO

©SHUTTERSTOCK.COM/WRIGHT STUDIO

Sliding-mode control (SMC) is a major field of automatic control that originated in Germany [1], [2] and the former Soviet Union [3], [4] approximately 70 years ago, during the development of relay systems (see the "Summary" for an article overview). Sliding-mode principles were later incorporated into the design of differentiators and observers [5], [6], and [7]. Higher-order sliding modes were introduced by Levant and coauthors in [6], [7], [8], [9], and [10]. They represented significant progress for control and differentiation, the most prominent algorithm being the supertwisting scheme. Overviews of SMC and its applications can be found in [11], [12], [13], [14], [15], and [16]. The mathematical foundations for sliding modes were developed mainly by Filippov [17], [18] within the framework of set-valued analysis and differential inclusions (DIs). Although other mathematical settings exist (maximal monotone operators being an important one, as shown in this article), Filippov's DIs remain the most popular mathematical concept in the field of automatic control. Later, discrete-time implementations were studied; see "Discrete-Time Sliding Modes in Control" for a brief historical summary of discrete-time sliding modes.

The *chattering* phenomenon has long been recognized as a major obstacle to the implementation of set-valued sliding-mode systems, hampering a wider utilization in applications.

Digital Object Identifier 10.1109/MCS.2025.3587510
Date of current version: 18 September 2025

A significant contribution of this work is the establishment of a clear connection between control methodologies and optimization algorithms.

Generally speaking, chattering is a high-frequency bang-bang input signal that harms actuators and can result in closed-loop instability [19], [20], [21] as well as high-frequency oscillations in the output signal. The time discretization of set-valued sliding-mode controllers is a major source of chattering when *explicit* (or *forward*-Euler) methods are used [22], [23], [24], [25], [26], [27], [28], [29], [30]. This has led to the introduction of a new discretization approach in the field of control, commonly known as the *implicit* (or the *backward*-Euler) method [31], [32], [33].

The main objective of this article is to show that the implicit and semi-implicit methods for the discrete-time implementation of set-valued sliding-mode controllers, observers, and differentiators are tightly related to the subject of optimization and that this close relationship allows us to consider the implicit/semi-implicit algorithms from a unified point of view. Within this setting, tools from convex analysis, complementarity theory, proximal-point algorithms and proximal operators, maximal monotone operators, and variational inequalities are crucial. This was alluded to in [21], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], and [42] and especially in [43], [44], and [45], where *resolvents* and *Yosida approximations* were used for the first time in this context. When implicitly or semi-implicitly discretized, sliding-mode controllers and differentiators yield new kinds of proximal-point algorithms, named *robust*, *time-varying*, *higher-order*, and *Hamiltonian* proximal point, all of which belong to a class of nonlinear difference equations.

Summary

This article investigates how classical sliding-mode methods, typically associated with discontinuous signals and high-frequency switching, can be more effectively understood and implemented using principles from optimization and convex analysis. For continuous-time systems, this study identifies energy dissipation as a fundamental aspect of the method. In discrete-time implementations, the research shows that employing the well-established backward-Euler discretization allows us to preserve robustness against matched disturbances as well as finite-time convergence while drastically reducing high-frequency oscillations (known as the chattering phenomenon).

A significant contribution of this work is the establishment of a clear connection between control methodologies and optimization algorithms. By creating this link, powerful optimization techniques can be utilized to simplify the analysis and enhance the design of control systems. The examples included in the article illustrate how this integrated approach yields robust and efficient strategies that can be applied for control, state observation, or signal differentiation. In addition, this study presents some variants of classical optimization algorithms specifically designed to address practical implementation challenges, facilitating the adoption of these advanced methods in real-world scenarios.

Control systems are mainly designed in a continuous-time setting, with their discretization being part of a subsequent step. When the discrete-time control is designed to emulate the continuous-time control—a procedure called *emulation*—it is usually expected that the continuous-time closed-loop properties (robustness, stability, passivity, etc.) will be preserved if the sampling time is short enough [46, Ch. 3]. It is well known, however, that the discretization process can degrade performance and modify the system's properties (location of zeroes [47], passivity [48], etc.).

Systems with sliding modes are particularly sensitive to the employed discretization method, some aspects of which are summarized in the flowchart of Figure 1. In fact, it has been verified analytically and experimentally that explicit emulation applied to sliding-mode set-valued systems yields ill-posed discrete-time systems that fail to approximate the set-valued control input, no matter how short the sampling time is.

This tutorial article is organized into three main parts. The first part is dedicated to first-order sliding-mode controllers, the second part focuses on higher-order sliding-mode systems (controllers, differentiators, state observers, etc.), and the third part discusses computational issues and presents experimental results. The article's main objectives are: 1) to shed new light on first-order schemes, in both continuous and discrete-time settings, using passivity as a guideline; 2) to show that proximal-point algorithms can be used as a unifying framework for the analysis and implementation of backward-Euler schemes applied to first-order and higher-order schemes; 3) to provide preliminary ideas on computational issues associated with the implementation of backward-Euler-based schemes (controllers, observers, differentiators, etc.); and 4) to survey the results (theoretical and experimental) that have been obtained so far concerning the backward-Euler discretization of set-valued sliding-mode systems.

SUBGRADIENT FLOWS, PROXIMAL ALGORITHMS, AND FINITE-TIME CONVERGENCE

The close relation between optimization and the implicit discretization of set-valued sliding-mode systems is illustrated in this section (see "Set-Valued Mappings and DIs" for a

Discrete-Time Sliding Modes in Control

The most widespread discrete-time algorithms for sliding-mode implementation consist of an emulation method coupled with a forward-Euler discretization, which leads to sliding dynamics similar to (7). This particular approach has long been known to produce high-frequency bang-bang controllers, and thus digital chattering [S1], [S2] (see Figure 8). Digital chattering has been deeply analyzed in [22] and [29] for various sliding-mode controllers. A common way to deal with such unwanted dynamical behavior consists of the *regularization* of the set-valued signum map (most of the time simply replacing it with a saturation function). Such an approach is essentially an engineering trick, and the necessary parameter tuning to find a correct tradeoff between sampling time and saturation width is often difficult. This has prompted researchers to design other types of discrete-time sliding-mode controllers, which should be designed directly in the discrete-time setting (whether the plant is a "pure" discrete-time system or results from the time discretization of a continuous-time plant). See, for instance, [S3] for a survey.

Let s_k be the discrete-time sliding variable. Imposing $s_{k+1} = 0$ yields a dead-beat input [S4]. This should be avoided in general; see Remark 6. To that end, various strategies have been proposed in the literature. A *quasi-sliding mode* is characterized by a tube defined by an inequality $|s_k| \leq \varepsilon$, $\varepsilon > 0$, see Figure S1. Designing control laws that render this tube invariant and attractive in a finite number of steps is a research topic initiated in [S5], [S6], [S7], and [S8] and active until recently; see, for instance, [S9], [S10], and [S11]. This method mainly consists of setting $s_{k+1} = f(s_k, x_k, k)$ for some f and then seeking a controller (a reaching law) guaranteeing this condition. Various reaching conditions have been stated in the related literature, see [49] and [S12]. Most of the reaching laws use signum functions [S8], [S9], [S10], [S11] and thus share close similarities with explicit-Euler schemes. Another branch of analysis was more recently tackled by Levant and Livne [50], [S13], [S14], [S15], concerning the discretization of higher-order and homogeneous sliding-mode schemes, with special attention paid to the accuracy of differentiators.

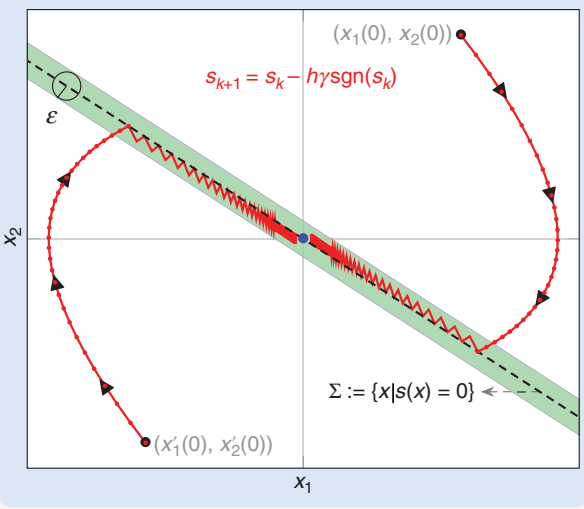

FIGURE S1 Discrete-time quasi-sliding trajectories. In discrete time, the concept of quasi-sliding refers to relaxing the exact sliding condition $x \in \Sigma := \{x \mid s(x) = 0\}$ by allowing the state to converge, in finite time, toward a neighborhood of the sliding surface (green area). This relaxation is mainly motivated by the fact that, for a discontinuous controller, no selection process is available, forcing the control input to switch at each time step. It is also important to note that, in this specific case, there are no external disturbances affecting the system. Thus, digital chatter arises due to the discontinuous nature of forward-Euler controllers.

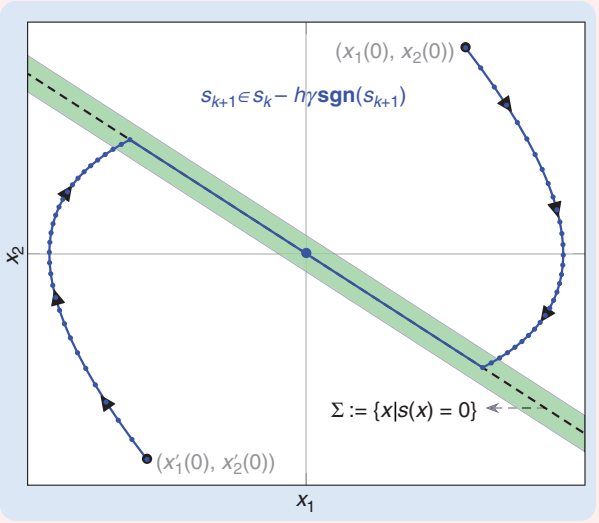

FIGURE S2 Discrete-time sliding trajectories. Contrary to the discontinuous case, set-valued controllers can attain the exact sliding surface in finite time for the nominal case without disturbances. This property is a consequence of the multivalued nature of the control map, allowing one to set up a *selection* process for computing the suitable input that will maintain the system in sliding mode. One simple way of setting a selection procedure is via the backward-Euler discretization of the multivalued terms as stated in Proposition 1. For the case when uncertainties are present, a selection strategy is presented in the "First-Order SMC: Discrete Time" section.

It is noteworthy that the first-order controllers proposed in [S1] and [S2] are backward-Euler algorithms (hence similar to the algorithms described in the "First-Order SMC: Discrete-Time" section), with the assumption that the equivalent controller is known, hence ignoring unknown disturbances. A minimum-operator scheme, as in (14b), is proposed in [49] to address unknown disturbances. Within the framework of so-called reaching laws [S8], minimum-operator controllers simply provide an alternative reaching law. However, no unknown disturbances were considered in a general context

(*Continued*)

Discrete-Time Sliding Modes in Control (*Continued*)

for backward (implicit) Euler discretization before [32], hence seriously limiting the applicability of such controllers (the link between minimum operators and backward-Euler methods was made much later in [51]). The key idea in [32] and subsequent works cited in this article is to keep the set-valued feature of sliding-mode controllers in the discrete-time design, hence showing that digital chattering is not unavoidable; see Figure S2 (contrary to some widely held beliefs that arise from experience with the forward-Euler scheme [S4]). Generalized equations, variational inequalities, complementarity problems, and maximal monotone operator theory were introduced in digital SMC design for the first time in [21], [31], [32], [33], [43], and [44]. The purpose of this article is precisely to give a broader perspective on this point of view.

REFERENCES

[S1] G. Bartolini, A. Ferrara, and V. I. Utkin, "Adaptive sliding mode control in discrete-time systems," *Automatica*, vol. 31, no. 5, pp. 769–773, 1995, doi: 10.1016/0005-1098(94)00154-B.

[S2] S. Drakunov and V. I. Utkin, "On discrete-times sliding modes," *IFAC Proc. Volumes*, pp. 273–278, Jun. 1989.

[S3] K. Furuta and Y. Pan, "Discrete-time variable structure control," In: Yu X and Xu, (Eds) *Variable Structure Systems: Towards the 21st Century* (Lecture Notes in Control and Information Sciences), vol.274. Springer, Berlin, Heidelberg, 2002, pp. 57–81.

[S4] S. Hui and S. H. Zak, "On discrete-time variable structure sliding mode control," *Syst. Control Lett.*, vol. 38, nos. 4–5, pp. 283–288, 1999, doi: 10.1016/S0167-6911(99)00075-4.

[S5] C. Milosavljevic, "General conditions for the existence of a quasis-liding mode on the switching hyperplane in discrete variable structure systems," *Autom. Remote Control*, vol. 46, pp. 307–314, Jan. 1985.

[S6] A. Bartosziewicz, "Discrete-time quasi-sliding-mode control strategies," *IEEE Trans. Ind. Electron.*, vol. 45, no. 4, pp. 633–637, Aug. 1998, doi: 10.1109/41.704892.

[S7] K. Furuta, "Sliding mode control of a discrete system," *Syst. Control Lett.*, vol. 14, no. 2, pp. 145–152, 1990, doi: 10.1016/0167-6911(90)90030-X.

[S8] W. Gao, Y. Wang, and A. Homaifa, "Discrete-time variable structure control systems," *IEEE Trans. Ind. Electron*, vol. 42, no. 2, pp. 117–122, Apr. 1995, doi: 10.1109/41.370376.

[S9] S. Chkrabarty and B. Bandyopadhyay, "A generalized reaching law for discrete time sliding mode control," *Automatica*, vol. 52, pp. 83–86, Feb. 2015, doi: 10.1016/j.automatica.2014.10.124.

[S10] H. Du, X. Yu, M. Z. Q. Chen, and S. Li, "Chattering-free discrete-time sliding mode control," *Automatica*, vol. 68, pp. 87–91, Jun. 2016, doi: 10.1016/j.automatica.2016.01.047.

[S11] H. Ma, Z. Xiong, Y. Li, and Z. Liu, "Sliding mode control for uncertain discrete-time systems using an adaptive reaching law," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 68, no. 2, pp. 722–726, Feb. 2021, doi: 10.1109/TCSII.2020.3005417.

[S12] A. J. Koshkouei and A. S. I. Zinober, "Sliding mode control of discrete-time systems," *J. Dyn. Syst. Meas. Control*, vol. 122, no. 4, pp. 793–802, 2000, doi: 10.1115/1.1321266.

[S13] M. Livne and A. Levant, "Proper discretization of homogeneous differentiators," *Automatica*, vol. 50, no. 8, pp. 2007–2014, 2014, doi: 10.1016/j.automatica.2014.05.028.

[S14] A. Levant and M. Livne, "Weighted homogeneity and robustness of sliding mode control," *Automatica*, vol. 72, no. 10, pp. 186–193, 2016, doi: 10.1016/j.automatica.2016.06.014.

[S15] A. Levant, M. Livne, and D. Lunz, "On discretization of high-order sliding modes," in *Recent Trends in Sliding Mode Control*, J. P. Barbot, L. Fridman, F. Plestan Eds., Stevenage, U.K.: IET, 2016, pp. 177–202.

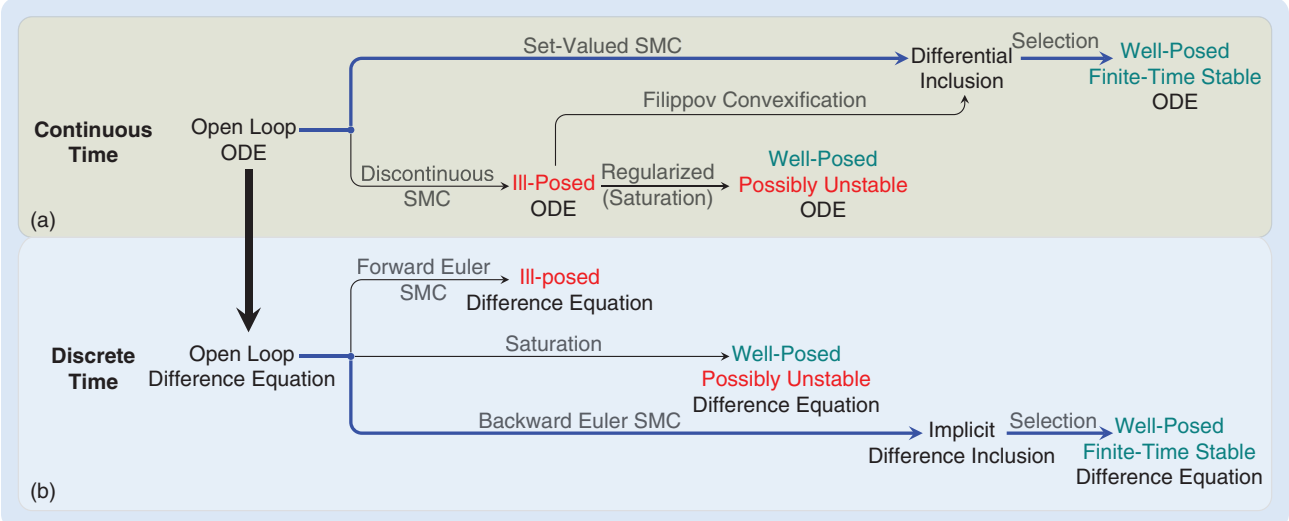

FIGURE 1 A flowchart of the implementation of control systems using sliding modes. From the point of view of traditional SMC, a discontinuous feedback law is enforced on an ordinary differential equation (ODE). A consistent notion of *solution* requires the conversion of the closed-loop discontinuous differential equation into a DI (Filippov's convexification). Alternatively, a set-valued feedback law can be directly formulated from the outset, yielding again a DI from which a continuous selection is uniquely chosen (thick blue line). The result is a well-posed ODE with a stable desired equilibrium. Some practitioners bypass Filippov's convexification by replacing the discontinuous law with an arbitrary approximation (typically a saturation). The outcome may be the destabilization of the desired equilibrium. A backward-Euler discretization of the sliding-mode controller results in a difference inclusion from which, in analogy with the continuous-time case, a selection can be made to form a well-posed difference equation with a stable desired equilibrium (thick blue line). The formulation of the difference inclusion, together with its solution techniques, is usually referred to as the *implicit* (or *backward-Euler*) method. (a) Continuous time. (b) Discrete time.

Set-Valued Mappings and DIs

The concept and properties of sliding modes are crucial to this article. Sliding modes are defined and analyzed through the lens of DIs, which, in turn, involve set-valued operators. The following material is taken from [S16], [S17], [S18], [S19], and [S20]. It aims at giving a concise introduction to DIs in finite dimensions.

We start with a few basic definitions. A set-valued operator or multifunction $\mathbf{F}:\mathbb{R}^n \rightrightarrows \mathbb{R}^m$ is a map that associates with any $x\in\mathbb{R}^n$, a subset $\mathbf{F}(x)\subset\mathbb{R}^m$. The *domain* of F is

$$\operatorname{dom}\mathbf{F}=\{\xi\in\mathbb{R}^n \mid \mathbf{F}(\xi)\neq\varnothing\}. \tag{S1}$$

The inverse mapping (set-valued or single-valued) of $\mathbf{F}$ is defined as $\mathbf{F}^{-1}:\mathbb{R}^m\rightrightarrows\mathbb{R}^n$ with $\mathbf{F}^{-1}(y)=\{\xi\in\mathbb{R}^n \mid y\in\mathbf{F}(\xi)\}$. Given a set $E\subset\mathbb{R}^m$, its inverse image by $\mathbf{F}$ is defined as $\mathbf{F}^{-1}(E)=\cup_{y\in E}\mathbf{F}^{-1}(y)$. The *range* of the operator F is

$$\operatorname{rge}\ \mathbf{F}=\{\eta\in\mathbb{R}^m \mid \eta\in\mathbf{F}(\xi)\ \text{for some}\ \xi\in\mathbb{R}^n\} \tag{S2}$$

and its *graph* is

$$\operatorname{gph}\mathbf{F}=\{(\xi,\eta)\in\mathbb{R}^n\times\mathbb{R}^m \mid \eta\in\mathbf{F}(\xi)\}. \tag{S3}$$

FILIPPOV'S DIs

We can now pass to the presentation of various types of DIs, which are central to this article. DIs provide a mathematical framework for the study of systems with discontinuous models. Namely, for a discontinuous differential equation of the form

$$\dot{x}(t)=f(x(t),t),\quad x(t_0)=x_0 \tag{S4}$$

where the single-valued map $f:\mathbb{R}^n\times\mathbb{R}_+\to\mathbb{R}^n$ is measurable in both arguments and locally bounded, it is well known that, in general, a solution in the sense of Carathéodory fails to exist [that is, an absolutely continuous function that satisfies (S4) almost everywhere (a.e.)]. Thus, to make sense of (S4), Filippov's convexification method considers instead the DI

$$\dot{x}(t)\in F(x(t),t),\quad x(t_0)=x_0 \tag{S5}$$

where $\mathbf{F}(x,t)=\cap_{\varepsilon>0}\cap_{\mu(\Omega)=0}\overline{\operatorname{co}}\{f((x+\varepsilon\mathcal{B})\setminus\Omega,t)\}$, $\overline{\operatorname{co}}$A denotes the closed convex-hull of A, and $\mu(\Omega)$ denotes the Lebesgue measure of the set $\Omega\subset\mathbb{R}^n$. Thus, absolutely continuous functions that satisfy (S5) a.e. are solutions of (S4) in the sense of Filippov, see, for instance, [17] and [18]. On the other hand, if instead of (S5) we consider the DI

$$\dot{x}(t)\in\mathbf{K}(x(t),t),\quad x(t_0)=x_0 \tag{S6}$$

where $\mathbf{K}(x,t)=\cap_{\varepsilon>0}\overline{\operatorname{co}}\{f(x+\varepsilon\mathcal{B},t)\}$, then, absolutely continuous functions that satisfy (S6) a.e. are said to be solutions of (S4) in the *sense of Krasovskii*. In general, Filippov and Krasovskii solutions are different; see, for instance, [52].

SEMI-CONTINUITY OF SET-VALUED MAPS

What are the general conditions that guarantee the existence of absolutely continuous solutions? Similarly to the Carathéodory conditions of ordinary differential equations (ODEs), continuity properties play a fundamental role in ensuring the existence of solutions of DIs.

Definition S1

The set-valued map $\mathbf{F}:\mathbb{R}^n\rightrightarrows\mathbb{R}^m$ is *outer semicontinuous (OSC) at* $x\in\operatorname{dom}\mathbf{F}$ if, for any sequence $\{(x_k,y_k)\}$ such that $(x_k,y_k)\in\operatorname{gph}\mathbf{F}$ for all k and $(x_k,y_k)\to(x,y)$, one has $(x,y)\in\operatorname{gph}\mathbf{F}$. The set-valued map is said to be OSC if it is OSC at every $x\in\operatorname{dom}\mathbf{F}$. The following characterization is similar to the usual $\epsilon-\delta$ formulation of continuity for single-valued functions.

Fact 1

The set-valued map $\mathbf{F}:\mathbb{R}^n\rightrightarrows\mathbb{R}^m$ is OSC in x_0 if, for all $\varepsilon>0$, there is a neighborhood $N(x_0)$ such that $x\in N(x_0)$ implies $F(x)\subset F(x_0)+\epsilon\mathcal{B}_2$ [where the unit ball $\mathcal{B}_2$ is defined in (S56) in "Convex Analysis Tools"].

Notice that OSC is also sometimes called upper semicontinuity, though OSC now seems to be a widely adopted terminology. Filippov's map in (S5) is OSC. Using Fact 1, it is easy to verify that the set-valued signum function is OSC (in particular, at $x_0=0$). Another set-valued map that is commonly used, namely the normal cone to a convex set (see Definition S8), is also OSC (this follows from Fact 1). Intuitively, OSC means that, when moving away from x_0, the set $F(x)$ does not "explode" suddenly. Fact 2 is helpful in establishing the OSC of monotone operators (see Fact 4 in "Maximal Monotone Operators").

Fact 2

If $\mathbf{F}:\mathbb{R}^n\rightrightarrows\mathbb{R}^m$ is OSC at x, then $\mathbf{F}(x)$ is a closed set (hence, an OSC set-valued map $\mathbf{F}$ is closed valued). Moreover, $\mathbf{F}$ is OSC if and only if $\operatorname{gph}\mathbf{F}$ is closed.

Set-valued maps can also be *inner semicontinuous* (ISC). However, we do not use ISC maps in this article. A set-valued map that is both OSC and ISC is said to be continuous.

WELL-POSEDNESS, SLIDING MODES, AND SELECTIONS

The conditions of the following theorem can be considered as an extension of Carathéodory conditions for DIs.

Theorem S1 ([S16, Thm. 5.1 and 5.2])

Let $\mathbf{F}:\mathbb{R}^n\times[0,T]\rightrightarrows\mathbb{R}^n$ be an OSC map for almost all $t\in[0,T]$ and with closed and convex images. Assume there exists a function $c\in L_1([0,T],R)$ such that, for all $t\in[0,T]$

$$\sup_{\eta\in F(x,t)}\|\eta\|\le c(t)(1+\|x\|).$$

(*Continued*)

Set-Valued Mappings and DIs (*Continued*)

Then, for every $x(0) = x_0 \in \mathbb{R}^n$, the DI $\dot{x}(t) \in \mathbf{F}(x(t),t)$ has an absolutely continuous solution on *[0, T]*.

Theorem S1 can be applied to (S5) and (S6). Since Filippov's and Krasovskii's DIs have been introduced previously, it is natural to present the notion of a sliding mode (or surface), which was the original motivation.

Definition S2

A solution $x:[t_1,t_2] \to \mathbb{R}^n$, $t_1 < t_2$ of the DI $\dot{x}(t) \in \mathbf{F}(x(t),t)$, is said to be a *sliding mode* if $\operatorname{int}\mathbf{F}(x(t),t) \neq \varnothing$ for all $t \in [t_1,t_2]$.

Transversal vector fields to a discontinuity manifold do not satisfy this definition [consider, for instance, $\dot{x}(t) = (1,-2)^\top$ if $x_1(t) < 0$ and $\dot{x}(t) = (1,2)^\top$ if $x_1(t) > 0$, $\dot{x}(t) \in \overline{\operatorname{co}}\{(1,-2),(1,2)\}$ if $x_1(t) = 0$], but repulsive manifolds may satisfy it [consider $\dot{x}(t) \in \mathbf{sgn}(x(t))$].

The notion of a selection is also central in this work, as it allows us to rigorously interpret the backward-Euler discretization. Inputs are selections of the set-valued right-hand side when the system is in a sliding mode, as stated in Filippov's mathematical framework.

Definition S3 ([S19, p. 35])

Let $\mathbf{F}:\mathbb{R}^n \rightrightarrows \mathbb{R}^m$ be a set-valued mapping. Then, a single-valued mapping $\lambda:\mathbb{R}^n \to \mathbb{R}^m$ satisfying $\lambda(x) \in \mathbf{F}(x)$ for all $x \in \mathbb{R}^n$ is a *selection*.

A solution of the DI must satisfy $\dot{x}(t) = \lambda(x(t))$ for all t with λ a selection. In the time-varying case with $F:\mathbb{R}^n \times \mathbb{R} \rightrightarrows \mathbb{R}^n$, one can augment the state with time as $\dot{\tau} = 1$, $\tau(0) = t_0$, and define a selection accordingly as $\lambda(x,\tau) \in F(x,\tau)$ for all x and τ. For instance, consider $\dot{x}(t) \in F(x,t) = -\mathbf{sgn}(x(t)) + \delta(t)$, $|\delta(t)| < 1$, $x(t) \in \mathbb{R}$, δ a continuous function. A selection is given by $\lambda(x,\tau) = -1 + \delta(\tau)$ if $x > 0$, $1 + \delta(\tau)$ if $x < 0$, and $\xi + \delta(\tau)$ with $\xi \in [-1,1]$ if $x = 0$.

SUBGRADIENT DIs

Apart from Filippov's or Krasovskii's DIs, other types of set-valued systems exist and are useful in the context of this article. DIs of the form

$$\dot{s}(t) \in -\partial f(s(t)) + \delta(t) \tag{S7}$$

can be formulated directly without calling upon a discontinuous differential equation. In (S7), f is assumed convex, proper, and lower semicontinuous (LSC, see Definition S6), ∂f is the subdifferential of f in the sense of convex analysis (see "Convex Analysis Tools"), and $\delta:\mathbb{R} \to \mathbb{R}^n$ is a measurable and uniformly bounded function accounting for external disturbances affecting the system. Such systems appear in (25) and (39). Notice that since the subdifferential in (S7) defines a maximal monotone mapping (see Theorem S2), the right-hand side of this DI is OSC, and Theorem S1 is applicable by Fact 3. It can be shown that, under some conditions on a nondifferentiable function $f:\mathbb{R}^n \to \mathbb{R}$, Filippov and Krasovskii's convexification processes on the corresponding gradient system yield ∂f [S24].

The previous results are presented for the case where $\operatorname{dom}(F) = \mathbb{R}^n$. They can be adapted to the case $\operatorname{dom}(F) \subset \mathbb{R}^n$ by adding some conditions [S16, Thm. 5.1 and 5.2] [in particular, $x(0) = x_0 \in \operatorname{dom}(F)$]. This is encountered in the context of maximal monotone right-hand sides like (S7) when f is the indicator function of a closed nonempty convex set $K \subset \mathbb{R}^n$ (see Definition S9).

A ROBUSTNESS RESULT

The set-valuedness of SMC systems (or differentiators) is extremely useful because it implies the intrinsic robustness of these schemes. This is manifested in the following.

Proposition S1

Consider the perturbed subgradient system (S7) with $f:\mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ a proper, convex, and LSC function such that $\operatorname{dom} f = \mathbb{R}^n s$. Suppose that there exists a fixed $\varepsilon > 0$ such that

$$\delta(t) + \varepsilon\mathcal{B}_2 \subseteq \partial f(0) \tag{S8}$$

for all $t \geq 0$. The origin is finite-time stable. Note that condition (S8) implies that ∂f is not a singleton at $s = 0$ and that $\delta(t)$ is uniformly bounded.

Proof

The DI (S7) is of the form $\dot{s}(t) \in -\mathbf{M}(s(t),t)$, where the operator $\mathbf{M}(\cdot,t)$ is maximal monotone for each t. Under mild conditions on δ, the DI possesses unique absolutely continuous solutions on $\mathbb{R}_+$ for any initial condition [S21, Thm. 3.4]. Consider the Lyapunov candidate function $V(s) = 1/2\,\|s\|^2$. The set-valued derivative [S22] of V along trajectories of (S7) is

$$\dot{\tilde{V}}(s) = \{-s^\top\vartheta + s^\top\delta \mid \vartheta \in \partial f(s)\}.$$

Since $(s,\vartheta) \in \operatorname{gph}\partial f$ and f is proper, convex, and LSC, it follows that

$$s^\top\vartheta = f(s) + f^*(\vartheta) \tag{S9}$$

where f^* denotes the conjugate function to f (see Definition S10 and Theorem S4 in "Convex Analysis Tools"). Consequently, from (S8), we see that $\delta(t) + \varepsilon\rho \in \partial f(0)$ for any $\rho \in \mathcal{B}_2$ and any $t \in \mathbb{R}_+$. By the definition of subdifferential (see Definition S7 in "Convex Analysis Tools")

$$\delta^\top s \leq -\varepsilon\rho^\top s + f(s) - f(0) \tag{S10}$$

(*Continued*)

Set-Valued Mappings and DIs (*Continued*)

for any $s \in \mathbb{R}^n$ and for any $\rho \in \mathcal{B}_2$. Since $\partial f(s)$ is compact for any $s \in \mathbb{R}^n$ [S23, Thm. 23.4], $\max \dot{\tilde{V}}(s)$ is well defined. Combining (S9) and (S10), we obtain

$$\max \dot{\tilde{V}}(s) \leq -\varepsilon \rho^\top s - f^*(\vartheta^*) - f(0) \tag{S11}$$

for any $\rho \in \mathcal{B}_2$ and some $\vartheta^* \in \partial f(x)$. Suppose that $s \neq 0$ and set $\rho = s/\|s\| \in \mathcal{B}_2$. Recalling that $f^*(\vartheta) + f(0) \geq 0$ for any $\vartheta \in \mathbb{R}^n$ gives

$$\max \dot{\tilde{V}}(s) \leq -\varepsilon \|s\| = -\varepsilon \sqrt{2V(s)}\,. \tag{S12}$$

For $s = 0$, the inequality follows trivially.

REFERENCES

[S16] K. Deimling, *Multivalued Differential Equations* (De Gruyter Series in Nonlinear Analysis and Applications). Berlin, Germany: de Gruyter, 1992.

[S17] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis* (Grundlehren Der Mathematischen Wissenschaften), vol. 317. Berlin, Germany: Springer-Verlag, 1998.

[S18] J. B. Hiriart Urruty, and C. Lemaréchal, *Fundamentals of Convex Analysis* (Grundlehren Text Editions). Berlin, Germany: Springer-Verlag, 2001.

[S19] G. V. Smirnov, *Introduction to the Theory of Differential Inclusions* (Graduate Studies in Mathematics), vol. 41. Providence, RI, USA: American Mathematical Society, 2002.

[S20] R. S. Burachik and A. N. Iusem, *Set-Valued Mappings and Enlargements of Monotone Operators*. New York, NY, USA: Springer-Verlag, 2008.

[S21] H. Brézis, "Opérateurs Maximaux Monotones et Semi-Groupes de Contractions dans les Espaces de Hibert, North-Holland," *Mathematics Stud.*, vol. 51973.

[S22] A. Bacciotti and F. Ceragioli, "Stability and stabilization of discontinuous systems and nonsmooth Lyapunov functions," *ESAIM: Control, Optim. Calculus Variations*, vol. 4, pp. 361–376, Aug. 1999, doi: 10.1051/cocv:1999113.

[S23] R. T. Rockafellar, *Convex Analysis. Princeton Landmarks in Mathematics*. Princeton, NJ, USA: Princeton Univ. Press, 1972.

[S24] M. Bivas, A. Daniilidis, and M. Quincampoix, "Characterization of Filippov representable maps and Clarke subdifferentials," *Math. Program.*, vol. 189, nos. 1–2, pp. 99–115, 2021, doi: 10.1007/s10107-020-01540-y.

definition of sliding motion). For the sake of clarity, we first restrict the presentation to the scalar case without perturbation.

The Continuous-Time System

Consider a scalar continuous-time dynamical system

$$\dot{x}(t) = ax(t) + bu(t) \tag{1}$$

with $a < 0$ and $0 < b$ constant and fixed. Suppose that the input is a feedback law $u(t) = u_{\text{sv}}(x(t))$. The set-valued feedback

$$-u_{\text{sv}}(x) \in \kappa\, \mathbf{sgn}(x), \quad \kappa > 0 \tag{2}$$

where $\mathbf{sgn}(\cdot)$ is the set-valued signum function (S50) achieves the finite-time convergence of the state toward the origin [53], [54]. The closed-loop system belongs to the class of gradient systems [see (S7)]

$$\dot{x}(t) \in -\partial f(x(t)) \tag{3}$$

with $f(x) = -(a/2)x^2 + b\kappa|x|$. According to Proposition S1 in "Set-Valued Mappings and DIs," the origin is finite-time stable.

The Time Discretization

Due to the computational power and flexibility of digital electronic devices, it is now common to consider the discrete-time version of the control law (2) for its implementation. To that end, consider the exact discrete-time dynamics associated with (1), that is, the exact solution of (1) evaluated at discrete times $t_k = kh$, where $k \in \mathbb{N}$ and $h > 0$ denotes the sampling time. By the variation of constants formula, it is inferred that

$$x(t) = e^{a(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{a(t-\tau)}bu(\tau)\,d\tau. \tag{4}$$

Thus, assuming that the control input $u(t)$ remains constant during the intervals $[t_k, t_{k+1})$, it is deduced that

$$x_{k+1} = \tilde{a}x_k + \tilde{b}u_k \tag{5}$$

where $x(t_k) = x_k$, $u_k = u(t_k)$, $\tilde{a} = e^{ah}$, and $\tilde{b} = ((e^{ah} - 1)b)/a$.

The Forward-Euler Method

Consider first the explicit (forward-Euler) discretization of (2), that is, $u_k = u_{\text{sv}}(x_k)$, where

$$-u_{\text{sv}}(x_k) \in \kappa \mathbf{sgn}(x_k). \tag{6}$$

The closed-loop discrete-time dynamics becomes

$$x_{k+1} \in \tilde{a}x_k - \tilde{b}\kappa \mathbf{sgn}(x_k). \tag{7}$$

It is worth emphasizing that, at $x_k = 0$, the inclusion (7) does not have a unique solution as the right-hand side is set-valued. Moreover, with such feedback, the origin is not finite-time stable, no matter how small h is. Indeed, the origin becomes unstable, and an asymptotically stable limit cycle appears, creating *numerical* (or *discretization* or *digital*) chattering. The amplitude of the limit cycle is proportional to the control gain κ and to h through $\tilde{b}$: if h decreases to zero, then the amplitude of the limit cycle also decreases, and at the limit, the oscillation disappears. On the other hand, if h increases, then $\tilde{b}$ increases, and the amplitude increases. This contrasts sharply with the implicit method described in the following, which is relatively insensitive to

an increase in the sampling time (see Figures 2 and 3). The existence of limit cycles in explicitly discretized sliding-mode systems has been analyzed in depth in [22], [23], [24], [25], [26], [27], [28], [29], [30], and [55]. It is inferred that digital chattering is intrinsically linked to the explicit discretization [not to be confused with the explicit forms of the implicit (backward-Euler) discretization, as detailed next].

The Backward-Euler Method

Consider now the backward-Euler discretization of (2)

$$-u_{sv}(x_k) \in \kappa \mathbf{sgn}(x_{k+1}) \tag{8}$$

which results in

$$x_{k+1} \in \tilde{a}x_k - \tilde{b}\kappa \mathbf{sgn}(x_{k+1}). \tag{9}$$

In this case, the closed loop is well-posed as the state x_{k+1} depends only on data available at time $t_k = kh$, (that is, $\tilde{a}$, $\tilde{b}$, κ, and x_k). In fact, (9) is equivalent to the following *generalized equation* in the unknown x_{k+1}:

$$\tilde{a}x_k \in (I_d + \tilde{b}\kappa \mathbf{sgn})(x_{k+1}) \tag{10}$$

where I_d is the identity mapping. In view of the maximal monotonicity of the sign multifunction (see "Maximal Monotone Operators"), the generalized (10) has a unique solution

$$x_{k+1} = \mathrm{Prox}_{\tilde{b}\kappa g}(\tilde{a}x_k) \tag{11}$$

where Prox_f denotes the proximal mapping of the function f and $g(\cdot) = |\cdot|$ (see "Proximal Mapping and Proximal-Point Algorithm" for more details). In this case

$$\mathrm{Prox}_{\tilde{b}\kappa g}(\tilde{a}x_k) = \tilde{a}x_k - \tilde{b}\kappa \mathbf{sgn}(x_k) \min\left\{\frac{\tilde{a}}{\tilde{b}\kappa}|x_k|, 1\right\} \tag{12}$$

or, equivalently

$$\mathrm{Prox}_{\tilde{b}\kappa g}(\tilde{a}x_k) = \begin{cases} 0 & \text{if } |x_k| \le \dfrac{\tilde{b}\kappa}{\tilde{a}} \\ \tilde{a}x_k - \tilde{b}\kappa\, \mathbf{sgn}(x_k) & \text{otherwise} \end{cases}. \tag{13}$$

The iteration (11) is an instance of the proximal-point algorithm (see Definition S5 in "Proximal Mapping and Proximal-Point Algorithm"), widely studied in the literature of convex optimization [56] and recognized as an effective discretization of the subgradient system (3) [57]. It is an explicit form of the algorithm implicitly defined by (9) and falls within the context of Krasnosel'skiĭ-Mann iterations [58]. Thus, its convergence is guaranteed by Krasnosel'skiĭ-Mann Theorem [58] for any proper convex LSC function g, whenever the set of minimizers of g is nonempty [58, Thm. 1.1]. Moreover, since, in particular, $g(\cdot) = |\cdot|$ implies $0 \in \mathrm{int}\,\partial g(0)$, the origin of (9) is globally finite-time stable for any $h > 0$ [59] and exhibits no oscillations at all. [Figure 2 shows the cobweb diagram of (9).] On the other hand, the explicit discretization (7) exhibits oscillations. [Figure 3 shows the cobweb diagram of (7).]

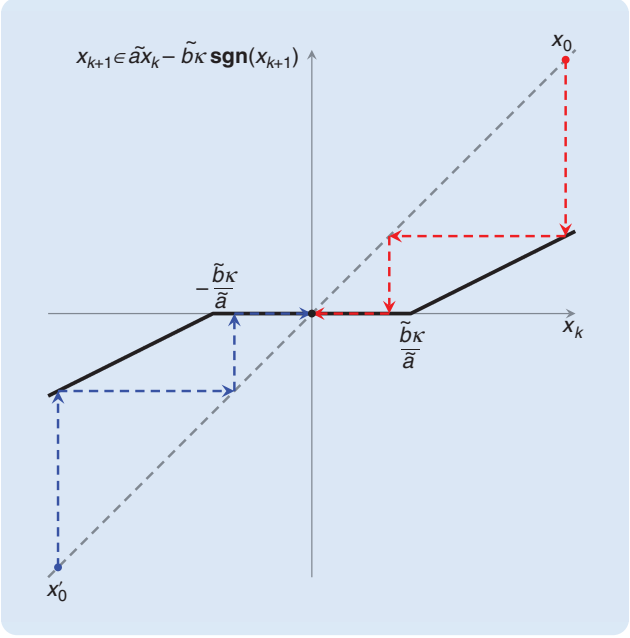

FIGURE 2 A cobweb diagram of (9) with parameters: $\tilde{a} = 0.5$, $\tilde{b}\kappa = 0.75$. The implicit discretization of the set-valued law (2) yields an iteration with a maximal monotone map such that, for any initial condition, the state converges toward the origin after a finite number of steps.

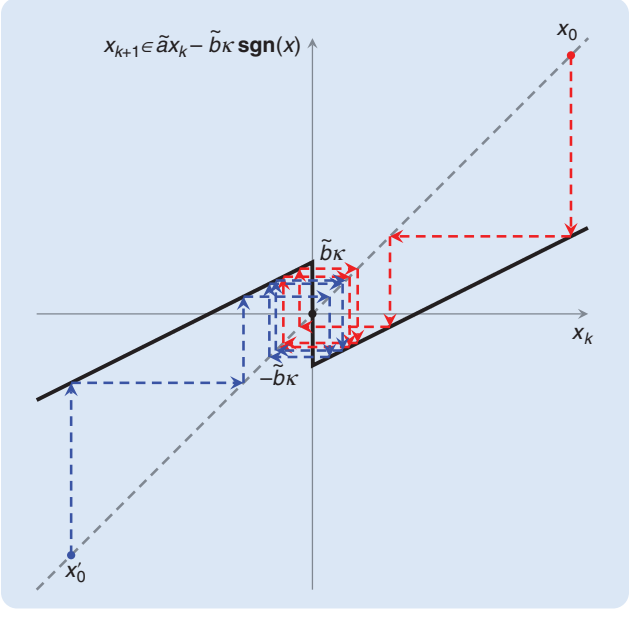

FIGURE 3 A cobweb diagram of (7) with parameters $\tilde{a} = 0.5$, $\tilde{b}\kappa = 0.75$ for two different initial conditions. The explicit discretization of the set-valued law (2) yields an iteration with a nonmonotone map such that, for almost all initial conditions, the closed-loop shows an oscillatory behavior. The amplitude of the oscillation is proportional to the control gain κ.

Maximal Monotone Operators

Since their introduction in 1960 by Minty [S25], maximal monotone operators have found application in several domains. From optimization to partial differential equations, maximal monotonicity appears as a fundamental property in the study of the convergence of numerical schemes via fixed-point methods.

Definition S4

A set-valued operator $\mathbf{M}:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is said to be *monotone* [respectively strongly monotone] if, for any two pairs $(x_i, y_i) \in \operatorname{gph}\mathbf{M}$, $i = 1,2$

$$\langle x_1 - x_2, y_1 - y_2 \rangle \geq 0 \quad (\text{resp.} \geq \alpha \|x_1 - x_2\|^2, \alpha > 0). \tag{S13}$$

The operator $\mathbf{M}$ is *maximal monotone* if it is monotone and its graph is not strictly contained inside the graph of any other monotone operator; see Figure S3.

Whenever the domain of an operator is not the whole ambient space, $\mathbf{M}:\operatorname{dom}(\mathbf{M}) \subseteq \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ (think of the normal-cone mapping), we will simplify the notation to $\mathbf{M}:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$. The Euclidean scalar product $\langle \cdot, \cdot \rangle$ is used in (S13). However, in the framework of this article (finite-dimensional systems defined on $\mathbb{R}^n$), it is possible to extend the definition to weighted inner products $\langle x, y \rangle_M = x^\top M y$ with $M = M^\top \succ 0$. The archetypal example of a maximal monotone map is the convex subdifferential of a convex function [see Definition S7 and (7) in "Convex Analysis Tools"].

Theorem S2

See [S28, Thm. 20.40]. Let $f:\mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, convex, and LSC function. Then, $\partial f:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is maximal monotone. The next fact states a strong property of maximal monotone maps.

Fact 3

Let $\mathbf{M}:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be maximal monotone, and let $x \in \mathbb{R}^n$. Then, $\mathbf{M}(x)$ is closed and convex.

As seen in "Set-Valued Mappings and DIs," outer semicontinuity (OSC) is an essential property in the context of DIs. It happens that OSC and maximal monotonicity are closely related. The following property is a consequence of Fact 2 in "Set-Valued Mappings and DIs."

Fact 4

A maximal monotone operator $\mathbf{M}:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$ has a closed graph; hence, it is OSC. Therefore, in finite dimensions, maximal monotone maps can be considered as OSC maps with an additional monotonicity (or incremental passivity) property.

It will be useful to consider $\operatorname{rint}(\operatorname{dom}\mathbf{F})$, the *relative interior* of dom $\mathbf{F}$, which is the interior when the ambient space is considered to be the affine hull of dom $\mathbf{F}$. For instance, the interior of a nontrivial planar disk in $D \subset \mathbb{R}^3$ is empty because D has no thickness. Let us embed D into the plane that contains it (which is its affine hull). Then, the interior of D, seen as a subset of this plane, is nonempty.

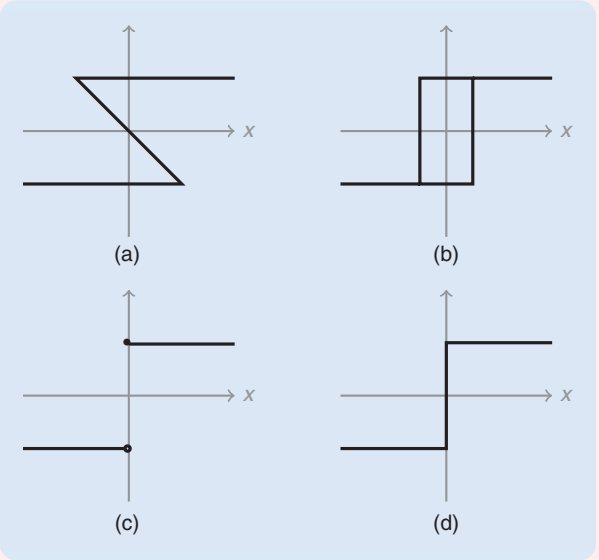

FIGURE S3 Four input–output characteristics. In the literature of sliding modes, the control input is associated with the action of switches. Some models, such as (a), (b), and (d), are multivalued, whereas others, such as (c), are discontinuous. Models (a) and (b) consider a hysteresis effect, and they are not monotone, whereas models (c) and (d) represent ideal switches. It is noteworthy that both ideal switches are monotone, but (c) is not maximal monotone.

Maximal monotonicity is preserved by elementary operations. Consider two maximal monotone operators $\mathbf{M}_i:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$, $i = 1,2$, and let $L \in \mathbb{R}^{n\times m}$ and $\gamma \geq 0$. Then

1) $\mathbf{M}_i^{-1}$ and $\gamma\mathbf{M}_i$ are maximal monotone.
2) $\mathbf{M}_1 + \mathbf{M}_2$ is maximal monotone whenever

$$\operatorname{rint}(\operatorname{dom}\mathbf{M}_1) \cap \operatorname{rint}(\operatorname{dom}\mathbf{M}_2) \neq \varnothing. \tag{S14}$$

3) $\mathbf{T}_i(x) = L^\top \mathbf{M}_i(Lx)$ are maximal monotone whenever

$$\operatorname{rge}(L) \cap \operatorname{rint}(\operatorname{dom}\mathbf{M}_i) \neq \varnothing. \tag{S15}$$

4) The mapping $x \mapsto -Ax + B(D + \mathbf{M}^{-1})^{-1}(Cx)$ is maximal monotone whenever

$$\operatorname{Im}(C) \cap \operatorname{rint}(\operatorname{Im}(D + \mathbf{M}^{-1})) \neq \varnothing \tag{S16}$$

$\mathbf{M}$ is maximal monotone, and the quadruple (A, B, C, D) is passive.

Items 1, 2, and 3 are classical [S17]. Inverse mappings are defined in "Set-Valued Mappings and DIs." Item 4 follows from [S26], building upon results in [S27] and [60] [there, monotonicity is understood with the weighted inner product defined by the solution $P = P^\top \succ 0$ of the linear matrix inequality (LMI)

(*Continued*)

Maximal Monotone Operators (*Continued*)

characterizing the passivity of the quadruple]. It means that the negative feedback interconnection of a passive linear invariant system with a maximal monotone static nonlinearity **M** preserves the maximal monotonicity. To some degree, this is an extension of [S17, Ex. 12.45 (b)]. The following theorem reveals one of the most important properties of monotonicity in the context of optimization algorithms (see "Proximal Mapping and Proximal-Point Algorithm").

Theorem S3

Let $\mathbf{M}: \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ be a monotone operator. Then, **M** is maximal monotone if and only if for any $\gamma > 0$, $\operatorname{rge}(I_d + \gamma \mathbf{M}) = \mathbb{R}^n$.

Given a maximal monotone operator **M**, we construct

$$\mathcal{J}_{\gamma \mathbf{M}} := (I_d + \gamma \mathbf{M})^{-1} \tag{S17}$$

called the *resolvent* of **M** of index $\gamma > 0$. Thus, an alternative statement of Theorem S3 is that $\operatorname{dom} \mathcal{J}_{\gamma \mathbf{M}} = \mathbb{R}^n$. An immediate corollary is that, for any $y \in \mathbb{R}^n$, the generalized equation

$$y \in x + \gamma \mathbf{M}(x) \tag{S18}$$

always admits a solution $x \in \operatorname{dom} \mathbf{M}$. It can be shown that $\mathcal{J}_{\gamma \mathbf{M}}$ is single-valued [S28, Corollary 23.10]. In other words, it is shown that the solution of (S18) is unique.

It follows from Theorem S2 that, if f is proper, convex, and LSC, then the generalized equation

$$y \in x + \gamma \partial f(x) \tag{S19}$$

has a unique solution $x = \mathcal{J}_{\gamma \partial f}(y)$ for any $y \in \mathbb{R}^n$. Such generalized equations appear mainly in convex optimization problems and the explicit computation of a control selection. Another important single-valued map is the *Yosida approximation* of **M** of index γ

$$\mathcal{Y}_{\gamma \mathbf{M}} := \frac{1}{\gamma}(I_d - \mathcal{J}_{\gamma \mathbf{M}}) \tag{S20}$$

which is Lipschitz continuous with a Lipschitz constant equal to $1/\gamma$. It is known that

$$\mathcal{Y}_{\gamma \mathbf{M}} = (\gamma I_d + \mathbf{M}^{-1})^{-1} \tag{S21}$$

that is, $\mathcal{Y}_{\gamma \mathbf{M}}(x) = \mathcal{J}_{(1/\gamma)\mathbf{M}^{-1}}((1/\gamma)x)$ [S17, Lemma 12.14]. A connection between monotonicity and passivity (see "Passive Systems") is expressed by the fact that, for any $x \in R^n$, we have [S29, Thm. 2, Ch. 3]

$$(\mathcal{J}_{\gamma \mathbf{M}}(x), \mathcal{Y}_{\gamma \mathbf{M}}(x)) \in \operatorname{gph} \mathbf{M}. \tag{S22}$$

REFERENCES

[S25] G. J. Minty, "Monotone networks," *Proc. R. Soc. London, Ser. A*, vol. 257, pp. 194–212, Sep. 1960, doi: 10.1098/rspa.1960.0144.

[S26] M. K. Camlibel and J. M. Schumacher, "Linear passive systems and maximal monotone mappings," *Math. Program.*, vol. 157, no. 2, pp. 397–420, 2016, doi: 10.1007/s10107-015-0945-7.

[S27] B. Brogliato and D. Goeleven, "Well-posedness, stability and invariance results for a class of multivalued Lur'e dynamical systems," *Nonlinear Anal. Theory Methods Appl.*, vol. 74, no. 1, pp. 195–212, 2011, doi: 10.1016/j.na.2010.08.034.

[S28] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Cham, Switzerland: Springer-Verlag, 2011.

[S29] J.-P. Aubin and A. Cellina, *Differential Inclusions: Set-Valued Maps and Viability Theory*. Berlin, Germany: Springer-Verlag, 2012.

Implementation of the Backward-Euler Method

For implementation purposes, an explicit expression for the selection of the control law in (8) can be computed with the help of (5), as shown in the following proposition.

Proposition 1

For the closed-loop (5) and (8), the explicit expression (selection) of the control input is as follows:

$$-u_{\mathrm{sv}}(x_k) = \operatorname{Proj}\left([-\kappa, \kappa]; \frac{\tilde{a}}{\tilde{b}} x_k\right) \tag{14a}$$

$$-u_{\mathrm{sv}}(x_k) = \kappa \operatorname{\mathbf{sgn}}(x_k) \min\left\{\frac{\tilde{a}}{\tilde{b}\kappa}|x_k|, 1\right\}. \tag{14b}$$

Proof

The proof follows directly from (11) and (12). Indeed, from (5), we have that

$$-u_{\mathrm{sv}}(x_k) = \frac{\tilde{a}x_k - x_{k+1}}{\tilde{b}} = \frac{1}{\tilde{b}}(I_{\mathrm{d}} - \operatorname{Prox}_{\tilde{b}\kappa g})(\tilde{a}x_k) \tag{15}$$

where we have used (11) to obtain the last equality. The substitution of (12) back into (15) yields

$$-u_{\mathrm{sv}}(x_k) = \kappa \operatorname{\mathbf{sgn}}(x_k) \min\left\{\frac{\tilde{a}}{\tilde{b}\kappa}|x_k|, 1\right\}. \tag{16}$$

Finally, the equivalence of the last expression with the projection map follows from direct computations. Clearly

$$\begin{aligned} \operatorname{Proj}\left([-\kappa, \kappa]; \frac{\tilde{a}}{\tilde{b}} x_k\right) &= \begin{cases} \frac{\tilde{a}}{\tilde{b}} x_k & \text{if } \frac{\tilde{a}}{\tilde{b}}|x_k| \le \kappa, \\ \kappa \operatorname{\mathbf{sgn}}(x_k) & \text{otherwise.} \end{cases} \\ &= \operatorname{\mathbf{sgn}}(x_k) \min\left\{\frac{\tilde{a}}{\tilde{b}}|x_k|, \kappa\right\} \\ &= \kappa \operatorname{\mathbf{sgn}}(x_k) \min\left\{\frac{\tilde{a}}{\tilde{b}\kappa}|x_k|, 1\right\}. \end{aligned} \tag{17}$$

■

Proximal Mapping and Proximal-Point Algorithm

Proximal operators and proximal-point algorithms are fundamental tools in optimization. The founding fathers of this field are Jean Jacques Moreau [S30], [S31], [S32], Kôsaku Yosida [S33], R. Tyrell Rockafellar [S17], [56], and Hedy Attouch [S34].

As seen in (11), (14), (76), (88), (98), and (99), proximal operators and proximal-point algorithms are important tools for the implicit discretization of subgradient systems. While the convergence properties of proximal-point algorithms have been thoroughly studied in the literature, the robustness of these algorithms in perturbed scenarios—which naturally arise in control applications—remains unknown in the field of optimization. Given a maximal monotone operator $\mathbf{M}:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$ (see "Maximal Monotone Operators"), its set of zeros is denoted by $\mathit{zero}\,\mathbf{M} = \mathbf{M}^{-1}(0) = \{\xi \in \mathbb{R}^n \mid 0 \in \mathbf{M}(\xi)\}$.

Definition S5

See [S28, p. 345]. Assume that $\mathbf{M}:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a maximal monotone operator satisfying $\mathit{zero}\,\mathbf{M} \neq \varnothing$, and let $x_0 \in \mathbb{R}^n$ and $\gamma > 0$. The *proximal-point algorithm* associated with $\mathbf{M}$ is defined as

$$x_{k+1} = \mathcal{J}_{\gamma\mathbf{M}}(x_k), \quad k \in \mathbb{N} \tag{S23}$$

where $\mathcal{J}_{\gamma\mathbf{M}}$ is the resolvent of $\mathbf{M}$.

Theorem 1 in [S35] establishes the convergence of (S23) toward the set zero $\mathbf{M}$. Given a convex, proper, and LSC (see Definition S6) function $f:\mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$, the operator

$$\mathit{Prox}_f = (I_d + \partial f)^{-1} \tag{S24}$$

is called the *proximal mapping* associated with f [S28, Def. 12.23]. By definition, when $\mathbf{M} = \partial f$ the algorithm (S23) is equivalent to any of the following iterations:

$$x_{k+1} = \mathrm{Prox}_{\gamma f}(x_k) \tag{S25}$$

$$x_{k+1} - x_k \in -\gamma \partial f(x_{k+1}) \tag{S26}$$

$$x_{k+1} = \operatorname*{argmin}_{\xi\in\mathbb{R}^n}\left\{ f(\xi) + \frac{1}{2\gamma} \| \xi - x_k \|^2 \right\}. \tag{S27}$$

The form (S25) gives the algorithm its name. In light of the usual first-order necessary conditions for optimality, convergence toward $\mathrm{zero}\,\mathbf{M} = \mathrm{zero}\,\partial f$ holds clear practical significance in optimization. Another useful property—readily seen from (S24)—is

$$\mathrm{zero}\,\mathrm{Prox}_f = \partial f(0). \tag{S28}$$

Also, note that if $f = \Psi_C$ (see Definition S9) with $C \subseteq \mathbb{R}^n$ a closed convex nonempty set, then

$$\mathrm{Prox}_f = \mathrm{Proj}\,(C;\cdot) = (I_d + N_C)^{-1}. \tag{S29}$$

When implicitly or semi-implicitly discretized, several higher-order differentiators and controllers involve the resolvent of the subdifferential of the convex, proper, and LSC function $f:\mathbb{R}\to\mathbb{R}$, $x \mapsto (1/2)\gamma x^2 + \Psi_{[-1,1]}(x)$, $\gamma > 0$, that is, $\partial f(x) = \gamma x + \mathbf{N}_{[-1,1]}(x)$, see "Convex Analysis Tools." This boils down to calculating the inverse of the set-valued function $\partial f' : x \mapsto (1+\gamma)x + \mathbf{N}_{[-1,1]}(x)$, which is a saturation

$$\mathcal{J}_{\partial f'}(y) = \begin{cases} -1 & \text{if } y \le -1-\gamma \\ \dfrac{1}{1+\gamma} y & \text{if } y \in [-1-\gamma, 1+\gamma] \\ 1 & \text{if } y \ge 1+\gamma \end{cases}. \tag{S30}$$

Remark S1

It can be readily seen from (S26) that the proximal-point algorithm corresponds to the backward (implicit) Euler discretization of the gradient system $\dot{x}(t) \in -\gamma/h\,\partial f(x(t))$, which is well known to be more efficient and have better approximation properties than its forward (explicit) Euler counterpart [57].

Implicit discrete-time sliding-mode controllers and differentiators will frequently yield perturbed iterations of the form

$$x_{k+1} = \mathcal{J}_{\gamma\mathbf{M}}(x_k + w_k). \tag{S31}$$

The next proposition is the discrete-time counterpart of Proposition S1.

Proposition S2

Robust proximal-point algorithm. Consider the perturbed proximal-point algorithm (S31) with $\gamma > 0$ and $\mathbf{M}:\mathbb{R}^n \rightrightarrows \mathbb{R}^n$ a maximal monotone operator such that

$$(\gamma\mathbf{M})^{-1}\left(\bigcup_{k\in\mathbb{N}} w_k\right) = \{0\}. \tag{S32}$$

The origin $x = 0$ is globally asymptotically stable. If, moreover

$$(\gamma\mathbf{M})^{-1}\left(\bigcup_{k\in\mathbb{N}} w_k + \varepsilon\mathcal{B}\right) = \{0\} \tag{S33}$$

for some $\varepsilon > 0$, then it is globally finite-time stable.

Proof

From the definition of the Yosida approximation (S20), it follows that $x_{k+1} = x_k + w_k - \gamma\mathcal{Y}_{\gamma\mathbf{M}}(x_k + w_k)$. Thus

$$\begin{aligned} \| x_{k+1} \|_2^2 &= x_{k+1}^\top (x_k + w_k - \gamma\mathcal{Y}_{\gamma\mathbf{M}}(x_k + w_k)) \\ &\le \frac{1}{2}\| x_{k+1} \|_2^2 + \frac{1}{2}\| x_k \|_2^2 \\ &\quad - \mathcal{J}_{\gamma\mathbf{M}}(x_k + w_k)^\top (\gamma\mathcal{Y}_{\gamma\mathbf{M}}(x_k + w_k) - w_k) \end{aligned} \tag{S34}$$

where we used the fact that $\langle \mathrm{x}, \mathrm{y}\rangle \le (1/2)\| \mathrm{x} \|_2^2 + (1/2)\| \mathrm{y} \|_2^2$. Setting the Lyapunov function candidate as $V_{k+1} = (1/2)\| x_{k+1} \|_2^2$, we obtain

$$V_{k+1} - V_k \le -\mathcal{J}_{\gamma\mathbf{M}}(x_k + w_k)^\top (\gamma\mathcal{Y}_{\gamma\mathbf{M}}(x_k + w_k) - w_k). \tag{S35}$$

In what follows, it is shown that the right-hand side of (S35) is negative definite. To that end, we point out that assumption (S33) implies that, for any $b \in \mathcal{B}$ and any $k \in \mathbb{N}$, $\varepsilon b - w_k \in \gamma\mathbf{M}(0)$. As a consequence, from (S22) and the maximal monotonicity of $\gamma\mathbf{M}$, for all $k \in \mathbb{N}$ it holds that

(*Continued*)

Proximal Mapping and Proximal-Point Algorithm (*Continued*)

$$\langle \mathcal{J}_{\gamma \mathbf{M}}(x_k + w_k), \gamma \mathcal{Y}_{\gamma \mathbf{M}}(x_k + w_k) - \varepsilon b - w_k \rangle \geq 0. \tag{S36}$$

After rearranging terms, we can see that

$$\langle \mathcal{J}_{\gamma \mathbf{M}}(x_k + w_k), \gamma \mathcal{Y}_{\gamma \mathbf{M}}(x_k + w_k) - w_k \rangle \geq \varepsilon \sup_{b \in \mathcal{B}} \langle b, \mathcal{J}_{\gamma \mathbf{M}}(x_k + w_k) \rangle. \tag{S37}$$

That is

$$\langle \mathcal{J}_{\gamma \mathbf{M}}(x_k + w_k), \gamma \mathcal{Y}_{\gamma \mathbf{M}}(x_k + w_k) - w_k \rangle \geq \varepsilon \| \mathcal{Y}_{\gamma \mathbf{M}}(x_k + w_k) \| \geq \varepsilon \| x_{k+1} \| \tag{S38}$$

where the last inequality follows from (S31). Therefore, $V_{k+1} - V_k \leq -\varepsilon \| x_{k+1} \|$, and the global asymptotic stability of the origin follows. Now, for the finite-time convergence, notice that (S31) is equivalent to

$$x_k + w_k \in (I_d + \gamma \mathbf{M})(x_{k+1}) \tag{S39}$$

whereas (S33) and the asymptotic stability of the origin imply that there is $k^* \in \mathbb{N}$ and $\varepsilon_2 > 0$ such that, for all $k \geq k^*$ and any $b \in \mathcal{B}$

$$w_k + x_k + \varepsilon_2 b \in (I_d + \gamma \mathbf{M})(0). \tag{S40}$$

Thus, (S39), (S40), and the maximal monotonicity of $I_d + \gamma \mathbf{M}$ imply that, for all $b \in \mathcal{B}$, we have

$$\varepsilon_2 \langle b, x_{k+1} \rangle \geq 0. \tag{S41}$$

It thus follows that $x_{k+1} = 0$ for all $k \geq k^*$. ■

REFERENCES

[S30] J. J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *C.R. Hebd. Seances Acad. Sci.*, vol. 255, pp. 2897–2899, Nov. 1962.

[S31] J. J. Moreau, "Propriétés des applications "prox," *C.R. Hebd. Seances Acad. Sci.*, vol. 256, pp. 1069–1071, Jan. 1963.

[S32] J. J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bull. Soc. Math.* France, Tome, vol. 93, no. 1, pp. 273–299, 1965, doi: 10.24033/bsmf.1625.

[S33] K. Yosida, *Functional Analysis*, Berlin, Germany: Springer-Verlag, 1974.

[S34] H. Attouch, G. Buttazzo, and G. Michaille, *Variational Analysis in Sobolev and BV Spaces*. Philadelphia, PA, USA: SIAM, 2014.

[S35] E. K. Ryu and W. Yin, *Large-Scale Convex Optimization via Monotone Operators*. Cambridge, U.K.: Cambridge Univ. Press, 2023.

The expressions in (13) and (14) can also be written using complementarity problems and variational inequalities, using equivalent formulations of normal cones [see (S51) for the variational formulation, and see "Projected Dynamical Systems" for the complementarity formulation of a class of nonsmooth systems that are close to the sliding-mode ones]. The signum multifunction can also be rewritten using various formalisms; see [61, "Set-Valued Signum Function"]. The $\min\{\cdot,\cdot\}$ function in (14b) and in (13) is a *complementarity function* (or *C*-function) [62], [63]. The expression (14a) shows that $u_{\mathrm{sv}}(x_k)$ is the solution of the quadratic optimization problem

$$-u_{\mathrm{sv}}(x_k) = \arg\min_{z \in [-\kappa, \kappa]} \frac{1}{2} \left\| z - \frac{\tilde{a}}{\tilde{b}} x_k \right\|^2. \tag{18}$$

This suggests that implicit controllers can be implemented online by solving quadratic programs with constraints.

Some Comments

It is noteworthy that the explicit form (14b) has been studied in [66, Ex. 3.1] via Lyapunov methods [see also [49], where the same algorithms were already presented, and [67], [68], and [69] for further analysis, where the form (14b) is called a minimum-operator-based input]. The connection with proximal maps via the implicit-Euler method was shown in [51] and [70]. It is worth emphasizing that, in the limit as $h \downarrow 0$, the trajectories of both schemes, (7) and (9), converge toward the trajectories of the continuous-time inclusion (1) and (2). However, the control input (14) is well defined at $x = 0$, whereas (6) is not. Thus, even though in the implementation of discrete-time controllers, the sampling time $h > 0$ cannot be arbitrarily small due to practical limitations, it is clear that the implicitly defined system (9) is more attractive than the explicit one. This distinction becomes more important when disturbances are considered.

To close this section, we emphasize that arbitrary regularizations of the set-valued sign function (that is, single-valued functions sufficiently close to the graph of the sign multifunction, like the saturation or the sigmoid functions) do not guarantee the global finite-time convergence property. Generically, such an approximation either breaks the global finite-time convergence toward zero (making the convergence asymptotic) or the stability of the iteration map, making the origin unstable and producing an oscillatory behavior (see "Proximal Mapping and Proximal-Point Algorithm").

The following section extends the ideas presented previously to the multivariable case, where nonvanishing external disturbances affecting the dynamics are also considered. Moreover, a control-theoretic interpretation for the use of implicitly defined controllers is presented under the general framework of passive systems.

FIRST-ORDER SMC: CONTINUOUS TIME

In this section, we argue that first-order SMC is a particular case of passivity-based control. This observation leads to a

Projected Dynamical Systems

Projected dynamical systems (PDSs) are a class of nonsmooth systems that, as seen later, are closely related to sliding-mode systems. They were introduced in [S36] and [S37]. Recently, they have been the subject of substantial interest from the automatic-control scientific community; see, for example, [S38] and [S39]. PDSs may be expressed under different equivalent forms [S40], [S41] (see definitions 8 and 11 in "Convex Analysis Tools"). Some examples of such equivalent dynamics are as follows:

$$\begin{aligned} &\dot{x} \in -f(x) - g(t) - \mathbf{N}_C(x), \\ &\text{with } \dot{x} = -f(x) - g(t) - \text{Proj}\left[\mathbf{N}_C(x); -f(x) - g(t)\right] \end{aligned} \tag{S42a}$$

$$\dot{x} \in -f(x) - g(t) - \mathbf{N}_{\mathbf{T}_C(x)}(\dot{x}) \tag{S42b}$$

$$\dot{x} = (I_d + \mathbf{N}_{\mathbf{T}_C(x)})^{-1}(-f(x) - g(t)) \tag{S42c}$$

$$\dot{x} = \text{Proj}[\mathbf{T}_C(x); -f(x) - g(t)]. \tag{S42d}$$

See [S40] and [S41] for details. The second condition in (S42a) means that solutions are "slow" [that is, with $-\dot{x}(t)$ of minimal norm inside $f(x) + g(t) + \mathbf{N}_C(x)$]. The resolvent of the normal cone to the tangent cone appears in (S42c). Use was made of Fact 5 in "Convex Analysis Tools" (S24) and the comment just after (S28) in "Proximal Mapping and Proximal-Point Algorithm" to obtain (S42d). The projection is a particular instance of Prox; see (S29), showing that PDSs belong to the class of proximal systems.

DISCRETE-TIME PDSs

Let us now discretize (S42a) as

$$x_{k+1} \in x_k - h(f(x_k) - g_k) - \mathbf{N}_C(x_{k+1}) \tag{S43}$$

$h > 0$. This is rewritten equivalently as

$$x_{k+1} = (I_d + \mathbf{N}_C)^{-1}(x_k - h(f(x_k) + g_k)) \tag{S44}$$

and as

$$x_{k+1} = \text{Proj}[C; x_k - h(f(x_k) + g_k)] \tag{S45}$$

where the same tools as described previously have again been used to pass from (S42c) to (S42d). We see once again that resolvents are ubiquitous in PDSs. The system (S45) furnishes a convenient way to obtain solutions whenever the projection onto the set C can be computed. This is the case when C is nonempty convex and polyhedral (S67). Indeed, in this case, the normal cone can be expressed as $\mathbf{N}_C = \{\xi \in \mathbb{R}^n \,|\, \xi = G^\top \lambda, 0 \le \lambda \perp -G\xi + b \ge 0\}$. In other words, the normal cone is generated by the outward normals to the active constraints at x. Thus, using (S43), the PDS is equivalently rewritten as a complementarity system

$$\begin{aligned} &x_{k+1} = x_k - h(f(x_k) - g_k) - G^\top \lambda_{k+1}, \\ &0 \le \lambda_{k+1} \perp w_{k+1} = -Gx_{k+1} + b \ge 0. \end{aligned} \tag{S46}$$

Note that the discrete-time PDS in (S46) is also a mixed linear complementarity problem (LCP) [S42] with unknowns x_{k+1} and λ_{k+1}. It gives the LCP

$$\begin{aligned} &0 \le \lambda_{k+1} \perp w_{k+1} = -G(x_k - h(f(x_k) - g_k)) \\ &\quad + GG^\top \lambda_{k+1} + b \ge 0 \end{aligned} \tag{S47}$$

Obviously, $GG^\top \succ 0$ if and only if G has row-rank m (which implies that $m \le n$); otherwise, $GG^\top \succcurlyeq 0$, and it is also a copositive-plus matrix. Using [S42, Thm. 3.8.6], if $b \in (\text{SOL}(GG^\top, 0))^*$, where $\text{SOL}(M, 0)$ denotes the set of solutions to $0 \le \lambda \perp M\lambda \ge 0$, and $(\text{SOL}(M, 0))^*$ is its dual cone, then the LCP in (S47) has a solution. Assuming a Slater condition [that is, that there exists λ such that $GG^\top \lambda - G(x_k - h(f(x_k) - g_k)) + b > 0$], then by [S42, Thm. 5.3.9], the LCP in (S47) has a solution. Also, if λ_1 and λ_2 are any two solutions, then $GG^\top(\lambda_1 - \lambda_2) = 0$ [S42, Thm. 3.1.7]. Therefore $\lambda_1 - \lambda_2 \in Ker(GG^\top) = Ker(G^\top)$ [64, Thm. 3.5.3]. It is easily deduced that, if

$$x^i_{k+1} = x^i_k + h(f(x^i_k) - g_k) - G^\top \lambda^i_{k+1} \tag{S48}$$

$i = 1, 2$, then $x^1_k = x^2_k$ implies $x^1_{k+1} = x^2_{k+1}$, $k \ge 0$. As expected, the projection onto C is unique. A quite interesting fact is that efficient algorithms can be used to solve the LCP in (S47) [S42], and [62], at each time step $k \ge 1$.

SLIDING MODES IN PDSs

Sliding modes can occur in PDSs when the single-valued vector field is such that the trajectory evolves persistently on the boundary of C. In this case, there exists an element of $\lambda(x, t) \in -\mathbf{N}_C(x)$ (a selection; see Definition S3 in "Set-Valued Mappings and DIs") that "compensates" for the part of $-f(x) - g(t)$, which tends to "push" the trajectory outside C (in a way quite similar to what occurs in contact mechanics where the interaction contact force balances the other forces, which otherwise would make the system leave the admissible domain). It is noteworthy that the set-valued controlled systems in [65, eq. (6)] can be transformed into a PDS after the classical state space transformation (see [S41, Sect. 3.4]). This guarantees robust output tracking.

REFERENCES

[S36] C. Henry, "Differential equations with discontinuous right-hand side for planning procedures," *J. Econ. Theory*, vol. 4, no. 3, pp. 545–551, 1972, doi: 10.1016/0022-0531(72)90138-X.

[S37] C. Henry, "An existence theorem for a class of differential equations with multivalued right- hand side," *J. Math. Anal. Appl.*, vol. 41, no. 1, pp. 179–186, 1973, doi: 10.1016/0022-247X(73)90192-3.

[S38] D. Gadjov and L. Pavel, "A passivity-based approach to Nash equilibrium seeking over networks," *IEEE Trans. Autom. Control*, vol. 64, no. 3, pp. 1077–1092, Mar. 2019, doi: 10.1109/TAC.2018.2833140.

[S39] L. Pavel, "Dissipativity theory in game theory: On the role of dissipativity and passivity in Nash equilibrium seeking," *IEEE Control Syst. Mag.*, vol. 42, no. 3, pp. 150–164, Jun. 2022, doi: 10.1109/MCS.2022.3157119.

[S40] B. Brogliato, A. Daniliidis, C. Lemaréchal, and V. Acary, "On the equivalence between complementarity systems, projected systems and differential inclusions," *Syst. Control Lett.*, vol. 55, no. 1, pp. 45–51, 2006, doi: 10.1016/j.sysconle.2005.04.015.

[S41] B. Brogliato and A. Tanwani, "Dynamical systems coupled with monotone set-valued operators: Formalisms, applications, well-posedness, and stability," *SIAM Rev.*, vol. 62, no. 1, pp. 3–129, 2020, doi: 10.1137/18M1234795.

[S42] R. W. Cottle, J. S. Pang, and R. E. Stone, *The Linear Complementarity Problem*. New York, NY, USA: Academic Press, 1992.

new method for designing sliding-mode controllers and establishes the grounds for the following section, which is about discrete-time SMC.

Consider the perturbed linear dynamics

$$\dot{x}(t) = Ax(t) + B(u(t) + \delta(t)) \tag{19}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $\delta(t) \in \mathbb{R}^m$ are, respectively, the state, the control input, and the perturbation at time t. Accordingly, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$. It is assumed that $\operatorname{rank}(B) = m$ and that $\delta(t)$ is uniformly bounded.

The Classical Approach

In the classical approach to SMC, the robust stabilization of (19) is undertaken in two steps.

1) Choose a linear map $\sigma : x \mapsto Cx$ and define the *sliding variable* $s = \sigma \circ x$. The matrix $C \in \mathbb{R}^{m\times n}$ is such that $\det(CB) \neq 0$ and such that $s(t) \equiv 0$ implies $x(t) \to 0$ as $t \uparrow +\infty$.
2) Design a control law $u(t)$ such that $s(t)$ converges to zero in finite time.

Step 1 is typically completed by putting the system in regular form and solving a lower-dimensional stabilization problem [53], [54]. To perform Step 2, the time derivative $\dot{s}(t) = CAx(t) + CB(u(t) + \delta(t))$ is first computed. By setting $u(t) = u_{\text{lin}}(x(t)) + (CB)^{-1}v(t)$, where

$$u_{\text{lin}}(x) = -(CB)^{-1}CAx \tag{20}$$

is a linear feedback and $v(t)$ is a new input; the dynamics for s are simplified to

$$\dot{s}(t) = v(t) + CB\delta(t). \tag{21}$$

The design is completed by fixing the new input as $v(t) = u_{\text{sv}}(x(t), t)$ with

$$u_{\text{sv}}(x, t) \in -K\,\mathbf{Sgn}(\sigma(x)), \quad K = K^\top \succ 0 \tag{22}$$

for all $t \in \mathbb{R}$. The term $u_{\text{sv}}(x, t)$ is a selection of the set-valued signum function [see Definition S3 in "Set-Valued Mappings and DIs" and (S50) and (S58) in "Convex Analysis Tools"]. With the controls (20) and (22), the sliding variable evolves according to the DI [which belongs to the class (S7)]

$$\dot{s}(t) \in -K\,\mathbf{Sgn}(s(t)) + CB\delta(t). \tag{23}$$

To verify that Step 2 is correctly achieved, note that the time derivative of the Lyapunov function $V_s(s) = (1/2)s^\top K^{-1}s$ along the closed-loop trajectories is

$$\begin{aligned}\dot{V}_s(s(t)) &= -s(t)^\top(\mathbf{Sgn}(s(t)) - K^{-1}CB\delta(t)) \\ &= -\|s(t)\|_1 + s(t)^\top K^{-1}CB\delta(t) \\ &\leq -(1 - \|K^{-1}CB\delta(t)\|_\infty)\|s(t)\|_1.\end{aligned}$$

Using classical arguments, the finite-time convergence of $V_s(s)$ to zero is inferred provided that $\|K^{-1}CB\delta(t)\|_\infty) < 1$ (using, for example, [64, Prop. 11.1.6]). Therefore, there exists some $t_{\min} < +\infty$ after which a sliding mode occurs. That is, $s(t) = 0$, and therefore

$$u_{\text{sv}}(x(t), t) = -CB\delta(t) \tag{24}$$

for all $t \geq t_{\min}$. In other words, the controller compensates exactly for the unknown perturbation. Certainly, this continuous-time property is a "miracle" of set-valued feedback control, adding to the *miracle of feedback stabilization* [71]. The exactness (equality) in (24) is obviously due to the idealization of the modeling. However, as demonstrated by the implicit discrete-time analysis in (100) and by experimental results, it is the idealization of an observed behavior. In particular, the insensitivity of the dynamics with respect to K during the sliding mode is experimentally observed in both differentiation and control when the implicit method is used. This emphasizes the fact that (24) is far from merely a mathematical curiosity.

A Passivity-Based Approach

We will now explore the classical approach through a passivity point of view (see "Passive Systems"). This will later be used to show the necessity of an implicit discretization.

Remark 1

Item 1 previously implies that the system with output s is feedback equivalent to a passive one. In fact, a system is passive feedback *if and only if* it is of relative degree $\{1, \ldots, 1\}$, that is, $\det(CB) \neq 0$ and minimum phase, that is, $s(t) \equiv 0$ implies $x(t) \to 0$. The linear feedback u_{lin} renders (19) passive with output s and input $v + CB\delta$ (see [74] for details on the unperturbed case). In this regard, first-order SMC is a particular instance of passivity-based control.

In addition to passivity, a fundamental notion is implicitly used in the previous analysis: maximal monotonicity (see "Maximal Monotone Operators"). Following [60], the change of variable $s' = K^{-(1/2)}s$ may be performed, transforming (23) into

$$\dot{s}'(t) \in -K^{\frac{1}{2}}\,\mathbf{Sgn}(K^{\frac{1}{2}}s'(t)) + K^{-\frac{1}{2}}CB\delta(t).$$

The DI can be written as

$$\dot{s}'(t) \in -\partial f(s'(t)) + K^{-\frac{1}{2}}CB\delta(t) \tag{25}$$

where $f(\cdot) = \|K^{1/2}\cdot\|_1$. The chain rule was used (see Theorem S2 in "Maximal Monotone Operators") to compute ∂f, the subdifferential of f (see Definition S7 in "Convex Analysis Tools"). The operator ∂f is a maximal monotone operator, f being a convex continuous proper

Convex Analysis Tools

Convexity allows us to draw connections among monotonicity, passivity, optimality, and geometry. The following results and definitions are taken from classical references [S18], [S17], [S23], [S28], [63]. Most of the tools described later extend to nonconvex sets and functions. However, it is essentially the convex case that is of interest to us in this article.

CONVEX FUNCTIONS AND SUBDIFFERENTIAL

In convex analysis, it is useful to extend the space $\mathbb{R}$ by allowing a function to take the value $+\infty$. A function $f: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is said to be *proper* if it is not identically infinite.

Definition S6

A proper convex function $f: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is LSC if its epigraph is closed. As alluded to in "Set-Valued Mappings and DIs" and Definition S1, semicontinuity properties exist for both single-valued and set-valued mappings. The following definition shows that, given an LSC single-valued function, we can construct a set-valued mapping. This set-valued mapping is, in turn, OSC, as can be verified using Theorem S2 and Fact 4.

Definition S7

Let $f: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, convex, and LSC function. The convex *subdifferential* of f at the point x is given by

$$\partial f(x) := \{\eta \in \mathbb{R}^n \mid \langle \eta, \xi - x \rangle \leq f(\xi) - f(x) \text{ for all } \xi \in \operatorname{dom} f\}. \tag{S49}$$

Note that $\partial f: \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is, in general, a set-valued mapping. Consider, for example, the real-valued function $f(x) = \kappa|x|$ with $\kappa > 0$. We have $\partial f(x) = \kappa \mathbf{sgn}(x)$ with

$$\mathbf{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}. \tag{S50}$$

We introduce another important set-valued mapping.

Definition S8

Let $C \subseteq \mathbb{R}^n$ be a closed convex set. Its *normal cone* is the set-valued mapping $\mathbf{N}_C: \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ defined by

$$\mathbf{N}_C(x) = \{\eta \in \mathbb{R}^n \mid \langle \eta, \xi - x \rangle \leq 0 \text{ for all } \xi \in C\} \tag{S51}$$

if $x \in C$ and $\mathbf{N}_\kappa(C) = \varnothing$ if $x \notin C$.

Note that the normal cone to a closed nonempty convex set is convex-valued. We now introduce a class of single-valued functions that undergo discontinuities (with infinitely large jumps). Despite their strong nonsmoothness, these functions possess a subdifferential. They are useful in mechanics and in optimization.

Definition S9

Let $C \subseteq \mathbb{R}^n$ be a set. Its *indicator function* is defined as $\psi_C(x) = 0$ if $x \in C$ and $\Psi_C(x) = +\infty$ if $x \notin C$.

The next fact shows that normal cones are, in fact, subdifferentials.

Fact 5

Let $C \subseteq \mathbb{R}^n$ be a closed convex set. Then, the indicator function Ψ_C is proper, convex, LSC, and $\partial \Psi_C(x) = \mathbf{N}_C(x)$ for all $x \in \mathbb{R}^n$.

It is seen throughout this article that normal cones, and hence indicator functions, are ubiquitous in generalized equations that appear in discrete-time backward-Euler schemes. The following is a simple consequence of Fact 5. It has interesting applications in constrained optimization problems and in the computation of discrete-time SMC laws.

Fact 6

The normal cone to a closed convex set defines a maximal monotone mapping. The calculation of normal cones is often necessary in applications. Suppose that C is finitely represented, that is, $C = \{\xi \in \mathbb{R}^n \mid g(\xi) \geq 0\}$ for some differentiable function $g: \mathbb{R}^n \to \mathbb{R}^m$. Assume further that the Mangasarian-Fromovitz constraint qualification holds [S17]. Then, $\mathbf{N}_C(x)$ is generated by the outward normals $\nabla g_i(x)$ to C at the active constraints $g_i(x) = 0$, $i \in \{1, \ldots, m\}$. In other words, $\mathbf{N}_C(x) = \{\eta \in \mathbb{R}^n \mid \eta = -\nabla g(x)\lambda,\ 0 \leq \lambda \perp g(x) \geq 0\}$. It is noteworthy that weighted inner products can be used in the normal-cone definition. For instance, $\langle \eta, \xi - x \rangle_M$ with $M = M^\top \succ 0$ can be used in (S51), in which case the normals are calculated as $M^{-1}\nabla g_i(x)$.

The next result, known as the *chain rule* of convex analysis, is useful in the computation of subdifferentials. It is used in several places in this article.

Fact 7

Let $f: \mathbb{R}^n \to \mathbb{R}$ be a proper convex LSC function, and let $A: \mathbb{R}^m \to \mathbb{R}^n$ be a linear mapping. Assume that f is a polyhedral function (that is, its epigraph is a polyhedral set) or that there exists x_0 with $Ax_0 \in \operatorname{dom}(f)$ such that $\operatorname{rge}(A) - \mathbb{R}_+(\operatorname{dom}(f) - Ax_0)$ is a vector subspace of $\mathbb{R}^n$. Then, the subdifferential in the sense of convex analysis of the composite function $f \circ A: \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ is given by $\partial(f \circ A)(x) = A^\top \partial f(Ax)$ for all $x \in \mathbb{R}^m$.

CONJUGATE FUNCTIONS, MAPPING INVERSION

The following concept of conjugacy is useful for inverting subdifferential mappings.

Definition S10

Let $f: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, convex, and LSC function. Its *convex conjugate*, $f^*: \mathbb{R}^n \to \mathbb{R}$, is defined as

$$f^*(y) = \sup_{\xi \in \mathbb{R}^n} \{\xi^\top y - f(\xi)\}. \tag{S52}$$

(*Continued*)

Convex Analysis Tools (*Continued*)

Fact 8

The convex conjugate of a proper convex LSC function is also proper, convex, and LSC. Moreover, $(f^\star)^\star = f$.

Another important result that relies on conjugacy (and is often used in calculations) is the following one.

Fact 9

Let $f:\mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ be proper, convex, and LSC. Then, $(\partial f)^{-1} = \partial f^\star$ and $\partial f = (\partial f^\star)^{-1}$. In other words, $y \in \partial f(x)$ if and only if $x \in \partial f^\star(y)$.

As an example, consider again the real-valued function $f(x) = \kappa|x|$ with $\kappa > 0$. Its convex conjugate is

$$f^\star(y) = \sup_{\xi\in\mathbb{R}} \{\xi^\top y - \kappa|\xi|\}. \tag{S53}$$

By concavity, the supremum is actually a maximum whenever the subdifferential of $\xi^\top y - \kappa|\xi|$ contains the origin, that is, whenever $y \in \kappa\mathbf{sgn}(\xi)$. Thus, when $y \in [-\kappa,\kappa]$, a maximum exists, and $f^\star(y) = 0$. On the other hand, when $y \notin [-\kappa,\kappa]$, there is no maximum, and by piecewise linearity, $f^\star(y) = +\infty$. In other words, the convex conjugate equals the indicator function of a closed interval, $f^\star = \Psi_{[-\kappa,\kappa]}$. Fact 5 is easy to verify in this example

$$\partial\Psi_{[-\kappa,\kappa]}(y) = \mathbf{N}_{[-\kappa,\kappa]}(y) = \begin{cases} (-\infty,0] & \text{if } y = -\kappa \\ 0 & \text{if } y \in (-\kappa,\kappa) \\ [0,+\infty) & \text{if } y = \kappa \\ \varnothing & \text{otherwise} \end{cases}. \tag{S54}$$

Verifying Fact 9 is also straightforward

$$\mathbf{N}_{[-\kappa,\kappa]} = (\kappa\mathbf{sgn})^{-1}. \tag{S55}$$

More generally, let $\|x\|_p$ be the p norm $(\Sigma_{i=1}^n |x_i|^p)^{1/p}$, $1 \le p < +\infty$, and $\|x\|_\infty = \max_{1\le i\le n}|x_i|$. The dual norm is $\|x\|_{p^\star}$. with $1/p + 1/p^\star = 1$. It is known that if $f = \kappa\|\cdot\|_p$, $\kappa > 0$, then $f^\star = \Psi_{\kappa\mathcal{B}_{p^\star}}$, where

$$\mathcal{B}_{p^\star} = \{\xi \in \mathbb{R}^n \,|\, \|\xi\|_{p^\star} \le 1\} \tag{S56}$$

is the unit ball in the $p^\star$ norm. Thus, by Fact 5 and Fact 9

$$\partial\kappa\|\cdot\|_p = (\partial\Psi_{\kappa\mathcal{B}_{p^\star}})^{-1} = (\mathbf{N}_{\kappa\mathcal{B}_{p^\star}})^{-1}. \tag{S57}$$

For example, setting $p = 1$ gives $p^\star = \infty$, $\mathcal{B}_\infty = [-1,1]^n$, and

$$\kappa\mathbf{Sgn}(x) = \mathbf{N}^{-1}_{\kappa\mathcal{B}_\infty}(x) \tag{S58}$$

with $\mathbf{Sgn}(x) = (\mathbf{sgn}(x_1), \mathbf{sgn}(x_2), \ldots, \mathbf{sgn}(x_m))^\top$. For $p = 2$ we have

$$\mathbf{N}^{-1}_{\kappa\mathcal{B}_2}(x) = \kappa\partial\|x\|_2 = \begin{cases} \kappa x/\|x\|_2 & \text{if } x \ne 0 \\ \kappa\mathcal{B}_2 & \text{if } x = 0 \end{cases}. \tag{S59}$$

The following theorem summarizes the inverse properties of the convex conjugate and is helpful in performing Lyapunov analysis of DIs with right-hand subgradient operators.

Theorem S4 [S28, Thm. 16.23]

See [S28, Thm. 16.23]. Let $f:\mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ be proper, convex, and LSC. Then, the following statements are equivalent:

$$\begin{aligned} &(x,y) \in \operatorname{gph}\partial f \\ &(y,x) \in \operatorname{gph}\partial f^\star \\ &x^\top y = f(x) + f^\star(y). \end{aligned} \tag{S60}$$

PROJECTIONS ONTO CONVEX SETS

The *projection* of a point $x \in \mathbb{R}^n$ onto the nonempty closed convex set $C \subseteq \mathbb{R}^n$ with inner product weighted by $M = M^\top \succ 0$ is defined by

$$\operatorname{Proj}_M[C;x] = \operatorname*{argmin}_{\xi\in C} \frac{1}{2}\langle \xi - x, \xi - x\rangle_M. \tag{S61}$$

For simplicity, write $\operatorname{Proj}[C;x]$ in place of $\operatorname{Proj}_I[C;x]$. The projection onto $\mathcal{B}_2$ is straightforward

$$\operatorname{Proj}[\mathcal{B}_2;x] = \begin{cases} \dfrac{x}{\|x\|_2} & \text{if } \|x\|_2 \ge 1 \\ x & \text{otherwise} \end{cases}. \tag{S62}$$

For $\mathcal{B}_\infty$, the projection is easily carried out componentwise

$$(\operatorname{Proj}[\mathcal{B}_\infty,x])_i = \min\{|x_i|,1\}\mathbf{sgn}(x_i), \quad i = 1,\ldots,n. \tag{S63}$$

Projections can also be computed using normal cones.

Proposition S3

Consider a vector $x \in \mathbb{R}^n$, a closed convex set $:C \subseteq \mathbb{R}^n$, and a weighted inner product $\langle\cdot,\cdot\rangle_M$. The following statements are equivalent:

$$y = \operatorname{Proj}_M[C;x]. \tag{S64a}$$

$$M(x-y) \in \mathbf{N}_C(y). \tag{S64b}$$

$$y = (M + \mathbf{N}_C)^{-1}(Mx). \tag{S64c}$$

Proof

The proof can be found in [63, p.79] inside another proof. It is reproduced here for convenience. The optimization problem (S61) can be equivalently written as $\operatorname{Proj}_M[C;x] = \operatorname{argmin}_{\xi\in\mathbb{R}^n}(1/2)\langle\xi - x,\xi - x\rangle_M + \Psi_C(\xi)$. By the convexity of the weighted norm and of the indicator function, it follows that $y = \operatorname{Proj}_M[C;x]$ if and only if $0 \in M(y-x) + \partial\Psi_C(y)$, that is, if and only if $M(x-y) \in \mathbf{N}_C(y)$, or equivalently, if and only if $Mx \in (M + \mathbf{N}_C)(y)$. Hence, $y \in (M + \mathbf{N}_C)^{-1}(Mx)$. ■

In addition, if C is a cone, then the statements (S64) are equivalent to [S23, Corollary 23.5.4]

$$C^\star \ni M(y-x) \perp y \in C \tag{S65}$$

(*Continued*)

Convex Analysis Tools (*Continued*)

where

$$C^* = \{\eta \in \mathbb{R}^n \,|\, \langle \eta, \xi \rangle \geq 0 \text{ for all } \xi \in C\} \tag{S66}$$

is the *dual cone* to C.

If C is a polyhedral set, then there exists a matrix $G \in \mathbb{R}^{m \times n}$ and a vector $b \in \mathbb{R}^m$ such that

$$C = \{\xi \in R^n \,|\, G\xi \leq b\}. \tag{S67}$$

For instance, if $C = \mathcal{B}_\infty$, then $G = [I_n, -I_n]^\top$ and $b = \mathbf{1}_{2n}$. For polyhedral sets, the projection (S64a) can be computed numerically by solving a conventional quadratic program. The following code shows an implementation in Python 3 using the open source modeling language for convex optimization CVXPY [72].

```
import cvxpy as cvx
xi = cvx.Variable(n)
objective = 0.5*cvx.quad_form(xi, M) -
    x.T@M@xi
cvx.Problem(cvx.Minimize(objective),
    G@xi <= b).solve()
y = xi.value
```

If $C = \mathcal{B}_1$, it is possible to use the previous code with $b = 1_n$, $G \in \mathbb{R}^{2n \times n}$, and the rows of G covering all possible vectors with entries in $\{-1, 1\}$. For large problems, a more efficient solution consists of splitting ξ into its positive and negative components, that is, $\xi = \xi^+ - \xi^-$, where $\xi^+, \xi^- \in \mathbb{R}^n_+$. In such a case, the following code solves the projection problem by setting:

$$H = \begin{bmatrix} \mathbf{1}_n^\top & \mathbf{1}_n^\top \\ -I_n & 0_{n \times n} \\ 0_{n \times n} & -I_n \end{bmatrix}, c = \begin{bmatrix} \mathbf{1} \\ 0_{n \times 1} \\ 0_{n \times 1} \end{bmatrix}.$$

```
import cvxpy as cvx
xi_p = cvx.Variable(n)
xi_n = cvx.Variable(n)
objective = 0.5*cvx.quad_form(xi_p -
xi_n, M) - x.T@M@(xi_p - xi_n)
cvx.Problem(cvx.Minimize(objective),
H@cvx.bmat([[xi_p], [xi_n]]) <= c).solve()
y = xi_p.value - xi_n.value
```

TANGENT CONES TO CONVEX SETS

Tangent cones are relevant in the context of the systems described in "Projected Dynamical Systems."

Definition S11

Let $C \subseteq \mathbb{R}^n$ be a nonempty closed convex set. Its *tangent cone* is the set-valued mapping $\mathbf{T}_C : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ defined by

$$\mathbf{T}_C(x) = \{\xi \in \mathbb{R}^n \,|\, \langle \eta, \xi \rangle \leq 0, \text{ for all } \eta \in \mathbf{N}_C(x)\}. \tag{S68}$$

Thus, $\mathbf{T}_C(x) = -(\mathbf{N}_C(x))^*$; see (S66) for the dual cone.

function (see Theorem S2 in "Maximal Monotone Operators"). The DI has the form (S7), so if the eigenvalues of K are large enough, it satisfies the conditions of Proposition S1 in "Set-Valued Mappings and DIs," and the origin $s' = 0$ is finite-time stable.

We now introduce a different and less-classical design directly based on passivity and monotonicity arguments. Later, this will prove useful in designing discrete-time sliding-mode controllers. Recall that the pair (A, B) is stabilizable if and only if there exists $P \in \mathbb{R}^{n \times n}$ such that $P = P^\top \succ 0$ and [75]

$$AP + PA^\top \prec BB^\top. \tag{26}$$

Similarly to [44] and [76], this inequality is exploited to construct the candidate storage function

$$V(x) = \frac{1}{2} x^\top P^{-1} x. \tag{27}$$

Its time derivative is

$$\dot{V}(x) = \frac{1}{2} x^\top (P^{-1}A + A^\top P^{-1}) x + x^\top P^{-1} B(u + \delta). \tag{28}$$

(When clear from context, the explicit dependence on t is omitted.) The only candidate passive output of relative degree equal to one and associated with the storage function (27) is $s(t) = \sigma(x(t))$ with

$$\sigma(x) = B^\top P^{-1} x \tag{29}$$

which corresponds to the passivity of the quadruple (A, B, C, D) with $C = B^\top P^{-1}$, $D = 0$, P^{-1} a solution of (S71), supply rate $u^\top y$; see "Passive Systems." This gives the energy balance

$$\dot{V}(x) = \frac{1}{2} x^\top (P^{-1}A + A^\top P^{-1}) x + s^\top (u + \delta). \tag{30}$$

Since (26) can be written equivalently as $Q = P^{-1}BB^\top P^{-1} - P^{-1}A - A^\top P^{-1} \succ 0$, and since we have $x^\top P^{-1} BB^\top P^{-1} x = s^\top s$, the energy balance is

$$\dot{V}(x) = s^\top \left(\frac{1}{2} s + u + \delta\right) - \frac{1}{2} x^\top Q x. \tag{31}$$

Passive Systems

Consider a finite-dimensional system with state $x(t) \in \mathbb{R}^n$, input $u(t) \in \mathbb{R}^m$, and output $y(t) \in \mathbb{R}^m$. As a prerequisite, it is assumed that the system is well posed, relying on classical results for ODEs with continuous solutions. The system is *passive* in Willems' sense [S43], [S44] if there exists a *storage function* $V : \mathbb{R}^n \to \mathbb{R}_+$ that satisfies the *dissipation inequality*

$$V(x(t_1)) - V(x(t_0)) \leq \int_{t_0}^{t_1} u^\top(t) y(t)\, \mathrm{d}t \tag{S69}$$

for all $t_1 \geq t_0$ along the continuous-time system trajectories. Roughly speaking, the dissipation inequality (S69) means that, on any time interval $[t_0, t_1]$, the energy that is internally stored in the system cannot increase beyond $\int_{t_0}^{t_1} u^\top(t) y(t)\, \mathrm{d}t$, the energy externally supplied throughout the power port (u, y).

Linear time-invariant systems

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{S70}$$

are passive if and only if the LMI

$$M_{\text{cont}} := \begin{pmatrix} -A^\top P - PA & -PB + C^\top \\ -B^\top P + C & D + D^\top \end{pmatrix} \succcurlyeq 0 \tag{S71}$$

has a positive semidefinite solution $P = P^\top$. Indeed, the candidate storage function

$$V(x) = \frac{1}{2} x^\top P x \tag{S72}$$

satisfies the energy balance

$$V(x(t_1)) - V(x(t_0)) = -\frac{1}{2} \int_{t_0}^{t_1} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix}^\top M_{\text{cont}} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \mathrm{d}t + \int_{t_0}^{t_1} u^\top(t) y(t)\, \mathrm{d}t.$$

Clearly, the inequality (S71) implies (S69). The LMI in (S71) has interesting consequences, which can be deduced from the Schur Complement Theorem [73, Sect. A.6.1] and its variants, $-A^\top P - PA \succcurlyeq 0$ and $D + D^\top \succcurlyeq 0$ [$-A^\top P - PA \succ 0$ and $D + D^\top \succ 0$, when the inequality in (S71) is strict]. It is also inferred from [64, Corollary 10.2.3, Prop. 10.2.5] that $\ker(D + D^\top) \subseteq \ker(PB - C^\top)$. The latter further implies the well-known input–output constraint $PB = C^\top$ when $D + D^\top = 0$.

Consider now the linear time-invariant discrete-time system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned} \tag{S73}$$

with $k \geq 0$. The system is passive if there exists a storage function (S72) such that the dissipation inequality

$$V(x_{k+1}) - V(x_k) \leq u_k^\top y_k \tag{S74}$$

holds for all $k \geq 0$ along the discrete-time system trajectories. Similar to the continuous-time case, we have

$$V(x_{k+1}) - V(x_k) = -\frac{1}{2} \begin{pmatrix} x_k \\ u_k \end{pmatrix}^\top M_{\text{disc}} \begin{pmatrix} x_k \\ u_k \end{pmatrix} + u_k^\top y_k. \tag{S75}$$

It can be shown that system (S73) is passive if and only if the LMI

$$M_{\text{disc}} := \begin{pmatrix} -A^\top PA + P & -A^\top PB + C^\top \\ -B^\top PA + C & D + D^\top - B^\top PB \end{pmatrix} \succcurlyeq 0 \tag{S76}$$

possesses a solution $P = P^\top \succcurlyeq 0$. Using the same tools as for the continuous-time LMI (S71), it is inferred that the LMI in (S76) implies $-A^\top PA + P \succcurlyeq 0$, $D + D^\top \succcurlyeq B^\top PB$ [$-A^\top PA + P \succ 0$, $D + D^\top \succ B^\top PB$ when the inequality in (S76) is strict] as well as $\ker(D + D^\top - B^\top PB) \subseteq \ker(-A^\top PB + C^\top)$ (hence $C = B^\top PA$ if $D + D^\top = B^\top PB$).

If (S70) or (S73) is passive, we say that the quadruple (A, B, C, D) is passive. The product $u^\top y$ is the *supply rate* of the system.

Many extensions of passivity have been studied in the automatic control literature in the past 50 years. *Strict state-passivity* holds for the continuous and discrete-time systems when $-A^\top P - PA \succ 0$ and $-A^\top PA + P \succ 0$, respectively. *Strong passivity* holds when $M_{\text{cont}} \succ 0$ and $M_{\text{disc}} \succ 0$. *Incremental passivity* holds if, for any pair of admissible inputs (u_1, u_2), corresponding trajectories (x_1, x_2), and outputs (y_1, y_2), the auxiliary system with state $x^\top = (x_1^\top, x_2^\top)$ is passive with supply rate $(u_1 - u_2)^\top (y_1 - y_2)$ and storage function $V(x_1, x_2) = (1/2)(x_1 - x_2)^\top P (x_1 - x_2)$.

For linear time-invariant systems, passivity and its variants are known to be closely related to the class of positive real systems. The celebrated Kalman-Yakubovich-Popov Lemma allows us to establish relationships between the frequency domain with positive-real transfer functions and the state-space domain with passive systems, as characterized by the previous dissipation inequalities and LMIs [73, Ch. 3].

REFERENCES

[S43] J. C. Willems, "Dissipative dynamical systems part I: General theory," *Arch. Ration. Mech. Anal.*, vol. 45, no. 5, pp. 21–351, 1972, doi: 10.1007/BF00276493.

[S44] J. C. Willems, "Dissipative dynamical systems, part II: Linear systems with quadratic supply rates," *Arch. Ration. Mech. Anal.*, vol. 45, no. 5, pp. 352–393, 1972, doi: 10.1007/BF00276494, doi: 10.1007/BF00276494.

This form suggests that, to render the system passive, a control law

$$u(t) = u_{\text{lin}}(x(t)) + v(t) \tag{32}$$

with

$$u_{\text{lin}}(x) = -\frac{1}{2}\sigma(x) \tag{33}$$

can be used since it leads to a choice of

$$\dot{V}(x) = s^\top (v + \delta) - \frac{1}{2} x^\top Q x. \tag{34}$$

It can be clearly seen that the system is passive with storage function (27), output (29), and new input $v + \delta$. Now set $v(t) = u_{\text{sv}}(x(t), t)$ with

$$u_{\mathrm{sv}}(x,t) \in -K\,\mathbf{Sgn}(Ks), \quad K = K^\top \succ 0 \tag{35}$$

for all $t \in \mathbb{R}$. Substituting this control law in (34) gives

$$\begin{aligned}\dot V(x) &= -s^\top K(\mathbf{Sgn}(Ks) - K^{-1}\delta) - \frac{1}{2}x^\top Qx \\ &\le -\|Ks\|_1(1 - \|K^{-1}\delta\|_\infty) - \frac{1}{2}x^\top Qx\end{aligned} \tag{36}$$

where the right-hand side is single-valued for all x. For suitable gains K, the origin of the closed-loop system is exponentially stable, regardless of the time-varying perturbation $\delta(t)$. We now prove that $s(t)$ converges to zero in finite time. The sliding variable evolves according to

$$\dot s \in B^\top P^{-1}\Big(\Big(A - \frac{1}{2}BB^\top P^{-1}\Big)x - BK\,\mathbf{Sgn}(Ks) + B\delta\Big). \tag{37}$$

Consider the quadratic Lyapunov function $V_s(s) = (1/2)s^\top (B^\top P^{-1}B)^{-1}s$. Its time derivative satisfies

$$\dot V_s(s) = -s^\top (K\,\mathbf{Sgn}(Ks) - \delta - Lx) \tag{38}$$

where $L = (B^\top P^{-1}B)^{-1}B^\top P^{-1}(A - (1/2)BB^\top P^{-1})$. In (37), the state $x(t)$ can be considered as an exogenous signal, uniformly bounded according to the previous analysis. Furthermore, it has already been established that x converges to zero exponentially fast. Using classical arguments, it is inferred that s converges to zero in finite time for a suitable K.

Alternatively, the dynamics (37) can be analyzed within the framework of DIs with maximal monotone right-hand sides, similarly to (25). To reveal the maximal monotone structure in (37), we perform the change of variables $s' = (B^\top P^{-1}B)^{-(1/2)}s$. The following is obtained:

$$\dot s' \in -\partial g(s') + (B^\top P^{-1}B)^{\frac{1}{2}}(\delta + Lx) \tag{39}$$

with $g(\cdot) = \|\cdot\|_1 \circ K(B^\top P^{-1}B)^{(1/2)}(\cdot)$ a proper, convex, and continuous function. Again, the DI (39) has the structure (S7), and if the eigenvalues of K are large enough, it satisfies the conditions of Proposition S1 in "Set-Valued Mappings and DIs," and the origin $s' = 0$ is finite-time stable. During the sliding mode, $s(t) = 0$ for all t, which means that there exists a selection of $\partial g(0)$ (see Definition S3 in "Set-Valued Mappings and DIs") that compensates exactly for the disturbance $(B^\top P^{-1}B)^{(1/2)}(\delta(t) + Lx(t))$ in (39) at $s' = 0$. Equivalently, consider (35) and note that there is a selection $u_{\mathrm{sv}}(x,t)$ in $K\mathbf{Sgn}(0)$ that compensates exactly for the equivalent disturbance in (37). That is, similar to (24)

$$u_{\mathrm{sv}}(x(t),t) = -(B^\top P^{-1}B)^{-1}B^\top P^{-1}\Big(A - \frac{1}{2}BB^\top P^{-1}\Big)x(t) - \delta(t). \tag{40}$$

We stress that neither (24) nor (8) means that the disturbance $\delta(t)$ is known. Finally, note that the well-posedness of (39) holds under mild measurability conditions on the perturbation, and from the analysis of the closed-loop dynamics

$$\dot x(t) \in Ax(t) + B\Big(-\frac{1}{2}\sigma(x(t)) - K\,\mathbf{Sgn}(\sigma(x(t)))\Big) + B\delta(t) \tag{41}$$

using the fact that the set-valued right-hand side has compact and convex images and that it is OSC. It is then inferred from Theorem S1 in "Set-Valued Mappings and DIs" that (41) has absolutely continuous solutions.

Recapitulation

Classical SMC is achieved by first choosing an output for which the system is of relative degree one and minimum phase, and then assigning a set-valued maximal monotone structure to the dynamics of the output variable. The main features of the resulting closed-loop system are robustness with respect to matched perturbations and finite-time convergence of the output to zero. Once the output reaches the origin, it stays at the origin in a *sliding motion*. During such motion, it follows from (37) that u_{sv} in (35) is selected as

$$u_{\mathrm{sv}}(x,t) = -Lx - \delta(t). \tag{42}$$

Note that during the sliding motion, the input is independent of the control gain K, as is well known. A similar conclusion holds for the feedbacks (20) and (22) [see (24)].

Incidentally, the DIs in (25) and (39) possess incremental passivity properties. Indeed, they have the generic form $\dot s'(t) \in -\mathbf{M}(s'(t)) + Nw(t)$ with $\mathbf{M}$ a maximal monotone mapping. The system is incrementally passive with storage function $V(s'_1, s'_2) = (1/2)(s'_1 - s'_2)^\top (s'_1 - s'_2)$ and output $N^\top s'$ [73].

It should be stressed that (35) can be written as $u_{\mathrm{sv}}(x) \in -\partial g(s)$ with $g(s) = \|Ks\|_p$ and $p = 1$. Naturally, other norms can be used. In general

$$\partial\|x\|_p = \frac{1}{\|x\|_p^{p-1}}\begin{pmatrix}|x_1|^{p-1}\mathbf{sgn}(x_1)\\ \vdots \\ |x_n|^{p-1}\mathbf{sgn}(x_n)\end{pmatrix} \tag{43}$$

if $x \neq 0$ and $\partial\|x\|_p = \mathcal{B}_{p^*}$ if $x = 0$; see (S55) in "Convex Analysis Tools." For example, the norm $p = 2$ yields the *unit control*

$$u_{\mathrm{sv}}(x,t) \in \begin{cases} -\dfrac{1}{\|Ks\|_2}K^2 s & \text{if } s \neq 0 \\ -K\mathcal{B}_2 & \text{otherwise}\end{cases}. \tag{44}$$

To close this section, the reader's attention is brought to the fact that, in the foregoing developments, no mention was made about the celebrated Filippov's convexification method, which guarantees the existence of absolutely continuous solutions by transforming a discontinuous differential equation into a specific DI. In fact, the previous closed-loop systems are introduced directly in a set-valued setting (see Figure 1 and Theorem S1 in "Set-Valued Mappings and DIs").

FIRST-ORDER SMC: DISCRETE-TIME

In this section, we exploit the passive properties of the sliding variable to derive a new design method for discrete-time SMC. We also show the necessity of defining any

discrete-time SMC implicitly. Consider a sampled-data model of (19)

$$x_{k+1} = \tilde{A}x_k + \tilde{B}(u_k + \tilde{\delta}_k) \tag{45}$$

where $\tilde{A} \in \mathbb{R}^{n\times n}$, $\tilde{B} \in \mathbb{R}^{n\times m}$, and it is assumed that rank $\tilde{B} = m$. In the spirit of the "A Passivity-Based Approach" section, an output s_k is constructed for which the system is feedback passive. In contrast to the continuous-time scenario, a necessary condition for a system to be feedback passive is that it has a relative degree $\{0,0,\ldots,0\}$ [77], [78], which ultimately results in the need for implicitly defined control laws.

A Discrete-Time Passivity-Based Approach

The discrete-time counterpart of (26) is

$$\tilde{A}\tilde{P}\tilde{A}^\top - \tilde{P} \prec \tilde{B}\tilde{B}^\top. \tag{46}$$

That is, the pair $(\tilde{A},\tilde{B})$ is stabilizable if and only if inequality (46) holds for some $\tilde{P} = \tilde{P}^\top \succ 0$ [75]. The forward difference $V(x_{k+1}) - V(x_k)$ of the storage function $V(x_k) = (1/2)x_k^\top \tilde{P}x_k$ is

$$\begin{aligned}\Delta V(x_k) = &\frac{1}{2}(\tilde{A}x_k + \tilde{B}(u_k+\tilde{\delta}_k))^\top \tilde{P}^{-1}(\tilde{A}x_k + \tilde{B}(u_k+\tilde{\delta}_k))\\ &-\frac{1}{2}x_k^\top \tilde{P}^{-1}x_k.\end{aligned} \tag{47}$$

By isolating the terms that do not depend on the inputs, the counterpart of (28) is obtained as

$$\begin{aligned}\Delta V(x_k) = &\frac{1}{2}x_k^\top(\tilde{A}^\top\tilde{P}^{-1}\tilde{A} - \tilde{P}^{-1})x_k\\ &+(u_k+\tilde{\delta}_k)^\top\tilde{B}^\top\tilde{P}^{-1}(\tilde{A}x_k + \tilde{B}(u_k+\tilde{\delta}_k))\\ &-\frac{1}{2}(u_k+\tilde{\delta}_k)^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}(u_k+\tilde{\delta}_k).\end{aligned} \tag{48}$$

The passive output candidate

$$s_{k+1} := \sigma(x_k, u_k + \tilde{\delta}_k) \tag{49}$$

with

$$\sigma(x,w) = \tilde{B}^\top\tilde{P}^{-1}(\tilde{A}x + \tilde{B}w) \tag{50}$$

is chosen, and the energy balance becomes

$$\begin{aligned}\Delta V(x_k) = &\frac{1}{2}x_k^\top(\tilde{A}^\top\tilde{P}^{-1}\tilde{A} - \tilde{P}^{-1})x_k + s_{k+1}^\top(u_k+\tilde{\delta}_k)\\ &-\frac{1}{2}(u_k+\tilde{\delta}_k)^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}(u_k+\tilde{\delta}_k)\end{aligned} \tag{51}$$

which corresponds to the dissipation equality (S76) of the quadruple $(\tilde{A},\tilde{B},\tilde{C},\tilde{D})$ with $\tilde{C} = \tilde{B}^\top\tilde{P}^{-1}\tilde{A}$, $\tilde{D} + \tilde{D}^\top = 2\tilde{B}\tilde{P}^{-1}\tilde{B}$, $\tilde{P}^{-1}$ solution of the LMI (S76), and supply rate $u^\top y$ (see "Passive Systems").

Remark 2

Equations (49) and (50) have an implicit (backward-Euler) flavor since

$$s_{k+1} = \tilde{B}^\top\tilde{P}^{-1}x_{k+1}. \tag{52}$$

Note that (50) is the counterpart of (29). In this context, the key difference is that discrete-time passive systems have relative degree zero, whereas continuous-time passive systems have relative degree one or zero [73].

Remark 3

Neither $s_{k+1} = B^\top P^{-1}x_k$ nor even $s_{k+1} = B^\top P^{-1}x_{k+1}$ was used, as would be the case in a pure emulation method. The sliding variable (and hence the sliding surface) are designed in the discrete-time context to preserve passivity. A similar remark holds for the linear controller (33), which can be discretized in various ways [by emulation, $u_{\text{lin}}(x_k) = -(1.2)B^\top P^{-1}x_k$, or directly in discrete time, as in (59) and (60) later]. Not all discretized controllers share similar closed-loop properties [21]. The same comments hold for the case of the classical first-order control in (93) and (94) later. See also the "Limitations and Modifications of Implicit Algorithms" section. The following theorem shows that stabilizability implies feedback equivalence to passivity.

Theorem 1

Suppose that system (45) is stabilizable. Consider the output (49) with $\tilde{P} = \tilde{P}^\top > 0$ a solution of (46). The control

$$u_k = u_{\text{lin}}(x_k) + v_k \tag{53}$$

with

$$u_{\text{lin}}(x_k) = -\frac{1}{2}\sigma(x_k, u_{\text{lin}}(x_k)) \tag{54}$$

renders the system passive with storage function $V(x) = (1/2)x^\top\tilde{P}^{-1}x$, output s_{k+1} given by (49), and new input $v_k + \tilde{\delta}_k$. Moreover, the energy balance

$$\begin{aligned}\Delta V(x_k) = &s_{k+1}^\top(v_k+\tilde{\delta}_k) - \frac{1}{2}(v_k+\tilde{\delta}_k)^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}(v_k+\tilde{\delta}_k)\\ &-\frac{1}{2}x_k^\top\tilde{Q}x_k\end{aligned} \tag{55}$$

holds for some positive definite matrix $\tilde{Q}$.

Notice that (54) is the counterpart of (33). The proof of this theorem requires the following lemma, which is inspired by [75, Ch. 6].

Lemma 1

Assume that the pair $(\tilde{A},\tilde{B})$ is stabilizable. The inequality (46) implies that

$$\tilde{A}^\top\Big(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top\Big)^{-1}\tilde{P}\Big(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top\Big)^{-1}\tilde{A} - \tilde{P}^{-1} \prec 0. \tag{56}$$

Proof

Using Schur complements, it can be seen that (46) is equivalent to

$$\begin{pmatrix}\tilde{P} & \tilde{P}\tilde{A}^\top\\ \tilde{A}\tilde{P} & \tilde{P} + \tilde{B}\tilde{B}^\top\end{pmatrix} \succ 0 \tag{57}$$

which implies

$$\begin{pmatrix} \tilde{P} & \tilde{P}\tilde{A}^\top \\ \tilde{A}\tilde{P} & \tilde{P}+\tilde{B}\tilde{B}^\top+\frac{1}{4}\tilde{B}\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{B}^\top \end{pmatrix} \succ 0 \tag{58}$$

equivalently, using again the Schur complement theorem, $\tilde{A}^\top(\tilde{P}+\tilde{B}\tilde{B}^\top+(1/4)\tilde{B}\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{B}^\top)^{-1}\tilde{A}-\tilde{P}^{-1}\prec 0$. Inequality (56) is recovered by noting that $\tilde{P}+\tilde{B}\tilde{B}^\top+(1/4)\tilde{B}\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{B}^\top = (\tilde{P}+(1/2)\tilde{B}\tilde{B}^\top)\tilde{P}^{-1}(\tilde{P}+(1/2)\tilde{B}\tilde{B}^\top)$. ■

Proof of Theorem 1

According to (50), the control (54) is implicitly defined by the equation

$$u_{\text{lin}}(x_k) = -\frac{1}{2}\tilde{B}^\top\tilde{P}^{-1}(\tilde{A}x_k+\tilde{B}u_{\text{lin}}(x_k)). \tag{59}$$

It can be solved explicitly as

$$u_{\text{lin}}(x_k) = -(2I+\tilde{B}^\top\tilde{P}^{-1}\tilde{B})^{-1}\tilde{B}^\top\tilde{P}^{-1}\tilde{A}x_k \tag{60}$$

which gives

$$\begin{aligned}\tilde{A}x_k+\tilde{B}u_{\text{lin}}(x_k) &= \tilde{A}x_k-\tilde{B}(2I+\tilde{B}^\top\tilde{P}^{-1}\tilde{B})^{-1}\tilde{B}^\top\tilde{P}^{-1}\tilde{A}x_k \\ &= \tilde{P}(\tilde{P}^{-1}-\tilde{P}^{-1}\tilde{B}(2I+\tilde{B}^\top\tilde{P}^{-1}\tilde{B})^{-1}\tilde{B}^\top\tilde{P}^{-1})\tilde{A}x_k.\end{aligned}$$

By the matrix inversion Lemma [64, Corollary 3.9.8]

$$\tilde{A}x_k+\tilde{B}u_{\text{lin}}(x_k) = \tilde{P}\Big(\tilde{P}+\frac{1}{2}\tilde{B}\tilde{B}^\top\Big)^{-1}\tilde{A}x_k. \tag{61}$$

It now follows from Lemma 1 that

$$(\tilde{A}x_k+\tilde{B}u_{\text{lin}}(x_k))^\top\tilde{P}^{-1}(\tilde{A}x_k+\tilde{B}u_{\text{lin}}(x_k))-x_k^\top\tilde{P}^{-1}x_k = -x_k^\top\tilde{Q}x_k \tag{62}$$

for some positive definite matrix $\tilde{Q}$. This shows passivity with the required ingredients. ■

Stability Analysis

It is deduced from (62) that the origin is exponentially stable if v_k is such that

$$s_{k+1}^\top(v_k+\tilde{\delta}_k)\le 0. \tag{63}$$

By continuing the analogy with the continuous-time case, we may be tempted to write

$$v_k = u_{\text{sv}}(x_k) \tag{64}$$

with $u_{\text{sv}}(x_k)\in -\tilde{K}\,\mathbf{Sgn}(\tilde{K}s_{k+1})$. This control satisfies (63), but unfortunately, the passive output (49) is not available to the controller because it depends directly on the unknown perturbation. To avoid this problem, we follow [32] and define the nominal output as

$$\tilde{s}_{k+1} := \sigma(x_k,u_k) \tag{65}$$

instead of (49). In what follows, it is shown that the full state of the closed-loop system defined by (45), (53), and (54) with the proposed nominal output $\tilde{s}_{k+1}$ is ultimately bounded.

Proposition 2

Consider the perturbed dynamics (45) with a feedback law given by (53) and (54), where

$$v_k = u_{\text{sv}}(x_k)\in -\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1}),\quad \tilde{K}=\tilde{K}^\top\succ 0 \tag{66}$$

$\tilde{s}_{k+1}$ is given by (65), and δ_k is uniformly bounded. Then, the trajectories of the closed loop are ultimately bounded.

Proof

Since $s_{k+1}=\tilde{s}_{k+1}+\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{\delta}_k$, the forward difference (5) becomes

$$\begin{aligned}\Delta V(x_k) = &(\tilde{s}_{k+1}+\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{\delta}_k)^\top(v_k+\tilde{\delta}_k) \\ &-\frac{1}{2}(v_k+\tilde{\delta}_k)^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}(v_k+\tilde{\delta}_k)-\frac{1}{2}x_k^\top\tilde{Q}x_k.\end{aligned} \tag{67}$$

By developing the right-hand side, one obtains

$$\begin{aligned}\Delta V(x_k) = &\tilde{s}_{k+1}^\top(v_k+\tilde{\delta}_k)+\delta_k^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}v_k \\ &+\delta_k^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{\delta}_k-\frac{1}{2}v_k^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}v_k-v_k^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{\delta}_k \\ &-\frac{1}{2}\tilde{\delta}_k^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{\delta}_k-\frac{1}{2}x_k^\top\tilde{Q}x_k\end{aligned} \tag{68}$$

which can be simplified as

$$\begin{aligned}\Delta V(x_k) = &\tilde{s}_{k+1}^\top(v_k+\delta_k)+\frac{1}{2}\tilde{\delta}_k^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{\delta}_k \\ &-\frac{1}{2}v_k^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}v_k-\frac{1}{2}x_k^\top\tilde{Q}x_k.\end{aligned} \tag{69}$$

Define the set-valued control (66) and note that it is a discrete-time counterpart of (35). The controller gives the inequality

$$\begin{aligned}\Delta V(x_k) \le &-\|\tilde{K}\tilde{s}_{k+1}\|_1(1-\|\tilde{K}^{-1}\tilde{\delta}_k\|_\infty)-\frac{1}{2}v_k^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}v_k \\ &-\frac{1}{2}x_k^\top\tilde{Q}x_k+\frac{1}{2}\tilde{\delta}_k^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{\delta}_k.\end{aligned} \tag{70}$$

Notice that since $\tilde{\delta}_k$ is uniformly bounded, $\Delta V(x_k)$ is negative for $\|x_k\|$ sufficiently large and $\tilde{K}$ such that $\|\tilde{K}^{-1}\tilde{\delta}_k\|_\infty<1$ for all $k\ge 0$. This establishes that the state is ultimately bounded (exponential stability is achieved only in the nominal case) with a bound proportional to the bound on $\tilde{\delta}_k$. ■

Discrete-Time Sliding Dynamics

In what follows, we show that the nominal output $\tilde{s}_{k+1}$ converges to zero in a finite number of steps. Once the nominal output reaches zero, it stays at zero in *discrete-time sliding motion*. To that end, note that the nominal output satisfies

$$\tilde{s}_{k+1}\in\tilde{B}^\top\tilde{P}^{-1}(\tilde{A}x_k+\tilde{B}u_{\text{lin}}(x_k)-\tilde{B}\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1})) \tag{71}$$

or, using (61)

$$\tilde{s}_{k+1} \in \tilde{B}^\top \left(\left(\tilde{P} + \frac{1}{2} \tilde{B}\tilde{B}^\top \right)^{-1} \tilde{A} x_k - \tilde{P}^{-1} \tilde{B} \tilde{K} \mathbf{Sgn}(\tilde{K} \tilde{s}_{k+1}) \right). \quad (72)$$

Defining $\tilde{L} = (\tilde{B}^\top \tilde{P}^{-1} \tilde{B})^{-(1/2)} \tilde{B}^\top (\tilde{P} + (1/2)\tilde{B}\tilde{B}^\top)^{-1} \tilde{A}$ and considering the change of coordinates $\tilde{s}'_k = (\tilde{B}^\top \tilde{P}^{-1} \tilde{B})^{-(1.2)} \tilde{s}_k$ yields the following:

$$\tilde{s}'_{k+1} \in -(\tilde{B}^\top \tilde{P}^{-1} \tilde{B})^{\frac{1}{2}} \tilde{K} \mathbf{Sgn}\left(\tilde{K} (\tilde{B}^\top \tilde{P}^{-1} \tilde{B})^{\frac{1}{2}} \tilde{s}'_{k+1} \right) + \tilde{L} x_k. \quad (73)$$

The generalized (73) shares the same structure as the DIs (25) and (39). Namely, using once again the chain rule of convex analysis (see Fact 7 in "Convex Analysis Tools"), one can see that

$$\tilde{s}'_{k+1} \in -\partial \tilde{g}(\tilde{s}'_{k+1}) + \tilde{L} x_k \quad (74)$$

with $\tilde{g}(\cdot) = \|\cdot\|_1 \circ \tilde{K} (\tilde{B}^\top \tilde{P}^{-1} \tilde{B})^{1/2}(\cdot)$ [formally the same function as $g(\cdot)$ in (39)]. This difference inclusion, which is of the implicit type since $\tilde{s}'_{k+1}$ appears on both sides, is equivalently rewritten as the equality

$$\tilde{s}'_{k+1} = (\mathrm{I}_d + \partial \tilde{g})^{-1} (\tilde{L} x_k). \quad (75)$$

Let us explain some of the implications of the right-hand side of (75). The structure of this difference equation will be encountered again in all the discretizations presented in the following sections (for both controllers and differentiators). Notice first that the right-hand side of (75) is, in fact, a proximal operator

$$\tilde{s}'_{k+1} = \mathrm{Prox}_{\tilde{g}}(\tilde{L} x_k) \quad (76)$$

[see (S24) in "Proximal Mapping and Proximal-Point Algorithm"]. The next proposition states a necessary and sufficient condition for the virtual sliding variable $\tilde{s}'_k$ to stay on the discrete sliding surface. This condition can be satisfied with a suitable choice of the feedback gain $\tilde{K}$ in (66); roughly speaking, it must be large enough.

Proposition 3

Consider the virtual sliding variable (76). Consequently, $\tilde{s}'_{k+1} = 0$ if and only if

$$\tilde{K}^{-1} (\tilde{B}^\top \tilde{P}^{-1} \tilde{B})^{-\frac{1}{2}} \tilde{L} x_k \in \mathcal{B}_\infty \quad (77)$$

where the unit ball $\mathcal{B}_\infty$ is defined in (S56).

Proof

The claim follows from (S28) in "Proximal Mapping and Proximal-Point Algorithm." Indeed $\mathrm{zeroProx}_{\tilde{g}} = \partial \tilde{g}(0) = (\tilde{B}^\top \tilde{P}^{-1} \tilde{B})^{(1/2)} \tilde{K} \mathcal{B}_\infty$, where the equality follows from Fact 7 and from (S58) in "Convex Analysis Tools," and the set of zeroes is defined in "Proximal Mapping and Proximal-Point Algorithm." Thus, from (77), we deduce that $\tilde{L} x_k \in \partial \tilde{g}(0)$, and the proof is complete. ■

If the eigenvalues of $\tilde{K}$ are chosen sufficiently large, then inclusion (77) holds once the ultimate bound in x_k is attained. This implies that the condition $\tilde{s}'_{k+1} = 0$ is achieved and maintained after a finite number of steps.

Remark 4

The difference inclusion (71) is the counterpart of the DI (37), while (73) and (74) are the counterparts of (39). The equality in (75) is the explicit (that is, forward in time) form of the implicit (backward-Euler) algorithm (74). Following [21], we define the discrete-time sliding motion in terms of the nominal variable $\tilde{s}_{k+1}$.

Definition 1

The closed-loop system (45), (53), (54), and (66) reaches a discrete-time sliding motion whenever there is a finite $k^* \in \mathbb{N}$ such that

$$\tilde{s}_k = 0 \quad (78)$$

holds for all $k \geq k^*$.

In analogy to the continuous-time case, the control compensates for the perturbation while the system trajectory is in a sliding motion. This is quite an important property of the backward-Euler method, which guarantees that the control input is a selection (see Definition S3 in "Set-Valued Mappings and DIs") of the set-valued closed-loop right-hand side.

Proposition 4

During the discrete-time sliding mode, the controller $u_k = u_{\mathrm{lin}}(x_k) + u_{\mathrm{sv}}(x_k)$ defined in (60) and (66) satisfies

$$u_k = -\tilde{\delta}_{k-1} - (\tilde{B}^\top \tilde{P}^{-1} \tilde{B})^{-1} \tilde{B} \tilde{P}^{-1} (\tilde{A} - I) x_k. \quad (79)$$

Proof

It follows from (71) and (52) that

$$\tilde{s}_{k+1} = s_k + \tilde{B}^\top \tilde{P}^{-1} (\tilde{A} - I) x_k + \tilde{B}^\top \tilde{P}^{-1} \tilde{B} u_k \quad (80)$$

whereas the definition of $\tilde{s}_{k+1}$ yields

$$s_{k+1} = \tilde{s}_{k+1} + \tilde{B}^\top \tilde{P}^{-1} \tilde{B} \tilde{\delta}_k. \quad (81)$$

Thus, the substitution of (81) back into (80), with a one-step delay, leads us to the expression

$$\tilde{s}_{k+1} = \tilde{s}_k + \tilde{B}^\top \tilde{P}^{-1} \tilde{B} \tilde{\delta}_{k-1} + \tilde{B}^\top \tilde{P}^{-1} (\tilde{A} - I) x_k + \tilde{B}^\top \tilde{P}^{-1} \tilde{B} u_k. \quad (82)$$

The conclusion follows directly by setting $\tilde{s}_{k+1} = \tilde{s}_k = 0$. ■

This last result shows that the exact compensation in continuous-time (42) can be well approximated by the implicit algorithm. The controller during the sliding phase is independent of $\tilde{K}$, as predicted by the continuous-time analysis.

Control Calculation and Input–Sliding-Variable Conjugacy

The feedback control and future sliding variable are implicitly defined by (65) and (66). Such expressions are useful for the stability analysis of closed-loop trajectories—but not for implementation. For the latter, an explicit expression is needed. To obtain such an expression, the duality between $\tilde{s}_{k+1}$ and $-u_{\mathrm{sv}}(x_k)$ can be exploited. Proposition 5 provides different descriptions of the sliding variable $\tilde{s}_{k+1}$ and the control input $u_{\mathrm{sv}}(x_k)$. Depending on the problem at hand, some descriptions may result in simpler expressions than others.

Proposition 5

Let $\Lambda_1 \in \mathbb{R}^{m\times n}$, $\Lambda_2 = \Lambda_2^\top \in \mathbb{R}^{m\times m}$ be nonsingular, $\mathbf{M}: \mathbb{R}^m \rightrightarrows \mathbb{R}^m$ maximal monotone, and $\tilde{\mathbf{M}} = \Lambda_2^{1/2} \circ \mathbf{M} \circ \Lambda_2^{1/2}$. Then, the control input and sliding output expressions shown in **i**, **ii**, **iii,** and **iv** in Figure 4 are equivalent.

Proof

The equivalence between the implicit expressions **i** and **ii** in Figure 4 is immediate. It suffices to invert Λ_1 and $\mathbf{M}$. To show the equivalence between **i** and **iii**, note that the following inclusions are equivalent:

$$\begin{aligned}
&\tilde{s}_{k+1} \in \Lambda_1 x_k - \Lambda_2 \mathbf{M}(\tilde{s}_{k+1})\\
&\tilde{s}_{k+1} \in (\Lambda_2^{-1} + \mathbf{M})^{-1}(\Lambda_2^{-1}\Lambda_1 x_k)\\
&\tilde{s}_{k+1} \in \left(\Lambda_2^{-\frac{1}{2}} \circ (I_d + \tilde{\mathbf{M}}) \circ \Lambda_2^{-\frac{1}{2}}\right)^{-1}(\Lambda_2^{-1}\Lambda_1 x_k)\\
&\tilde{s}_{k+1} \in \left(\Lambda_2^{\frac{1}{2}} \circ (I_d + \tilde{\mathbf{M}})^{-1} \circ \Lambda_2^{-\frac{1}{2}}\right)(\Lambda_1 x_k)\\
&\tilde{s}_{k+1} = \Lambda_2^{\frac{1}{2}} \mathcal{J}_{\tilde{\mathbf{M}}}\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right) \qquad (83)
\end{aligned}$$

where the fact that the resolvent $\mathcal{J}_{\tilde{\mathbf{M}}}$ is single-valued in the last expression is used. Now, the explicit expression for the control input is given as

$$\begin{aligned}
-u_{\mathrm{sv}}(x_k) &= \Lambda_2^{-1}(\Lambda_1 x_k - \tilde{s}_{k+1})\\
&= \Lambda_2^{-\frac{1}{2}}(I_d - \mathcal{J}_{\tilde{\mathbf{M}}})\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right)\\
&= \Lambda_2^{-\frac{1}{2}} \mathcal{Y}_{\tilde{\mathbf{M}}}\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right). \qquad (84)
\end{aligned}$$

To complete the proof, we show the equivalence between **ii** and **iv** in Figure 4. To this end, we proceed in a similar manner as previously; that is, the following inclusions are equivalent:

$$\begin{aligned}
&-u_{\mathrm{sv}}(x_k) \in \Lambda_2^{-1}\Lambda_1 x_k - \Lambda_2^{-1}\mathbf{M}^{-1}(-u_{sv}(x_k))\\
&-u_{\mathrm{sv}}(x_k) \in (\Lambda_2 + \mathbf{M}^{-1})^{-1}(\Lambda_1 x_k)\\
&-u_{\mathrm{sv}}(x_k) \in \left(\Lambda_2^{\frac{1}{2}} \circ (I_d + \tilde{\mathbf{M}}^{-1}) \circ \Lambda_2^{\frac{1}{2}}\right)^{-1}(\Lambda_1 x_k)\\
&-u_{\mathrm{sv}}(x_k) \in \Lambda_2^{-\frac{1}{2}} \circ (I_d + \tilde{\mathbf{M}}^{-1})^{-1} \circ \Lambda_2^{-\frac{1}{2}}(\Lambda_1 x_k)\\
&-u_{\mathrm{sv}}(x_k) = \Lambda_2^{-\frac{1}{2}} \mathcal{J}_{\tilde{\mathbf{M}}^{-1}}\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right). \qquad (85)
\end{aligned}$$

Finally, the substitution of (85) into the sliding output yields

$$\begin{aligned}
\tilde{s}_{k+1} &= \Lambda_1 x_k - \Lambda_2^{\frac{1}{2}} \mathcal{J}_{\tilde{\mathbf{M}}^{-1}}\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right)\\
&= \Lambda_2^{\frac{1}{2}}(I_d - \mathcal{J}_{\tilde{\mathbf{M}}^{-1}})\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right)\\
&= \Lambda_2^{\frac{1}{2}} \mathcal{Y}_{\tilde{\mathbf{M}}^{-1}}\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right). \qquad (86)
\end{aligned}$$

■

The controller $u_{\mathrm{sv}}(x_k)$ in (65) and (66) is calculated with the help of Proposition 5 by setting $\mathbf{M}^{-1} = \partial(\Psi_{\mathcal{B}_\infty} \circ \tilde{K}^{-1})$, $\Lambda_1 = \tilde{B}^\top(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top)^{-1}\tilde{A}$ and $\Lambda_2 = \tilde{B}^\top \tilde{P}^{-1}\tilde{B}$, where the inversion of set-valued mappings and (S58) were used to compute $\mathbf{M}^{-1}$ (see Fact 9 in "Convex Analysis Tools"). To write the explicit expression for $-u_{\mathrm{sv}}(x_k)$, we make use of the implicit expression **ii** in the lower part of Figure 4 by premultiplying by $\tilde{K}^{-1}$. Consequently, after rearranging terms

$$\begin{aligned}
&\tilde{K}^{-1}\Lambda_2^{-1}\Lambda_1 x_k + \tilde{K}^{-1}u_{\mathrm{sv}}(x_k)\\
&\in (\tilde{K}\Lambda_2\tilde{K})^{-1}\mathbf{N}_{\mathcal{B}_\infty}(-\tilde{K}^{-1}u_{\mathrm{sv}}(x_k)). \qquad (87)
\end{aligned}$$

A consequence of (S64) is that

$$\begin{aligned}
-u_{\mathrm{sv}}(x_k) &= \tilde{K}(I_d + M^{-1}\mathbf{N}_{\mathcal{B}_\infty})^{-1}(\tilde{K}^{-1}\varphi(x_k))\\
&= \tilde{K}\mathrm{Proj}_M(\mathcal{B}_\infty; \varphi(x_k)) \qquad (88)
\end{aligned}$$

where

$$M = M^\top := \tilde{K}(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})\tilde{K} \qquad (89)$$

$$\begin{aligned}
\varphi(x_k) &:= \tilde{K}^{-1}(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-1}\tilde{B}^\top\left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top\right)^{-1}\tilde{A}x_k\\
&= \tilde{K}^{-1}(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-\frac{1}{2}}\tilde{L}x_k. \qquad (90)
\end{aligned}$$

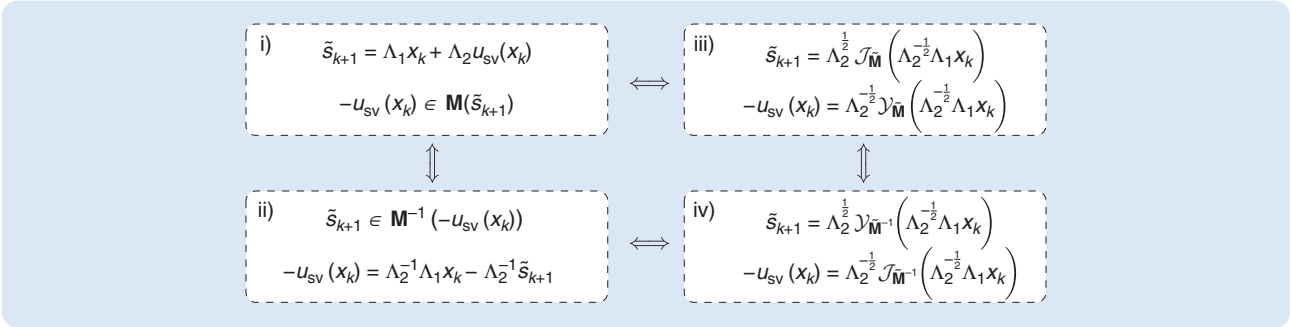

FIGURE 4 Implicit (left-hand side) and explicit (right-hand side) equivalent descriptions of the closed-loop sliding dynamics. Implicit expressions are useful for stability analysis, whereas explicit ones are used for the numerical implementation of the controller.

This extends the scalar case in (14). A simple choice for the gain is $\tilde{K} = \gamma(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-1/2}$, $\gamma > 0$, in which case $M = \gamma^2 I$ and (S63) can be applied to obtain

$$\begin{aligned} w_i &= \min\{\,|\,\varphi(x_k)_i\,|\,,1\}\mathbf{sgn}(\varphi(x_k)_i) \quad i = 1,\dots,n \\ u_{\mathrm{sv}}(x_k) &= -\gamma(\tilde{B}\top\tilde{P}^{-1}\tilde{B})^{-1/2}w. \end{aligned} \tag{91}$$

Otherwise, the controller can be computed as the solution of a quadratic program under constraints, similarly to (18)

$$\begin{aligned} &\mathrm{Proj}_M(\mathcal{B}_\infty; \tilde{K}^{-1}\varphi(x_k)) \\ &\quad = \arg\min_{\xi\in\mathcal{B}_\infty} \frac{1}{2}(\xi - \tilde{K}^{-1}\varphi(x_k))^\top M(\xi - \tilde{K}^{-1}\varphi(x_k)). \end{aligned} \tag{92}$$

See "Convex Analysis Tools" for $\mathcal{B}_\infty$, (S64a), and the associated piece of code for the computation of the projection.

Discrete-Time Classical Approach

Let us pass to the discretization of the first controller in (20) and (22). To start with, consider (45) and set $\tilde{s}_{k+1} = \sigma(x_k, u_k)$ as in (65) but with $\sigma(x,w) = \tilde{C}(\tilde{A}x + \tilde{B}w)$, so that $s_{k+1} = \sigma(x_k, u_k + \tilde{\delta}_k) = \tilde{C}x_{k+1}$. We define u_{lin} by imposing the condition

$$\tilde{C}(\tilde{A}x_k + \tilde{B}u_{\mathrm{lin}}(x_k)) = \tilde{C}x_k = s_k \tag{93}$$

and assume as usual that $\tilde{C}\tilde{B} \in \mathbb{R}^{m\times m}$ has rank m. In such a case, u_{lin} can be computed as

$$u_{\mathrm{lin}}(x_k) = (\tilde{C}\tilde{B})^{-1}\tilde{C}(I_n - \tilde{A})x_k. \tag{94}$$

The rationale behind the choice for u_{lin} is to subsequently ensure that $\tilde{s}_{k+1}$ depends only on s_k and u_k. Further, setting $u_k = u_{\mathrm{lin}}(x_k) + (\tilde{C}\tilde{B})^{-1}v_k$ gives

$$s_{k+1} = s_k + v_k + \tilde{C}\tilde{B}\tilde{\delta}_k \tag{95a}$$

$$\tilde{s}_{k+1} = s_k + v_k. \tag{95b}$$

From (95b), it is natural to define the control action $v_k = u_{\mathrm{sv}}(x_k) \in -\tilde{K}\mathbf{Sgn}(\tilde{s}_{k+1})$. This results in

$$s_{k+1} \in s_k - \tilde{K}\mathbf{Sgn}(\tilde{s}_{k+1}) + \tilde{C}\tilde{B}\tilde{\delta}_k \tag{96a}$$

$$\tilde{s}_{k+1} \in s_k - \tilde{K}\mathbf{Sgn}(\tilde{s}_{k+1}). \tag{96b}$$

A generalized equation with unknown $\tilde{s}_{k+1}$ is obtained in (96b). The resemblance between equalities in (96b) and (72) is clear. Now we define $\tilde{s}'_k = \tilde{K}^{-1/2}\tilde{s}_k$ and note that (96b) is equivalent to

$$\tilde{s}'_{k+1} \in \tilde{K}^{-\frac{1}{2}}s_k - \tilde{K}^{\frac{1}{2}}\mathbf{Sgn}(\tilde{K}^{\frac{1}{2}}\tilde{s}'_{k+1}) = \tilde{K}^{-\frac{1}{2}}s_k - \partial f(\tilde{s}'_{k+1}) \tag{97}$$

with $f(\,\cdot\,) = \|\,\cdot\,\|_1 \circ \tilde{K}^{1/2}$. Once again, the chain rule has been used (see Fact 7 in "Convex Analysis Tools"). Notice that the functions f in (97) and (25) share the same structure and are both proper, convex, and LSC. Thus, the expression (97) is also a generalized equation with unknown $\tilde{s}'_{k+1}$, rewritten equivalently as

$$\begin{aligned} \tilde{s}'_{k+1} &= \mathrm{Prox}_f(\tilde{K}^{-\frac{1}{2}}s_k) \\ &= \mathrm{Prox}_f\big(\tilde{K}^{-\frac{1}{2}}(\tilde{s}'_k + \tilde{C}\tilde{B}\tilde{\delta}_{k-1})\big) \\ &= \mathrm{Prox}_{\|\tilde{K}^{\frac{1}{2}}(\,\cdot\,)\|_1}(\tilde{s}'_k + \tilde{C}\tilde{B}\tilde{\delta}_{k-1}). \end{aligned} \tag{98}$$

This algorithm may be named a *robust* or *perturbed proximal-point algorithm*. Using (95), it is also inferred that

$$s_{k+1} = \tilde{K}^{\frac{1}{2}}\mathrm{Prox}_{\|\tilde{K}^{\frac{1}{2}}(\,\cdot\,)\|_1}(s_k) + \tilde{C}\tilde{B}\tilde{\delta}_k. \tag{99}$$

Remark 5

The previous control strategy has been proposed in [21] and [32]. For instance, (96b) is exactly [21, eq. (9)].

Stability Analysis and Sliding Mode

We briefly recall the main stability properties of (96). It is proved in [21, Prop. 2] that the $\{\tilde{s}_k\}_{k\in\mathbb{N}}$ solution of (96b) converges to zero in a finite number $k_{\min} < +\infty$ of steps (cf. Proposition S2 in "Proximal Mapping and Proximal-Point Algorithm" as well), while the $\{s_k\}_{k\in\mathbb{N}}$ solution of (96a) is bounded. During the sliding mode ($\tilde{s}_k = 0$ for $k \geq k_{\min}$), we have $s_{k+1} = \tilde{C}\tilde{B}\tilde{\delta}_k$ since $s_k + u_{\mathrm{sv}}(x_k) = 0$ is equivalent to

$$u_{\mathrm{sv}}(x_k) = -s_k = -\tilde{C}\tilde{B}\tilde{\delta}_{k-1}. \tag{100}$$

As a fundamental result, the control (24) is approximated by (100); in discrete time, the implicit method allows us to design an input that compensates for the perturbation with a one-step delay, in a similar way to (42). Once again, the set-valued controller does not depend on the gain $\tilde{K}$ during the sliding phase.

Control-Input Calculation

Using Proposition S3 in "Convex Analysis Tools," (96), and mapping inversion [see "Set-Valued Mappings and DIs," Fact 9 and (S59) in "Convex Analysis Tools"], we can see that

$$\tilde{K}(\tilde{K}^{-1}s_k - (-\tilde{K}^{-1}u_{\mathrm{sv}}(x_k)) \in \mathbf{N}_{\mathcal{B}_\infty}(-\tilde{K}u_{\mathrm{sv}}(x_k)) \tag{101}$$

is equivalent to $-u_{\mathrm{sv}}(x_k) = \tilde{K}(I_{\mathrm{d}} + \tilde{K}^{-1}\mathbf{N}_{\mathcal{B}_\infty})^{-1}(\tilde{K}^{-1}s_k)$, which in turn is equivalent to

$$-u_{\mathrm{sv}}(x_k) = \tilde{K}\mathrm{Proj}_{\tilde{K}}(\mathcal{B}_\infty; \tilde{K}^{-1}s_k). \tag{102}$$

Again, the projection can be computed by solving a quadratic program or, similarly to (91), conveniently choosing $\tilde{K} = \gamma I$ so that

$$(u_{\mathrm{sv}}(x_k))_i = -\gamma\min\left\{\frac{|\,(s_k)_i\,|}{\gamma}, 1\right\}\mathbf{sgn}((s_k)_i). \tag{103}$$

See "Convex Analysis Tools" for $\mathcal{B}_\infty$, (S64a) and the associated piece of code for the computation of the projection in a

general setting. The controller in (103) is in the minimum-operator formalism [49], [51], [66], [67], [68].

Remark 6

Dead-beat controller. The linear controllers in (20) and in (94) have different structures. Suppose that $u_{\text{lin}}(x_k)$ is computed as $u_{\text{lin}}(x_k) = -(\tilde{C}\tilde{B})^{-1}\tilde{C}\tilde{A}x_k$, which would be a naive implementation of (20). Then

$$\tilde{C}x_{k+1} = \tilde{C}\tilde{A}x_k + \tilde{C}\tilde{B}(u_{\text{lin}}(x_k) + \tilde{\delta}_k) + v_k = v_k + \tilde{C}\tilde{B}\tilde{\delta}_k \tag{104}$$

which is different from (95). Applying the previous $u_{\text{sv}}(x_k)$ yields $\tilde{s}_{k+1} \in -\tilde{K}\mathbf{Sgn}(\tilde{s}_{k+1})$, which is equivalent to $\tilde{s}'_{k+1} = (I_d + \partial f)^{-1}(0)$. Since the operator $(I_d + \partial f)^{-1}$ is single-valued and its graph contains $(0,0)$, we infer that $\tilde{s}_{k+1} = 0$. Therefore, this algorithm is a one-step dead-beat controller. The implicit algorithms introduced in [21], [31], and [32] are not of the dead-beat type. It is noteworthy that a major discrepancy exists between (94) and the dead-beat input in the framework of the discretization of continuous systems. Consider, for instance, the exact zero-order hold (ZOH) discretization with constant sampling time $h > 0$, where $\tilde{A} = e^{Ah}$, $\tilde{B} = e^{Ah}\int_0^h e^{-A\tau}B d\tau$. Then, as $h \to 0$, we have $(\tilde{C}\tilde{B})^{-1}\tilde{C}(I_n - \tilde{A}) = O(1)$, while $(\tilde{C}\tilde{B})^{-1}\tilde{C}\tilde{A} = O(h^{-1})$. Clearly, the dead-beat input grows unbounded as $h \to 0$ and cannot be a good candidate for convergence toward its continuous-time counterpart. This shows that this type of dead-beat controller is well suited to purely discrete-time systems but not for discretized continuous-time systems.

Recapitulation

Here, we summarize some of the connections between SMC, passivity, and optimization.

Link With Optimization

It is interesting to note that operators of the form $(I_d + \mathbf{M})^{-1}$ are ubiquitous in the previous discrete-time algorithms; they are the well-known resolvent mappings, very close to Yosida approximations [see (S17) and (S20) in "Maximal Monotone Operators"]. They have been exhibited in (10), (11), (75), (88), (98), and (102). They are a direct consequence of the presence of a maximal monotone mapping $\mathbf{M}$ on the right-hand side of the DIs in (25) and (39). These operators are resolvents of mappings of the form $\mathbf{M} = \partial f$ for some proper convex LSC function f. This shows the strong connection between the implicit discretization method and optimization [see also (88)]. Thus, a common feature between model-predictive control (MPC) and set-valued sliding-mode implicit discretization is that an optimization problem has to be solved to compute the controller. This is formalized in "Implicit Sliding-Mode and MPC," where the implicitly discretized scheme is reinterpreted as a one-step

Implicit Sliding-Mode and MPC

Implicit discrete-time SMC and differentiation require the solution of an optimization problem at each time step. MPC requires the solution of an optimization problem over longer time intervals. Besides this feature, there exists a closer and more formal link between implicit sliding modes and MPC.

Consider an optimization problem over a horizon event of one step

$$\begin{aligned} &\min_{(s_{k+1},u_k)} f(s_{k+1} - h\delta_k) + \frac{h}{2}\|u_k\|^2, \\ &\text{such that } s_{k+1} = s_k + hu_k + h\delta_k \end{aligned} \tag{S77}$$

where the function $f : \mathbb{R}^m \to \mathbb{R}$ is proper convex and LSC. At time $t_k = kh$, s_k is treated as a known parameter, whereas δ_k is treated as an unknown parameter. The target is to find the optimal control u_k and the optimal state value s_{k+1} that minimize the given cost function. The Lagrangian associated with (S77) is

$$\begin{aligned} L(s_{k+1}, u_k, \lambda_{k+1}) = f(s_{k+1} - h\delta_k) + \frac{h}{2}\|u_k\|^2 \\ - \lambda_{k+1}^\top(s_{k+1} - s_k - hu_k - h\delta_k) \end{aligned} \tag{S78}$$

while the Karush-Kuhn-Tucker (KKT) conditions associated with (S77) are

1) $0 \in \partial_{s_{k+1}} L(s^*_{k+1}, u^*_k)$.
2) $0 = \nabla_{u_k} L(s^*_{k+1}, u^*_k)$.
3) $s^*_{k+1} = s_k + hu^*_k + h\delta_k$.

From KKT-i, it is seen that $0 \in \partial f(s^*_{k+1} - h\delta_k) - \lambda^*_{k+1}$ is equivalent to $\lambda^*_{k+1} \in \partial f(s^*_{k+1} - h\delta_k)$, whereas from KKT-ii, it is seen that $0 = hu^*_k + h\lambda^*_{k+1}$ is equivalent to $\lambda^*_{k+1} = -u^*_k$. The combination of both relations together with condition KKT-iii) yields

$$h\lambda^*_{k+1} = -hu^*_k = s_k + h\delta_k - s^*_{k+1} \in h\partial f(s^*_{k+1} - h\delta_k). \tag{S79}$$

Consequently

$$s_k \in (I + h\partial f)(s^*_{k+1} - h\delta_k) \Leftrightarrow s^*_{k+1} - h\delta_k = \text{Prox}_{hf}(s_k) \tag{S80}$$

and the substitution of (S80) back into KKT-iii leads to

$$s^*_{k+1} - h\delta_k = s_k + hu^*_k = \text{Prox}_{hf}(s_k). \tag{S81}$$

Equivalently, we have

$$u^*_k = -\frac{1}{h}(s_k - \text{Prox}_{hf}(s_k)). \tag{S82}$$

Setting the dummy variable $\tilde{s}_{k+1}$ as $\tilde{s}_{k+1} = s^*_{k+1} - h\delta_k$ yields

$$\begin{aligned} s^*_{k+1} &= \tilde{s}_{k+1} + h\delta_k \\ \tilde{s}_{k+1} &= s_k + hu^*_k, \\ -u^*_k &\in \partial f(\tilde{s}_{k+1}) \end{aligned} \tag{S83}$$

which are the same equations obtained from the backward-Euler discretization of the system $\dot{s}(t) \in -\partial f(s(t)) + \delta(t)$; see (95).

The discretization with an implicit scheme turns out to be necessary for assuring passivity (a fact that was not noticed in previous works on the topic).

MPC. The explicit discretization [that is, choosing $u_{sv}(x_k) \in -\tilde{K}\mathbf{Sgn}(s_k)$] yields operators of the form $(I_d - \partial f)$, which are neither monotone nor do they possess interesting properties in terms of stability or robustness.

The previous analysis is closely related to the analysis of proximal-point algorithms (Definition S5 in "Proximal Mapping and Proximal-Point Algorithm"). However, a peculiarity of discrete SMC is the notion of a *robust* proximal-point algorithm, formulated in (S31) in "Proximal Mapping and Proximal-Point Algorithm" and shown to converge in a finite number of steps, given sufficiently large gains. It is worth emphasizing that (66) has the form $u_{sv}(x_k) \in -\partial \|\tilde{K}\tilde{s}_{k+1}\|_p$ with $p = 1$. Just as with the continuous-time case, other norms can be used (or other proper LSC convex functions for that matter). For arbitrary p, it suffices to replace $\mathcal{B}_\infty$ in (88) by $\mathcal{B}_{p^*}$. For example, the implicit discretization of the unit control is

$$-u_{sv}(x_k) = \tilde{K}\mathrm{Proj}_M(\mathcal{B}_2; \tilde{K}^{-1}\xi(x_k)). \tag{105}$$

Again, it is possible to choose $\tilde{K}$ such that $M = \gamma^2 I$ and use (S62) to compute an explicit form for the projection.

Link With Passivity

Both control design methodologies described previously follow the celebrated passivity plus zero-state detectability or observability results [73], [74], [79] (see [77] for the discrete-time version). Regarding the direct passivity-based design, Theorem 1 shows that a triplet $(V(x_k), s_{k+1}, v_k + \tilde{\delta}_k)$ can be chosen, defining a passifiable system. Hence, it is necessary that the system has relative degree zero and is of minimum phase. The system is actually rendered passive with $u_{\mathrm{lin}}(x_k)$, so that

$$\begin{aligned} x_{k+1} &= \tilde{A}x_k + \tilde{B}u_{\mathrm{lin}}(x_k) + \tilde{B}(v_k + \tilde{\delta}_k) \\ s_{k+1} &= \tilde{C}x_{k+1} \end{aligned} \tag{106}$$

happens to be zero-state detectable [77, Def. 2.2]. In this setting, the discretization with an implicit scheme turns out to be necessary for assuring passivity (a fact that was not noticed in previous works on the topic). The next step is, in fact, an extension of [77, Thm. 2.6], using a set-valued static output feedback controller. The principle is the same for the discrete-time classical controller design, although in most of the classical literature, the zero-state detectability property is assumed a priori rather than shown as it was done in Theorem 1. A notion that is closely related to feedback passivity is that of control Lyapunov functions. The connection between sliding variables and control Lyapunov functions has been explicitly recognized and exploited, for example, in [76] and [80], although it is worth mentioning that the idea already appears in its early stages in [81] and [82].

Closed-Loop Properties

It is well known that, in continuous time, a sliding-mode controller gives rise to a DI. For a properly designed controller, closed-loop solutions converge to a sliding surface $\{\xi \in \mathbb{R}^n \mid \sigma(\xi) = 0\}$ in finite time. Moreover, once the state has reached the sliding surface, a solution stays on the surface and becomes independent from any perturbations matched by the control. The latter is a remarkable property that can be explained only by the set-valued nature of the closed-loop vector field. A suitably discretized controller should approximately replicate these properties. This is the case, for example, with the implicitly defined control law (66) since the condition $\sigma(x_k, u_k) = 0$ is attained in finite time and maintained thereafter, regardless of any bounded matched perturbation.

The discretization $u_{sv}(x_k) \in -\tilde{K}\mathbf{Sgn}(\tilde{K}\tilde{s}_k)$ produces digital chattering and loses the robustness and stability properties of its continuous-time counterpart, no matter how small $h > 0$ is. The implicit discretization, on the other hand, maintains these properties, even for a relatively large h (a fact verified experimentally). At the same time, implicit discretizations are consistent in the sense that discrete-time solutions and inputs converge to their continuous-time counterparts as $h \downarrow 0$ [21], [44]. When written explicitly, the implicit (backward-Euler) discretization takes the form of a projection function of the state on a set, which in some simple instances can be computed using saturation functions. Note, however, that when using the implicit discretization, correct parameters for the saturation functions are automatically provided by the method; no trial-and-error tuning is required.

The properties described in the previous two paragraphs can be formulated using asymptotic notation. Consider, for example, the nominal case. If the state is inside a neighborhood of order two of the sliding surface, $\|s_k\| = O(h^2)$, then at the following step, we have only $\|s_{k+1}\| = O(h)$, and the state exits the neighborhood [21, Lemma 10]. If, on the other hand, the discretization is implicit, the sliding variable remains inside a neighborhood of the same order, $\|s_{k+1}\| = O(h^2)$ [21, Lemma 11].

The implicit discretization method results in a controlled system that is also robust with respect to uncertainty in the system parameters [43], [44]. With the

Contrary to a widely spread idea, SMC is not intrinsically related to discontinuous and infinitely fast switching inputs (bang-bang-like controllers).

appropriate choice of the maximal monotone operators, it is also possible to obtain robustness with respect to unmatched external perturbations as well [45], [65]. The explicit computation of the control law can be carried out in several ways, depending on the complexity of the problem. In some cases, it is possible to obtain an analytic expression [for example, as in (102)]. In more complex scenarios, it is always possible to resort to the numerical machinery described in the "Computational Issues" section.

On the Nature of SMC

Contrary to a widely spread idea, SMC is not intrinsically related to discontinuous and infinitely fast switching inputs (bang-bang-like controllers). The implicit discretization shows this fact as it allows us to correctly approximate not only the closed-loop system's output s but, most importantly, the set-valued input. Two fundamental notions are prominent: the selection of a set-valued map (Definition S3 in "Set-Valued Mappings and DIs") and the calculation of the controller with optimization tools, as detailed later. See Figure 8 for a typical illustration of discontinuous inputs and selections of set-valued control inputs.

HIGHER-ORDER SET-VALUED CONTROLLERS AND DIFFERENTIATORS

The goal of this section is to show how the foregoing developments extend to "modern" sliding-mode algorithms dedicated to control, state observation, or differentiation. Most importantly, higher-order schemes, introduced by Levant and coauthors [8], [9], [10], have witnessed considerable attention in the past 20 years. Roughly speaking, first-order schemes permit the designer to guarantee that a sliding variable (denoted s in the "First-Order SMC: Continuous Time" section) vanishes in finite time. Then, the origin is attained asymptotically while the state trajectories evolve on the sliding surface. Higher-order schemes of order r (order-r schemes) guarantee the finite-time convergence of the sliding variable s and its derivatives up to the $(r-1)$th order to zero. This is called an r-sliding surface, and the r-sliding mode is defined by the invariance of $\{x \in \mathbb{R}^n \mid s(x) = \dot{s}(x) = \ldots = s^{(r-1)}(x) = 0\}$ along a system trajectory. From a practical point of view, the higher-order sliding-mode method offers several advantages: 1) in the extreme case $r = n > 1$, the origin is attained in finite time, a feature not possible with first-order schemes; 2) by adding integrators at the plant's input, it is possible to steer the plant using arbitrarily smooth controls; and 3) given a signal with the bounded $(r+1)$th time derivative, it is possible to recover the first r derivatives.

The main principle used in the discretization schemes presented in previous sections is structure preservation, that is, the preservation of passivity and maximal monotonicity. Structure preservation presents several challenges in the case of r-SMC. First, a necessary condition for establishing an r-sliding mode on s is that the sliding output has relative degree r. Thus, for $r > 1$, the plant cannot be the feedback equivalent of a passive system. Second, most higher-order SMC algorithms are not defined by maximal monotone DIs. Finally, the main underlying structure is homogeneity [83], a property that is not meaningful in discrete time. However, we will show in this section that maximal monotonicity still plays an essential role in the backward-Euler discretization of such schemes.

The first part of this section is dedicated to the supertwisting algorithm [10], which is certainly the most widespread modern sliding-mode scheme. Then, other higher-order differentiators and controllers are presented. The section ends with a short discussion about homogeneity-based schemes and on how proximal-point algorithms appear in this setting.

Supertwisting Algorithm

The supertwisting is an order-2 algorithm. It has proven useful for control, differentiation, and state observation. The following presents all three application fields.

Supertwisting Differentiator

Consider a signal $f : \mathbb{R} \to \mathbb{R}$ and suppose that we wish to differentiate it in real time. A common solution in practice is to replace $\dot{f}$ by its *dirty derivative*, $\hat{g}$, which is obtained by passing f through a linear filter of the form

$$\begin{aligned} \dot{z}(t) &= -\gamma\,(z(t) + \gamma f(t)) \\ \hat{g}(t) &= z(t) + \gamma f(t) \end{aligned} \tag{107}$$

with $\gamma \gg 1$. It can be easily shown that as γ approaches $+\infty$, the transfer function of the filter converges to that of a differentiator. This high-gain strategy benefits from simplicity, but it has two important drawbacks: noise amplification and phase lag. If $\dot{f}$ is Lipschitz continuous with the Lipschitz constant $L > 0$, then a supertwisting differentiation algorithm can be used. The

advantages of such a scheme are 1) exact differentiation with finite gains in the absence of noise, 2) limited noise amplification, and 3) no phase lag.

We will make use of the operator $\lceil w \rfloor^\alpha := |w|^\alpha \mathbf{sgn}(w)$, $\alpha \geq 0$, $w \in \mathbb{R}$. Notice that $\lceil w \rfloor^\alpha = \partial \zeta_\alpha(w)$ with

$$\zeta_\alpha(w) = \frac{1}{\alpha+1} \mid w \mid^{\alpha+1}. \tag{108}$$

The convexity of ζ_α immediately guarantees the maximal monotonicity of $\lceil \cdot \rfloor^\alpha$; see Theorem S2 in "Maximal Monotone Operators." Notice also that $\lceil \cdot \rfloor^\alpha$ is single-valued for $\alpha > 0$ but multivalued at the origin when $\alpha = 0$.

The supertwisting differentiation algorithm [6] reads as

$$\begin{aligned} \dot{z}_0(t) &= z_1(t) - \gamma_0 L^{\frac{1}{2}} s_0(t)^{\frac{1}{2}} \\ \dot{z}_1(t) &\in -\gamma_1 L \mathbf{sgn}(s_0(t)) \end{aligned} \tag{109}$$

where $s_0(t) := z_0(t) - f(t)$ and γ_0, γ_1 are positive gains. The differentiator properties are best appreciated under the coordinate transformation $\sigma_0(t) := (1/L)(z_0(t) - f(t))$ and $\sigma_1(t) := (1/L)(z_1(t) - \dot{f}(t))$. In these coordinates, system (109) is rewritten in recursive form as [84]

$$\begin{aligned} \dot{\sigma}_0(t) &= \sigma_1(t) - \gamma_0 \sigma_0(t)^{\frac{1}{2}} \\ \dot{\sigma}_1(t) &\in -\gamma_1 \mathbf{sgn}(\sigma_0(t)) - \ddot{f}(t) \end{aligned} \tag{110}$$

If γ_0 and γ_1 are appropriately chosen, then the origin $(\sigma_0, \sigma_1) = (0,0)$ is reached in finite time, and thus the condition $s_0(t) = \dot{s}_0(t) = 0$ is attained and maintained after a finite time [85]. Hence, the differentiator attains a two-sliding surface. On the sliding surface $s_0(t) \equiv 0$, we have $z_1(t) \equiv \dot{f}(t)$, which shows that the signal's derivative is available in finite time. The backward-Euler (or implicit-Euler) discretization of (109) is

$$s_{0,k+1} = s_{0,k} + h z_{1,k+1} - h\gamma_0 L^{\frac{1}{2}} \lceil s_{0,k+1} \rfloor^{\frac{1}{2}} - \Delta f_k \tag{111a}$$

$$z_{1,k+1} \in z_{1,k} - h\gamma_1 L \mathbf{sgn}(s_{0,k+1}) \tag{111b}$$

where $\Delta f_k := f_{k+1} - f_k$. Our purpose now is to derive an explicit representation of (111). Substituting (111b) into (111a) and solving for $s_{0,k+1}$ yields

$$s_{0,k+1} = (I_d + h\mathbf{M}_h)^{-1}(s_{0,k} + h z_{1,k} - \Delta f_k) \tag{112}$$

where

$$\mathbf{M}_h(w) = \gamma_0 L^{\frac{1}{2}} \lceil w \rfloor^{\frac{1}{2}} + h\gamma_1 L \mathbf{sgn}(w) \tag{113}$$

is a maximal monotone operator (being the sum of two maximal monotone operators, both with domain $\mathbb{R}$). The iteration (112) is compactly written as the resolvent-based algorithm

$$s_{0,k+1} = \mathcal{J}_{h\mathbf{M}_h}(s_{0,k} + h z_{1,k} - \Delta f_k) \tag{114}$$

[see (S31) and Proposition S2 in "Proximal Mapping and Proximal-Point Algorithm"], where the resolvent is defined in (S17) in "Maximal Monotone Operators" [compare with the classical algorithm (98) and the general iteration (S31)]. It is interesting to note that choosing an exact discretization as in [86] and neglecting unmeasurable terms also yields a form like (114) [86, eq. (24)]. Together with (10), (75), and (98), this shows the ubiquity of resolvent-based algorithms in the implicit discretization of sliding-mode systems. The algorithms provided in [86] and [87] also correspond to the resolvent in (114). As seen later in this article, there is no unique way to compute the resolvents numerically.

With an explicit expression for $s_{0,k+1}$ available in (114), we can readily solve for the state $z_{1,k+1}$ for all $h > 0$ in (111a)

$$z_{1,k+1} = \frac{1}{h}\left(s_{0,k+1} - s_{0,k} + h\gamma_0 L^{\frac{1}{2}} \lceil s_{0,k+1} \rfloor^{\frac{1}{2}} + \Delta f_k\right). \tag{115}$$

Therefore, the pair of (114) and (115) forms a nonlinear difference equation with the state vector $(s_{0,k} \;\; z_{1,k})^\top$ and allows us to advance the discretized supertwisting differentiator from step k to step $k+1$. Notice in passing that, having computed the future state $z_{1,k+1}$ in (115), the equation $z_{1,k+1} = z_{1,k} - h\gamma_1 L \lambda_{k+1}$ can be used to compute a selection $\lambda_{k+1} \in \mathbf{sgn}(s_{0,k+1})$ such that (111b) holds.

Remark 7

In real-time applications, it is possible to substitute Δf_k by Δf_{k-1} in (111a), and hence in (115), to render the differentiator nonanticipative.

Semi-implicit discretization is also possible. Replace (111) by

$$s_{0,k+1} \in s_{0,k} + h z_{1,k+1} - h\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} sgn(s_{0,k+1}) - \Delta f_k \tag{116a}$$

$$z_{1,k+1} \in z_{1,k} - h\gamma_1 L sgn(s_{0,k+1}) \tag{116b}$$

where the implicit terms have been restricted to the set-valued functions. Solving again for $s_{0,k+1}$ now leads to

$$s_{0,k+1} = (I_d + (h\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} + h^2\gamma_1 L)\mathbf{sgn})^{-1}(s_{0,k} + h z_{1,k} - \Delta f_k). \tag{117}$$

Compared to (113), the maximal monotone operator is modified to

$$\mathbf{M}_{k,h}(w) = (\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} + h\gamma_1 L)\mathbf{sgn}(w). \tag{118}$$

Calculating $z_{1,k+1}$ is somewhat more complicated than before. First, define the selection $\lambda_{k+1} \in \mathbf{sgn}(s_{0,k+1})$ and recall that it holds if and only if $s_{0,k+1} \in \mathbf{N}_{[-1,1]}(\lambda_{k+1})$ [see Definition S3 in "Set-Valued Mappings and DIs" and Fact 9 and (S55) in "Convex Analysis Tools"]. By inverting the operators in (116), the system in the new coordinates can be rewritten as

$$s_{0,k} + hz_{1,k+1} - h\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} \lambda_{k+1} - \Delta f_k \in \mathbf{N}_{[-1,1]}(\lambda_{k+1}) \tag{119a}$$

$$z_{1,k+1} = z_{1,k} - h\gamma_1 L \lambda_{k+1} \tag{119b}$$

and then solved for λ_{k+1}. To do so, first substitute (119b) in (119a) to obtain

$$\lambda_{k+1} \in \frac{1}{h^2\gamma_1 L}\Big(s_{0,k} + hz_{1,k} - h\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} \lambda_{k+1} - \Delta f_k - \mathbf{N}_{[-1,1]}(\lambda_{k+1})\Big). \tag{120}$$

The inclusion is equivalent to

$$\lambda_{k+1} = \mathcal{J}_{\mathbf{M}_{k,h}}\left(\frac{s_{0,k} + hz_{1,k} - \Delta f_k}{h^2\gamma_1 L}\right) \tag{121}$$

$$\mathbf{M}_{k,h}(w) = \frac{\gamma_0}{h\gamma_1 L^{\frac{1}{2}}} |s_{0,k}|^{\frac{1}{2}} w + \mathbf{N}_{[-1,1]}(w). \tag{122}$$

Finally, $z_{1,k+1}$ is recovered from (119b) as

$$z_{1,k+1} = z_{1,k} - h\gamma_1 L\, \mathcal{J}_{\mathbf{M}_{k,h}}\left(\frac{s_{0,k} + hz_{1,k} - \Delta f_k}{h^2\gamma_1 L}\right). \tag{123}$$

The resolvent $\mathcal{J}_{\mathbf{M}_{k,h}}(\cdot)$ can be calculated as indicated in "Proximal Mapping and Proximal-Point Algorithm," after (S29). Resolvents of this kind are basic tools found in all the higher-order algorithms in this section.

Super-Twisting Controller

When the supertwisting algorithm is used for control (as opposed to differentiation), a perturbed integrator $\dot{x}_1(t) = u(t) + \varphi(t)$ is considered. Such dynamics may stem from some transformation of the plant's dynamics via feedback, or x_1 may represent a sliding variable similar to s in the first-order schemes; hence, it may be a combination of the plant's state variables [88]. The proposed controller is given by

$$\begin{aligned} u(t) &= \nu(t) - \gamma_0 L^{\frac{1}{2}} \lfloor x_1(t) \rceil^{\frac{1}{2}}, \\ \dot{\nu}(t) &\in -\gamma_1 L \mathbf{sgn}(x_1(t)). \end{aligned} \tag{124}$$

The control u in (124) can be interpreted as a nonlinear nonsmooth set-valued proportional-integral input. The main interest in this controller stems from the fact that, contrary to the first-order case, the applied control u is a continuous function of the state. By making the time-varying change of variable $x_2(t) = \nu(t) + \varphi(t)$, one obtains the closed-loop system

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) - \gamma_0 L^{\frac{1}{2}} \lfloor x_1(t) \rceil^{\frac{1}{2}} \\ \dot{x}_2(t) &\in -\gamma_1 L \mathbf{sgn}(x_1(t)) + \delta(t) \end{aligned} \tag{125}$$

where $\delta(t) = \dot{\varphi}(t)$. The closed-loop dynamics (125) is the same as the differentiator's dynamics (110). The finite-time stability of the origin $x_1 = x_2 = 0$ is guaranteed under similar assumptions on the gains [85]; hence, $x_1 = \dot{x}_1 = 0$ defines the two-sliding surface. Notice that, similarly to (24) and (42), the controller compensates exactly for the perturbation on the sliding surface $\{0\} \subset \mathbb{R}^2$, along which $u(t) = \nu(t) = -\varphi(t)$.

The objective now is to compute a discrete-time version of the controller, similar to the explicit expressions (13) and (14). A discrete-time model for the plant is

$$x_{1,k+1} = x_{1,k} + hu_k + h\bar{\varphi}_k \tag{126a}$$

$$\varphi_{k+1} = \varphi_k + h\bar{\delta}_k. \tag{126b}$$

Here, $\bar{\varphi}_k$ and $\bar{\delta}_k$ are discretized versions of $\varphi(t)$ and $\delta(t)$ (see [35] for details). The implicit discretization can now be achieved by following the same steps as in (111). Introduce, for instance, a nominal virtual variable $\tilde{x}_{1,k} := x_{1,k} - h\bar{\varphi}_{k-1}$, as done in (65) and (95); see [35, eq. (8) and (9)].

According to (126a), the nominal variable evolves by

$$\tilde{x}_{1,k+1} = x_{1,k} + hu_k \tag{127}$$

while the nominal backward-Euler discretization of (124) is

$$\begin{aligned} u_k &\in -\gamma_0 L^{\frac{1}{2}} |\tilde{x}_{1,k+1}|^{\frac{1}{2}} \mathbf{sgn}(\tilde{x}_{1,k+1}) + \nu_{k+1}, \\ \nu_{k+1} &= \nu_k - h\gamma_1 L \mathbf{sgn}(\tilde{x}_{1,k+1}). \end{aligned} \tag{128}$$

Substitution of (128) in (127) gives

$$\begin{aligned} \tilde{x}_{1,k+1} &= (I_{\mathrm{d}} + h\mathbf{M}_h)^{-1}(x_{1,k} + h\nu_k) \\ &= \mathcal{J}_{h\mathbf{M}_h}(x_{1,k} + h\nu_k) \end{aligned} \tag{129}$$

with $\mathbf{M}_h$ as in (113). The explicit expression for the control in (128) is as follows [35], [89]:

$$\begin{aligned} u(x_{1,k}, \nu_k) = &-\gamma_0 L^{\frac{1}{2}} \beta_k \mathbf{sgn}(x_{1,k} + h\nu_k) \\ &+ \nu_k - h\gamma_1 L \mathrm{Proj}\Big([-1,1]; \frac{x_{1,k} + h\nu_k}{h^2\gamma_0 L}\Big) \end{aligned} \tag{130}$$

with

$$\begin{aligned} \beta_k &= \frac{-h\gamma_0 L^{\frac{1}{2}}}{2} + \sqrt{\frac{h^2\gamma_0^2 L}{4} + \rho_k} \\ \rho_k &= \max\{0, |x_{1,k} + h\nu_k| - h^2\gamma_1 L\} \end{aligned} \tag{131}$$

while

$$\nu_{k+1} = \nu_k - h\gamma_1 \mathrm{Proj}\Big([-1,1]; \frac{x_{0,k} + h\nu_k}{h\mu_k}\Big). \tag{132}$$

Alternatively, we may rewrite the control as

$$u(x_{1,k}, \nu_k) = \frac{1}{h}(I_d + h\mathbf{M}_h)^{-1}(x_{1,k} + h\nu_k) - \frac{x_{1,k}}{h}. \tag{133}$$

It is noteworthy that this boils down once again to the calculation of a resolvent and of a projection (the saturation function). That is, (129) defines a resolvent-based algorithm similar to (114) and (116).

As mentioned previously, a semi-implicit method can be applied [36]. The method yields

$$\tilde{x}_{1,k+1} = (I_d + h\mathbf{M}_{k,h})^{-1}(x_{1,k} + h\nu_k) \tag{134}$$

but now with

$$\mathbf{M}_{k,h}(w) = (\gamma_0 L^{\frac{1}{2}} |x_{1,k}|^{\frac{1}{2}} + h\gamma_1 L)\,\mathbf{sgn}(w) \tag{135}$$

similar to (118). Another choice is

$$\mathbf{M}_{k,h}(w) = (\gamma_0 L^{\frac{1}{2}} |\tilde{x}_{1,k}|^{\frac{1}{2}} + h\gamma_1 L)\,\mathbf{sgn}(w). \tag{136}$$

Which alternative should be chosen is an open question. Note that, during the sliding motion, $\tilde{x}_{1,k+1} = \tilde{x}_{1,k} = 0$, and the inclusion

$$\nu_k + \bar{\varphi}_k \in \frac{1}{h}(I_d + h\mathbf{M}_h)(0) = h\gamma_0 L[-1,1] \tag{137}$$

follows from (129). Using (130) and (131), we obtain

$$u(x_{1,k}, \nu_k) = -\frac{x_{1,k}}{h} = -\frac{\tilde{x}_{1,k} + h\bar{\varphi}_{k-1}}{h} = -\bar{\varphi}_{k-1}. \tag{138}$$

It is inferred that, similarly to first-order controllers [see (100)], the implicit supertwisting scheme compensates for the perturbation with a one-step delay.

Multivariable Super-Twisting

The previous supertwisting scheme in (124) extends to the multivariable supertwisting [90], with $x_1,\ x_2 \in \mathbb{R}^n$. One replaces (108) by

$$\zeta_\alpha(w) = \frac{1}{\alpha+1}\|w\|_2^{\alpha+1} \tag{139}$$

so that instead of $|x_1|^{\frac{1}{2}}\mathbf{sgn}(x_1)$, one has

$$\|x_1\|_2^{\frac{1}{2}}\partial\|x_1\|_2 = \|x_1\|_2^{\frac{1}{2}}\mathbf{N}_{\mathcal{B}_2}^{-1}(x_1) \tag{140}$$

[see (S59) in "Convex Analysis Tools"], and $\mathbf{sgn}(x_1)$ by $\partial\|x_1\|_2$ (or the subdifferential of any other norm), in (124). This yields the resolvent $\mathcal{J}_{h\mathbf{M}_h}$, to be computed at each step with

$$\mathbf{M}_h(w) = \alpha_1\|w\|_2^{\frac{1}{2}}\partial\|w\|_2 + h\alpha_2\partial\|w\|_2 \tag{141}$$

$\alpha_1 > 0$, $\alpha_2 > 0$. Note that $\mathbf{M}_h = \partial(\alpha_1\zeta_{1/2} + h\alpha_2\zeta_0)$. It follows from the convexity of ζ_α and Theorem S2 in "Maximal Monotone Operators" that $\mathbf{M}_h$ is maximal monotone.

Generalized Observer

A further interesting class of state observers is analyzed next. So far, we have seen that the supertwisting algorithm is suitable for differentiating a signal. Differentiation and state observation are notions close to each other. The equivalence between uniform observability and the ability to recover the complete state by differentiating the system output is well known. In view of (109), it is not surprising that the supertwisting algorithm can be used to observe planar systems. Recall that a planar system is uniformly observable if and only if there exists a coordinate transformation such that the system can be written as follows [91]:

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, u) + x_2 \\ \dot{x}_2 &= f_2(x_1, x_2, u) \\ y &= x_1 \end{aligned} \tag{142}$$

from where it is clear that the remaining state x_2 can be recovered by differentiating the output y. Consider the class of set-valued supertwisting observers presented in [92, eq. (12.3)–(12.5)]

$$\dot{\hat{x}}_1 = f_1(\hat{x}_1, u) + \hat{x}_2 - l_1\gamma\phi_1(e_1) \tag{143a}$$

$$\dot{\hat{x}}_2 \in f_2(\hat{x}_1, \hat{x}_2, u) - l_2\gamma^2\phi_2(e_1) \tag{143b}$$

$$\phi_1(e_1) = \mu_1\lceil e_1\rfloor^{\frac{1}{2}} + \mu_2\lceil e_1\rfloor^q \tag{143c}$$

$$\phi_2(e_1) = \frac{\mu_1^2}{2}\,\mathbf{sgn}(e_1) + \mu_1\mu_2\lceil e_1\rfloor^{q-\frac{1}{2}} + \mu_2^2\lceil e_1\rfloor^{2q-1} \tag{143d}$$

where $\mu_1 \geq 0$, $\mu_2 \geq 0$, $\mu_1\mu_2 \neq 0$, $q \geq (1/2)$, $l_1 > 0$, $l_2 > 0$, $\gamma > 0$, and $e_1 = \hat{x}_1 - x_1$ is the state estimation error. An implicit discretization of (143) reads

$$\hat{x}_{1,k+1} = \hat{x}_{1,k} + hf_1(\hat{x}_{1,k+1}, u_k) + h\hat{x}_{2,k+1} - hl_1\gamma\phi_1(e_{1,k+1}) \tag{144a}$$

$$\hat{x}_{2,k+1} \in \hat{x}_{2,k} + hf_2(\hat{x}_{1,k+1}, \hat{x}_{2,k+1}, u_k) - hl_2\gamma^2\phi_2(e_{1,k+1}) \tag{144b}$$

with $e_{1,k+1} = \hat{x}_{1,k+1} - y_k$. Noting that $\phi_2(e_{1,k+1})$ is a set-valued function of $e_{1,k+1}$, it is assumed that (144b) can be solved to get $\hat{x}_{2,k+1} \in \Phi(e_{1,k+1}, y_k, \hat{x}_{2,k}, u_k, h\phi_2(e_{1,k+1}))$ for some function Φ. Inserting the latter in (144a) yields

$$\begin{aligned} e_{1,k+1} \in\ & y_{k-1} - y_k + e_{1,k} + hf_1(e_{1,k+1}, y_k, u_k) \\ & + h\Phi(e_{1,k+1}, y_k, \hat{x}_{2,k}, u_k, h\phi_2(e_{1,k+1})) - hl_1\gamma\phi_1(e_{1,k+1}). \end{aligned} \tag{145}$$

It is inferred that

$$\begin{aligned} e_{1,k+1} \in\ & (I_d - hf_1(\cdot, y_k, u_k) + h\Phi(\cdot, y_k, \hat{x}_{2,k}, u_k, h\phi_2(\cdot)) \\ & + hl_1\gamma\phi_1(\cdot))^{-1}(y_{k-1} - y_k + e_{1,k}). \end{aligned} \tag{146}$$

Then, provided that the operators $x \mapsto -f_1(x, y_k, u_k) + l_1\gamma\phi_1(x)$ and $x \mapsto \Phi(x, y_k, \hat{x}_{2,k}, u_k, h\phi_2(x))$ are maximal monotone (notice that ϕ_1 and ϕ_2 are strongly monotone for $q > (1/2)$), the inclusion in (146) is an equality, and the right-hand side is a resolvent. Depending on the nonlinearities, the closed form of this resolvent may be impossible to obtain, in which case the numerical solutions described later are necessary. The observed variable $\hat{x}_{2,k+1}$ is obtained from (144a) and (146). Of course, semi-implicit methods can also be applied, making computations easier and allowing for relaxed assumptions.

Higher-Order Differentiators

The supertwisting differentiator calculates only the first-order derivative. It is sometimes necessary to compute higher-order derivatives of f. A naive approach is to cascade several differentiators of the form (109) (which may provide reasonably good results for the second derivative [93]). A more efficient solution is to directly construct a differentiator of order $r-1$, $r\geq 2$ by enforcing an r-sliding mode. The arbitrary-order supertwisting differentiator (AO-STD) has the dynamics in nonrecursive form as follows [7], [84]:

$$\begin{aligned}&\dot{z}_i(t)=z_{i+1}(t)-\gamma_i L^{\frac{i+1}{r+1}}\lceil s_0(t)\rfloor^{\frac{r-i}{r+1}},\quad i=0,\ldots,r-1,\\&\dot{z}_r(t)\in-\gamma_r L\,\mathbf{sgn}(s_0(t))\end{aligned}\tag{147}$$

with $s_0(t)=z_0(t)-f(t)$. The differentiator (147) is the counterpart of (109), while the counterpart of (110) is written in the recursive form

$$\begin{aligned}&\dot{\sigma}_i(t)=\sigma_{i+1}(t)-\gamma_i\lceil\sigma_0(t)\rfloor^{\frac{r-i}{r+1}},\quad i=0,\ldots,r-1,\\&\dot{\sigma}_r(t)\in-\gamma_r\mathbf{sgn}(\sigma_0(t))+f^{(r+1)}(t)\end{aligned}\tag{148}$$

where $\sigma_i(t):=(1/L)(z_i(t)-f^{(i)}(t))$. The accuracy and robustness of these differentiators have been studied by exploiting the homogeneity of (148) [50], [84]. As shown in [87], [86], and [94], their discrete-time versions lend themselves to transformations that yield the same structures as in (129) or (114). In particular, it is shown in [87, Table 2] that "classical" higher-order differentiators known as uniform robust exact differentiator (URED) [95], AO-STD [7], [84], homogeneous discrete-time differentiator, generalized homogeneous discrete-time differentiator [96], first-order differentiator with first-order sliding-mode filtering [5], [97], and arbitrary-order differentiator with first-order sliding-mode filtering [97], [98] yield, when implicitly or semi-implicitly discretized, the equation

$$s_{0,k+1}=\mathcal{J}_{h\mathbf{M}_h}(-b_k)\tag{149}$$

where $\mathbf{M}_h=\mathbf{M}_{\mathrm{sing},h}+\mathbf{M}_{\mathrm{set},h}$ with $\mathbf{M}_{\mathrm{set},h}$ set-valued maximal monotone, $\mathbf{M}_{\mathrm{sing},h}$ single-valued monotone, and $b_k=b(s_{0,k},z_{1,k},\ldots,z_{n,k},\Delta f_k)$, $\Delta f_k=f_{k+1}-f_k$.

We now provide the implicit URED operators, which consist of the supertwisting operators with additional higher-order terms [87, Table 2, eq. (39)]. Its defining operators are

$$\mathbf{M}_{\mathrm{sing},h}(w)=\gamma_0 L^{\frac{1}{2}}\left(\lceil w\rfloor^{\frac{1}{2}}+\mu\lceil w\rfloor^{\frac{3}{2}}\right)+\frac{3}{2}h\mu^2\gamma_1 L\lceil w\rfloor^2+2h\gamma_1 L\mu w\tag{150a}$$

$$\mathbf{M}_{\mathrm{set},h}(w)=\frac{1}{2}h\gamma_1 L\,\mathbf{sgn}(w)\tag{150b}$$

and the argument is

$$b(s_{0,k},z_{1,k},\Delta f_k)=-s_{0,k}-hz_{1,k}+\Delta f_k\tag{150c}$$

where λ_0, λ_1, μ are parameters. Similarly to the supertwisting differentiator, the URED has an additional state variable x_1; see, for example, [87, eq. (6)]. The URED counterpart of (115) takes the form

$$z_{1,k+1}=\frac{1}{h}(s_{0,k+1}-s_{0,k}+\Delta f_k+h\gamma_0 L^{\frac{1}{2}}\lceil s_{0,k+1}\rfloor^{\frac{1}{2}}+hL^{\frac{1}{2}}\mu\lceil s_{0,k+1}\rfloor^{\frac{3}{2}}).\tag{151}$$

We now derive the semi-implicit version of this algorithm. First, $s_{0,k+1}=\mathcal{J}_{h\mathbf{M}_{k,h}}(-b_k)$ is computed with

$$\begin{aligned}\mathbf{M}_{k,h}(w)=\Big(&\gamma_0 L^{\frac{1}{2}}\left(|s_{0,k}|^{\frac{1}{2}}+\mu|s_{0,k}|^{\frac{3}{2}}\right)+\frac{3}{2}h\mu^2\gamma_1 L s_{0,k}^2\\&+h\gamma_L|s_{0,k}|+\frac{1}{2}h\gamma_1 L\Big)\,\mathbf{sgn}(w)\end{aligned}\tag{152}$$

which is the URED counterpart of (117). Let $\lambda_{k+1}\in\mathbf{sgn}(s_{0,k+1})$ be a selection of the set-valued sign function. Solving for $s_{0,k+1}\in\mathbf{N}_{[-1,1]}(\lambda_{k+1})$ gives

$$\begin{aligned}&s_{0,k}+hx_{1,k}-\Delta f_k-h\gamma_0 L^{\frac{1}{2}}\left(|s_{0,k}|^{\frac{1}{2}}+\mu|s_{0,k}|^{\frac{3}{2}}\right)\lambda_{k+1}-\\&\quad h^2\gamma_1 L\left(2\mu|s_{0,k}|+\frac{3}{2}\mu^2 s_{0,k}^2+\frac{1}{2}\right)\lambda_{k+1}\in\mathbf{N}_{[-1,1]}(\lambda_{k+1})\end{aligned}\tag{153}$$

so that $\lambda_{k+1}=\mathcal{J}_{\mathbf{M}_{k,h}}\left(-\dfrac{2b_k}{h^2\gamma_1 L}\right)$ with

$$\begin{aligned}\mathbf{M}_{k,h}=\frac{2}{h\lambda_1 L^{\frac{1}{2}}}\Big(&\gamma_0\left(|s_{0,k}|^{\frac{1}{2}}+\mu|s_{0,k}|^{\frac{3}{2}}\right)\\&+h\gamma_1 L^{\frac{1}{2}}\left(2\mu|s_{0,k}|+\frac{3}{2}\mu^2 s_{0,k}^2+\frac{1}{2}\right)\Big)I_d+\mathbf{N}_{[-1,1]}.\end{aligned}\tag{154}$$

Finally, we can recover

$$z_{1,k+1}=z_{1,k}+h\gamma_1 L\left(2\mu|s_{0,k}|+\frac{3}{2}\mu^2 s_{0,k}^2+\frac{1}{2}\right)\lambda_{k+1}\tag{155}$$

which is the URED counterpart of (123). Thus, (114) and (115), (117) and (123), (149) and (151), and (152) and (155) make nonlinear difference equations with the state vector $(s_{0,k}\ \ z_{1,k})^\top$. Some of these may be named *higher-order* proximal-point algorithms since inserting, for example, (115) into (114) yields a difference equation with $s_{0,k}$ and $s_{0,k-1}$ in its right-hand side. From the point of view of the resolvent calculation at step k, all terms depending only on measurements at k are considered as being constants.

Similar developments can be carried out for the AO-STD in both implicit and semi-implicit cases [87, Sect. 5.1 and 5.8]. For instance, the AO-STD of order 3 yields $s_{0,k+1}=\mathcal{J}_{h\mathbf{M}_{k,h}}(-b_k)$ with

$$\mathbf{M}_{k,h}(w)=\sum_{l=0}^{2}h^l\gamma_l L^{\frac{l+1}{4}}\lceil w\rfloor^{\frac{3-l}{4}}+h^3\gamma_3 L\,\mathbf{sgn}(w)\tag{156}$$

for the implicit method, and

$$\mathbf{M}_{k,h}(w)=\left(\sum_{l=0}^{2}h^l\gamma_l L^{\frac{l+1}{4}}|s_{0,k}|^{\frac{3-l}{4}}+h^3\gamma_3 L\right)\mathbf{sgn}(w)\tag{157}$$

for the semi-implicit method. The remaining state variables $z_{1,k+1}$, $z_{2,k+1}$, $z_{3,k+1}$ (which are the higher-order derivatives of f) are calculated similarly as for the supertwisting and the URED, yielding other higher-order proximal-point algorithms.

Relay Polynomial Controllers

As mentioned previously, an interesting feature of higher-order sliding modes is the possibility for the state to attain the origin in finite time. There exist several classes of algorithms achieving this objective (for instance, *quasi-continuous* or *nested* controllers [99]). The following algorithm is one of the simplest, which is why we present it. Consider a perturbed chain of integrators

$$\begin{aligned}\dot{x}_i(t) &= x_{i+1}(t), \quad i = 1,\ldots,n-1,\\ \dot{x}_n(t) &= u(t) + \delta(t).\end{aligned} \tag{158}$$

The *relay polynomial controller* $u(t) = \hat{u}(x(t))$ with

$$\hat{u}(x) \in -\gamma_n \mathbf{sgn}\left(\lceil x_n \rfloor^n + \gamma_{n-1}\lceil x_{n-1}\rfloor^{\frac{n}{2}} + \cdots + \gamma_1 x_1\right) \tag{159}$$

establishes an n-sliding mode at the origin in finite time, provided that the perturbation $\delta(t)$ is uniformly bounded and that the gains $\gamma_i > 0$ are appropriately chosen [80], [100]. The forward-Euler discretization of the open-loop system is

$$\begin{aligned}x_{i,k+1} &= x_{i,k} + h x_{i+1,k}, \quad i = 1,\ldots,n-1,\\ x_{n,k+1} &= x_{n,k} + h u_k + \delta_k.\end{aligned} \tag{160}$$

The backward-Euler discretization of (159) takes the implicit form $u_k \in -\gamma_n \mathbf{M}_{k,h}(u_k)$ with

$$\begin{aligned}\mathbf{M}_{k,h}(w) = \mathbf{sgn}\Big(&\lceil x_{n,k} + h w^n \rfloor + \gamma_{n-1}\lceil x_{n-1,k} + h x_{n,k}\rfloor^{\frac{n}{2}}\\ &+ \cdots + \gamma_1 (x_{1,k} + h x_{2,k})\Big).\end{aligned} \tag{161}$$

It is not difficult to see that, for every x_k, $\mathbf{M}_{k,h}$ is maximal monotone; hence, a unique solution $u_k(x_k) = (I_d + \gamma_n \mathbf{M}_{k,h})^{-1}(0)$ exists. In fact, it is possible to write an analytic expression for the solution. Note that $u_k \in -\gamma_n \mathbf{M}_{k,h}(u_k)$ is equivalent to

$$u_k \in -\gamma_n \mathbf{sgn}(u_k - \bar{u}(x_k)) \tag{162}$$

where

$$\begin{aligned}\bar{u}(x_k) = -\frac{1}{h}\Big(&\lceil \gamma_{n-1}\lceil x_{n-1,k} + h x_{n,k}\rfloor^{\frac{\alpha}{2}} + \cdots\\ &+ \gamma_1 \lceil x_{1,k} + h x_{2,k}\rfloor^{\frac{\alpha}{n}}\rfloor^{\frac{1}{\alpha}} + x_{n,k}\Big).\end{aligned} \tag{163}$$

The control action is thus $u_k = \bar{u}(x_k) - \mathcal{J}_{\gamma_n \mathbf{sgn}}(\bar{u}(x_k))$, that is

$$u_k = \min\{|\bar{u}(x_k)|, \gamma_n\}\mathbf{sgn}(\bar{u}(x_k)), \tag{164}$$

which once again is a form of the minimum-operator formalism as in (14b). Now, we consider the ZOH discrete model of the chain of integrators

$$x_{i,k+1} = x_{i,k} + \sum_{j=1}^{n-i} \frac{h^j}{j!} x_{i+j,k} + \frac{h^{n+1-i}}{(n+1-i)!}(u_k + \bar{\delta}_{i,k}) \tag{165}$$

where $i = 1,\ldots n$ and $\bar{\delta}_i$ are (possibly unmatched) disturbances resulting from $\delta(t)$.

We also consider the inclusion $u_k \in -\gamma_n M_{k,h}(u_k)$, where

$$\begin{aligned}\mathbf{M}_{k,h}(w) = \mathbf{sgn}\Big(&\lceil x_{n,k} + h w^n \rfloor + \gamma_{n-1}\Big\lceil x_{n-1,k} + h x_{n,k} + \frac{h^2}{2!} w\Big\rfloor^{\frac{n}{2}}\\ &+ \cdots + \gamma_1 (x_{1,k} + \sum_{j=1}^{n-1} \frac{h^j}{j!} x_{1+j,k} + \frac{h^n}{n!} w)\Big)\end{aligned} \tag{166}$$

results from the exact ZOH discretization of (158). For each fixed x_k, the operator $\mathbf{M}_{k,h}$ is again maximal monotone, so the resolvent $(I_d + \mathbf{M}_{k,h})^{-1}$ has domain $\mathbb{R}$ and is single-valued. While an analytic expression is no longer available, it is still possible to compute the resolvent numerically by using the techniques described in the "Computational Issues" section.

HIGHER-ORDER SCHEMES: STABILITY ANALYSIS

The systems (114) and (115), or (116) or (149) and (151), or their semi-implicit version in (152) and (155), are higher-order proximal-point algorithms that do not fit within the robust proximal-point algorithm in (S31) in "Proximal Mapping and Proximal-Point Algorithm." The boundedness of the closed-loop state with (130) has been proved only in the absence of perturbations in [35]. As shown in [89], the algorithm is not robust with respect to an unbounded perturbation. A modification to (130) is proposed in [89] that enhances robustness. Some results concerning higher-order differentiators of the implicit AO-STD type can be found in [87]. Some properties of the implicit higher-order differentiators (sliding surface invariance, accuracy, and finite convergence) are shown in [87] and [94]. The stability of the semi-implicit supertwisting controller is analyzed in [36]; see also the semi-implicit discretization [32, eq. (60)] of the supertwisting velocity observer [101].

HOMOGENEITY AND MAXIMAL MONOTONICITY

It is clear that maximal monotonicity is an essential property for the analysis and the implementation of backward-Euler discretizations (implementation issues are discussed in detail in the "Computational Issues" section). On the other hand, homogeneity is a powerful tool for control design in continuous time [102]. Homogeneity plays an important role in all of the higher-order sliding-mode algorithms presented previously. It is therefore legitimate to tackle the relationships between these two notions. A first—very preliminary—analysis is made in [103, Lemma 1]. Its conclusion is that homogeneous operators and maximal monotone operators rarely match. In this section, we

continue this path of investigation by examining two examples of consistent discretization of homogeneous systems.

While there is no consensus on what homogeneity should be in discrete time, it is possible to construct consistent discretizations by keeping track of the maximal monotone operators that appear in the homogeneous vector fields. Consistent discretizations of finite-time convergent systems have been presented in [19, Def. 2.1 and 2.2]. For example, consider the systems analyzed in [19, Sect. 6.2]. The first analysis concerns the discretization of a generalized homogeneous closed-loop system with homogeneity degree $d=1$ (see [102] for definitions). After a suitable transformation into another homogeneous system, the consistent discrete-time dynamics

$$x_{k+1}=x_k-h\|x_k\|(I_n-\tilde{A})x_{k+1} \tag{167}$$

is obtained [19, Sect. 6.2, eq. (6.8)]. Similar to the discrete dynamics presented in the previous sections, this dynamics is given by the resolvent of a maximal monotone operator. Indeed, these are equivalent to

$$x_{k+1}=(I_d+h\|x_k\|(I_n-\tilde{A}))^{-1}(x_k),\quad k\geq 0 \tag{168}$$

with $\tilde{A}^\top P+P\tilde{A}=0$ for some $P=P^\top\succ 0$ [19, Def. 2.1]. If we define $R=R^\top\succ 0$ and $R^2=P$, then we obtain $R^{-1}(\tilde{A}-I_n)^\top R+R(\tilde{A}-I_n)R^{-1}=-2I_n\prec 0$, from where it follows that $R(I_n-\tilde{A})R^{-1}\succ 0$. Using the new state variable $\bar{x}_k=Rx_k$, the system dynamics takes the form

$$\bar{x}_{k+1}=\bar{x}_k-h\|R^{-1}\bar{x}_k\|R(I_n-\tilde{A})R^{-1}\bar{x}_{k+1} \tag{169}$$

which is equivalent to

$$\bar{x}_{k+1}=(I_d+h\|R^{-1}\bar{x}_k\|R(I_n-\tilde{A})R^{-1})^{-1}(\bar{x}_k),\quad k\geq 0. \tag{170}$$

Therefore, at each step k, the scheme is updated by calculating the resolvent $\mathcal{J}_{\mathbf{M}_k}$ of the maximal monotone operator $\mathbf{M}_k:w\mapsto\|R^{-1}\bar{x}_k\|R(I_n-\tilde{A})R^{-1}w$. The proximal algorithm that is obtained is $\bar{x}_{k+1}=\mathcal{J}_{\mathbf{M}_k}(\bar{x}_k)$, which may be named a *time-varying proximal-point* algorithm.

Another case with homogeneity degree $d=-1$ is treated in [19, Sect. 6.2, eq. (6.4)]. It yields the consistent discretization

$$x_{k+1}\in x_k-h(I_n-\tilde{A})\tilde{F}(x_{k+1}) \tag{171}$$

which is equivalent to

$$x_{k+1}=(I_d+h(I_n-\tilde{A})\tilde{F})^{-1}(x_k) \tag{172}$$

with

$$\tilde{F}(x)=\begin{cases}\dfrac{x}{\|x\|_P} & \text{if } x\neq 0\\ \mathcal{B}_P & \text{if } x=0\end{cases} \tag{173}$$

and $\|x\|_P=\sqrt{x^\top Px}$. Using again the state variable $\bar{x}_k=Rx_k$, one obtains

$$\begin{aligned}\bar{x}_{k+1}&\in\bar{x}_k-hR(I_n-\tilde{A})R^{-1}R\tilde{F}(x_{k+1}),\\&=\bar{x}_k-hR(I_n-\tilde{A})R^{-1}\partial\|\bar{x}_{k+1}\|_2.\end{aligned} \tag{174}$$

For the same reasons given in the first case ($d=1$), it follows that $\Lambda=R(I_n-\tilde{A})R^{-1}\succ 0$. However, $\Lambda\neq\Lambda^\top$ in general. Now, we proceed as in the proof of [19, Thm. 4.1]. For all $\bar{x}_k\neq 0$

$$\begin{aligned}\bar{x}_k^\top\bar{x}_k&=\bar{x}_{k+1}^\top\bar{x}_{k+1}+2h\bar{x}_{k+1}^\top\Lambda\partial\|\bar{x}_{k+1}\|_2\\&\quad+h^2(\Lambda\partial\|z_{k+1}\|_2)^\top(\Lambda\partial\|z_{k+1}\|_2)\end{aligned} \tag{175}$$

$$\begin{aligned}&=\bar{x}_{k+1}^\top\bar{x}_{k+1}+2h\frac{\bar{x}_{k+1}^\top\Lambda\bar{x}_{k+1}}{\|\bar{x}_{k+1}\|_2}\\&\quad+h^2(\Lambda\partial\|\bar{x}_{k+1}\|_2)^\top\Lambda\partial\|\bar{x}_{k+1}\|_2\end{aligned} \tag{176}$$

$$>\bar{x}_{k+1}^\top\bar{x}_{k+1}+h^2(\Lambda\partial\|\bar{x}_{k+1}\|_2)^\top\Lambda\partial\|\bar{x}_{k+1}\|_2 \tag{177}$$

$$=\bar{x}_{k+1}^\top\bar{x}_{k+1}+h^2\frac{\bar{x}_{k+1}^\top\Lambda^\top\Lambda\bar{x}_{k+1}}{\|\bar{x}_{k+1}\|_2^2}. \tag{178}$$

Since $\Lambda^\top\Lambda\succ 0$, we have

$$\inf_{\xi\neq 0}h^2\frac{\xi^\top\Lambda^\top\Lambda\xi}{\|\xi\|_2^2}>0 \tag{179}$$

for all $h>0$, from where the convergence of $\bar{x}_k$ to zero in a finite number of steps is inferred. Thus, in spite of the fact that neither the operator inside (172) nor the inside $\bar{x}_{k+1}=(I_d+h\Lambda\partial\|\cdot\|_2)^{-1}(\bar{x}_k)$ is maximal monotone in general, finite-time convergence of this algorithm is guaranteed [19, Sect. 6.2]. As a side note, note that (174) is equivalent to

$$\bar{x}_{k+1}\in\bar{x}_k-h(\Lambda_{\text{sym}}+\Lambda_{\text{sk}})\partial\|\bar{x}_{k+1}\|_2 \tag{180}$$

with $\Lambda_{\text{sym}}=(1/2)(\Lambda+\Lambda^\top)$ and $\Lambda_{\text{sk}}=(1/2)(\Lambda-\Lambda^\top)$. Therefore, (174) is interpreted as a discrete-time Hamiltonian system

$$\bar{x}_{k+1}\in\bar{x}_k+h(J-R)\partial H(\bar{x}_{k+1}) \tag{181}$$

with $J=-\Lambda_{\text{sk}}$, $R=\Lambda_{\text{sym}}\succ 0$, and Hamiltonian function $H(x)=\|x\|_2$. This is the implicit discretization of the Hamiltonian dynamics

$$\dot{x}(t)\in(J-R)\partial\|x(t)\|_2. \tag{182}$$

Thus, *Hamiltonian proximal-point* algorithms with resolvents $\mathcal{J}_{h\mathbf{M}}=(I_d-h(J-R)\partial\|\cdot\|)^{-1}$, $\mathbf{M}=-(J-R)\partial\|\cdot\|$ may be introduced. More details on the discretization of Hamiltonian systems can be found in [104]. The previous analysis suggests that maximal monotone operators and proximal-point algorithms can be derived from the consistent discretization of homogeneous systems.

FURTHER IMPLICIT ALGORITHMS

Other Classes of Set-Valued Algorithms

The implicit discretization for first-order sliding-mode controllers, supertwisting controllers, observers, differentiators,

and higher-order differentiators has been studied in the previous sections. The implicit discretization of other classes of algorithms has been analyzed in the literature: terminal SMC for first- and second-order systems [42], [105], [106], twisting controllers and differentiators [107], [108], switching between a twisting and a first-order controller [109], proxy-based sliding mode [33], [97], [110], homogeneous differentiators and observers [39], [40], supertwisting velocity observer [32], other types of differentiators [97], [98], [111], [112], [113], [114], [115], the differentiator proposed in [5] discretized in [87], SMC of nonsmooth actuators [37], homogeneous systems and controllers [19], [116], nested controllers for second-order systems with unmatched perturbations [45], first-order controllers applied to Lagrangian nonlinear systems with parameter uncertainties [43] and to linear hyperbolic infinite dimensional systems [117], fixed-time or prescribed-time schemes [19], [105], [118], [119], adaptive sliding-mode algorithms [120], [121], [122], SMC with reaching law [123], fractional-order differentiators [124], [125], multivariable supertwisting [126], and modified supertwisting using minimum-operator-based algorithms [similar to (14b)] [127].

The articles mentioned previously study implicit or semi-implicit schemes. The solvability of the one-step nonsmooth problem (that is, do the generalized equations possess a solution to advance the scheme from step k to step $k+1$?) is usually analyzed. Some also analyze the boundedness/stability properties, and only a few of them analyze the discrete solutions' convergence toward the solutions of the continuous-time closed-loop. The analysis made in the previous sections focuses on the first step, which is necessary for implementation.

Limitations and Modifications of Implicit Algorithms

The fact that implicit and semi-implicit discretization methods supersede the explicit one is widely accepted and validated (see the "Experimental Results and Validations" section). However, open issues have emerged that show that there is room for significant improvement. The implicit method has been studied in [107] for the twisting scheme and in [35] for the supertwisting one. However, the straightforward implicit discretization of the twisting scheme (that is, the implicit or backward-Euler *emulation*) does not guarantee finite-time convergence to the origin [128]; a modification of the emulated implicit method is proposed in [107], which guarantees this property. As noted in [89], the implicit emulation of the supertwisting scheme (129) and (130) does not yield acceptable robustness (a fact also noticed experimentally in [129]; see also [130] for the influence of the plant discretization). Again, a suitable modification of the implicit emulation allows us to enhance it.

In addition, the discretization of implicit emulation of some higher-order differentiators may still undergo numerical chattering, and suitable modifications have to be made to avoid it [131]. All these numerous ad hoc modifications motivate the conception of controllers and observers directly in the discrete-time setting to overcome all the possible issues linked to the emulation of the continuous-time controller. In [132], this approach has been used for the design of supertwisting-like controllers, leading to a closed loop that does not have the limitations of pure backward-Euler emulation, preserving the finite-time and robustness properties of the continuous-time case. The necessary a priori knowledge of the plant parameters may decrease the performance in some cases [133]. Finally, finite-time stability may be lost after an implicit emulation discretization has been performed and may be recovered if a suitable state transformation is applied before implicitly discretizing [19].

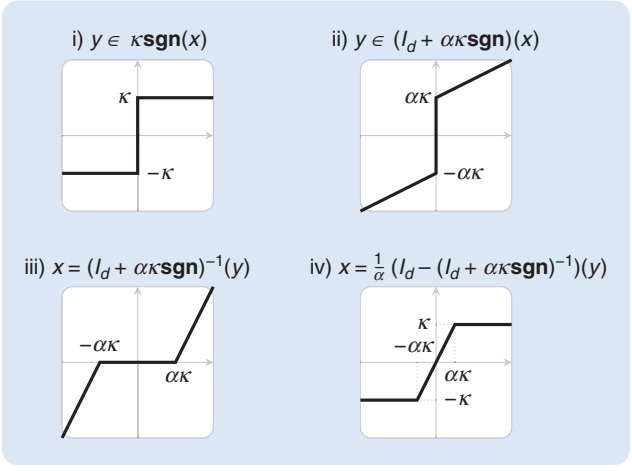

FIGURE 5 The graphical computation of resolvent and Yosida approximation (lower left and right figures, respectively) for the **sgn** multifunction in the scalar case. Note that, at each step, a maximal monotone operator is obtained.

COMPUTATIONAL ISSUES

The analysis of closed-loop systems with implicit discrete-time controllers and the computation of output differentiators rely on computing proximal maps to convex functions, or more generally, resolvents of maximal monotone operators (to solve the associated generalized equations). Figure 5 depicts the steps to follow to compute the resolvent of the sign multifunction. However, as seen in the previous sections, some algorithms yield nontrivial resolvents. In general, finding closed-form expressions for such proximal maps is a hard task, so that the computation of the control law has to rely on numerical methods from convex optimization. Moreover, the numerical method used to calculate the resolvent can significantly influence closed-loop behavior in some applications [41], [86], [93], [134].

Solving Generalized Equations With Proximal Operators

The computation of the variable of interest (control input or differentiator output) is reduced to solving either the generalized equation

$$\begin{aligned}\zeta_{k+1} &= \varphi(x_k) + \rho\eta(x_k),\\ -\eta(x_k) &\in \mathbf{M}(\zeta_{k+1})\end{aligned} \tag{183a}$$

with unknown ζ_{k+1}, or the generalized equation

$$\begin{aligned}-\eta(x_k) &= \frac{1}{\rho}\varphi(x_k) - \frac{1}{\rho}\zeta_{k+1},\\ \zeta_{k+1} &\in \mathbf{M}^{-1}(-\eta(x_k))\end{aligned} \tag{183b}$$

with unknown $\eta(x_k)$, where $\varphi:\mathbb{R}^n\to\mathbb{R}^m$ is a known function of the state, $\rho>0$, and $\mathbf{M}:\mathbb{R}^m\rightrightarrows\mathbb{R}^m$ is maximal monotone.

As an example, note that (73) has this form with $\zeta_{k+1}=\tilde{s}'_{k+1}$, $\eta(x_k)=u_{\mathrm{sv}}(x_k)$

$$\mathbf{M} = \partial\|\cdot\|_1 \circ \tilde{K}(\tilde{B}^\top\tilde{P}^{-1}\tilde{B})^{\frac{1}{2}} \tag{184}$$

and $\varphi(x_k)=\tilde{L}x_k$. Likewise, (97) with $\zeta_{k+1}=\tilde{s}'_{k+1}$, $\eta(x_k)=u_{\mathrm{sv}}(x_k)$, $\mathbf{M}=\partial\|\cdot\|_1\circ\tilde{K}^{\frac{1}{2}}$, and $\varphi(x_k)=\tilde{K}^{-\frac{1}{2}}s_k$ belongs to this class. Equations (126) through (131) also have this form with $\zeta_{k+1}=\tilde{x}_{0,k+1}$, $\eta(x_k)=u_k(x_{0,k},\nu_k)$, $\mathbf{M}$ as in (113), $\varphi(x_k)=x_{0,k}+h\nu_k$. Equations (149)–(150) are an instance of (183) with $\zeta_{k+1}=s_{0,k+1}$, $\mathbf{M}=\mathbf{M}_{\mathrm{sing}}+\mathbf{M}_{\mathrm{set},h}$, $\varphi(x_k)=-b_k$. The unknown ζ_{k+1} for the generalized equation in (183a) corresponds to a "virtual," "nominal," or "unperturbed" sliding variable, which can be used to determine precisely whether or not a sliding mode is occurring in discrete time. The generalized equation in (183b) has the unknown $\eta(x_k)$ and can be interpreted as the dual—or reciprocal—of (183a). From (183a), the *selection* $\eta(x_k)$ is explicitly given as

$$-\eta(x_k) = \frac{1}{\rho}(I_d - \mathcal{J}_{\rho\mathbf{M}})(\varphi(x_k)) = \mathcal{Y}_{\rho\mathbf{M}}(\varphi(x_k)) \tag{185}$$

[see (S24) in "Proximal Mapping and Proximal-Point Algorithm"], whereas (183b) gives

$$-\eta(x_k) = \mathcal{J}_{\frac{1}{\rho}\mathbf{M}^{-1}}\left(\frac{1}{\rho}\varphi(x_k)\right). \tag{186}$$

The expressions (185) and (186) yield the same control law. The choice between them depends on the proximal map to be computed. That is, for a certain $\mathbf{M}$, it could be easier to first compute the inverse map $\mathbf{M}^{-1}$ and then (186), while for other cases, (185) could be a simpler choice. In any case, a convex optimization problem has to be solved at each time step.

For $\mathbf{M}=\partial\|\cdot\|_p$, the calculation of the control inputs can be performed according to (S57) through (S59) in "Convex Analysis Tools" so that

$$-\eta(x_k) = \mathrm{Proj}\left(\mathcal{B}_{p^\star};\frac{1}{\rho}\varphi(x_k)\right). \tag{187}$$

If $p=1$, then $p^\star=\infty$, and according to (S63), the control can be computed componentwise as

$$\eta(x_k)_i = -\,\mathbf{sgn}(\varphi(x_k)_i)\min\left\{\frac{1}{\rho}\mid\varphi(x_k)_i\mid,1\right\},\quad i=1,\ldots,n$$

which is a minimum-operator formalism, whereas if $p=p^\star=2$, then, according to (S62)

$$\eta(x_k) = \begin{cases} -\dfrac{1}{\rho}\varphi(x_k) & \text{if } \|\varphi(x_k)\|\le\rho,\\ -\dfrac{1}{\|\varphi(x_k)\|_2}\varphi(x_k) & \text{otherwise.}\end{cases}$$

For SMC, it is also common to find maps f given as the composition of norm functions with linear maps. In this case, the projectors change accordingly by considering instead a weighted norm. Indeed, if $\mathbf{M}=\partial(\|\cdot\|_p\circ R)$, where R is nonsingular, then it follows from [S28, Prop. 23.23] that

$$\begin{aligned}\mathcal{J}_{\rho\mathbf{M}}(\varphi(x_k)) &= \left(I_d - R^\top\circ\left(RR^\top + (\partial\rho\|\cdot\|_p)^{-1}\right)^{-1}\circ R\right)(\varphi(x_k)),\\ &= (I_d - R^\top\circ(RR^\top + \mathbf{N}_{\rho\mathcal{B}_{p^\star}})^{-1}\circ R)(\varphi(x_k)),\\ &= \varphi(x_k) - R^\top\mathrm{Proj}_{RR^\top}(\rho\mathcal{B}_{p^\star};R^{-\top}\varphi(x_k))\end{aligned} \tag{188}$$

where we have made use of (S57) to get the second equality and (S64c) to get the last equality, both in "Convex Analysis Tools."

In the more general case in which $\mathbf{M}=\partial(g\circ R)$ with g, a proper convex LSC function, we have

$$\mathcal{J}_{\frac{1}{\rho}\mathbf{M}^{-1}}\left(\frac{1}{\rho}\varphi(x_k)\right) = R^\top\theta \tag{189}$$

where

$$\begin{aligned}\theta = \operatorname*{argmin}_{\xi\in\mathbb{R}^n}\{&g^\star(\xi)\\ &+\frac{\rho}{2}\left(\xi-\frac{1}{\rho}R^{-\top}\varphi(x_k)\right)RR^\top\left(\xi-\frac{1}{\rho}R^{-\top}\varphi(x_k)\right)\Big\}.\end{aligned} \tag{190}$$

The following proposition will be useful for computing proximal maps associated with higher-order sliding-mode controllers and differentiators as well as their associated splittings.

Proposition 6

Let $\mathbf{M}:\mathbb{R}\rightrightarrows\mathbb{R}$ be the maximal monotone map

$$\mathbf{M}(x) = \left(\sum_{i=0}^{N} a_i\,|\,x\,|^{\frac{p_i}{q_i}}\right)\mathbf{sgn}(x) + b_0 \tag{191}$$

where $a_i\ge 0$ and p_i/q_i is a nonnegative rational number for all $i\in\{1,\ldots,N\}$ and $p_0=0$. Then, for $\rho>0$, the resolvent $\mathcal{J}_{\rho\mathbf{M}}$ has the form

$$\mathcal{J}_{\rho\mathbf{M}}(x) = \beta_{\mathbf{M}}(x,b_0,\rho)^C\,\mathbf{sgn}(x-\rho b_0) \tag{192}$$

where C is the least common denominator of the set of fractions $\{(p_i/q_i)\mid i=1,\ldots,N\}$, and $\beta_{\mathbf{M}}:\mathbb{R}\times\mathbb{R}\times\mathbb{R}_{++}\to\mathbb{R}_+$ is the unique nonnegative root of the polynomial

$$\beta^C + \rho\sum_{i=1}^{N} a_i\beta^{r_i} - \max\{0,\mid x-\rho b_0\mid -\rho a_0\} \tag{193}$$

with indeterminate β and powers $r_i = C(p_i/q_i)\in\mathbb{Z}_+$.

Proof

From the definition of resolvent in (S17), it follows that $y = \mathcal{J}_{\rho \mathbf{M}}(x)$ if and only if $x \in y + \rho \mathbf{M}(y)$. That is

$$x - \rho b_0 \in \left(|y| + \rho a_0 + \rho \sum_{i=1}^{N} a_i |y|^{\frac{p_i}{q_i}} \right) \mathbf{sgn}(y). \tag{194}$$

Since all the coefficients a_i are nonnegative, we can see from (194) that $y = 0$ if and only if $|x - \rho b_0| \leq \rho a_0$. Hence, if $|x - \rho b_0| > \rho a_0$, then $y \neq 0$, and from (194), we conclude that $\mathbf{sgn}(x - \rho b_0) = \mathbf{sgn}(y)$, so that (194) becomes single-valued as

$$|y| + \rho \sum_{i=1}^{N} a_i |y|^{\frac{p_i}{q_i}} - (|x - \rho b_0| - \rho a_0) = 0. \tag{195}$$

Let $C > 0$ be the least common denominator of the set $\{p_i/q_i \mid i = 1, \ldots N\}$. The change of variables $|y| = \beta^C$, where $\beta > 0$, transforms (195) into a root-finding problem for the polynomial

$$\beta^C + \rho \sum_{i=1}^{N} a_i \beta^{r_i} - (|x - \rho b_0| - \rho a_0) \tag{196}$$

where $r_i = C(p_i/q_i)$. Hence, to solve (195), we look for the positive root of (196). Note that the nonnegativity of all a_i, $i \in \{1, \ldots, N\}$ and Descartes' rule of signs [135, Theorem 2.23] imply the uniqueness of such a positive root of (196) in the cases when $|x - \rho b_0| > \rho a_0$. The polynomial (193) includes both previous cases. Indeed, if $|x - \rho b_0| \leq \rho a_0$, then the unique nonnegative root of (193) is $\beta = 0$, implying $y = 0$. Finally, it follows that

$$\mathcal{J}_{\rho \mathbf{M}}(x) = y = |y| \, \mathbf{sgn}(y) = \beta_{\mathbf{M}}(x, b_0, \rho)^C \mathbf{sgn}(x - \rho b_0). \tag{197}$$

The proof is complete. ■

Instances of Proposition 6 have been independently developed in [87] and [86] in the context of higher-order differentiators, exploiting the fact that (192) is the unique solution of the generalized equation

$$0 \in y + \sum_{i=0}^{N} a_i |y|^{\frac{p_i}{q_i}} \mathbf{sgn}(y) + b_0 - x. \tag{198}$$

Computation of Proximal Maps via Splittings

For complex strategies such as higher-order SMC and differentiation, the associated monotone operator is composed by adding several simpler operators. In this situation, the splitting algorithms in "Proximal Splitting Algorithms" can be used to compute the associated proximal map in an iterative way.

Example 1

The splitting (S92) can be used to compute the URED proximal map in (149) and (150) as follows. Consider the optimization problem (S87) with $A = 1$ and

$$f(\xi) = \frac{2}{3} h \gamma_0 L^{\frac{1}{2}} |\xi|^{\frac{3}{2}} + \frac{1}{2} h^2 \gamma_1 L |\xi| \tag{199a}$$

$$g(\xi) = \frac{1}{2} h^2 \mu^2 \gamma_1 L |\xi|^3 \tag{199b}$$

$$h(\xi) = \frac{2}{5} h \gamma_0 L^{\frac{1}{2}} \mu |\xi|^{\frac{5}{2}} + h^2 \gamma_1 L \mu \xi^2 + \frac{1}{2} (\xi + b_k)^2. \tag{199c}$$

With the given parameters, the solution of (S87) gives the proximal map $\mathcal{J}_{h\mathbf{M}_h}(-b_k)$, where $\mathbf{M}_h$ is as in (150). Also, note that $\nabla h(\cdot)$ is Lipschitz continuous in any bounded set of $\mathbb{R}$ since

$$\frac{\partial^2}{\partial \xi^2} h(\xi) = \frac{3}{2} h \gamma_0 L^{\frac{1}{2}} \mu |\xi|^{\frac{1}{2}} + (1 + 2h^2 \gamma_1 L \mu). \tag{200}$$

Notice that, with such decomposition, each map appearing in (S92) is proximable. Indeed, it follows from Proposition 6 that

$$\mathrm{Prox}_{\alpha_1 f}(x) = \mathcal{J}_{\alpha_1 \partial f}(x) = \beta_{\partial f}(x, \alpha_1)^2 \mathbf{sgn}(x) \tag{201}$$

where

$$\beta_{\partial f}(x, \alpha_1) = \frac{-\alpha_1 h \gamma_0 L^{\frac{1}{2}} + \sqrt{\left(\alpha_1 h \gamma_0 L^{\frac{1}{2}}\right)^2 + 4\rho(x)}}{2} \tag{202}$$

and $\rho(x) = \max\{0, |x| - (\alpha_1 h^2 \gamma_1 L)/2\}$. Now, to compute the proximal map associated with $g^*(\cdot)$ in (S92), we make use of Moreau's decomposition (see, for instance, [S28, Thm. 14.3] in "Maximal Monotone Operators")

$$x = \mathrm{Prox}_{\alpha_2 g^*}(x) + \alpha_2 \mathrm{Prox}_{\frac{1}{\alpha_2} g}\left(\frac{x}{\alpha_2}\right). \tag{203}$$

Let us compute $\mathrm{Prox}_{\frac{1}{\alpha_2} g}(x/\alpha_2)$ using once again Proposition 6, so that

$$\mathrm{Prox}_{\frac{1}{\alpha_2} g}\left(\frac{x}{\alpha_2}\right) = \beta_{\partial g}\left(\frac{x}{\alpha_2}, \frac{1}{\alpha_2}\right) \mathbf{sgn}(x) \tag{204}$$

and

$$\beta_{\partial g}(x, \rho) = \frac{-1 + \sqrt{1 + 6h^2 \mu^2 \gamma_1 L \rho |x|}}{3h^2 \mu^2 \gamma_1 L \rho}. \tag{205}$$

Finally, it follows from (203) and (204) that

$$\begin{aligned} \mathrm{Prox}_{\alpha_2 g^*}(x) &= x - \alpha_2 \mathrm{Prox}_{\frac{1}{\alpha_2} g}\left(\frac{x}{\alpha_2}\right), \\ &= x - \alpha_2 \beta_{\partial g}(x, \rho) \mathbf{sgn}(x). \end{aligned} \tag{206}$$

Therefore, by using the expressions (201) and (206) in the iteration (S92) and by selecting the parameters α_1 and α_2 sufficiently small such that (S93) holds, we have that $\xi_{j+1} \to \mathcal{J}_{h\mathbf{M}_h}(-b_k)$ as $j \uparrow +\infty$.

It is worth mentioning that, for problems with more than two maximal monotone terms, parallel algorithms, such as those described in [137] and [S35], can be used.

Proximal Splitting Algorithms

Consider the following problem. Given two maximal monotone operators $\mathbf{M}_1, \mathbf{M}_2 : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, find $\xi \in \mathbb{R}^n$ such that

$$0 \in \mathbf{M}_1(\xi) + \mathbf{M}_2(\xi). \tag{S84}$$

Under constraint-qualification assumptions guaranteeing that $\mathbf{M}_1 + \mathbf{M}_2$ is maximal monotone, it is inferred from (S17) in "Maximal Monotone Operators" that problem (S84) is equivalent to finding $\xi \in \mathbb{R}^n$ such that

$$\xi = \mathcal{J}_{\rho(\mathbf{M}_1+\mathbf{M}_2)}(\xi) \tag{S85}$$

for any $\rho > 0$. The formulation (S85) allows for the search of a solution of (S84) in an iterative way. Indeed, from the contents of "Proximal Mapping and Proximal-Point Algorithm," it is deduced that the fixed-point iteration

$$\xi_{k+1} = \mathcal{J}_{\rho(\mathbf{M}_1+\mathbf{M}_2)}(\xi_k) \tag{S86}$$

converges toward a solution of (S85). However, an explicit expression for the resolvent map of the sum $\mathbf{M}_1 + \mathbf{M}_2$ in (S86) may not be available. To solve such an issue, proximal splitting schemes are applied to compute alternative iterations that converge to a solution of (S84) and that involve only the individual resolvents of $\mathbf{M}_1$ and $\mathbf{M}_2$ at each iteration. In what follows, two popular splitting schemes are discussed: the Condat-Vũ and the Douglas-Rachford splittings.

CONDAT-VŨ SPLITTING

The Condat-Vũ splitting was independently proposed in [S45] and [S46] for solving the optimization problem

$$\min_{\xi \in \mathbb{R}^n} f(\xi) + g(A\xi) + h(\xi) \tag{S87}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^m \to \mathbb{R}$, and $h : \mathbb{R}^n \to \mathbb{R}$ are proper convex LSC functions. The function h is continuously differentiable, and $A \in \mathbb{R}^{m \times n}$. Recalling that the proximal map is itself the solution of a convex optimization problem [see (S25) and (S27) in "Proximal Mapping and Proximal-Point Algorithm"], the Condat-Vũ splitting can be used for approximating the values of elaborated proximal maps, such as those appearing in higher-order sliding-mode controllers, observers, and differentiators [see, for instance, (129) and (149)]. The splitting relies on the fact that $g(A\xi) = \sup_{\mu \in \mathbb{R}^m}\{\langle A\xi, \mu \rangle - g^*(\mu)\}$ (Fenchel-Moreau theorem [S28, Thm. 13.32]), so that, as pointed out in [S45], the primal problem (S87) can be rewritten as

$$\min_{\xi \in \mathbb{R}^n} \sup_{\mu \in \mathbb{R}^m} \mathcal{L}(\xi, \mu) \tag{S88}$$

where $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is of the form

$$\mathcal{L}(\xi, \mu) = f(\xi) + h(\xi) + \langle \mu, A\xi \rangle - g^*(\mu). \tag{S89}$$

Thus, the saddle points of $\mathcal{L}$ are characterized by the following generalized equations:

$$0 \in \nabla h(\xi) + A^\top \mu + \partial f(\xi) \tag{S90a}$$

$$0 \in -A\xi + \partial g^*(\mu). \tag{S90b}$$

Proceeding in a similar way as mentioned previously, (S90) is equivalent to the following fixed-point conditions:

$$\xi = \mathrm{Prox}_{\alpha_1 f}(\xi - \alpha_1(\nabla h(\xi) + A^\top \mu)) \tag{S91a}$$

$$\mu = \mathrm{Prox}_{\alpha_2 g^*}(\mu + \alpha_2 A\xi) \tag{S91b}$$

for any $\alpha_1, \alpha_2 > 0$. It is well known that if $\xi^\star$ and $\mu^\star$ are solutions of the primal problem (S87) and its associated dual problem, respectively, then they also satisfy (S91); see, for instance, [S28, Prop. 19.18]. The converse is true if, in addition, a constraint qualification such as $0 \in \mathrm{rint}(A \,\mathrm{dom}\, f - \mathrm{dom}\, g)$ holds. Under such a total duality assumption, the following iteration was proposed in [S45] and [S46]:

$$\xi_{j+1} = \mathrm{Prox}_{\alpha_1 f}(\xi_j - \alpha_1(\nabla h(\xi_j) + A^\top \mu_j)) \tag{S92a}$$

$$\mu_{j+1} = \mathrm{Prox}_{\alpha_2 g^*}(\mu_j + \alpha_2 A(2\xi_{j+1} - \xi_j)). \tag{S92b}$$

Thus, if the gradient of h is Lipschitz continuous with constant ℓ and

$$\frac{\alpha_1 \ell}{2} + \alpha_1 \alpha_2 \lambda_{\max}(A^\top A) < 1 \tag{S93}$$

then the iteration (S92) converges toward solutions of (S91), implying that the sequence $\{\xi_k\}_{k \in \mathbb{N}}$ converges to a solution of (S87).

DOUGLAS-RACHFORD SPLITTING

Similar to the previous approach, Douglas-Rachford splitting can be used to approximate proximal maps composed of the sum of two maximal monotone operators. We start considering the problem of finding $\xi \in \mathbb{R}^n$ such that

$$0 \in \partial f(\xi) + \partial g(\xi) \tag{S94}$$

where, as before, f and g are proper convex LSC functions. Thus, as pointed out in [S35, p. 44], the following equivalence is true:

$$\begin{aligned} 0 \in \partial f(\xi) + \partial g(\xi) \\ \Leftrightarrow z = (\mathcal{R}_{\alpha\partial f} \circ \mathcal{R}_{\alpha\partial g})(z), \text{and } \xi = \mathrm{Prox}_{\alpha g}(z) \end{aligned} \tag{S95}$$

where $\mathcal{R}_{\alpha\partial f} = 2\mathrm{Prox}_{\alpha f} - I_d$ is the reflected resolvent. Now, since the operator $\mathcal{R}_{\alpha\partial f} \circ \mathcal{R}_{\alpha\partial g}$ is simply nonexpansive, a Picard iteration using (S95) may not converge. To overcome such an issue, the Douglas-Rachford splitting uses the following Krasnosel'skiĭ-Mann iteration

$$\begin{aligned} z_{j+1} &= \frac{1}{2} z_j + \frac{1}{2}(\mathcal{R}_{\alpha\partial f} \circ \mathcal{R}_{\alpha\partial g})(z_j), \\ &= \mathrm{Prox}_{\alpha f} \circ (2\mathrm{Prox}_{\alpha g} - I_d)(z_j) + (I_d - \mathrm{Prox}_{\alpha g})(z_j) \end{aligned} \tag{S96}$$

(*Continued*)

Proximal Splitting Algorithms (*Continued*)

the convergence of which is guaranteed for any $\alpha > 0$ [136, Theorem 1]. Thus, the iteration

$$z_{j+1} = \mathrm{Prox}_{\alpha f} \circ (2\mathrm{Prox}_{\alpha g} - I_d)(z_j) + (I_d - \mathrm{Prox}_{\alpha g})(z_j) \quad \text{(S97a)}$$

$$\xi_{j+1} = \mathrm{Prox}_{\alpha g}(z_j) \quad \text{(S97b)}$$

converges, and ξ_{j+1} approaches a solution of (S94) as $j \uparrow +\infty$.

REFERENCES

[S45] L. Condat, "A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, 2013, doi: 10.1007/s10957-012-0245-9.

[S46] C. B. Vu, "A splitting algorithm for dual monotone inclusions involving cocoercive operators," *Adv. Comput. Math.*, vol. 38, pp. 667–681, Nov. 2013.

Example 2

The Douglas-Rachford splitting can also be used to compute the proximal map in (149) and (150). This time, a splitting of the following form is considered:

$$f(\xi) = \frac{2}{5}h\gamma_0 L^{\frac{1}{2}}\mu|\xi|^{\frac{5}{2}} + \frac{2}{3}h\gamma_0 L^{\frac{1}{2}}|\xi|^{\frac{3}{2}} + \frac{1}{2}h^2\gamma_1 L|\xi| \quad (207)$$

$$g(\xi) = \frac{1}{2}h^2\mu^2\gamma_1 L|\xi|^3 + h^2\gamma_1 L\mu\xi^2 + \frac{1}{2}(\xi + b_k)^2. \quad (208)$$

From Proposition 6, the expressions

$$\mathrm{Prox}_{\alpha f}(x) = \beta_{\partial f}(x,\alpha)^2 \mathbf{sgn}(x) \quad (209)$$

$$\mathrm{Prox}_{\alpha g}(x) = \beta_{\partial g}(x, b_k, \alpha)\mathbf{sgn}(x - \alpha b_k) \quad (210)$$

are obtained, where $\beta_{\partial f}(x,\alpha)$ is the unique positive root of the cubic polynomial

$$\alpha h\lambda_0 L^{\frac{1}{2}}\mu\beta^3 + \beta^2 + \alpha\lambda_0 L^{\frac{1}{2}}\beta - \max\left\{0, |x| - \frac{\alpha h^2\lambda_1 L}{2}\right\} \quad (211)$$

that is

$$\beta_{\partial g}(x, b_k, \alpha) = \frac{-2(1 + h^2\mu\lambda_1 L)}{3h^2\mu^2\lambda_1 L} + \frac{\sqrt{4(1 + h^2\mu\lambda_1 L)^2 + 6h^2\mu^2\lambda_1 L|x - b_k|}}{3h^2\mu^2\lambda_1 L}. \quad (212)$$

Hence, by taking the Douglas-Rachford iteration (S97) with $\mathrm{Prox}_{\alpha f}$ and $\mathrm{Prox}_{\alpha g}$ as in (209) through (212), we have that ζ_j converges to the resolvent in (149) and (150) as $j \uparrow +\infty$.

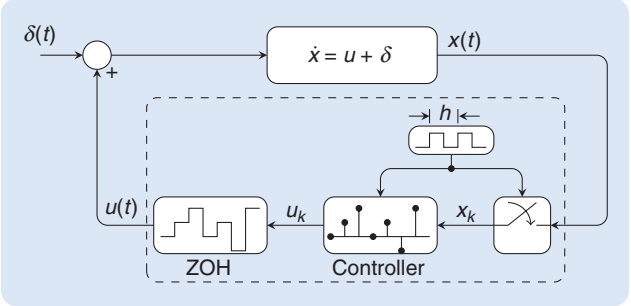

FIGURE 6 A closed-loop system. The state of the continuous-time dynamics is sampled every h seconds. Such information is used for computing the control input u_k, which passes through a classical zero-order-hold mechanism to obtain the feedback input signal $u(t) = u_k$ for $kh < t \le (k+1)h$.

Relaxation

One of the main objectives of SMC concerns the finite-time stability of the sliding variable, whose discrete-time dynamics is of the form

$$s_{k+1} = s_k + \rho\, u_{\mathrm{sv}}(x_k) + \delta_k. \quad (213)$$

Notice that, as the disturbance δ_k is assumed unknown, the finite-time stability of s_k cannot be achieved, so that a new variable $\tilde{s}_k$ is introduced. The form of such "virtual" sliding variable is largely open. One option consists of considering a copy of the nominal behavior as in (183). One way of accelerating the convergence of $\tilde{s}_k$ toward the origin consists of setting the control law as

$$-u_{\mathrm{sv}}(x_k) \in \partial f(\tilde{s}_{k+1}) \quad (214)$$

$$\tilde{s}_{k+1} = s_k + \mu\, u_{\mathrm{sv}}(x_k) \quad (215)$$

where $\mu > 0$, so that the closed-loop is given as

$$\tilde{s}_{k+1} = \mathrm{Prox}_{\mu f}(s_k) \quad (216a)$$

$$s_{k+1} = s_k + \frac{\rho}{\mu}(\tilde{s}_{k+1} - s_k) + \delta_k. \quad (216b)$$

The iteration (216) is known as a relaxed iteration in the literature of proximal-point algorithms [138]. In the cases where $\delta_k \equiv 0$, it has been shown that (216) converges toward a minimum of $f(\cdot)$ [S28, Corollary 5.16]. Moreover, for certain values of $(\rho/\mu) \in (0,2)$, the algorithm (216) exhibits a faster convergence compared to the case $\rho = \mu$; see, for instance, [138].

Numerical Experiments

Taking advantage of the fact that the supertwisting controller can be calculated in the closed form (130), it is used as a benchmark to test algorithms that approximate implicit or semi-implicit discrete-time inputs. It appears that there are several ways to implement a given controller or differentiator, and it is certainly crucial to launch a research effort to tackle such issues. Consider a perturbed integrator in feedback with a discrete-time controller, as shown in Figure 6. The target amounts to driving the state $x_0(t)$ toward the origin in the presence of the persistent disturbance $\delta(t) = \cos(\pi t)\sin(5\sqrt{t/3})$. To this end, consider the

supertwisting algorithm, taking into account the two main approaches for its discrete-time implementation, that is, emulation of the continuous-time controller and discrete-time design.

For the discretization using the emulation part, three options are considered: 1) the explicit (forward) Euler discretization, 2) the implicit (backward) Euler discretization in [35], and 3) the semi-implicit discretization in [36]. For the first case, the control law takes the form

$$\begin{aligned} u_k &= -\gamma_0 L^{\frac{1}{2}} |x_{0,k}|^{\frac{1}{2}} \mathbf{sgn}(x_{0,k}) + \nu_k, \\ \nu_{k+1} &= \nu_k - h\gamma_1 L \mathbf{sgn}(x_{0,k}). \end{aligned} \tag{217}$$

For the second case, the control law takes the form

$$\begin{aligned} u_k &= -\gamma_0 L^{\frac{1}{2}} \beta_k \mathbf{sgn}(x_{0,k} + h\nu_k) + \nu_k, \\ &\quad - h\gamma_1 L \mathrm{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h^2\gamma_1 L}\right), \\ \beta_k &= \frac{-h\gamma_0 L^{\frac{1}{2}}}{2} + \sqrt{\frac{h^2\gamma_0^2 L}{4} + \max\{0, |x_{0,k} + h\nu_k| - h^2\gamma_1 L\}}, \\ \nu_{k+1} &= \nu_k - h\gamma_1 L \mathrm{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h^2\gamma_1 L}\right). \end{aligned} \tag{218}$$

For the third case, it takes the form

$$\begin{aligned} u_k &= \nu_k - \mu_k \mathrm{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h\mu_k}\right), \\ \nu_{k+1} &= \nu_k - h\gamma_1 \mathrm{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h\mu_k}\right) \\ \mu_k &= \gamma_0 L^{\frac{1}{2}} |x_{0,k}|^{\frac{1}{2}} + h\gamma_1 L. \end{aligned} \tag{219}$$

For the direct design in discrete time, we consider the following controller [89]:

$$\begin{aligned} u_k &= -\gamma_0 \beta_k \mathbf{sgn}(x_{1,k}) - 2h\gamma_1 \mathrm{Proj}\left([-1,1]; \frac{x_{1,k}}{h^2\gamma_1}\right) + \nu_k, \\ \beta_k &= -\frac{h\gamma_0}{2} + \frac{1}{2}\sqrt{(h\gamma_0)^2 + 4\max\{0, |x_{1,k}| - h^2\gamma_1\}}, \\ \nu_{k+1} &= \nu_k - h\gamma_1 \mathrm{Proj}\left([-1,1]; \frac{x_{1,k}}{h^2\gamma_1}\right). \end{aligned} \tag{220}$$

The controller (220) is in explicit form, though strongly motivated by the implicit controller obtained from the backward-Euler discretization (218). In addition, the splitting schemes (S92) and (S97), applied to the implicit supertwisting algorithm in (112), are also considered. That is, each splitting scheme is tailored to solve the optimization problem

$$\min_{\xi \in R} \frac{2}{3} h\gamma_0 L^{\frac{1}{2}} |\xi|^{\frac{3}{2}} + h^2\gamma_1 L|\xi| + \frac{1}{2}(\xi - (x_{0,k} + h\nu_k))^2. \tag{221}$$

As a last scheme to compare, we solve the problem (221) numerically at each time step by using the interior point solver *Clarabel* [139]. Listings 1 and 2 show the code implementation in Python 3 used for the Douglas-Rachford and Condat-Vũ splittings, respectively.

Listing 1: Douglas-Rachford Splitting for the Supertwisting Controller (218)

```
import numpy as np
#====================================
# Definition of bracket function
#====================================
def bracket(x, a):
    return np.sign(x)*np.abs(x)**a
#====================================
# Resolvent of operator A_0(x) =
|x|^{0.5}sgn(x)
#====================================
def J_A0(alpha, x, h_c, g0):
  beta_A = (-alpha*h_c*g0 +
    np.sqrt((alpha*h_c*g0)**2 +
    4.0*np.abs(x)))/2.0
  return (beta_A**2)*np.sign(x)
#====================================
#Resolvent of operator A_1(x) = g1*sgn(x)
#====================================
def J_A1(x, h_c, g1):
  return 0 if np.abs(x) <=(h_c**2)*g1 else
      v-(h_c**2)*g1*np.sign(x)
#====================================
#Douglas-Rachford splitting
#====================================
def sta_dgl_rfd(h_c, g0, g1, x_0_k, nu,
    x_0_tilde, w, N):
    #================================
    #Initialization of variables
    #================================
r = x_0_k + h_c*nu[-1]
z = w[-1]
#====================================
#Main iteration loop
#====================================
for i in range(N):
    R_A1 = 2.0*J_A1(z, h_c, g1) - z
    z = 0.5*(z - R_A1) + J_A0(0.5, 0.5*(r +
        R_A1), h_c, g0)
#====================================
#Computation of the next state x̃_{k+1}
#====================================
x_0_next_tilde = J_A1(z, h_c, g1)
#====================================
# Computation of the control input
#====================================
u_k = -(x_0_k - x_0_next_tilde)/h_c
nu_next = u_k + g0*bracket(x_0_next_tilde,
    0.5)
#====================================
#Storage of variables
#====================================
x_0_tilde.append(x_0_next_tilde)
nu.append(nu_next)
w.append(z)
return u_k
```

For simulations, the initial condition $x_0(0) = 5$ is taken, and several discretization steps $100\ \mu s \leq h \leq 1\ s$ are considered. For each simulation, a final time of $t_f = 20\ s$ is chosen.

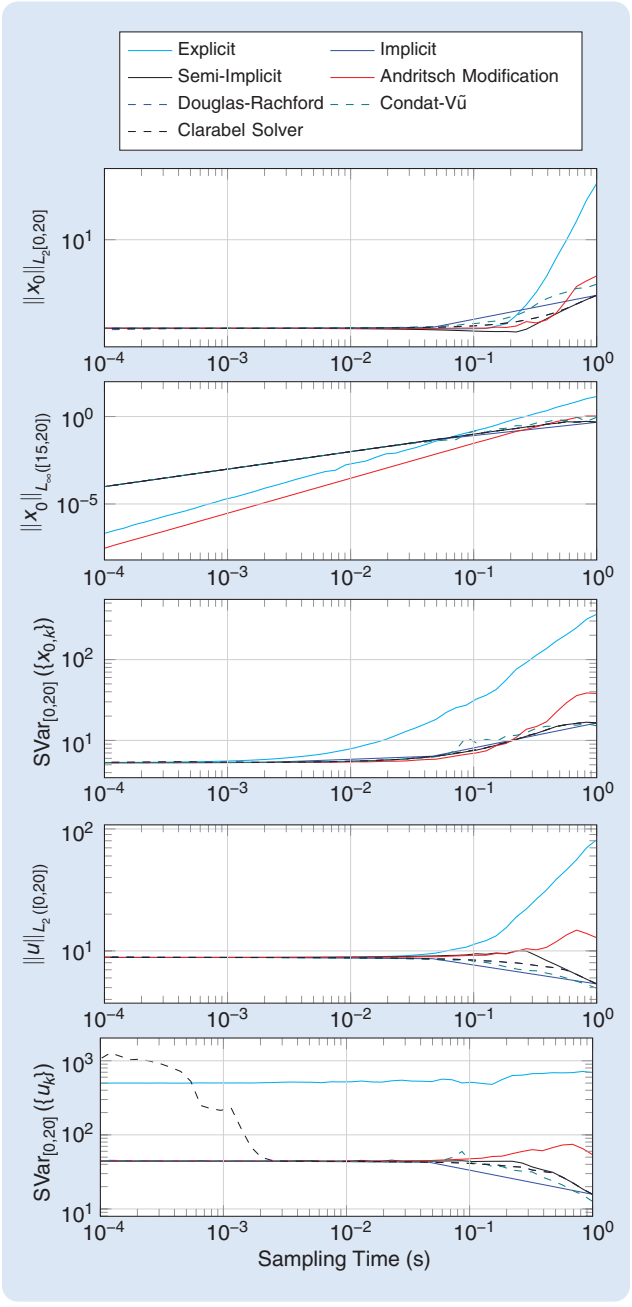

FIGURE 7 Norms and total variations of state and control signals for several discrete-time supertwisting controllers. In terms of the L_2-norm of the state x_0 and the control input u, the implicit controller (218) (and its associated splittings), the semi-implicit controller (219), and the numerical solver for (221) have a similar behavior for all sampling times. In terms of the step-to-step variation (222) of the control-input sequence, the explicit controller shows the larger values. It is noteworthy that for sampling times $h < 2$ ms, the numerical solver presents an increment in the variation of the controller u, indicating the presence of numerical chattering. In terms of the $\|x_0\|_{L_\infty([15,20])}$, the controller (220) shows the best performance for $h \leq 200$ ms.

Figure 7 displays the L_2-norm, the L_∞-norm, and the step-to-step variation SVar of the state and control signals for different sampling times. The latter is defined for a finite sequence $\{\xi_k\}$ of N samples as

$$\text{SVar}(\{\xi_k\}) = \sum_{k=0}^{N-2} |\xi_{k+1} - \xi_k|. \tag{222}$$

For large sampling times, $0.5\ s < h < 1\ s$, all controllers lead to similar values on the L_2-norm of the state x_0. Notice that, in terms of the L_2-norm of the control input and the step-to-step variations, the controllers (220) and (217) are less performant. Indeed, for the explicit controller (217), $\text{SVar}_{[0,20]}(\{u_k\}) \approx 500$ units, regardless of the sampling time. For smaller sampling times, that is, $2\ \text{ms} < h < 20\ \text{ms}$, all controllers perform in a similar way in terms of the L_2-norm of the state and the input, even though the step-to-step variation of the state with the controller (217) is also higher than in the other cases. Finally, when $h < 2$ ms, all the controllers show the same performance, except for the numerical approach, which shows a sudden increment in the step-to-step variation of the input u [that at some point even surpasses that of the explicit controller (217)]. Such an increment in the step-to-step variation, while maintaining the L_2-norm unchanged, is a sign of the presence of chattering in the input.

EXPERIMENTAL RESULTS AND VALIDATIONS

A Quick Review

Implicit and semi-implicit control and differentiation algorithms have been experimentally tested, mostly on laboratory setups. It is noteworthy that the fundamental properties, like insensitivity with respect to the gain during sliding phases [stemming from (79), (100), and (138)], moderate deterioration of the performance with increasing sampling times, and significant chattering alleviation at both outputs and inputs, are validated experimentally; see, for instance, [21], [34], [140], and [141]. It is also often observed that explicit and implicit algorithms behave similarly for very small sampling times, though no clear explanation of this fact has been given yet. The classical first-order sliding-mode controller [34], [140], [141], the twisting controller [140], and several higher-order and homogeneous differentiators have been implemented and compared using an electropneumatic system [39], [93], [134]. Homogeneous differentiators have been tested on an RLC circuit [142] and on cable-driven parallel robots [143] to estimate velocities and accelerations; differentiators have been implemented on dc-dc buck converters in [144] and [145]. Various discretizations of the classical first-order SMC [21], a switching input upgrading linear control [146], and several higher-order differentiators [93], [134], [147] have been tested and compared in closed loops on the inverted

pendulum and the rotary inverted pendulum (including cascaded first-order differentiators). Set-valued admittance controllers [114], [148], [149], [150], [151] and first-order differentiators [97] have been tested on robotic systems. The supertwisting observer and sliding mode observers have been validated in silico for pancreas and diabetes control, and glucose regulation with insulin infusion, in [152], [153], and [154].

A parabolic differentiator is implemented on a dc servomotor equipped with an optical encoder in [113], and homogeneous differentiators are used to estimate output derivatives for the control of an airfoil section [155]. A set-valued antisway controller is applied to an overhead crane in [156]. A supertwisting observer is applied to a magneto-rheological clutch in [157] and [158], while a supertwisting controller semi-implicitly discretized is applied to an electromechanical system with time delay in [159] and on a linear motor speed control with saturation in [160] and [161]. The backward-Euler discretizations of the supertwisting (with suitable modification of the emulation version) and of first-order SMC are applied on the speed control of dc motors in [162] and extensively compared to forward-Euler schemes. Various controllers (supertwisting, first order, etc.) are applied on a floating-wind turbine in [163], [164], and [165] and on a magnetic resonance image patient-positioning system in [166], with extensive comparisons with other types of controllers. The supertwisting controller is applied to functional electrical stimulation of the index finger's metacarpophalangeal joint in [167]. A first-order sliding-mode set-valued input is designed for the stabilization of a nonsmooth model of a hydraulic actuator in [37]. It is applied to an industrial commercial excavator [168], [169], [170] and to a force controller for hydraulic actuators [171]. Sliding-mode observers are tested on a surface permanent magnet synchronous motor in [133] and extensively compared to conventional observers (robustness, chattering, execution time, etc.).

Toolboxes

A toolbox for automatic gain-tuning of higher-order differentiators implemented with the backward-Euler method, as introduced in the "Higher-Order Differentiators" section, is proposed in [87] and [172]. See also [173] for a toolbox dedicated to differentiators with the forward-Euler discretization and [174] for the implicit discretization of a Levant differentiator.

Typical Experimental Results

We present a few experimental results that have been obtained in [21], [34], [128], [140], [141], [175], and [176] in two setups: an electropneumatic system and an inverted pendulum. Two controllers are tested: the classical first-order SMC and the twisting algorithm. They illustrate some important properties of backward-Euler schemes. Our aim

Listing 2: Condat-Vũ Splitting for the Supertwisting Controller (218)

```
import numpy as np
#====================================
# Definition of bracket function
#====================================
def bracket(x, a):
    return np.sign(x)*np.abs(x)**a
#====================================
# Resolvent of operator B_0(x) =
|x|^{0.5}sgn(x)
#====================================
def J_B0(x, h_c, g0):
    beta = (-h_c*g0 + np.sqrt((h_c*g0)**2 +
        4.0*np.abs(x)))/2.0
    return (beta**2)*np.sign(x)
#====================================
# Resolvent of operator B_1(x) =
(a sgn(x))^{*}
#====================================
def J_B1(x, a):
    return x if np.abs(x) <=a else a*np.
    sign(x)
#====================================
#Condat-Vu splitting
#====================================
def sta_cdt_vu(h_c, g0, g1, x_0_k, nu,
     x_0_tilde, mu, N, a1, a2):
#====================================
# Initialization of variables
#====================================
z = x_0_tilde[-1]
m = mu[-1]
#====================================
#Main iteration
#====================================
for i in range(N):
    z_next = J_B0((1 - a1)*z + a1*(x_0_k +
        h_c*nu[-1] - m, h_c, g0)
    m = J_B1(m + a2*(2.0*z_next - z),
        g1*h_c**2)
    z = z_next
#====================================
#Update of virtual state variable
#====================================
x_0_next_tilde = z
#====================================
# Computation of control law
#====================================
u_k = -(x_0_k - x_0_next_tilde)/h_c
nu_next = u_k +
    g0*bracket(x_0_next_tilde,0.5)
#====================================
#Storage of variables
#====================================
x_0_tilde.append(x_0_next_tilde)
nu.append(nu_next)
mu.append(m)
return u_k
```

is to illustrate the main specific features without presenting details that are available in the cited articles.

The first property is digital chattering alleviation and magnitude decrease in the input (see Figure 8 for first-order SMC applied to the electropneumatic setup [34] and [177] for a video showing the setup). The explicit controller [Figure 8(a)] switches between the two extremes of the sign function at each sampling step; this is a high-frequency bang-bang signal. In contrast, the values of the implicit controller are selected from the graph of the sign-set-valued map at each sampling step. Consequently, when the sliding surface has been reached, the amplitude of the explicit controller depends on the feedback gain, whereas the amplitude of the implicit controller is independent of such gain, and it only depends on the disturbance [in this case, a square force signal; see Figure 8(b)]; we are therefore exactly in a situation where (100) [or (79) for the passivity-based first-order SMC] holds.

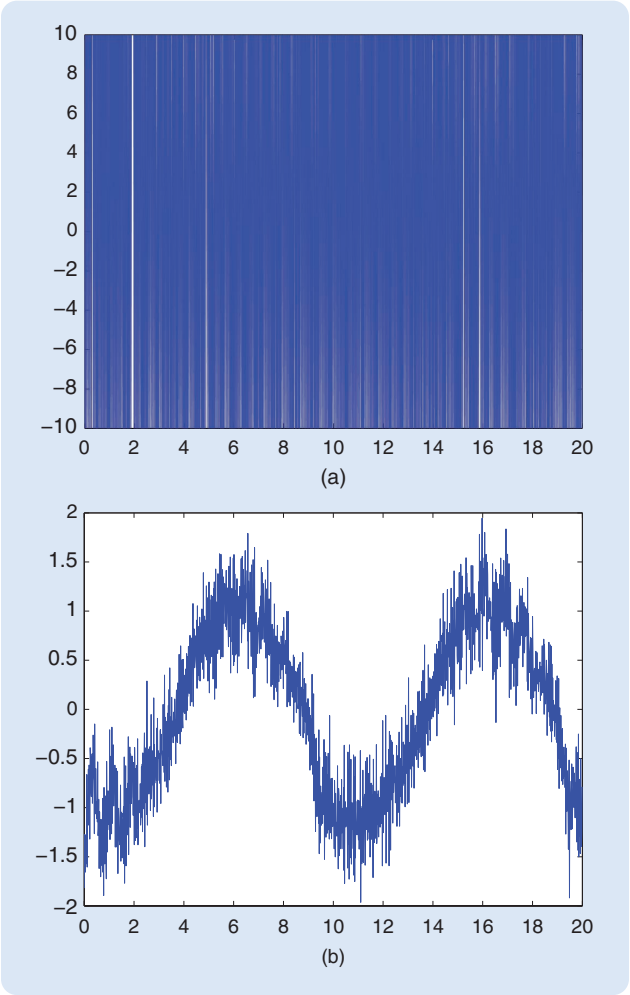

FIGURE 8 Typical control inputs for (a) explicit and (b) implicit first-order SMC with the same gain and sampling period [175, Figures 4 and 5]; see also [34] and [176]. The horizontal axis corresponds to time (s). See [177] for further illustration of chattering alleviation.

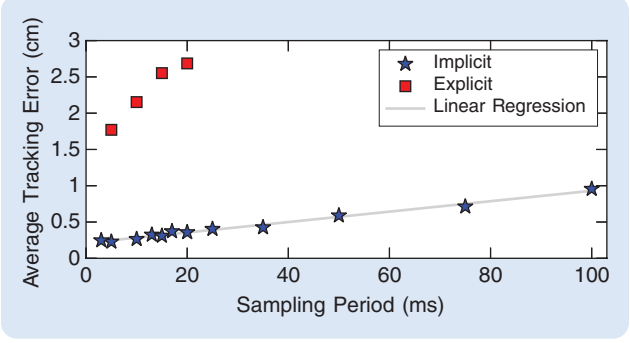

FIGURE 9 Twisting controller: the evolution of the average tracking error with respect to the sampling time for both implicit and explicit discretizations [128, Figure 4.3].

Figures 9 and 10 depict experiments on the same electropneumatic system, using the twisting algorithm for a trajectory tracking task [107], [128], [140]. In Figure 9, it is observed that the performance of the explicit implementation deteriorates as h increases, whereas the implicit one maintains a very good performance level. In Figure 10, the gain insensitivity property [as a consequence of (24) and (100)] is shown for three different sampling times h and a large range of control gains [equivalent to K in (22)]. Very similar results on gain insensitivity are obtained in [21] (first-order SMC and inverted pendulum) and [34] (first-order SMC and electropneumatic system). Figures 11 and 12 show the evolution of the system's output for two different sampling times (first-order SMC and electropneumatic system [34]). Explicit and implicit controllers produce similar outputs for very small h [Figures 11(a) and 12(a)]. However, it is striking to notice that although the explicit method displays large chattering for $h = 15$ ms, the implicit one remains with very low chattering for both $h = 2$ ms and $h = 15$ ms. This confirms the results depicted in Figure 9. This also demonstrates that both methods yield similar outputs for very small h; the inputs are, however, still quite different from each other (they always look like the signals in Figure 8). Quite similar results are obtained with a first-order SMC applied to a six-component thrust generator installed on the head of a floating-wind turbine [164, Figs. 7–10], where codimension $n \geq 2$ sliding surfaces are considered.

CONCLUSIONS AND PERSPECTIVES

The implicit (or backward-Euler) discretization of set-valued sliding-mode systems has received significant attention during the last decade since it was shown in [32] that it applies to systems with unknown matched disturbances and is an efficient way to significantly alleviate the digital chattering at both the output and the input. This article aims to show the close link between implicit/semi-implicit discretization of set-valued SMC and differentiation, and optimization. In particular, proximal-point algorithms using the resolvents of maximal monotone operators are shown to be at the core of the

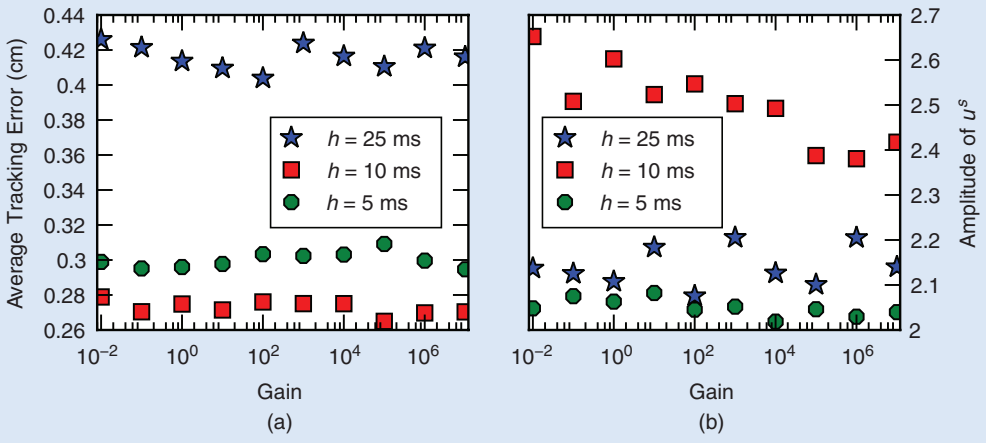

FIGURE 10 (a) and (b) Twisting controller: the evolution of the average tracking error and the implicit control magnitude when the gain varies, for three sampling periods [128, Figure 4.8] (gain $= 10^5$ in Figure 9).

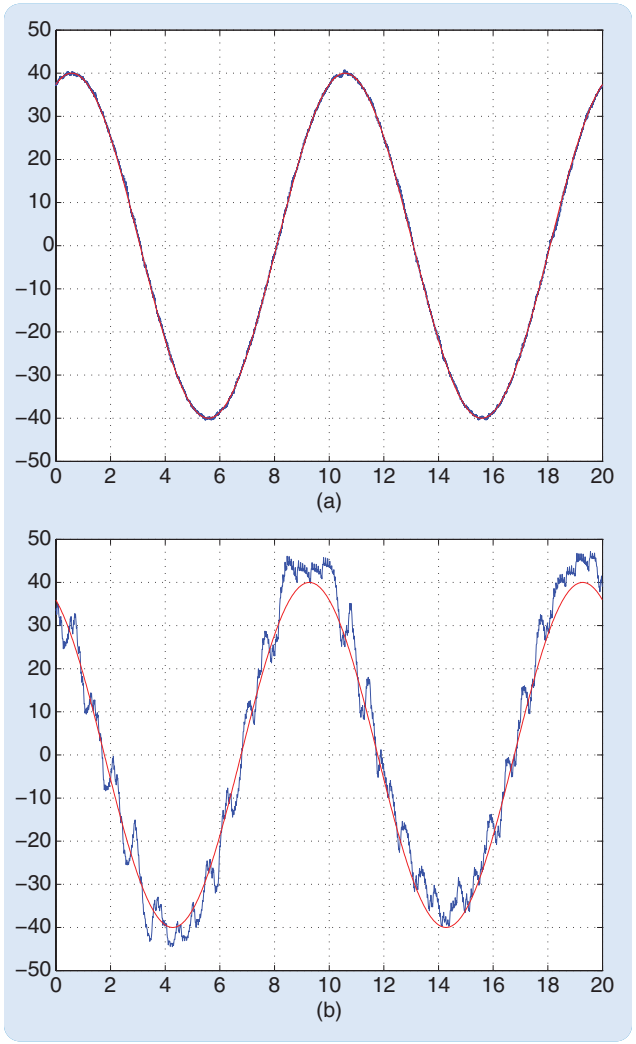

FIGURE 11 The explicit method. (a) $h = 2$ ms and (b) $h = 15$ ms: constant control gains [175, Figures 2 and 3] and [34]. The signals y and y_{ref} are the position output and the desired output of the electropneumatic setup. The horizontal axis corresponds to time (s).

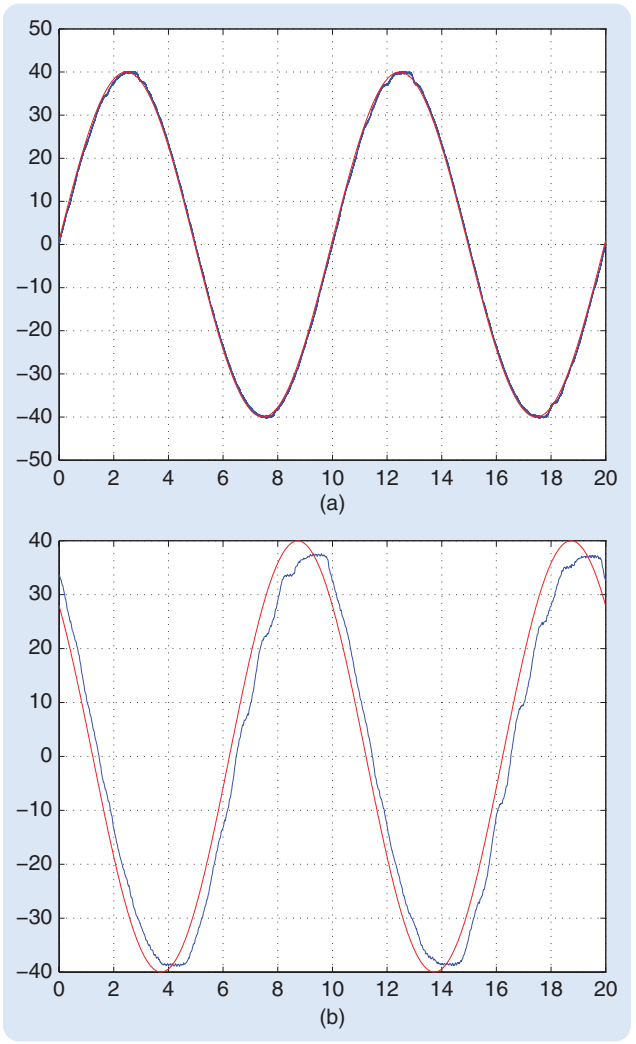

FIGURE 12 The implicit method. (a) $h = 2$ ms and (b) $h = 15$ ms: constant control gains [175, Figures 2 and 3] and [34] and [141]. The signals y and y_{ref} are the position output and the desired output of the electropneumatic setup. The horizontal axis corresponds to time (s).

analysis and of the implementation of implicit methods. The crucial role played by passivity in the design of implicit discretization of first-order sliding-mode algorithms is highlighted. Robust, time-varying, Hamiltonian, and higher-order versions of the proximal-point algorithm are introduced. Computational issues, which are crucial for real-time implementations, are analyzed. Splitting and relaxation algorithms, which provide systematic ways to solve the generalized equations associated with the one-step problems, are reviewed. Some numerical experiments illustrate the presentation.

It is our belief that the analysis of this class of algorithms is still in its infancy, but existing results and experiments show its great potential. The implicit discretizations of infinite-dimensional systems (maximal monotonicity being well suited to the study of some classes of partial differential equations), output feedback control, fixed-time and prescribed-time convergent systems, and the influence of neglected dynamics combined with the discretization method (which is the prominent effect?) are still largely open problems. The problem's complexity (which can stem from the sliding surface codimension or from the controller/differentiator's structure) also deserves future studies. Finally, it appears that a gap exists between monotonicity (extensively used in this article) and homogeneity (recently extensively used in continuous-time sliding-mode systems design). Analyzing the link between both deserves future attention as well. The choice of solvers for the one-step generalized equations also represents an important question for real-time implementation.

ACKNOWLEDGMENT

The work of the second author, performed during his sabbatical leave from Cinvestav-IPN, Mexico, was supported in part by Inria and Laboratoire Jean Kuntzman, Grenoble. We warmly thank Dr. Nicholas Anton Collins-Craft for his careful reading of the manuscript.

AUTHOR INFORMATION

Félix A. Miranda-Villatoro (felix.miranda-villatoro@inria.fr) received the B.Eng. degree in electronic engineering from the Universidad Autónoma del Estado de México (UAEMex), México. Afterwards, he obtained the master's and Ph.D. degrees in automatic control, both from CINVESTAV-IPN, México, in 2013 and 2017, respectively. From March 2017 to September 2020, he was a research associate in the Control Group of the University of Cambridge, U.K. Since November 2020, he has been a part of the TRIPOP team at INRIA Grenoble-Rhône-Alpes, CNRS, LJK, Grenoble INP, 38000 Grenoble, France. His research interests include nonsmooth dynamical systems, dissipative systems, nonlinear control, and nonequilibrium behaviors.

Fernando Castaños received the B.Eng. in electric and electronic engineering from the Universidad Nacional Autónoma de México (UNAM). He received the M.Eng. in control engineering from UNAM and the Ph.D. degree from the Université Paris-Sud XI. He was a postdoctoral fellow at McGill's Center for Intelligent Machines for two years. He joined the Automatic Control Department, Cinvestav-IPN, Mexico City 07360, Mexico, in 2011. His research interests include variable structure systems, passivity-based control, nonlinear control, port-Hamiltonian systems, and robust control. He currently serves as an editor for *International Journal of Robust and Nonlinear Control*. He is a Member of IEEE.

Bernard Brogliato graduated from the Mechanical Engineering Department of the École Normale Supèrieure de Cachan, France. He received the Ph.D. and Habilitation degrees from Grenoble INP in January 1991 and November 1995, respectively. He is with the University Grenoble Alpes, Inria, CNRS, LJK, Grenoble INP, 38000 Grenoble, France. His research interests include the modeling, analysis, and control of nonsmooth dynamical systems. He is the author of *Nonsmooth Mechanics, Models, Dynamics and Control* (Springer CCE, Third Edition, 2016) and coauthor of *Dissipative Systems Analysis and Control* (Springer CCE, Third Edition, 2020), *Numerical Methods for Nonsmooth Dynamical Systems* (Springer LNACM 35, 2008), *Multiple Impacts in Dissipative Granular Chains* (Springer LNACM 72, 2014), and *Nonsmooth Modeling and Simulation for Switched Circuits* (Springer LNEE 69, 2011).

REFERENCES

[1] J. André and P. Seibert, "Über stückweise lineare Differentialgleichungen, die bei Regelungsproblemen auftreten I," *Arch. Math.*, vol. 7, no. 2, pp. 148–156, 1956, doi: 10.1007/BF01899571.

[2] J. André and P. Seibert, "Über stückweise lineare Differentialgleichungen, die bei Regelungsproblemen auftreten II," *Arch. Math.*, vol. 7, no. 3, pp. 157–164, 1956, doi: 10.1007/BF01899832.

[3] Y. Z. Tsypkin, *Theory of Relay Control Systems*. Moscow, Russia: Gostechizdat, 1955.

[4] A. Aizerman and F. R. Gantmakher, "On certain switching specifics in non-linear automatic control systems with a piecewise-smooth characteristics of nonlinear element," *Automatika i Telemekhanika*, vol. 18, no. 11, pp. 1017–1028, 1957.

[5] J. J. Slotine, J. K. Hedrick, and E. A. Misawa, "On sliding observers for nonlinear systems," *J. Dyn. Syst. Meas. Control*, vol. 109, no. 3, pp. 245–252, 1987, doi: 10.1115/1.3143852.

[6] A. Levant, "Robust exact differentiation via sliding mode technique," *Automatica*, vol. 34, no. 3, pp. 379–384, 1998, doi: 10.1016/S0005-1098(97)00209-4.

[7] A. Levant, "Higher-order sliding modes, differentiation and output-feedback control," *Int. J. Control*, vol. 76, nos. 9–10, pp. 924–941, 2003, doi: 10.1080/0020717031000099029.

[8] L. V. Levantovsky, "Second order sliding algorithms: Their realization," Institute for System Studies, Moscow (in Russian) *Dyn. Heterogeneous Syst.*, vol. 31, no. 4, pp. 32–43, 1985.

[9] S. V. Emelyanov, S. K. Korovin, and L. V. Levantovsky, "Higher order sliding modes in the binary control systems," *Soviet Phys. Doklady*, vol. 31, no. 4, pp. 291–293, 1986.

[10] A. Levant, "Sliding order and sliding accuracy in sliding mode control," *Int. J. Control*, vol. 58, no. 6, pp. 1247–1263, 1993, doi: 10.1080/00207179308923053.

[11] H. Komurcugil, S. Biricik, S. Bayhan, and Z. Zhang, "Sliding mode control: Overview of its applications in power converters," *IEEE Ind. Electron. Mag.*, vol. 15, no. 1, pp. 40–49, Mar. 2021, doi: 10.1109/MIE.2020.2986165.

[12] A. Pisano and E. Usai, "Sliding mode control: A survey with applications in math," *Math. Comput. Simul.*, vol. 81, no. 5, pp. 954–979, 2011, doi: 10.1016/j.matcom.2010.10.003.

[13] G. Bartolini, A. Pisano, E. Punta, and E. Usai, "A survey of applications of second-order sliding mode control to mechanical systems," *Int. J. Control*, vol. 76, nos. 9–10, pp. 875–892, 2003, doi: 10.1080/0020717031000099010.
[14] S. J. Gambhire, D. R. Kishore, P. S. Londhe, and S. N. Pawar, "Review of sliding mode based control techniques for control system applications," *Int. J. Dyn. Control*, vol. 9, no. 1, pp. 363–378, 2021, doi: 10.1007/s40435-020-00638-7.
[15] Y. Mousavi, G. Bevan, I. B. Kucukdemiral, and A. Fekih, "Sliding mode control of wind energy conversion systems: Trends and applications," *Renewable Sustainable Energy Rev.*, vol. 167, Oct. 2022, Art. no. 112734, doi: 10.1016/j.rser.2022.112734.
[16] F. Faraz Ahmad, C. Ghenai, A. K. Hamid, and M. Bettayeb, "Application of sliding mode control for maximum power point tracking of solar photovoltaic systems: A comprehensive review," *Annu. Rev. Control*, vol. 49, pp. 173–196, May 2020, doi: 10.1016/j.arcontrol.2020.04.011.
[17] A. F. Filippov, "Differential equations with discontinuous right-hand side," *Matematicheskii Sbornik. Novaya Seriya*, vol. 51, no. 1, pp. 99–128, 1960.
[18] A. F. Filippov, *Differential Equations With Discontinuous Righthand Sides*. Dordrecht, The Netherlands, 1988.
[19] A. Polyakov, D. Efimov, and B. Brogliato, "Consistent discretization of finite-time and fixed-time stable systems," *SIAM J. Control Optim.*, vol. 57, no. 1, pp. 78–103, 2019, doi: 10.1137/18M1197345.
[20] A. Levant, "On fixed and finite time stability in sliding mode control," in *Proc. 52nd IEEE Conf. Decis. Control*, 2013, pp. 4260–4265, doi: 10.1109/CDC.2013.6760544.
[21] O. Huber, V. Acary, and B. Brogliato, "Lyapunov stability and performance analysis of the implicit discrete sliding mode control," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 3016–3030, Oct. 2016, doi: 10.1109/TAC.2015.2506991.
[22] X. Yu and G. Chen, "Discretization behaviors of equivalent control based sliding-mode control systems," *IEEE Trans. Autom. Control*, vol. 48, no. 9, pp. 1641–1646, Sep. 2003, doi: 10.1109/TAC.2003.816970.
[23] X. Yu, B. Wang, Z. Galias, and G. Chen, "Discretization effect on equivalent control-based multi-input sliding-mode control systems," *IEEE Trans. Autom. Control*, vol. 53, no. 6, pp. 1563–1569, Jul. 2008, doi: 10.1109/TAC.2008.928311.
[24] Y. Yan, Z. Galias, X. Yu, and C. Sun, "Euler's discretization effect on a twisting algorithm based sliding mode control," *Automatica*, vol. 68, pp. 203–208, Jun. 2016, doi: 10.1016/j.automatica.2016.01.051.
[25] Y. Yan, S. Yu, and X. Yu, "Quantized super-twisting algorithm based sliding mode control," *Automatica*, vol. 105, pp. 43–48, Jul. 2019, doi: 10.1016/j.automatica.2019.03.002.
[26] Y. Yan, X. Yu, and C. Sun, "Discretization behaviors of a super-twisting algorithm based sliding mode control system," in *Proc. Int. Workshop Recent Adv. Sliding Modes (RASM)*, Istanbul, Turkey, 2015, pp. 1–5, doi: 10.1109/RASM.2015.7154656.
[27] Z. Galias and X. Yu, "Complex discretization behaviors of a simple sliding-mode control system," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 53, no. 8, pp. 652–656, Aug. 2006, doi: 10.1109/TCSII.2006.875377.
[28] Z. Galias and X. Yu, "Euler's discretization of single input sliding-mode control systems," *IEEE Trans. Autom. Control*, vol. 52, no. 9, pp. 1726–1730, Sep. 2007, doi: 10.1109/TAC.2007.904289.
[29] Z. Galias and X. Yu, "Analysis of zero-order holder discretization of two-dimensional sliding-mode control systems," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 55, no. 12, pp. 1269–1273, Dec. 2008, doi: 10.1109/TCSII.2008.2008069.
[30] P. Mayr, S. Koch, M. Reichhartinger, and A. Pisano, "Boundary sliding mode control of a diffusion process under sampling and hold: Chattering analysis," in *Proc. Workshop Variable Struct. Syst.*, Abu Dhabi, UAE, 2024, pp. 7–12, doi: 10.1109/VSS61690.2024.10753410.
[31] V. Acary and B. Brogliato, "Implicit Euler numerical scheme and chattering-free implementation of sliding mode systems," *Syst. Control Lett.*, vol. 59, no. 5, pp. 284–293, 2010, doi: 10.1016/j.sysconle.2010.03.002.
[32] V. Acary, B. Brogliato, and Y. Orlov, "Chattering-free digital sliding-mode control with state observer and disturbance rejection," *IEEE Trans. Autom. Control*, vol. 57, no. 5, pp. 1087–1101, May 2012, doi: 10.1109/TAC.2011.2174676.
[33] R. Kikuuwe, S. Yasukouchi, H. Fujimoto, and M. Yamamoto, "Proxy-based sliding mode control: A safer extension of PID position control," *IEEE Trans. Robot.*, vol. 26, no. 4, pp. 670–683, Aug. 2010, doi: 10.1109/TRO.2010.2051188.
[34] O. Huber, B. Brogliato, V. Acary, A. Boubakir, F. Plestan, and B. Wang, "Experimental results on implicit and explicit time-discretization of equivalent-control-based sliding mode control," in J. P. Barbot L. Fridman and F. Plestan, Eds., *Recent Trends in Sliding Mode Control*, Stevenage, U.K.: IET, 2016, pp. 207–235.
[35] B. Brogliato, A. Polyakov, and D. Efimov, "The implicit discretization of the supertwisting sliding-mode control algorithm," *IEEE Trans. Autom. Control*, vol. 65, no. 8, pp. 3707–3713, Aug. 2020, doi: 10.1109/TAC.2019.2953091.
[36] X. Xiong, G. Chen, Y. Lou, R. Huang, and S. Kamal, "Discrete-time implementation of super-twisting control with semi-implicit Euler method," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 69, no. 1, pp. 99–103, Jan. 2022, doi: 10.1109/TCSII.2021.3078526.
[37] R. Kikuuwe, Y. Yamamato, and B. Brogliato, "Implicit implementation of nonsmooth controllers to nonsmooth actuators," *IEEE Trans. Autom. Control*, vol. 67, no. 9, pp. 4645–4657, Sep. 2022, doi: 10.1109/TAC.2022.3163124.
[38] X. Xiong, R. Kikuuwe, S. Kamal, and S. Jin, "Implicit-Euler implementation of super-twisting observer and twisting controller for second-order systems," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 67, no. 11, pp. 2607–2611, Nov. 2020, doi: 10.1109/TCSII.2019.2957271.
[39] L. Michel, S. Selvarajan, M. Ghanes, F. Plestan, Y. Aoustin, and J. P. Barbot, "An experimental investigation of discretized homogeneous differentiators: Pneumatic actuator case," *IEEE J. Emerg. Sel. Topics Ind. Electron.*, vol. 2, no. 3, pp. 227–236, Jul. 2021, doi: 10.1109/JESTIE.2021.3061924.
[40] L. Michel, M. Ghanes, F. Plestan, Y. Aoustin, and J. P. Barbot, "Semi-implicit Euler discretization for homogeneous observer-based control: One dimensional case," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 5135–5140, 2020, doi: 10.1016/j.ifacol.2020.12.1152.
[41] A. Winkler, G. Grabmair, and J. Reger, "*On implementing the implicit discrete-time super-twisting observer on mechanical systems*," *Int. J. Robust Nonlinear Control*, vol. 33, no. 13, pp. 7532–7562, 2023.
[42] X. Xiong, Y. Chu, A. D. Udai, S. Kamal, S. Jin, and Y. Lou, "Implicit discrete-time terminal sliding mode control for second-order systems," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 68, no. 7, pp. 2508–2512, Jul. 2021, doi: 10.1109/TCSII.2021.3053318.
[43] F. Miranda-Villatoro, B. Brogliato, and F. Castanos, "Multivalued robust tracking control of Lagrange systems: Continuous and discrete-time algorithms," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4436–4450, Sep. 2017, doi: 10.1109/TAC.2017.2662804.
[44] F. Miranda-Villatoro, B. Brogliato, and F. Castanos, "Set-valued sliding-mode control of uncertain linear systems: Continuous and discrete-time analysis," *SIAM J. Control Optim.*, vol. 56, no. 3, pp. 1756–1793, 2018, doi: 10.1137/16M1077362.
[45] F. Miranda-Villatoro, F. Castanos, and B. Brogliato, "Continuous and discrete-time stability of a robust set-valued nested controller," *Automatica*, vol. 107, pp. 406–417, Sep. 2019, doi: 10.1016/j.automatica.2019.06.003.
[46] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital Control of Dynamic Systems*, vol. 3. Menlo Park, CA, USA: Addison-Wesley, 1998.
[47] K. J. Åström, P. Hagander, and J. Sternby, "Zeros of sampled systems," *Automatica*, vol. 20, no. 1, pp. 31–38, 1984, doi: 10.1016/0005-1098(84)90062-1.
[48] S. Greenhalgh, V. Acary, and B. Brogliato, "On preserving dissipativity properties of linear complementarity dynamical systems with the θ-method," *Numerische Mathematik*, vol. 125, no. 4, pp. 601–637, 2013, doi: 10.1007/s00211-013-0553-5.
[49] G. Golo and C. Milosavljevic, "Robust discrete-time chattering free sliding mode control," *Syst. Control Lett.*, vol. 41, no. 1, pp. 19–28, 2000, doi: 10.1016/S0167-6911(00)00033-5.
[50] A. Levant and M. Livne, "Globally convergent differentiators with variable gains," *Int. J. Control*, vol. 91, no. 9, pp. 1994–2008, 2018, doi: 10.1080/00207179.2018.1448115.
[51] B. Brogliato, "Comments on "Finite-time stability of discrete autonomous systems [Automatica 122 (2020) 109282]," *Automatica*, vol. 156, Oct. 2023, Art. no. 111206, doi: 10.1016/j.automatica.2023.111206.
[52] O. Hájek, "Discontinuous differential equations I," *J. Differential Equ.*, vol. 32, no. 2, pp. 149–170, 1979, doi: 10.1016/0022-0396(79)90056-1.
[53] V. Utkin, J. Guldner, and J. Shi, *Sliding Mode Control in Electro-Mechanical Systems*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2009.
[54] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant, *Sliding Mode Control and Observation*. New York, NY, USA: Birkhäuser, 2013.
[55] B. Wang, N.-V. Truong, B. Brogliato, and S. Khoo, "The oscillation behaviors in Euler discretized terminal sliding mode control sytems," in *Proc. Int. Conf. Control, Automat. Inf. Sci. (ICCAIS)*, 2013, pp. 201–205, doi: 10.1109/ICCAIS.2013.6720554.

[56] R. T. Rockafellar, "Monotone operators and the proximal point algorithm," *SIAM J. Control Optim.*, vol. 14, no. 5, pp. 877–898, 1976, doi: 10.1137/0314056.
[57] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014, doi: 10.1561/2400000003.
[58] Q.-L. Dong, Y. J. Cho, S. He, P. M. Pardalos, and T. M. Rassias, *The Krasnosel'skiĭ-Mann Iterative Method: Recent Progress and Applications*. Cham, Switzerland: Springer-Verlag, 2022.
[59] B. Baji and A. Cabot, "An inertial proximal algorithm with dry friction: Finite convergence results," *Set Valued Anal.*, vol. 14, no. 1, pp. 1–23, 2006, doi: 10.1007/s11228-005-0002-4.
[60] B. Brogliato, "Absolute stability and the Lagrange-Dirichlet theorem with monotone multivalued mappings," *Syst. Control Lett.*, vol. 51, no. 5, pp. 343–353, 2004, doi: 10.1016/j.sysconle.2003.09.007.
[61] B. Brogliato, "Dissipative dynamical systems with set-valued feedback loops: Well-posed set-valued Lur'e dynamical systems," *IEEE Control Syst. Mag.*, vol. 42, no. 3, pp. 93–114, Jun. 2022, doi: 10.1109/MCS.2022.3157160.
[62] V. Acary and B. Brogliato, *Numerical Simulation for Nonsmooth Dynamical Systems. Applications in Mechanics and Electronics* (Lectures Notes in Applied and Computational Mechanics), vol. 35. Berlin, Germany: Springer-Verlag, 2008.
[63] F. Facchinei and J. S. Pang, *Finite-Dimensional Variational Inequalities and Complementarity Problems* (Springer Series in Operations Research), vol. I. New York, NY, USA: Springer-Verlag, 2003.
[64] D. S. Bernstein, *Scalar, Vector and Matrix Mathematics: Theory, Facts and Formulas*, expanded ed. Princeton, NJ, USA: Princeton Univ. Press, 2018.
[65] F. A. Miranda and F. Castaños, "Robust output regulation of linear passive systems using maximally monotone controls," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Osaka, Japan, 2015, pp. 6897–6902, doi: 10.1109/CDC.2015.7403306.
[66] W. M. Haddad and J. Lee, "Finite-time stability of discrete autonomous systems," *Automatica*, vol. 122, Dec. 2020, Art. no. 109282, doi: 10.1016/j.automatica.2020.109282.
[67] P. Prasun, S. Kamal, S. Pandey, A. Bartoszewicz, and S. Ghosh, "A minimum operator-based discrete-time sliding mode control," *IEEE Trans. Autom. Control*, vol. 69, no. 11, pp. 7871–7876, Nov. 2024, doi: 10.1109/TAC.2024.3397911.
[68] P. Prasun, S. Kamal, S. Ghosh, and T. N. Dinh, "A minimum operator based discrete variable structure control," *Eur. J. Control*, vol. 76, Mar. 2024, Art. no. 100953, doi: 10.1016/j.ejcon.2024.100953.
[69] V. Pandey, S. Kamal, and S. Ghosh, "Discrete-time sliding mode controller for magnetic levitation system using minima operator," in *Proc. Int. Workshop Variable Struct. Syst.*, Abu Dhabi, UAE, 2024, pp. 69–74.
[70] F. A. Miranda-Villatoro, F. Castaños, and B. Brogliato, "Finite-time convergent discrete-time algorithms: From explicit to backward schemes," *Automatica*, vol. 174, Apr. 2025, Art. no. 112080, doi: 10.1016/j.automatica.2024.112080.
[71] D. S. Bernstein, "Facing future challenges in feedback control of aerospace systems through scientific experimentation," *J. Guid., Control Dyn.*, vol. 45, no. 12, pp. 2202–2210, 2022, doi: 10.2514/1.G006785.
[72] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 83, pp. 1–5, 2016.
[73] B. Brogliato, R. Lozano, B. Maschke, and O. Egeland, *Dissipative Systems Analysis and Control: Theory and Applications*. London, U.K.: Springer-Verlag, 2020.
[74] C. I. Byrnes, A. Isidori, and J. C. Willems, "Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems," *IEEE Trans. Autom. Control*, vol. 36, no. 11, pp. 1228–1240, Nov. 1991, doi: 10.1109/9.100932.
[75] G. R. Duan and H. H. Yu, *LMIs in Control Systems, Analysis, Design and Applications*. Boca Raton, FL, USA: CRC Press, 2013.
[76] M. A. Estrada, J. A. Moreno, and L. Fridman, "Sliding mode controllers design based on control Lyapunov functions for uncertain LTI systems," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 615–1620, 2023, doi: 10.1016/j.ifacol.2023.10.1863.
[77] W. Lin and C. I. Byrnes, "Passivity and absolute stabilization of a class of discrete-time nonlinear systems," *Automatica*, vol. 31, no. 2, pp. 263–267, 1995, doi: 10.1016/0005-1098(94)00075-T.
[78] E. M. Navarro-López and E. Fossas-Colet, "Feedback passivity of nonlinear discrete-time systems with direct input–output link," *Automatica*, vol. 40, no. 8, pp. 1423–1428, 2004, doi: 10.1016/j.automatica.2004.03.009.
[79] D. J. Hill and P. J. Moylan, "Dissipative dynamical systems: Basic input-output and state properties," *J. Franklin Inst.*, vol. 30, no. 5, pp. 327–357, 1980.
[80] E. Cruz-Zavala and J. A. Moreno, "Homogeneous high order sliding mode design: A Lyapunov approach," *Automatica*, vol. 80, pp. 232–238, Jun. 2017, doi: 10.1016/j.automatica.2017.02.039.
[81] S. Gutman, "Uncertain dynamical systems–a Lyapunov min-max approach," *IEEE Trans. Autom. Control*, vol. 24, no. 3, pp. 437–443, Jun. 1979, doi: 10.1109/TAC.1979.1102073.
[82] G. Leitmann, "Guaranteed asymptotic stability for some linear systems with bounded uncertainties," *J. Dyn. Syst. Meas. Control*, vol. 101, no. 3, pp. 212–216, 1979, doi: 10.1115/1.3426427.
[83] A. Levant, "Homogeneity approach to high-order sliding mode design," *Automatica*, vol. 41, no. 5, pp. 823–830, 2005, doi: 10.1016/j.automatica.2004.11.029.
[84] A. Levant and M. Livne, "Robust exact filtering differentiators," *Eur. J. Control*, vol. 55, pp. 33–44, Sep. 2020, doi: 10.1016/j.ejcon.2019.08.006.
[85] J. A. Moreno and M. Osorio, "Strict Lyapunov functions for the super-twisting algorithm," *IEEE Trans. Autom. Control*, vol. 57, no. 4, pp. 1035–1040, Apr. 2012, doi: 10.1109/TAC.2012.2186179.
[86] J. E. Carvajal-Rubio, J. D. Sánchez-Torres, M. Defoort, M. Djemai, and A. G. Loukianov, "Implicit and explicit discrete-time realizations of homogeneous differentiators," *Int. J. Robust Nonlinear Control*, vol. 31, no. 9, pp. 3606–3630, 2021.
[87] M. R. Mojallizadeh, B. Brogliato, and V. Acary, "Time-discretization of differentiators: Design of implicit algorithms, and comparative analysis," *Int. J. Robust Nonlinear Control*, vol. 31, no. 16, pp. 7679–7723, 2021, doi: 10.1002/rnc.5710.
[88] Y. Shtessel, M. Taleb, and F. Plestan, "A novel adaptive-gain *supertwisting* sliding mode controller: Methodology and application," *Automatica*, vol. 48, no. 5, pp. 759–769, 2012, doi: 10.1016/j.automatica.2012.02.024.
[89] B. Andritsch, L. Watermann, S. Koch, M. Reichhartinger, J. Reger, and M. Horn, "Modified implicit discretization of the super-twisting controller," *IEEE Trans. Autom. Control*, vol. 69, no. 8, pp. 5620–5626, Aug. 2024, doi: 10.1109/TAC.2024.3370494.
[90] I. Nagesh and C. Edwards, "A multivariable super-twisting sliding mode approach," *Automatica*, vol. 50, no. 3, pp. 984–988, Mar. 2014, doi: 10.1016/j.automatica.2013.12.032.
[91] G. Besançon, *Nonlinear Observers and Applications* (Lecture Notes in Control and Information Sciences), vol. 363. Berlin, Germany: Springer-Verlag, 2007.
[92] J. A. Moreno et al., "On discontinuous observers for second order systems: Properties, analysis and design," in *Advances in Sliding Mode Control*, B. Bandyopadhyay, Ed., pp. 243–265. Berlin, Germany: Springer-Verlag, 2013, pp. 243–265.
[93] M. R. Mojallizadeh et al., "Discrete-time differentiators in closed-loop control systems: Experiments on electro-pneumatic system and rotary inverted pendulum," INRIA, Montbonnot-Saint-Martin, France, Feb. 2023. [Online]. Available: https://inria.hal.science/hal-03125960v2/document
[94] J. E. Carvajal-Rubio, M. Defoort, J. D. Sánchez-Torres, M. Djemai, and A. G. Loukianov, "Implicit and explicit discrete-time realizations of the robust exact filtering differentiator," *J. Franklin Inst.*, vol. 359, no. 8, pp. 3951–3978, 2022, doi: 10.1016/j.jfranklin.2022.03.007.
[95] E. Cruz-Zavala, J. A. Moreno, and L. M. Fridman, "Uniform robust exact differentiator," *IEEE Trans. Autom. Control*, vol. 56, no. 11, pp. 2727–2733, Nov. 2011, doi: 10.1109/TAC.2011.2160030.
[96] S. Koch, M. Reichhartinger, M. Horn, and L. Fridman, "Discrete-time implementation of homogeneous differentiators," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 757–762, Feb. 2020, doi: 10.1109/TAC.2019.2919237.
[97] R. Kikuuwe, R. Pasaribu, and G. Byun, "A first-order differentiator with first-order sliding mode filtering," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 771–776, 2019, doi: 10.1016/j.ifacol.2019.12.056.
[98] G. Byun and R. Kikuuwe, "An improved sliding mode differentiator combined with sliding mode filter for estimating first and second-order derivatives of noisy signals," *Int. J. Control Autom. Syst.*, vol. 18, no. 12, pp. 3001–3014, 2020, doi: 10.1007/s12555-019-0688-y.
[99] A. Levant and A. Michael, "Adjustment of high-order sliding-mode controllers," *Int. J. Robust Nonlinear Control*, vol. 19, no. 15, pp. 1657–1672, 2009, doi: 10.1002/rnc.1397.
[100] S. Ding, A. Levant, and S. Li, "Simple homogeneous sliding-mode controller," *Automatica*, vol. 67, pp. 22–32, May 2016, doi: 10.1016/j.automatica.2016.01.017.

[101] J. Davila, L. Fridman, and A. Levant, "Second-order sliding-mode observer for mechanical systems," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1785–1789, Nov. 2005, doi: 10.1109/TAC.2005.858636.
[102] A. Polyakov, *Generalized Homogeneity in Systems and Control* (Communications and Control Engineering). Cham, Switzerland: Springer-Verlag, 2020.
[103] B. Brogliato and A. Polyakov, "Digital implementation of sliding-mode control via the implicit method: A tutorial," *Int. J. Robust Nonlinear Control*, vol. 31, no. 9, pp. 3528–3586, 2021, doi: 10.1002/rnc.5121.
[104] F. Castaños, F. M. Villatoro, and B. Brogliato, "Multivalued Hamiltonian systems with multivalued dissipation: Analysis of the backward-Euler discretisation," INRIA, Grenoble, France, Jun. 2024. [Online]. Available: https://inria.hal.science/hal-04625231v1/document
[105] C. Wang, H. Xia, and S. Ren, "An implicit discretization-based adaptive reaching law for discrete-time sliding mode control systems," *J. Vib. Control*, vol. 29, nos. 5-6, pp. 1117–1127, 2023, doi: 10.1177/10775463211057652.
[106] C. Wang, H. Xia, Y. Wang, and S. Ren, "Implicit discrete-time fast terminal sliding mode control with disturbance compensation," *Asian J. Control*, vol. 25, no. 1, pp. 637–643, 2023, doi: 10.1002/asjc.2784.
[107] O. Huber, V. Acary, and B. Brogliato, "Lyapunov stability analysis of the implicit discrete-time twisting control algorithm," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2619–2626, Jun. 2020, doi: 10.1109/TAC.2019.2940323.
[108] B. Singh, X. Xiong, D. Ghosh, and S. Kamal, "Numerical integrator based on implicit Euler discretization of twisting control algorithm," in *Proc. 17th Int. Workshop Variable Struct. Syst.*, Abu Dhabi, UAE, 2024, pp. 46–50, doi: 10.1109/VSS61690.2024.10753376.
[109] X. Xiong, A. Sachan, R. Huang, P. Anil Kumar, S. Kamal, and W. Chen, "Implicit-Euler based digital implementation for constrained stabilization of second-order systems," *Int. J. Robust Nonlinear Control*, vol. 31, no. 11, pp. 5086–5100, 2021, doi: 10.1002/rnc.5525.
[110] R. Kikuuwe, "Sliding motion accuracy of proxy-based sliding mode control subjected to measurement noise and disturbance," *Eur. J. Control*, vol. 58, pp. 114–122, Mar. 2021, doi: 10.1016/j.ejcon.2020.07.005.
[111] Z. Lv, S. Jin, X. Xiong, and J. Yu, "A new quick-response sliding mode tracking differentiator with its chattering-free discrete-time implementation," *IEEE Access*, vol. 7, pp. 130,236–130,245, 2019, doi: 10.1109/ACCESS.2019.2940262.
[112] S. Jin, Z. Lv, X. Xiong, and J. Yu, "A chattering-free sliding mode filter enhanced by first order derivative feedforward," *IEEE Access*, vol. 8, pp. 41,175–41,185, 2020, doi: 10.1109/ACCESS.2020.2976737.
[113] M. T. S. Aung, Z. Shi, and R. Kikuuwe, "A new parabolic sliding mode filter augmented by a linear low-pass filter and its application to position control," *J. Dyn. Syst. Meas. Control*, vol. 140, no. 4, 2018, Art. no. 041005.
[114] S. Jin, R. Kikuuwe, and M. Yamamoto, "Improving velocity feedback for position control by using a discrete-time sliding mode filtering with adaptive windowing," *Adv. Rob.*, vol. 28, no. 14, pp. 943–953, 2014, doi: 10.1080/01691864.2014.899161.
[115] L. Michel, M. Ghanes, Y. Aoustin, and J. P. Barbot, "An interconnected discrete time cascaded semi-implicit differentiation," in *Proc. 17th Int. Workshop Variable Struct. Syst.*, Abu Dhabi, UAE, 2024, pp. 164–169, doi: 10.1109/VSS61690.2024.10753413.
[116] S. Jin, X. Xiong, D. Zhao, and C. Jin, "Unified framework for implicit-Euler implementation of second-order sliding mode controllers," *Commun. Nonlinear Sci. Numer. Simul.* , vol. 111, Aug. 2022, Art. no. 106430, doi: 10.1016/j.cnsns.2022.106430.
[117] T. Liard, I. Balogoun, S. Marx, and F. Plestan, "Boundary sliding mode control of a system of linear hyperbolic equations: A Lyapunov approach," *Automatica*, vol. 135, Jan. 2022, Art. no. 109964, doi: 10.1016/j.automatica.2021.109964.
[118] B. Brogliato and A. Polyakov, "Globally stable implicit Euler time-discretization of a nonlinear single-input sliding-mode control system," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, 2015, pp. 5426–5431, doi: 10.1109/CDC.2015.7403069.
[119] X. Xiong, H. Chen, Y. Lou, Z. Liu, S. Kamal, and M. Yamamoto, "Implicit discrete-time adaptive first-order sliding mode control with predefined convergence time," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 68, no. 12, pp. 3562–3566, Dec. 2021, doi: 10.1109/TCSII.2021.3070435.
[120] C. Hettiger et al., "On discretization methods for indirect adaptive sliding mode control," in *Proc. IEEE 61st Conf. Decis. Control (CDC)*, Cancun, Mexico, 2022, pp. 4930–4936, doi: 10.1109/CDC51059.2022.9993161.
[121] X. Xiong, S. Kamal, and S. Jin, "Adaptive gains to super-twisting technique for sliding mode design," *Asian J. Control*, vol. 23, no. 1, pp. 362–373, 2021, doi: 10.1002/asjc.2202.
[122] D. Luo, X. Xiong, S. Jin, and S. Kamal, "Adaptive gains of dual level to super-twisting algorithm for sliding mode design," *IET Control Theory Appl.*, vol. 12, no. 17, pp. 2347–2356, 2018, doi: 10.1049/iet-cta.2018.5380.
[123] C. Wang, H. Xia, Y. Wang, and S. Ren, "Discrete-time sliding mode control with adaptive reaching law via implicit Euler method," *Int. J. Control Autom. Syst.*, vol. 21, no. 1, pp. 109–116, 2023, doi: 10.1007/s12555-021-0478-1.
[124] R. K. Sharma, X. Xiong, S. Kamal, and S. Ghosh, "Discrete-time super-twisting fractional-order differentiator with implicit Euler method," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 68, no. 4, pp. 1238–1242, Apr. 2021, doi: 10.1109/TCSII.2020.3027733.
[125] X. Xiong, R. K. Sharma, S. Kamal, S. Ghosh, Y. Bai, and Y. Lou, "Discrete-time super-twisting fractional-order observer with implicit Euler method," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 69, no. 6, pp. 2787–2791, Jun. 2022, doi: 10.1109/TCSII.2021.3131369.
[126] X. Yang, X. Xiong, Z. Zou, Y. Lou, S. Kamal, and J. Li, "Discrete-time multivariable super-twisting algorithm with semi-implicit Euler method," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 69, no. 11, pp. 4443–4447, Nov. 2022, doi: 10.1109/TCSII.2022.3182772.
[127] P. Prasun, S. Pandey, S. Kamal, S. Ghosh, and X. Xiong, "A minimum operator based discrete-time super-twisting-like algorithm," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 71, no. 1, pp. 286–290, 2024, doi: 10.1109/TCSII.2023.3298834.
[128] O. Huber, "Analysis and implementation of discrete-time sliding mode control," Ph.D. thesis, Université Grenoble Alpes, Saint-Martin-d'Hères, France, May 2015. [Online]. Available: https://inria.hal.science/tel-01194430v1/document
[129] A. Gonzalez-Garcia, H. Castañeda, and J. De León-Morales, "Unmanned surface vehicle robust tracking control using an adaptive super-twisting controller," *Control Eng. Pract.*, vol. 149, Aug. 2024, Art. no. 105985, doi: 10.1016/j.conengprac.2024.105985.
[130] X. Xiong, Y. Bai, R. Shi, S. Kamal, Y. Wang, and Y. Lou, "Discrete-time twisting algorithm implementation with implicit-Euler ZOH discretization method," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 69, no. 8, pp. 3435–3439, Aug. 2022, doi: 10.1109/TCSII.2022.3144197.
[131] R. Seeber and S. Koch, "Structural conditions for chattering avoidance in implicitly discretized sliding mode differentiators," *IEEE Control Syst. Lett.*, vol. 7, pp. 2065–2070, 2023, doi: 10.1109/LCSYS.2023.3282886.
[132] F. A. Miranda-Villatoro, "A variational approach to the design of multivariable discrete-time supertwisting-like algorithms," *IEEE Trans. Autom. Control*, vol. 70, no. 4, pp. 2495–2506, Apr. 2025, doi: 10.1109/TAC.2024.3484308.
[133] N. Baiomy, M. M. Abdelhafiz, and A. Mahgoub, "Implicit Euler discretization for enhanced sliding mode observer in sensorless permanent magnet synchronous motor control," *IEEE Access*, vol. 13, pp. 4304–4314, 2025, doi: 10.1109/ACCESS.2024.3524159.
[134] M. R. Mojallizadeh et al., "A survey on the discrete-time differentiators in closed-loop control systems: Experiments on an electro-pneumatic system," *Control Eng. Pract.*, vol. 136, Jul. 2023, Art. no. 105546, doi: 10.1016/j.conengprac.2023.105546.
[135] O. Güler, *Foundations of Optimization* (Graduate Texts in Mathematics), vol. 258. New York, NY, USA: Springer-Verlag, 2010.
[136] P. L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM J. Numer. Anal.*, vol. 16, no. 6, pp. 964–979, 1979, doi: 10.1137/0716071.
[137] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, Eds., New York, NY, USA: Springer-Verlag, 2011, pp. 185–212.
[138] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi, "Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists," *SIAM Rev.*, vol. 65, no. 2, pp. 375–435, 2023, doi: 10.1137/20M1379344.
[139] P. Goulart and Y. Chen. "Clarabel solver." Clarabel. Accessed: Nov. 23, 2023. [Online]. Available: https://clarabel.org/stable/
[140] O. Huber, V. Acary, B. Brogliato, and F. Plestan, "Implicit discrete-time twisting controller without numerical chattering: Analysis and experimental results," *Control Eng. Pract.*, vol. 46, pp. 129–141, Jan. 2016, doi: 10.1016/j.conengprac.2015.10.013.
[141] B. Wang, B. Brogliato, V. Acary, A. Boubakir, and F. Plestan, "Experimental comparisons between implicit and explicit implementations of discrete-time sliding mode controllers: Toward input and output chattering

suppression," *IEEE Trans. Control Syst. Technol.* , vol. 23, no. 5, pp. 2071–2075, Sep. 2015, doi: 10.1109/TCST.2015.2396473.
[142] L. Michel, M. Ghanes, Y. Aoustin, and J. P. Barbot, "A third order semi-implicit homogeneous differentiator: Experimental results," in *Proc. 16th Int. Workshop Variable Struct. Syst. (VSS)*, Rio de Janeiro, Brazilm, 2022, pp. 77–82, doi: 10.1109/VSS57184.2022.9902069.
[143] L. Michel et al., "A semi-implicit homogeneous discretized differentiator based on two projectors: Experimental validation on a cable-driven parallel robot," *Mechanics Industry*, vol. 25, no. 11, pp. 1–10, 2024.
[144] M. A. Alarcón-Carbajal, J. E. Carvajal-Rubio, J. D. Sánchez-Torres, D. E. Castro-Palazuelos, and G. J. Rubio-Astorga, "An output feedback discrete-time controller for the DC-DC Buck converter," *Energies*, vol. 15, no. 14, 2022, Art. no. 5288, doi: 10.3390/en15145288.
[145] J. E. Carvajal-Rubio et al., "Robust discrete-time output feedback stabilization of integrator chains," *Int. J. Robust Nonlinear Control*, vol. 33, no. 15, pp. 9233–9250, 2023.
[146] G. Perozzi, A. Polyakov, and B. Brogliato, and F. Miranda-Villatoro, "Upgrading a linear controller to a sliding mode one: Theory and experiments," *Control Eng. Pract.*, vol. 123, Jun. 2022, Art. no. 105107, doi: 10.1016/j.conengprac.2022.105107.
[147] M. R. Mojallizadeh and B. Brogliato, "Effect of Euler explicit and implicit time discretizations on variable-structure differentiators," in *Sliding-Mode Control and Variable-Structure Systems: Studies in Systems Decision and Control*, T. R. Oliveira, L. Fridman, and L. Hsu, Eds., Cham, Switzerland: Springer-Verlag, , 2023, pp. 165–180.
[148] R. Mae and R. Kikuuwe, "An admittance controller with a jerk limiter for position-controlled robots," *J. Robot. Mechatron.*, vol. 36, no. 2, pp. 483–493, 2024, doi: 10.20965/jrm.2024.p0483.
[149] R. Kikuuwe, "Torque-bounded admittance control realized by a set-valued algebraic feedback," *IEEE Trans. Robot.*, vol. 35, no. 5, pp. 1136–1149, Oct. 2019, doi: 10.1109/TRO.2019.2920069.
[150] X. Yuan, Y. Ding, X. Xiong, and Y. Lou, "Torque-bounded admittance control with implicit Euler realization of set-valued operators," *IEEE/ASME Trans. Mechatronics*, vol. 29, no. 5, pp. 3360–3369, 2023, doi: 10.1109/TMECH.2023.3342479.
[151] M. T. S. Aung and R. Kikuuwe, "Stability enhancement of admittance control with acceleration feedback and friction compensation," *Mechatronics*, vol. 45, pp. 110–118, Aug. 2017, doi: 10.1016/j.mechatronics.2017.06.011.
[152] I. Sala-Mira, J. L. Díez, B. Ricarte, and J. Bondia, "Sliding-mode disturbance observers for an artificial pancreas without meal announcement," *J. Process Control*, vol. 78, pp. 68–77, Jun. 2019, doi: 10.1016/j.jprocont.2019.03.008.
[153] S. Faccioli et al., "Super–twisting-based meal detector for type 1 diabetes management: Improvement and assessment in a real-life scenario," *Comput. Methods Programs Biomed.*, vol. 219, Jun. 2022, Art. no. 10673, doi: 10.1016/j.cmpb.2022.106736.
[154] I. Sala-Mira, M. Siket, L. Kovács, G. Eigner, and J. Bondia, "Effect of model, observer and their interaction on state and disturbance estimation in artificial pancreas: An in-silico study," *IEEE Access*, vol. 9, pp. 143,549–143,563, 2021, doi: 10.1109/ACCESS.2021.3120880.
[155] L. Michel, C. Braud, J.-P. Barbot, F. Plestan, D. Peaucelle, and X. Boucher, "Comparison of different feedback controllers on an airfoil benchmark," *Wind Energy Sci.*, vol. 10, no. 1, pp. 177–191, 2025, doi: 10.5194/wes-10-177-2025.
[156] R. Nishimoto and R. Kikuuwe, "Position-commanding anti-sway controller for 2-D overhead cranes under velocity and acceleration constraints," *IEEE Access*, vol. 11, pp. 35,069–35,079, 2023, doi: 10.1109/ACCESS.2023.3265586.
[157] G. Chen, X. Xiong, and Y. Lou, "Multi-state modelling and observation of magneto-rheological clutch with rate-dependent hysteresis characteristic," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2445–2452, Apr. 2021, doi: 10.1109/LRA.2021.3061064.
[158] G. Chen, X. Xiong, Y. Lou, and Z. Li, "Modeling and observation of rate-dependent hysteresis and creep phenomena in magnetorheological clutch," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 4, pp. 2053–2061, Aug. 2022, doi: 10.1109/TMECH.2022.3172515.
[159] X. Xiong, Z. Zou, Y. Lou, X. Yang, X. Zhu, and F. Zheng, "Semi-implicit Euler realization of time-delayed super-twisting algorithm with modified smith predictor," in *Proc. IEEE 19th Int. Conf. Automat. Sci. Eng. (CASE)*, 2023, pp. 1–6, doi: 10.1109/CASE56687.2023.10260432.
[160] X. Yang, X. Xiong, Z. Zou, and Y. Lou, "Semi-implicit Euler digital implementation of conditioned super-twisting algorithm with actuation saturation," *IEEE Trans. Ind. Electron.*, vol. 70, no. 8, pp. 8388–8397, Aug. 2023, doi: 10.1109/TIE.2022.3229380.
[161] R. Seeber, "Discussion on "Semi-implicit Euler digital implementation of conditioned super-twisting algorithm with actuation saturation," *IEEE Trans. Ind. Electron.*, vol. 71, no. 4, pp. 4304–4304, Apr. 2024, doi: 10.1109/TIE.2023.3274855.
[162] B. Andritsch, V. Holzner, S. Koch, M. Reichhartinger, and M. Horn, "Experimental comparison of sliding mode control dedicated discretization approaches," in *Proc. 17th Int. Workshop Variable Struct. Syst. (VSS)*, Abu Dhabi, UAE, 2024, pp. 40–45, doi: 10.1109/VSS61690.2024.10753395.
[163] M. R. Mojallizadeh et al., "Control design for thrust generators with application to wind turbine wave-tank testing: A sliding-mode control approach with Euler backward time-discretization," *Control Eng. Pract.*, vol. 146, May 2024, Art. no. 105894, doi: 10.1016/j.conengprac.2024.105894.
[164] M. R. Mojallizadeh, F. Bonnefoy, F. Plestan, M. A. Hamida, and J. Ohana, "Euler implicit time-discretization of multivariable sliding-mode controllers," *ISA Trans.*, vol. 147, pp. 140–152, Apr. 2024, doi: 10.1016/j.isatra.2024.01.031.
[165] F. Bonnefoy, V. Leroy, M. R. Mojallizadeh, S. Delacroix, V. Arnal, and J.-C. Gilloteaux, "Multidimensional hybrid software-in-the-loop modeling approach for experimental analysis of a floating offshore wind turbine in wave tank experiments," *Ocean Eng.*, vol. 309, Oct. 2024, Art. no. 118390, doi: 10.1016/j.oceaneng.2024.118390.
[166] M. Zapf, B. Andritsch, N. Contartese, S. Koch, M. Reichhartinger, and M. Horn, "Application of discretized super-twisting control to magnetic resonance imaging patient positioning systems," in *Proc. 17th Int. Workshop Variable Struct. Syst. (VSS)*, Abu Dhabi, UAE, 2024, pp. 51–56, doi: 10.1109/VSS61690.2024.10753412.
[167] H. Chen, X. Xiong, K. Honda, S. Okunami, and M. Yamamoto, "FES control of a finger MP joint with a proxy-based super-twisting algorithm," *Appl. Sci.*, vol. 14, no. 11, 2024, Art. no. 4905, doi: 10.3390/app14114905.
[168] R. Kikuuwe, T. Okada, H. Yoshihara, T. Doi, T. Nanjo, and K. Yamashita, "A nonsmooth quasi-static modeling approach for hydraulic actuators," *J. Dyn. Syst. Meas. Control*, vol. 143, no. 12, Dec. 2021, Art. no. 121002, doi: 10.1115/1.4051894.
[169] Y. Yamamoto et al., "A sliding-mode set-point position controller for hydraulic excavators," *IEEE Access*, vol. 9, pp. 153,735–153,749, 2021, doi: 10.1109/ACCESS.2021.3128215.
[170] Y. Yamamoto, J. Qiu, T. Doi, T. Nanjo, K. Yamashita, and R. Kikuuwe, "A position controller for hydraulic excavators with deadtime and regenerative pipelines," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 855–871, 2025, doi: 10.1109/TASE.2024.3354952.
[171] Y. Yamamoto and R. Kikuuwe, "A force controller for valve-manipulated hydraulic actuators," *Control Eng. Pract.*, vol. 156, Mar. 2025, Art. no. 106234, doi: 10.1016/j.conengprac.2024.106234.
[172] M. R. Mojallizadeh, "Differentiation toolbox," Univ. Grenoble Alpes, INRIA, Grenoble, France, 2020. [Online]. Available: https://github.com/Mojallizadeh/DifferentiationToolbox
[173] B. Andritsch, M. Horn, S. Koch, H. Niederwieser, M. Wetzlinger, and M. Reichhartinger, "The robust exact differentiator toolbox revisited: Filtering and discretization features," in *Proc. IEEE Int. Conf. Mechatron. (ICM)*, 2021, *pp.* 01–06, doi: 10.1109/ICM46511.2021.9385675.
[174] R. Seeber, "Implicit discrete-time implementation of robust exact differentiators – A toolbox," *at - Automatisierungstechnik*, vol. 72, no. 8, pp. 757–768, 2024, doi: 10.1515/auto-2024-5068.
[175] B. Wang, B. Brogliato, V. Acary, A. Boubakir, and F. Plestan, "Comparisons between implicit and explicit discrete-time implementations of equivalent-control-based sliding mode controllers: Input and output chattering suppression via the implicit method," INRIA, Grenoble, France, Res. Rep. hal-01087400, Nov. 2014. [Online]. Available: https://inria.hal.science/hal-01087400v1/document
[176] B. Wang, B. Brogliato, V. Acary, A. Boubakir, and F. Plestan, "Experimental comparisons between implicit and explicit implementations of discrete-time sliding mode controllers: Towards chattering suppression in output and input signals," in *Proc. 13th Int. Workshop Variable Struct. Syst. (VSS)*, 2014, pp. 1–6, doi: 10.1109/VSS.2014.6881159.
[177] O. Huber. *Sliding Mode Controller on an Electro-Pneumatic Benchmark in Nantes*. (2016). Accessed: Aug. 3, 2025. [Online Video]. http://xhub.github.io/pages/videos.html