When proximal-point algorithms meet set-valued systems. An Optimization point of view of discrete-time sliding modes

Félix Miranda-Villatoro, Fernando Castanos, Bernard Brogliato

When proximal-point algorithms meet set-valued systems. An Optimization point of view of discrete-time sliding modes

FÉLIX MIRANDA-VILLATORO[†], FERNANDO CASTAÑOS[⋆], AND BERNARD BROGLIATO[†]
[†]Univ. Grenoble Alpes, Inria, CNRS, LJK, Grenoble INP, Grenoble, 38000 France.
[⋆] Automatic Control Department, Cinvestav-IPN, 2508 Av. IPN, 07360, Mexico City, Mexico.

June 26, 2024

Sliding-mode control is a major field of Automatic Control that originated in Germany [1, 2] and in the former Soviet Union [3, 4], about 70 years ago during the development of relay systems. Sliding-mode principles were later incorporated in the design of differentiators and observers [5, 6, 7]. Higher-order sliding modes were introduced by Levant in [6, 7, 8, 9, 10]. They represented a significant progress for control and differentiation, the prominent algorithm being the super-twisting scheme. Overviews of sliding-mode control and its applications can be found in [11, 12, 13, 14, 15, 16]. Mathematical foundations for sliding-modes were set mainly by Filippov [17, 18] within the framework of set-valued analysis and differential inclusions. Though other mathematical settings exist (maximal monotone operators being an important one as shown in this article), Filippov's differential inclusions remain the most popular mathematical concept in the automatic control field. Later, discrete-time implementations were studied, see "Sidebar II" for a short 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 dissemination in applications. It refers to high-frequency bang-bang-like input signals which harm actuators and possibly yield closed-loop instability [19, 20, 21], as well as high-frequency oscillations of 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]. This has motivated the introduction of a new discretization approach, broadly known as the *implicit* (or, the *backward* Euler) method [30, 31, 32].

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 [31, 21, 30, 33, 34, 35, 36, 37, 32, 38, 39, 40, 41], and especially in [42, 43, 44], 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.

Control systems are mainly designed in a continuous-time setting, 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 small enough [45, Chapter 3]. It is well-known, however, that the discretization process can deteriorate the performance and modify the system's properties (zeroes [46], passivity [47], *etc*).

Systems with sliding modes are particularly susceptible 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 small the sampling time is.

This tutorial paper is organized in 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), 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, both in 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), 4) to sur-

Systems with sliding modes are particularly susceptible to the employed discretization method.

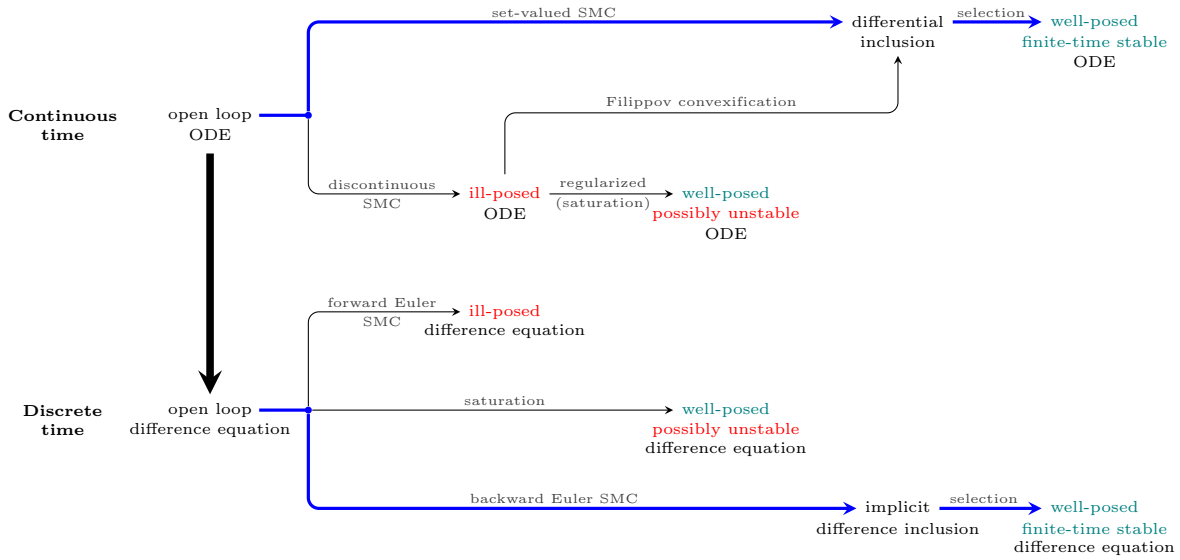

Figure 1: Flowchart of the implementation of control systems using sliding modes. From the point of view of traditional sliding-mode control (SMC), a discontinuous feedback law is enforced on an ordinary differential equation (ODE). A consistent notion of *solution* requires the conversion of the closed-loop discontinuous ODE into a differential inclusion (Filippov's convexification). Alternatively, a set-valued feedback law can be directly formulated from the outset, yielding again a differential inclusion 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 chosen 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.

vey the results (theoretical and experimental) that have been obtained so far concerning the backward-Euler discretization of set-valued sliding-mode systems.

1 SUBGRADIENT FLOWS, PROXIMAL ALGORITHMS, FINITE-TIME CONVERGENCE

The intimate relation between optimization and the implicit discretization of set-valued sliding-mode systems is illustrated in this section (see "Sidebar III" for a definition of sliding motion). For the sake of simplicity, the presentation is restricted to the scalar case without perturbation. The disturbed case is analysed in the following section in a more general framework. 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 $u(t) = u_{\mathrm{sv}}(x(t))$. The set-valued feedback

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

where $\mathbf{sgn}(\cdot)$ is the set-valued signum function (30), achieves the finite-time convergence of the state towards the origin [51, 52]. Indeed, the closed-loop system belongs to the class of gradient systems

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

with $f(x) = -\frac{a}{2}x^2 + b\kappa|x|$. According to Proposition 2 in "Sidebar III", the origin is finite-time stable. Due to the

Sidebar I: Summary

This article is largely concerned with the relationships between the implicit (backward) discretization of set-valued sliding-mode systems (for control, state observation, or differentiation), and proximal-point algorithms of optimization. It is shown that using well-known notions like maximal monotonicity, proximal and resolvent operators, and Yosida approximations, it is possible to embed implicit time-discretization methods into the unified framework of proximal-point algorithms. This is illustrated with first-order and higher-order sliding-mode systems. Passivity is shown to play an important role. Extensions of the classical proximal-point algorithm are introduced. Particular attention is given to the use of splitting methods to resolve computational issues related to implementation.

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 in a digital computer. To that end, let us 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. Indeed, by the variation of constants formula it follows 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 follows 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 $\tilde{b} = \frac{(e^{ah}-1)b}{a}$. Let us consider first the explicit (forward-Euler) discretization of (2), that is, $u_k = u_{\rm sv}(x_k)$, where

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

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 difference inclusion (7) does not have a unique solution, as the right-hand side becomes a non-singleton. 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, in the limit, the oscillation disappears. On the other hand, if h increases then $\tilde{b}$ increases and the amplitude increases. This stands in sharp contrast with the implicit method described below, 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 deeply analyzed in [22, 23, 24, 25, 26, 27, 28, 29, 53]. 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 below).

Let us now consider the backward-Euler discretization of (2),

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

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 $t_k = kh$, (*i.e.*, $\tilde{a}$, $\tilde{b}$, κ, and x_k). Indeed, (9) is equivalent to the following *generalized equation* in the unknown x_{k+1}:

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

where $\mathrm{I_d}$ is the identity mapping. In view of the maximal monotonicity of the sign multifunction (see "Sidebar V"), the generalized equation (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 "Sidebar VII" for more details). In this case,

$$\begin{aligned}\mathrm{Prox}_{\tilde{b}\kappa g}(\tilde{a}x_k) &= \tilde{a}x_k - \tilde{b}\kappa\,\mathbf{sgn}(x_k)\min\left\{\frac{\tilde{a}}{\tilde{b}\kappa}|x_k|, 1\right\} \\ &= \begin{cases} 0 & \text{if } |x_k| \le \frac{\tilde{b}\kappa}{\tilde{a}} \\ \tilde{a}x_k - \tilde{b}\kappa\ \mathbf{sgn}(x_k) & \text{otherwise} \end{cases}.\end{aligned} \tag{12}$$

The iteration (11) is an instance of the proximal-point algorithm (see Definition 12 in "Sidebar VII"), vastly studied in the literature of Convex Optimization [54] and recognized as an effective discretization of the subgradient system (3) [55]. It is an explicit form of the algorithm implicitly defined by (9) and falls within the context of Krasnosel'skiĭ-Mann iterations [56]. Thus, its convergence is guaranteed by Krasnosel'skiĭ-Mann Theorem [56] for any proper, convex, lower semicontinuous (LSC) function g, whenever the set of minimizers of g is nonempty [56, Theorem 1.1]. Moreover, since $g(\cdot) = |\cdot|$ implies $0 \in \mathrm{int}\,\partial g(0)$,

Sidebar II: Discrete-Time Sliding-Modes in Control

The most widespread discrete-time algorithms for sliding-mode implementation consist of an emulation method coupled to a forward-Euler discretization (see (6)), with or without a saturation function replacing the set-valued signum function. The latter is essentially an engineering trick, and the necessary parameter tuning to find a correct trade-off between sampling time and saturation width is often uneasy. Forward-Euler has long been known to produce digital chattering [S7, S4], thus high-frequency bang-bang-like controllers (see Fig. 8). Such unwanted dynamical behavior has been deeply analyzed in [22]–[29] for various sliding-mode controllers. This has prompted researchers to design other types of discrete-time sliding-mode controllers, which should be designed directly in the discrete-time setting (should they be "pure" discrete-time systems or resulting from the time-discretization of a continuous-time plant). See, *e.g.*, [S9] for a survey.

Let s_k be the discrete-time sliding variable. Imposing $s_{k+1} = 0$ yields a dead-beat input [S2]. 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 $|s_k| \leq \varepsilon$, $\varepsilon > 0$. Designing control laws that render it invariant and attractive in a finite number of steps is a research topic initiated in [S1, S3, S8, S10], and active until recently, see, *e.g.*, [S5, S6, S15]. It mainly consists of setting $s_{k+1} = f(s_k, x_k, k)$ for some f, then looking for a controller (a reaching law) guaranteeing it. Various reaching conditions have been stated in the related literature, see [48, S11]. Most of the reaching laws use signum functions [S5, S6, S8, S15] and thus share close similarities with explicit Euler schemes. Another branch of analysis was more recently tackled by Levant [S12]–[S14], [49], concerning the discretization of higher-order and homogeneous sliding-mode schemes, with special attention paid to accuracy of differentiators.

It is noteworthy that the first-order controllers proposed in [S4] and [S7] are backward-Euler algorithms (hence similar to the algorithms described in section "3"), with the assumption that the equivalent controller is known, hence ignoring unknown disturbances. A minimum-operator scheme as in (13c) is proposed in [48] considering unknown disturbances (in the framework of so called reaching laws [S8], minimum-operator controllers merely provide another reaching law). However, no unknown disturbances were considered in a general context for backward (implicit) Euler discretization before [31], hence seriously limiting the applicability of such controllers (the link between minimum-operators and backward-Euler methods was made much later in [50]). The key idea in [31] 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 (contrarily to some widely spread conclusions led in an explicit, forward-Euler context [S2]). Mathematical tools like generalized equations, variational inequalities, complementarity problems, maximal monotone operator theory, were introduced in digital sliding-mode control design for the first time in [21, 30, 31, 32, 42, 43]. The topic of this article is precisely to give a broader extent to this point of view.

References

[S1] K. Furuta, Sliding mode control of a discrete system, *Systems and Control Letters*, 14:145-152, 1990.

[S2] S. Hui, S.H. Zak, On discrete-time variable structure sliding mode control, *Systems and Control Letters*, 38:283-288, 1999.

[S3] A. Bartosziewicz, Discrete-time quasi-sliding-mode control strategies, *IEEE Trans. Industrial Electronics*, 45(4):633-637, 1998.

[S4] G. Bartolini, A. Ferrara, V.I. Utkin, Adaptive sliding mode control in discrete-time systems, *Automatica*, 31(5):769-773, 1995.

[S5] S. Chkrabarty, B. Bandyopadhyay, A generalized reaching law for discrete time sliding mode control, *Automatica*, 52:83-86, 2015.

[S6] H. Du, X. Yu, M.Z.Q. Chen, S. Li, Chattering-free discrete-time sliding mode control, *Automatica*, 68:87-91, 2016.

[S7] S. Drakunov, V.I. Utkin, On discrete-times sliding modes, *IFAC Nonlinear Control Systems Design*, Capri, Italy, 273-278, 1989.

[S8] W. Gao, Y. Wang, A. Homaifa, Discrete-time variable structure control systems, *IEEE Trans. Ind. Electon.*, 42(4):117-122, 1995.

Sidebar (cont.)

[S9] K. Furuta, Y. Pan, Discrete-time variable structure control, In: Yu, X., Xu, J.X. (Eds) *Variable Structure Systems: Towards the 21st Century.* Lecture Notes in Control and Information Sciences, vol.274. Springer, Berlin, Heidelberg.

[S10] C. Milosavljevic, General conditions for the existence of a quasi-sliding mode on the switching hyperplane in discrete variable structure systems, *Automatic Remote Control*, 46:307–314, 1985.

[S11] A.J. Koshkouei, A.S.I. Zinober, Sliding mode control of discrete-time systems, *ASME. J. Dyn. Sys., Meas., Control*, 122(4):793–802, 2000.

[S12] M. Livne, A. Levant, Proper discretization of homogeneous differentiators, *Automatica*, 50(8):2007-2014, 2014.

[S13] A. Levant, M. Livne, Weighted homogeneity and robustness of sliding mode control, *Automatica*, 72(10):186-193, 2016.

[S14] A. Levant, M. Livne, D. Lunz, On discretization of high order sliding modes, in J.P. Barbot, L. Fridman, F. Plestan (Eds.) *Recent Trends in Sliding Mode Control*, 177-202, IET, 2016.

[S15] H. Ma, Z. Xiong, Y. Li and Z. Liu, Sliding mode control for uncertain discrete-time systems using an adaptive reaching law, *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(2):722-726, 2021.

the origin of (9) is globally finite-time stable for any $h > 0$ [57] and, contrary to the explicit discretization (7), it exhibits no oscillations at all. Figure 3 shows the cobweb diagram of (9).

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. *Consider the input implicitly defined by* (8) *together with the discrete-time system* (5)*. The following statements are equivalent:*

$$-u_{\mathrm{sv}}(x_k) \in \kappa\,\mathbf{sgn}(x_{k+1}). \tag{13a}$$

$$-u_{\mathrm{sv}}(x_k) = \mathrm{Proj}\left([-\kappa,\kappa];\frac{\tilde{a}}{\tilde{b}}x_k\right). \tag{13b}$$

$$-u_{\mathrm{sv}}(x_k) = \kappa\,\mathbf{sgn}(x_k)\min\left\{\frac{\tilde{a}}{\tilde{b}\kappa}|x_k|,1\right\}. \tag{13c}$$

Proof. The proof uses various results presented in "Sidebar VI", "Sidebar III", "Sidebar V", and "Sidebar VII". It follows from (37) and (5) that (13a) is equivalent to $\tilde{a}x_k + \tilde{b}u_{\mathrm{sv}}(x_k) \in \mathbf{N}_{[-\kappa,\kappa]}\left(-u_{\mathrm{sv}}(x_k)\right)$. Since cones are invariant under multiplication by positive scalars, the latter is equivalent to $\frac{\tilde{a}}{\tilde{b}}x_k \in \left(\mathrm{I_d} + \mathbf{N}_{[-\kappa,\kappa]}\right)\left(-u_{\mathrm{sv}}(x_k)\right)$. Proposition 6 in "Sidebar VI" then establishes the equivalence between (13a) and (13b). The equivalence between (13b) and (13c) follows from direct computations. Indeed,

$$\begin{aligned}\mathrm{Proj}\left([-\kappa,\kappa];\frac{\tilde{a}}{\tilde{b}}x_k\right) &= \begin{cases}\frac{\tilde{a}}{\tilde{b}}x_k & \text{if } \frac{\tilde{a}}{\tilde{b}}|x_k| \le \kappa\\ \kappa\,\mathbf{sgn}(x_k) & \text{otherwise.}\end{cases}\\ &= \mathbf{sgn}(x_k)\min\left\{\frac{\tilde{a}}{\tilde{b}}|x_k|,\kappa\right\}\\ &= \kappa\,\mathbf{sgn}(x_k)\min\left\{\frac{\tilde{a}}{\tilde{b}\kappa}|x_k|,1\right\}.\end{aligned} \tag{14}$$

□

The expressions in (12) and (13) can also be written using complementarity problems and variational inequalities, using equivalent formulations of normal cones (see (31) for the variational formulation, and "Sidebar IX" for the complementarity formulation of a class of nonsmooth systems which are close to the sliding-mode ones). The signum multifunction can also be rewritten using various formalisms, see [58, "Set-Valued Signum Function"]. The $\min\{\cdot,\cdot\}$ function in (13c) and in (12) is a *complementarity function* (or C-function) [59, 60]. The expression (13b) shows that $u_{\mathrm{sv}}(x_k)$ is the solution of the quadratic optimization problem

$$-u_{\mathrm{sv}}(x_k) = \underset{z\in[-\kappa,\kappa]}{\arg\min}\ \frac{1}{2}\left\|z - \frac{\tilde{a}}{\tilde{b}}x_k\right\|^2. \tag{15}$$

This suggests that implicit controllers can be implemented online by solving quadratic programs (QPs) with constraints. It is noteworthy that the explicit form (13c) has been studied in [61, Example 3.1] via Lyapunov methods (see also [48], where the same algorithms were already presented, and [62, 63] for further analysis, where

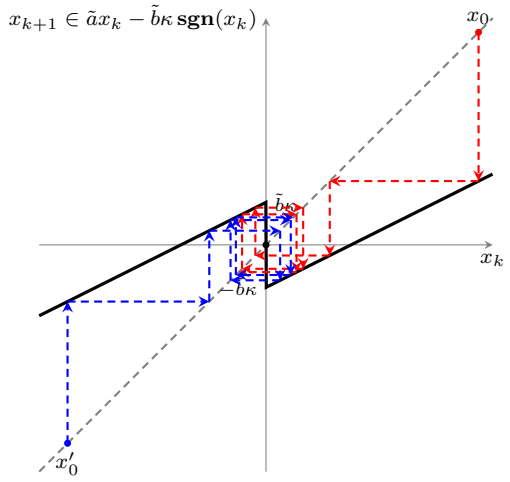

Figure 2: Cobweb diagram of (7) with parameters: $\tilde{a} = 0.5$, $\tilde{b}\kappa = 0.75$ for two different initial conditions. The explicit discretization of the set-valued law (2) yields an iteration with a non-monotone map such that, for almost all initial conditions, the closed-loop shows an oscillatory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

the form (13c) is called a minimum-operator-based input), and that the connection with proximal maps via the implicit Euler method was shown in [50]. It is worth to emphasize that, in the limit as $h \downarrow 0$, the trajectories of both schemes, (7) and (9), converge towards trajectories of the continuous time inclusion (1)-(2). However, the control input (13) 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 because of practical limitations, it is clear that the implicitly defined system (9) is more attractive than the explicit one. Such distinction becomes more important when disturbances are considered.

To close this section, it is also worth to 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 towards zero (making the convergence asymptotic), or the stability of the iteration map, making the origin unstable and producing an oscillatory behavior (see "Sidebar VII").

The following section extends the ideas presented above to the multivariable case, where non-vanishing 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.

2 FIRST-ORDER SLIDING-MODE CONTROL: CONTINUOUS-TIME

In this section we argue that first-order sliding-mode control is a particular case of passivity-based control. This observation leads to a new method for designing sliding-mode controllers and establishes the grounds for the following section, which is about discrete-time sliding-mode control.

Consider the perturbed linear dynamics

$$\dot{x}(t) = Ax(t) + B(u(t) + \delta(t)), \tag{16}$$

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 $\delta(t)$ is uniformly bounded.

2.1 The Classical Approach

In the classical approach to sliding-mode control, the robust stabilization of (16) 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 accomplished by putting the system in regular form and solving a lower-dimensional stabilization problem [51, 52]. To accomplish 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

$$u_{\text{lin}}(x) = -(CB)^{-1}CAx \tag{17}$$

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

$$\dot{s}(t) = v(t) + CB\delta(t). \tag{18}$$

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

$$u_{\text{sv}}(x, t) \in -K\,\mathbf{Sgn}(\sigma(x)), \quad K = K^\top \succ 0, \tag{19}$$

for all $t \in \mathbb{R}$. The term $u_{\text{sv}}(x, t)$ is a selection of the set-valued signum function (see Definition 3 in "Sidebar III", and (30) (40) in "Sidebar VI"). With the controls (17) and (19), the sliding variable evolves according to the differential inclusion

$$\dot{s}(t) \in -K\,\mathbf{Sgn}(s(t)) + CB\delta(t). \tag{20}$$

The time derivative of the Lyapunov function $V_s(s) = \frac{1}{2}s^\top K^{-1}s$ along closed-loop trajectories is

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

The computation of the implicit first-order SMC with a single sliding surface, boils down to projecting a scalar onto a bounded interval: it is therefore extremely simple to implement.

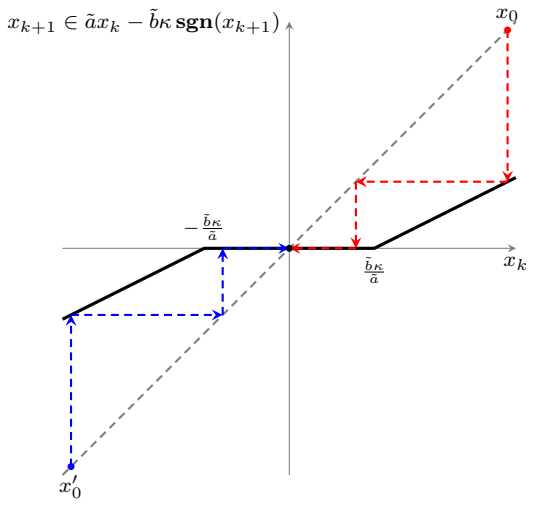

Figure 3: Cobweb diagram of (9) with parameters: $\tilde{a} = 0.5$, $\tilde{b}\kappa = 0.75$. The implicit discretization of the set-valued law (2) yields an iteration with a maximal monotone map such that, for any initial condition, the state converges towards the origin after a finite number of steps.

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, [65, Proposition 11.1.6]). Therefore, there exists some $t_{\min} < +\infty$ after which a sliding mode occurs. That is: $s(t) = 0$ and therefore

$$u_{\mathrm{sv}}(x(t), t) = -CB\delta(t) \tag{21}$$

for all $t \geq t_{\min}$. In other words, the controller compensates exactly for the unknown perturbation. Certainly, this continuous-time property is a kind of miracle of set-valued feedback control, adding to the *miracle of feedback stabilization* [66]. The exactness (equality) in (21) is obviously due to modeling idealization. However, as demonstrated by the implicit discrete-time analysis in (98) and by experimental results, it is the idealization of an *observed behavior*. In particular, insensitivity of the dynamics with respect to K during the sliding-mode is observed experimentally in both differentiation and control when the implicit method is used. This witnesses the fact that (21) is far from being just a mathematical fuss.

2.2 A Passivity-Based Approach

Let us ponder the classical approach through a passivity point-of-view. This will later be used to show the necessity for an implicit discretization.

Remark 1. *Item 1) above implies that the system with output s is feedback equivalent to a passive one. Indeed, a system is feedback passive if, and only if, it is of relative degree $\{1, \ldots, 1\}$, i.e., $\det(CB) \neq 0$ and minimum phase, i.e., $s(t) \equiv 0$ implies $x(t) \to 0$. The linear feedback u_{lin} renders (16) passive with output s and input $v + CB\delta$ (see [68] for details on the unperturbed case). In this regard, first-order sliding-mode control is a particular instance of passivity-based control.*

Besides passivity, a fundamental notion is implicitly used in the above analysis: maximal monotonicity (see "Sidebar V"). Following [69] the change of variable $s' = K^{-\frac{1}{2}}s$ may be performed, transforming (20) 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 differential inclusion can be written as

$$\dot{s}'(t) \in -\partial f(s'(t)) + K^{-\frac{1}{2}}CB\delta(t), \tag{22}$$

where $f(\cdot) = \|K^{\frac{1}{2}}\cdot\|_1$ and where the chain rule was used (see Theorem 3 in "Sidebar V") to compute ∂f, the subdifferential of f (see "Sidebar VI"). The operator ∂f is a maximal monotone operator, f being a convex continuous proper function (see "Sidebar VI"). The differential inclusion has the form (7) so, if the eigenvalues of K are large enough, it satisfies the conditions of Proposition 2 in "Sidebar III" and the origin $s' = 0$ is finite-time stable.

Let us now introduce a different, less-classical design directly based on passivity and monotonicity arguments. 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 [70]

$$AP + PA^\top \prec BB^\top. \tag{23}$$

Similarly to [43, 71], this inequality is exploited to construct the storage function candidate

$$V(x) = \frac{1}{2}x^\top P^{-1}x. \tag{24}$$

Its time-derivative is

$$\dot{V}(x) = \frac{1}{2}x^\top\left(P^{-1}A + A^\top P^{-1}\right)x + x^\top P^{-1}B(u+\delta) \tag{25}$$

Sidebar III: Set-Valued Mappings and Differential Inclusions

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 $\mathbf{F}$ is given by

$$\operatorname{dom} \mathbf{F} = \{\xi \in \mathbb{R}^n \mid \mathbf{F}(\xi) \neq \emptyset\}. \tag{1}$$

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 $\mathbf{F}^{-1}(E) = \bigcup_{y \in E} \mathbf{F}^{-1}(y)$. It will be useful to consider $\operatorname{rint}(\operatorname{dom} \mathbf{F})$, the *relative interior* of $\operatorname{dom} \mathbf{F}$. The *range* of the operator F is given by

$$\operatorname{rge} \mathbf{F} = \{\eta \in \mathbb{R}^m \mid \eta \in \mathbf{F}(\xi) \text{ for some } \xi \in \mathbb{R}^n\} \tag{2}$$

and its *graph* by

$$\operatorname{gph} \mathbf{F} = \{(\xi, \eta) \in \mathbb{R}^n \times \mathbb{R}^m \mid \eta \in \mathbf{F}(\xi)\}. \tag{3}$$

Semicontinuity properties (inner, upper, outer) are fundamental for set-valued maps analysis. The material below is taken from [S16, S17, S18, S19].

Differential inclusions provide a mathematical framework for the study of systems with discontinuous models. Namely, for a discontinuous ODE of the form

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0, \tag{4}$$

where the single-valued map $f : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^n$ 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 (*i.e.*, an absolutely continuous function that satisfies (4) a.e.). Thus, in order to make sense of (4), the Filippov's convexification method considers instead the differential inclusion

$$\dot{x}(t) \in \mathbf{F}(t, x(t)), \quad x(t_0) = x_0, \tag{5}$$

where $\mathbf{F}(t, x) = \bigcap_{\varepsilon > 0} \bigcap_{\mu(\Omega)=0} \overline{co}\, \{f\,(t, (x + \varepsilon \mathcal{B}) \setminus \Omega)\}$, $\overline{co} A$ 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 (5) a.e., are solutions of (4) in the sense of Filippov, see *e.g.*, [17, 18]. On the other hand, let instead of (5) the following differential inclusion be considered

$$\dot{x}(t) \in \mathbf{K}(t, x(t)), \quad x(t_0) = x_0, \tag{6}$$

where $\mathbf{K}(t, x) = \bigcap_{\varepsilon > 0} \overline{co}\{f(t, x + \varepsilon \mathcal{B})\}$. Then, absolutely continuous functions that satisfy (6) a.e. are solutions of (4) in the sense of Krasovskii. In general, Filippov and Krasovskii solutions are different, see, *e.g.*, [64].

Definition 1. *The set-valued map $\mathbf{F} : \mathbb{R}^n \rightrightarrows \mathbb{R}^m$ is* outer semicontinuous *(OSC) if, for any $x \in \operatorname{dom} \mathbf{F}$ and 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 $(x, y) \in \operatorname{gph} \mathbf{F}$.*

Fact 1. *If $\mathbf{F}$ 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.*

Theorem 1. *[S17, Theorem 5.1] Let $\mathbf{F} : \mathbb{R}^n \times [0, T] \rightrightarrows \mathbb{R}^n$ be an OSC map with closed and convex images. Assume there exists a function $c : [0, T] \to \mathbb{R}$, $c \in L_1([0, T])$, such that for all $t \in [0, T]$,*

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

Then, the differential inclusion $\dot{x}(t) \in \mathbf{F}(x(t), t)$ has an absolutely continuous solution on $[0, T]$.

Definition 2. *A solution $x : [0, T] \to \mathbb{R}^n$, $T > 0$ of the differential inclusion $\dot{x}(t) \in \mathbf{F}(x(t), t)$ is said to be a* sliding motion *if $\operatorname{int} \mathbf{F}(x(t), t) \neq \emptyset$ for all $t \in [0, T]$.*

Definition 3. *[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.

Sidebar (cont.)

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. Differential inclusions of the form

$$\dot{s}(t) \in -\partial f(s(t)) + \delta(t) \tag{7}$$

are of interest, where ∂f is the subdifferential of f in the sense of convex analysis (see "Sidebar VI") and $\delta : \mathbb{R} \to \mathbb{R}^m$ is a measurable and uniformly bounded function accounting for external disturbances affecting the system. Indeed they appear in (22) and (36).

Proposition 2. *Consider the perturbed subgradient system* (7) *with* $f : \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ *a proper, convex, and LSC function such that* $\operatorname{dom} f = \mathbb{R}^m$. *Suppose that there exists a fixed* $\varepsilon > 0$ *such that*

$$\delta(t) + \varepsilon \mathcal{B}_2 \subseteq \partial f(0) \tag{8}$$

for all $t \geq 0$ *(see* (38) *in "Sidebar VI"). The origin is finite-time stable.*

Note that the condition (8) implies that ∂f is not a singleton at $s = 0$ and that $\delta(t)$ is uniformly bounded. See Definition 6 for LSC functions.

Proof. The differential inclusion (7) 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 differential inclusion possesses unique absolutely continuous solutions on $\mathbb{R}_+$, for any initial condition [S20, Theorem 3.4]. Consider the Lyapunov function candidate $V(s) = \frac{1}{2}\|s\|^2$. The set-valued derivative [S21] of V along trajectories of (7) 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

$$s^\top \vartheta = f(s) + f^\star(\vartheta), \tag{9}$$

where $f^\star$ denotes the conjugate function to f (see Definition 11 and Theorem 5 in "Sidebar VI"). It follows from (8) 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 7 in "Sidebar VI"),

$$\delta^\top s \leq -\varepsilon \rho^\top s + f(s) - f(0) \tag{10}$$

for any $s \in \mathbb{R}^n$ and for any $\rho \in \mathcal{B}_2$. Since $\partial f(s)$ is compact for any $s \in \mathbb{R}^m$ [S22, Theorem 23.4], $\max \dot{\tilde{V}}(s)$ is well defined. Combining (9) and (10), it follows that

$$\max \dot{\tilde{V}}(s) \leq -\varepsilon \rho^\top s - f^\star(\vartheta^\star) - f(0) \tag{11}$$

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

$$\max \dot{\tilde{V}}(s) \leq -\varepsilon \|s\| = -\varepsilon \sqrt{2V(s)}. \tag{12}$$

For $s = 0$ the inequality follows trivially.

□

References

[S16] J.B. Hiriart Urruty, C. Lemaréchal, Fundamentals of Convex Analysis, Grundlehren Text Editions, Springer-Verlag Berlin Heidelberg, 2001.

[S17] K. Deimling, Multivalued Differential Equations, De Gruyter Series in Nonlinear Analysis and Applications, Berlin, New York, 1992.

[S18] R.T. Rockafellar and R.J.B. Wets, Variational Analysis, Springer-Verlag Berlin Heidelberg, Grundlehren der mathematischen Wissenschaften, vol.317, 1998.

[S19] G.V. Smirnov, Introduction to the Theory of Differential Inclusions, Graduate Studies in Mathematics vol. 41, American Mathematical Society, 2002.

Sidebar (cont.)

[S20] H. Brézis, Opérateurs Maximaux Monotones et Semi-Groupes de Contractions dans les Espaces de Hibert, North-Holland, Mathematics Studies 5,1973.

[S21] A. Bacciotti and F. Ceragioli, Stability and stabilization of discontinuous systems and non-smooth Lyapunov function, *ESAIM: Control, Optimisation and Calculus of Variations*, 4:361-376, 1999.

[S22] R.T. Rockafellar, Convex Analysis, Princeton Landmarks in Mathematics, 1970.

(when clear from context, explicit dependence on t is omitted). The only candidate passive output of relative degree equal to one and associated with the storage function (24) is $s(t) = \sigma(x(t))$ with

$$\sigma(x) = B^\top P^{-1} x, \tag{26}$$

which corresponds to the passivity of the quadruple (A, B, C, D) with $C = B^\top P^{-1}$, $D = 0$, P^{-1} a solution to (16), supply rate $u^\top y$, see "Sidebar IV". This gives the energy balance

$$\dot V(x) = \frac{1}{2} x^\top \left(P^{-1} A + A^\top P^{-1} \right) x + s^\top (u + \delta). \tag{27}$$

Since (23) can be equivalently written as $Q = P^{-1} B B^\top P^{-1} - P^{-1} A - A^\top P^{-1} \succ 0$, and since we have $x^\top P^{-1} B B^\top P^{-1} x = s^\top s$, the energy balance is

$$\dot V(x) = s^\top \left(\frac{1}{2} s + u + \delta \right) - \frac{1}{2} x^\top Q x. \tag{28}$$

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

$$u(t) = u_{\rm lin}(x(t)) + v(t) \tag{29}$$

with

$$u_{\rm lin}(x) = -\frac{1}{2} \sigma(x) \tag{30}$$

can be used, since it gives

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

It can be readily seen that the system is passive with storage function (24), output (26), and new input $v + \delta$. Now let us set $v(t) = u_{\rm sv}(x(t), t)$ with

$$u_{\rm sv}(x, t) \in -K \, \mathbf{Sgn}(Ks), \quad K = K^\top \succ 0, \tag{32}$$

for all $t \in \mathbb{R}$. Substituting this control law in (31) gives

$$\begin{aligned} \dot V(x) &= -s^\top K \left(\mathbf{Sgn}(Ks) - K^{-1} \delta \right) - \frac{1}{2} x^\top Q x \\ &\leq -\|Ks\|_1 (1 - \|K^{-1} \delta\|_\infty) - \frac{1}{2} x^\top Q x, \end{aligned} \tag{33}$$

where, as above, 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 $\delta(t)$. Let us now prove that $s(t)$ converges to zero in finite time. The sliding variable evolves according to

$$\dot s \in B^\top P^{-1} \left(\left(A - \frac{1}{2} B B^\top P^{-1} \right) x - BK \, \mathbf{Sgn}(Ks) + B\delta \right). \tag{34}$$

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

$$\dot V_s(s) = -s^\top \left(K \, \mathbf{Sgn}(Ks) - \delta - Lx \right), \tag{35}$$

where $L = (B^\top P^{-1} B)^{-1} B^\top P^{-1} \left(A - \frac{1}{2} B B^\top P^{-1} \right)$. In (34), the state $x(t)$ can be considered as an exogenous signal, uniformly bounded according to the above 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 suitable K. Alternatively, the dynamics (34) can be analyzed within the framework of differential inclusions with maximal monotone right-hand sides, similarly to (22). To reveal the maximal monotone structure in (34), let us perform the change of variables $s' = (B^\top P^{-1} B)^{-\frac{1}{2}} s$. The following is obtained:

$$\dot s' \in -\partial g(s') + (B^\top P^{-1} B)^{\frac{1}{2}} (\delta + Lx) \tag{36}$$

with $g(\cdot) = \|\cdot\|_1 \circ K (B^\top P^{-1} B)^{\frac{1}{2}} (\cdot)$ a proper, convex, continuous function. Again, the differential inclusion (36) has the structure (7) and, if the eigenvalues of K are large enough, it satisfies the conditions of Proposition 2 in "Sidebar III", 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 3 in "Sidebar III") which compensates exactly for the disturbance $(B^\top P^{-1} B)^{\frac{1}{2}} (\delta(t) + Lx(t))$ in (36). Equivalently, consider (32) and note that there is a selection $u_{\rm sv}(x, t)$ in $K \, \mathbf{Sgn}(0)$ which compensates exactly for the equivalent disturbance in (34). That is, similarly to (21),

$$\begin{aligned} &u_{\rm sv}(x(t), t) = \\ &-(B^\top P^{-1} B)^{-1} B^\top P^{-1} \left(A - \frac{1}{2} B B^\top P^{-1} \right) x(t) - \delta(t). \end{aligned} \tag{37}$$

Let us stress once again that neither (21) nor (37) mean that the disturbance $\delta(t)$ is known. Finally, mark that

Sidebar IV: Passive Systems

Consider a linear time-invariant system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{13}$$

with state $x(t) \in \mathbb{R}^n$, input $u(t) \in \mathbb{R}^m$, and output $y(t) \in \mathbb{R}^m$. The system is passive in Willems' sense [S23, S24] if there exists a symmetric matrix $P \in \mathbb{R}^{n\times n}$, $P \succcurlyeq 0$ such that the *storage function*

$$V(x) = \frac{1}{2} x^\top P x \tag{14}$$

satisfies the *dissipation inequality*

$$V(x(t_1)) - V(x(t_0)) \leq \int_{t_0}^{t_1} u^\top(t) y(t) \mathrm{d}t \tag{15}$$

for all $t_1 \geq t_0$ along the continuous-time system trajectories. The system is passive if and only if the the LMI

$$M_{\text{cont}} := \begin{pmatrix} -A^\top P - PA & -PB + C^\top \\ -B^\top P + C & D + D^\top \end{pmatrix} \succcurlyeq 0 \tag{16}$$

has a positive-definite solution $P = P^\top$ [67]. Indeed, if the inequality (16) holds, then

$$V(x(t_1)) - V(x(t_0)) = -\frac{1}{2} \int_{t_0}^{t_1} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix}^\top M_{\text{c}} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \mathrm{d}t + \int_{t_0}^{t_1} u^\top(t) y(t) \mathrm{d}t. \tag{17}$$

The LMI in (16) implies $-A^\top P - PA \succcurlyeq 0$, $D + D^\top \succcurlyeq 0$, and $\ker(D + D^\top) \subseteq \ker(PB - C^\top)$. The latter further implies that $PB = C^\top$ when $D + D^\top = 0$ [65, Corollary 10.2.3, Proposition 10.2.5]. Similarly, a discrete-time system

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k \end{aligned} \tag{18}$$

is passive if there exists a storage function (14) such that

$$V(x_{k+1}) - V(x_k) \leq u_k^\top y_k \tag{19}$$

holds for all $k \geq 0$ along the discrete-time system trajectories. The system is passive if and only if the LMI

$$M_{\text{disc}} := \begin{pmatrix} -A^\top PA + P & -A^\top PB + C^\top \\ -B^\top PA + C & D + D^\top - B^\top PB \end{pmatrix} \succcurlyeq 0 \tag{20}$$

possesses a solution $P = P^\top \succcurlyeq 0$, in which case we have

$$V(x_{k+1}) - V(x_k) = -\frac{1}{2} \begin{pmatrix} x_k \\ u_k \end{pmatrix}^\top M_{\text{disc}} \begin{pmatrix} x_k \\ u_k \end{pmatrix} + u_k^\top y_k. \tag{21}$$

The LMI in (20) implies $-A^\top PA + P \succcurlyeq 0$, $D + D^\top \succcurlyeq B^\top PB$, and $\ker(D + D^\top - B^\top PB) \subseteq \ker(-A^\top PB + C^\top)$ (hence $C = B^\top PA$ if $D + D^\top = B^\top PB$),.

If (13) or (18) are passive, we say that the quadruple (A, B, C, D) is passive. The product $u^\top y$ is the *supply rate* of the system. *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 $V(x_1, x_2) = \frac{1}{2}(x_1 - x_2)^\top P (x_1 - x_2)$.

The celebrated Kalman-Yakubovich-Popov Lemma allows us to establish relationships between positive-real transfer functions and passive systems as characterized by the above LMIs [67, Chapter 3].

Sidebar (cont.)

References

[S23] J.C. Willems, Dissipative dynamical systems, part I: General theory, *Arch. Ration. Mech. Anal.*, 45(5):21-351, 1972, doi: 10.1007/BF00276493.

[S24] J.C. Willems, Dissipative dynamical systems, part II: Linear systems with quadratic supply rates, *Arch. Ration. Mech. Anal.*, 45(5):352-393, 1972.

the well-posedness of (36) holds under mild measurability conditions on the perturbation, and from the analysis of the closed-loop dynamics

$$\dot{x}(t) \in Ax(t)+B\left(-\frac{1}{2}\sigma(x(t)) - K\mathbf{Sgn}(\sigma(x(t)))\right)+B\delta(t), \tag{38}$$

using the fact that the set-valued right-hand side has compact and convex images, and that it is outer semicontinous. It is then inferred from Theorem 1 (see "Sidebar III") that (38) has absolutely continuous solutions.

2.3 Recapitulation

Classical sliding-mode control 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 (34) that $u_{\rm sv}$ in (32) is selected as

$$u_{\rm sv}(x,t) = -Lx - \delta(t). \tag{39}$$

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 (17) and (19) (see (21)). Incidentally, the differential inclusions in (22) and (36) 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 $V(s'_1, s'_2) = \frac{1}{2}(s'_1 - s'_2)^\top(s'_1 - s'_2)$ and output $N^\top s'$ [67].

It is worth stressing that (32) can be written as $u_{\rm sv}(x) \in -\partial g(s)$ with $g(s) = \|Ks\|_p$ and $p = 1$. Other norms can be used. In general:

$$\partial\|x\|_p = \frac{1}{\|x\|_p^{p-1}}\begin{pmatrix} |x_1|^{p-1}\,\mathbf{sgn}(x_1) \\ \vdots \\ |x_n|^{p-1}\,\mathbf{sgn}(x_n)\end{pmatrix} \tag{40}$$

if $x \neq 0$ and $\partial\|x\|_p = \mathcal{B}_{p^*}$ if $x = 0$, see (37) in "Sidebar VI". For example, the norm $p = 2$ yields the *unit control*

$$u_{\rm sv}(x,t) \in \begin{cases} -\frac{1}{\|Ks\|_2}K^2 s & \text{if } s \neq 0 \\ -K\mathcal{B}_2 & \text{otherwise}\end{cases}. \tag{41}$$

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 ODE into a specific differential inclusion. In fact, the above closed-loop systems are introduced directly in a set-valued setting (see Figure 1 and Theorem 1 in "Sidebar III").

3 FIRST-ORDER SLIDING-MODE CONTROL: DISCRETE-TIME

In this section we exploit the passive properties of the sliding variable to derive a new design method for discrete-time sliding-mode control. We also show the necessity of defining any discrete-time sliding-mode control implicitly. Consider a sampled-data model of (16),

$$x_{k+1} = \tilde{A}x_k + \tilde{B}(u_k + \tilde{\delta}_k), \tag{42}$$

where $\tilde{A} \in \mathbb{R}^{n\times n}$, $\tilde{B} \in \mathbb{R}^{n\times m}$, and it is assumed that $\operatorname{rank}\tilde{B} = m$. In the spirit of Section "2.2", an output s_k is constructed for which the system is feedback passive. In contrast with the continuous-time scenario, a necessary condition for a system to be feedback passive is that it has relative degree $\{0, 0, \ldots, 0\}$ [72, 73], which ultimately results in the necessity for implicitly defined control laws.

3.1 A Discrete-Time Passivity-Based Approach

The discrete-time counterpart of (23) is

$$\tilde{A}\tilde{P}\tilde{A}^\top - \tilde{P} \prec \tilde{B}\tilde{B}^\top. \tag{43}$$

That is, the pair $(\tilde{A}, \tilde{B})$ is stabilizable if, and only if, inequality (43) holds for some $\tilde{P} = \tilde{P}^\top \succ 0$ [70]. The forward difference $V(x_{k+1}) - V(x_k)$ of the storage function

Sidebar V: 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 convergence of numerical schemes via fixed-point methods.

Definition 4. *A set-valued operator* $\mathbf{M} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *is said to be* monotone *(resp. strongly monotone) if, for any two pairs* $(x_i, y_i) \in \operatorname{gph} \mathbf{M}$*,* $i = 1, 2$*,*

$$\langle x_1 - x_2, y_1 - y_2 \rangle \geq 0 \quad \textit{(resp, } \geq \alpha \|x_1 - x_2\|^2, \ \alpha > 0\textit{).} \tag{22}$$

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.*

The Euclidean scalar product $\langle \cdot, \cdot \rangle$ is used in (22). 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 $M = M^\top \succ 0$.

Fact 2. *Let* $\mathbf{M} : \mathbb{R} \rightrightarrows \mathbb{R}$ *be maximal monotone and let* $x \in \mathbb{R}$*. Then* $\mathbf{M}(x)$ *is closed and convex.*

The following follows from Fact 1 in "Sidebar III".

Fact 3. *A maximal monotone operator* $\mathbf{M}$ *has a closed graph, hence it is OSC.*

Maximal monotone maps can be considered as OSC maps with an additional monotonicity (or incremental passivity) property. 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 $\operatorname{rint}(\operatorname{dom} \mathbf{M}_1) \cap \operatorname{rint}(\operatorname{dom} \mathbf{M}_2) \neq \emptyset$.
3. $\mathbf{T}_i(x) = L^\top \mathbf{M}_i(Lx)$ are maximal monotone whenever $\operatorname{rge}(L) \cap \operatorname{rint}(\operatorname{dom} \mathbf{M}_i) \neq \emptyset$.
4. The mapping $x \mapsto -Ax + B(D + \mathbf{M}^{-1})^{-1}(Cx)$ is maximal monotone whenever $\operatorname{Im}(C) \cap \operatorname{rint}(\operatorname{Im}(D + \mathbf{M}^{-1})) \neq \emptyset$, $\mathbf{M}$ is maximal monotone, and the quadruple (A, B, C, D) is passive.

Items 1, 2 and 3 are classical [S18]. Inverse mappings are defined in "Sidebar III". Item 4 follows from [S26], building upon results in [S27] and [69] (there, monotonicity is understood with the weighted inner product defined by the solution $P = P^\top \succ 0$ of the linear matrix inequality 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 $\mathbf{M}$ preserves the maximal monotonicity. This is a kind of extension of [S18, Example 12.45 (b)]. One of the most important properties of maximal monotonicity is stated in the following theorem.

Theorem 2. *Let* $\mathbf{M} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *be a monotone operator. Then,* $\mathbf{M}$ *is maximal monotone if, and only if, for any* $\gamma > 0$*,* $\operatorname{rge}(\mathrm{I_d} + \gamma \, \mathbf{M}) = \mathbb{R}^n$*.*

Given a maximal monotone operator $\mathbf{M}$ let us construct

$$\mathcal{J}_{\gamma \, \mathbf{M}} := (\mathrm{I_d} + \gamma \, \mathbf{M})^{-1}, \tag{23}$$

the *resolvent* of $\mathbf{M}$ of index $\gamma > 0$. Thus, an alternative statement of Theorem 2 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

$$y \in x + \gamma \, \mathbf{M}(x) \tag{24}$$

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, that the solutions to (24) are unique.
The archetypal example of a maximal monotone map is the convex subdifferential of a proper, convex, and LSC function (see Definitions 6 and 7 in "Sidebar VI").

Theorem 3. *[S28, Theorem 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.*

Sidebar (cont.)

It follows from the previous theorem that, if f is proper, convex, and LSC, the generalized equation

$$y \in x + \gamma \partial f(x) \tag{25}$$

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 $\mathbf{M}$ of index γ,

$$\mathcal{Y}_{\gamma\,\mathbf{M}} := \frac{1}{\gamma}(\mathrm{I_d} - \mathcal{J}_{\gamma\,\mathbf{M}}), \tag{26}$$

which is Lipschitz continuous with Lipschitz constant equal to $\frac{1}{\gamma}$. It is known that

$$\mathcal{Y}_{\gamma\,\mathbf{M}} = (\gamma\,\mathrm{I_d} + \mathbf{M}^{-1})^{-1}, \tag{27}$$

i.e., $\mathcal{Y}_{\gamma\,\mathbf{M}}(x) = \mathcal{J}_{\frac{1}{\gamma}\,\mathbf{M}^{-1}}(\frac{1}{\gamma}x)$ [S18, Lemma 12.14]. Moreover, for any $x \in \mathbb{R}^n$, we have [S29, Theorem 2, Chapter 3]

$$(\mathcal{J}_{\gamma\,\mathbf{M}}(x), \mathcal{Y}_{\gamma\,\mathbf{M}}(x)) \in \mathrm{gph}\,\mathbf{M}\,. \tag{28}$$

References

[S25] G.J. Minty, Monotone Networks, *Proc. R. Soc. Lond. A* 257:194–212, 1960.

[S26] M.K. Camlibel, J.M. Schumacher, Linear passive systems and maximal monotone mappings, *Mathematical Programming*, Ser. B, 157(2):397–420, 2016.

[S27] B. Brogliato, D. Goeleven, Well-posedness, stability and invariance results for a class of multivalued Lur'e dynamical systems, *Nonlinear Analysis: Theory, Methods and Applications*, 74(1):195-212, 2011.

[S28] H.H. Bauschke, P.L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, Canadian Mathematical Society, Société mathématique du Canada, Springer Science+Business Media, 2011.

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

$V(x_k) = \frac{1}{2} x_k^\top \tilde{P} x_k$ is

$$\Delta V(x_k) = \frac{1}{2} \left(\tilde{A}x_k + \tilde{B}(u_k + \tilde{\delta}_k) \right)^\top \tilde{P}^{-1} \left(\tilde{A}x_k + \tilde{B}(u_k + \tilde{\delta}_k) \right) - \frac{1}{2} x_k^\top \tilde{P}^{-1} x_k. \quad (44)$$

By isolating the terms that do not depend on the inputs, the counterpart of (25) is obtained as

$$\begin{aligned} \Delta V(x_k) = & \frac{1}{2} x_k^\top \left(\tilde{A}^\top \tilde{P}^{-1} \tilde{A} - \tilde{P}^{-1} \right) x_k \\ & + (u_k + \tilde{\delta}_k)^\top \tilde{B}^\top \tilde{P}^{-1} \left(\tilde{A}x_k + \tilde{B}(u_k + \tilde{\delta}_k) \right) \\ & - \frac{1}{2} (u_k + \tilde{\delta}_k)^\top \tilde{B}^\top \tilde{P}^{-1} \tilde{B} (u_k + \tilde{\delta}_k). \end{aligned} \quad (45)$$

The passive output candidate

$$s_{k+1} := \sigma(x_k, u_k + \tilde{\delta}_k) \quad (46)$$

with

$$\sigma(x, w) = \tilde{B}^\top \tilde{P}^{-1} \left(\tilde{A}x + \tilde{B}w \right) \quad (47)$$

is unveiled and the energy balance becomes

$$\begin{aligned} \Delta V(x_k) = & \frac{1}{2} x_k^\top \left(\tilde{A}^\top \tilde{P}^{-1} \tilde{A} - \tilde{P}^{-1} \right) x_k + s_{k+1}^\top (u_k + \tilde{\delta}_k) \\ & - \frac{1}{2} (u_k + \tilde{\delta}_k)^\top \tilde{B}^\top \tilde{P}^{-1} \tilde{B} (u_k + \tilde{\delta}_k), \end{aligned} \quad (48)$$

which corresponds to the dissipation equality (20) 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 to the LMI (20), supply rate $u^\top y$ (see "Sidebar IV").

Remark 2. *The equations* (46), (47) *have an implicit (backward-Euler) flavour since*

$$s_{k+1} = \tilde{B}^\top \tilde{P}^{-1} x_{k+1}. \quad (49)$$

Note that (47) is the counterpart of (26). In this context, the key difference is that discrete-time passive systems have relative degree zero whereas continuous-time passive systems have relative degree either one or zero [67].

Remark 3. *Neither $s_{k+1} = B^\top P^{-1} x_k$ nor even $s_{k+1} = B^\top P^{-1} x_{k+1}$ were used, as it would be the case in a pure emulation method. The sliding variable (hence the sliding surface) is designed in the discrete-time context to preserve passivity. A similar remark holds for the linear controller (30), which can be discretized in various ways (by emulation, $u_{\text{lin}}(x_k) = -\frac{1}{2} B^\top P^{-1} x_k$, or directly in discrete time, as in (55) and (56) below). Not all discretized controllers share similar closed-loop properties [21]. The same comments hold for the case of the classical first-order control in (91) and (92) below. See also Section "5.2".*

The following theorem shows that stabilizability implies feedback equivalence to passivity.

Theorem 4. *Suppose that system* (42) *is stabilizable. Consider the output* (46) *with $\tilde{P} = \tilde{P}^\top > 0$ a solution of* (43)*. The control*

$$u_k = u_{\text{lin}}(x_k) + v_k, \quad (50)$$

with

$$u_{\text{lin}}(x_k) = -\frac{1}{2} \sigma(x_k, u_{\text{lin}}(x_k)) \quad (51)$$

renders the system passive with storage function (24)*, output s_{k+1} given by* (46)*, and new input $v_k + \tilde{\delta}_k$.*

Notice that (51) is the counterpart of (30). The proof of this theorem requires the following lemma, which is inspired by [70, Chapter 6].

Lemma 1. *The inequality* (43) *implies that*

$$\tilde{A}^\top \left(\tilde{P} + \frac{1}{2} \tilde{B}\tilde{B}^\top \right)^{-1} \tilde{P} \left(\tilde{A} + \frac{1}{2} \tilde{B}\tilde{B}^\top \right)^{-1} \tilde{A} - \tilde{P}^{-1} \prec 0. \quad (52)$$

Proof. Using Schur complements, it can be seen that (43) is equivalent to

$$\begin{pmatrix} \tilde{P} & \tilde{P}\tilde{A}^\top \\ \tilde{A}\tilde{P} & \tilde{P} + \tilde{B}\tilde{B}^\top \end{pmatrix} \succ 0, \quad (53)$$

which implies

$$\begin{pmatrix} \tilde{P} & \tilde{P}\tilde{A}^\top \\ \tilde{A}\tilde{P} & \tilde{P} + \tilde{B}\tilde{B}^\top + \frac{1}{4}\tilde{B}\tilde{B}^\top \tilde{P}^{-1} \tilde{B}\tilde{B}^\top \end{pmatrix} \succ 0, \quad (54)$$

that is, $\tilde{A}^\top \left(\tilde{P} + \tilde{B}\tilde{B}^\top + \frac{1}{4}\tilde{B}\tilde{B}^\top \tilde{P}^{-1} \tilde{B}\tilde{B}^\top \right)^{-1} \tilde{A} - \tilde{P}^{-1} \prec 0$. Inequality (52) is recovered by noting that $\tilde{P} + \tilde{B}\tilde{B}^\top + \frac{1}{4}\tilde{B}\tilde{B}^\top \tilde{P}^{-1} \tilde{B}\tilde{B}^\top = \left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top \right) \tilde{P}^{-1} \left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top \right)$. □

Proof of Theorem 4. According to (47), the control (51) is implicitly defined by the equation

$$u_{\text{lin}}(x_k) = -\frac{1}{2} \tilde{B}^\top \tilde{P}^{-1} \left(\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k) \right). \quad (55)$$

It can be solved explicitly as

$$u_{\text{lin}}(x_k) = -\left(2I + \tilde{B}^\top \tilde{P}^{-1} \tilde{B} \right)^{-1} \tilde{B}^\top \tilde{P}^{-1} \tilde{A}x_k, \quad (56)$$

which gives

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

By the Matrix Inversion Lemma [65, Corollary 3.9.8],

$$\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k) = \tilde{P} \left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top \right)^{-1} \tilde{A}x_k. \quad (57)$$

It now follows from Lemma 1 that

$$\left(\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k)\right)^\top \tilde{P}^{-1}\left(\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k)\right) - x_k^\top \tilde{P}^{-1}x_k = -x_k^\top \tilde{Q}x_k \quad (58)$$

for some positive definite matrix $\tilde{Q}$. The dissipation equality (48) can be written as

$$\Delta V(x_k) = s_{k+1}^\top(v_k+\tilde{\delta}_k) - \frac{1}{2}(v_k+\tilde{\delta}_k)^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}(v_k+\tilde{\delta}_k) - \frac{1}{2}x_k^\top \tilde{Q}x_k, \quad (59)$$

which is the counterpart of (31). This shows passivity with the required ingredients. □

3.1.1 Stability Analysis

It follows from (59) that the origin is exponentially stable if v_k is such that

$$s_{k+1}^\top(v_k + \tilde{\delta}_k) \leq 0. \quad (60)$$

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

$$v_k = u_{\text{sv}}(x_k) \quad (61)$$

with $u_{\text{sv}}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(\tilde{K}s_{k+1})$. Such control does satisfy (60) but, unfortunately, the passive output (46) is not available to the controller, because it depends directly on the unknown perturbation. To avoid this problem let us follow [31] and define the nominal output

$$\tilde{s}_{k+1} := \sigma(x_k, u_k) \quad (62)$$

instead of (46). In what follows it is shown that, the full state of the closed-loop (42), (50)-(51), with the proposed nominal output $\tilde{s}_{k+1}$ is ultimately bounded.

Proposition 3. *Consider the perturbed dynamics* (42) *with feedback law given by* (50)-(51) *where*

$$v_k = u_{\text{sv}}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1}). \quad (63)$$

where $\tilde{s}_{k+1}$ is given by (62)*. 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 (59) becomes

$$\Delta V(x_k) = \left(\tilde{s}_{k+1} + \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k\right)^\top \left(v_k + \tilde{\delta}_k\right) - \frac{1}{2}(v_k+\tilde{\delta}_k)^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}(v_k+\tilde{\delta}_k) - \frac{1}{2}x_k^\top \tilde{Q}x_k. \quad (64)$$

By developing the right-hand side one obtains

$$\Delta V(x_k) = \tilde{s}_{k+1}^\top\left(v_k + \tilde{\delta}_k\right) + \delta_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}v_k + \delta_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k - \frac{1}{2}v_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}v_k - v_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k - \frac{1}{2}\tilde{\delta}_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k - \frac{1}{2}x_k^\top \tilde{Q}x_k, \quad (65)$$

which can be simplified as

$$\Delta V(x_k) = \tilde{s}_{k+1}^\top(v_k + \delta_k) + \frac{1}{2}\tilde{\delta}_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k - \frac{1}{2}v_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}v_k - \frac{1}{2}x_k^\top \tilde{Q}x_k. \quad (66)$$

Define the set-valued control

$$u_{\text{sv}}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1}), \quad \tilde{K} = \tilde{K}^\top \succ 0, \quad (67)$$

as a discrete-time counterpart of (32). It gives the inequality

$$\Delta V(x_k) \leq -\|\tilde{K}\tilde{s}_{k+1}\|_1(1-\|\tilde{K}^{-1}\tilde{\delta}_k\|_\infty) - \frac{1}{2}v_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}v_k - \frac{1}{2}x_k^\top \tilde{Q}x_k + \frac{1}{2}\tilde{\delta}_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k. \quad (68)$$

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 $k \geq 0$. 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 $\tilde{\delta}_k$. □

3.1.2 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 it reaches zero, it stays at zero in a *discrete-time sliding-motion*. To that end, notice that the nominal output satisfies

$$\tilde{s}_{k+1} \in \tilde{B}^\top \tilde{P}^{-1}\left(\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k) - \tilde{B}\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1})\right) \quad (69)$$

or, using (57),

$$\tilde{s}_{k+1} \in \tilde{B}^\top\left(\left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top\right)^{-1}\tilde{A}x_k - \tilde{P}^{-1}\tilde{B}\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1})\right). \quad (70)$$

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

$$\tilde{s}'_{k+1} \in -(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{\frac{1}{2}}\tilde{K}\,\mathbf{Sgn}\left(\tilde{K}(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{\frac{1}{2}}\tilde{s}'_{k+1}\right) + \tilde{L}x_k. \quad (71)$$

The generalized equation (71) shares the same structure as the differential inclusions (22) and (36). Namely, using once again the chain rule of convex analysis (see Fact 8 in "Sidebar VI") one can see that

$$\tilde{s}'_{k+1} \in -\partial\tilde{g}(\tilde{s}'_{k+1}) + \tilde{L}x_k \quad (72)$$

with $\tilde{g}(\cdot) = \|\cdot\|_1 \circ \tilde{K}(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{\frac{1}{2}}(\cdot)$ (formally the same function as $g(\cdot)$ in (36)). This difference inclusion, which is of the implicit type since $\tilde{s}'_{k+1}$ appears on both sides, is equivalently rewritten as the equality

$$\tilde{s}'_{k+1} = (\mathrm{I_d} + \partial\tilde{g})^{-1}(\tilde{L}x_k). \quad (73)$$

Let us stop and explain the meaning of the right-hand side of (73) using the material in the "Sidebar VII", "Sidebar V", and "Sidebar VI" sidebars. Unquestionably, 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 (73) is in fact a proximal operator,

$$\tilde{s}'_{k+1} = \mathrm{Prox}_{\tilde{g}}(\tilde{L}x_k) \tag{74}$$

(see (50) in "Sidebar VII").

Proposition 4. *Consider the virtual sliding variable* (74)*. It follows that* $\tilde{s}'_{k+1} = 0$ *if, and only if,*

$$\tilde{K}^{-1}(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-\frac{1}{2}}\tilde{L}x_k \in \mathcal{B}_\infty, \tag{75}$$

where the unit ball $\mathcal{B}_\infty$ *is defined in (38).*

Proof. The claim follows from (54) in "Sidebar VII". Indeed $\mathrm{zero}\,\mathrm{Prox}_{\tilde{g}} = \partial\tilde{g}(0) = (\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{\frac{1}{2}}\tilde{K}\mathcal{B}_\infty$, where the equality follows from Fact 8 and from (40) in "Sidebar VI", and the set of zeroes is defined in "Sidebar VII". Thus, it follows from (75) 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 the inclusion (75) 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 (69) is the counterpart of the differential inclusion (34), while (71) or (72) are the counterparts of (36). The equality in (73) is the explicit form of the implicit (backward-Euler) algorithm (72).*

Following [21] we define the discrete-time sliding motion in terms of the nominal variable $\tilde{s}_{k+1}$.

Definition 5. *The closed-loop system* (42), (50), (51), (63) *reaches a discrete-time sliding motion whenever there is a finite* $k^* \in \mathbb{N}$ *such that*

$$\tilde{s}_k = 0, \tag{76}$$

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

Proposition 5. *During the discrete-time sliding mode, the controller* $u_k = u_{\mathrm{lin}}(x_k) + u_{\mathrm{sv}}(x_k)$ *satisfies*

$$u_k = -\tilde{\delta}_{k-1} - (\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-1}\tilde{B}\tilde{P}^{-1}(\tilde{A} - I)x_k. \tag{77}$$

Proof. It follows from (69) and (49) that

$$\tilde{s}_{k+1} = s_k + \tilde{B}^\top \tilde{P}^{-1}(\tilde{A} - I)x_k + \tilde{B}^\top \tilde{P}^{-1}\tilde{B}u_k, \tag{78}$$

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

$$s_{k+1} = \tilde{s}_{k+1} + \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k. \tag{79}$$

Thus, the substitution of (79) back into (78), with a one-step delay, leads us to the expression

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

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 (39) 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.

3.1.3 Control Calculation and Input–Sliding-Variable Conjugacy

The feedback control and future sliding-variable are implicitly defined by (62), (67). Such expressions are useful for stability analysis of closed-loop trajectories, but not for implementation. For the latter, an explicit expression is needed. To obtain such expression, the duality between $\tilde{s}_{k+1}$ and $-u_{\mathrm{sv}}(x_k)$ can be exploited. Proposition 7 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 expressions may result simpler than others.

Proposition 7. *Let* $\Lambda_1 \in \mathbb{R}^{m\times n}$, $\Lambda_2 = \Lambda_2^\top \in \mathbb{R}^{m\times m}$ *be non-singular, and let* $\mathbf{M} : \mathbb{R}^m \rightrightarrows \mathbb{R}^m$ *be maximal monotone. Then, the control input and sliding output expressions shown in Figure 4 are equivalent.*

Proof. The equivalence between the implicit expressions i) and ii) is immediate. It suffices to invert Λ_1 and $\mathbf{M}$. Let us show the equivalence between i) and iii). To that end, let $\tilde{\mathbf{M}} = \Lambda_2^{\frac{1}{2}} \circ \mathbf{M} \circ \Lambda_2^{\frac{1}{2}}$. The following inclusions are equivalent:

$$\begin{aligned}
&\tilde{s}_{k+1} \in \Lambda_1 x_k - \Lambda_2\, \mathbf{M}(\tilde{s}_{k+1}) \\
&\tilde{s}_{k+1} \in \left(\Lambda_2^{-1} + \mathbf{M}\right)^{-1}\left(\Lambda_2^{-1}\Lambda_1 x_k\right) \\
&\tilde{s}_{k+1} \in \left(\Lambda_2^{-\frac{1}{2}} \circ \left(\mathrm{I_d} + \tilde{\mathbf{M}}\right) \circ \Lambda_2^{-\frac{1}{2}}\right)^{-1}\left(\Lambda_2^{-1}\Lambda_1 x_k\right) \\
&\tilde{s}_{k+1} \in \left(\Lambda_2^{\frac{1}{2}} \circ \left(\mathrm{I_d} + \tilde{\mathbf{M}}\right)^{-1} \circ \Lambda_2^{-\frac{1}{2}}\right)\left(\Lambda_1 x_k\right) \\
&\tilde{s}_{k+1} = \Lambda_2^{\frac{1}{2}}\, \mathcal{J}_{\tilde{\mathbf{M}}}\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right),
\end{aligned} \tag{81}$$

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

$$\begin{aligned}
-u_{\mathrm{sv}}(x_k) &= \Lambda_2^{-1}\left(\Lambda_1 x_k - \tilde{s}_{k+1}\right) \\
&= \Lambda_2^{-\frac{1}{2}}\left(\mathrm{I_d} - \mathcal{J}_{\tilde{\mathbf{M}}}\right)\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right) \\
&= \Lambda_2^{-\frac{1}{2}}\, \mathcal{Y}_{\tilde{\mathbf{M}}}\left(\Lambda_2^{-\frac{1}{2}}\Lambda_1 x_k\right).
\end{aligned} \tag{82}$$

Sidebar VI: Convex Analysis Tools

The following results and definitions are taken from classical references [S16, S18, S22, S28, 60]. Most of the tools below extend to non-convex sets and functions. However, it is essentially the convex case that is of interest to us in this article. 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 6. *A proper convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is* lower semicontinuous *(LSC) if its epigraph is closed.*

Definition 7. *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*

$$\partial f(x) := \{\eta \in \mathbb{R}^n \mid \langle \eta, \xi - x\rangle \leq f(\xi) - f(x) \text{ for all } \xi \in \operatorname{dom} f\}. \tag{29}$$

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 $\kappa > 0$. Then, $\partial f(x) = \kappa\, \mathbf{sgn}(x)$ with

$$\mathbf{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}. \tag{30}$$

Definition 8. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a closed convex set. Its* normal cone *is the set-valued mapping $\mathbf{N}_{\mathcal{C}} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ defined by*

$$\mathbf{N}_{\mathcal{C}}(x) = \{\eta \in \mathbb{R}^n \mid \langle \eta, \xi - x\rangle \leq 0 \text{ for all } \xi \in \mathcal{C}\} \tag{31}$$

if $x \in \mathcal{C}$ and $\mathbf{N}_K(\mathcal{C}) = \emptyset$ if $x \notin \mathcal{C}$.

The normal cone to a closed convex set is convex-valued. Suppose that $\mathcal{C}$ is finitely represented, that is, $\mathcal{C} = \{\xi \in \mathbb{R}^n \mid g(\xi) \geq 0\}$ for some differentiable function $g : \mathbb{R}^n \to \mathbb{R}^m$. Assume further that the Mangasarian-Fromovitz constraint qualification holds [S18]. Then, $\mathbf{N}_{\mathcal{C}}(x)$ is generated by the outward normals $\nabla g_i(x)$ to $\mathcal{C}$ at the active constraints $g_i(x) = 0$, $i \in \{1, \ldots, m\}$. In other words, $\mathbf{N}_{\mathcal{C}}(x) = \{\eta \in \mathbb{R}^n \mid \eta = -\nabla g(x)\lambda,\ 0 \leq \lambda \perp g(x) \geq 0\}$. 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 (31), in which case the normals are calculated as $M^{-1}\nabla g_i(x)$.

Definition 9. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a closed convex set. Its* tangent cone *is the set-valued mapping $\mathbf{T}_{\mathcal{C}} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ defined by*

$$\mathbf{T}_{\mathcal{C}}(x) = \{\xi \in \mathbb{R}^n \mid \langle \eta, \xi\rangle \leq 0, \text{ for all } \eta \in \mathbf{N}_{\mathcal{C}}(x)\}. \tag{32}$$

This means that the normal and tangent cones are polar to each other.

Fact 4. *The normal cone to a closed convex set defines a maximal monotone mapping.*

Definition 10. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a set. Its* indicator function *is defined as $\psi_{\mathcal{C}}(x) = 0$ if $x \in \mathcal{C}$ and $\Psi_{\mathcal{C}}(x) = +\infty$ if $x \notin \mathcal{C}$.*

Fact 5. *Let $\mathcal{C} \subseteq \mathbb{R}^n$ be a closed convex set. Then, $\Psi_{\mathcal{C}}$ is proper, convex, LSC, and $\partial\Psi_{\mathcal{C}}(x) = \mathbf{N}_{\mathcal{C}}(x)$ for all $x \in \mathbb{R}^n$.*

Definition 11. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, convex, and LSC function. Its* convex conjugate, *$f^\star : \mathbb{R}^n \to \mathbb{R}$, is defined as*

$$f^\star(y) = \sup_{\xi \in \mathbb{R}^n} \left\{\xi^\top y - f(\xi)\right\}. \tag{33}$$

Sidebar (cont.)

Fact 6. *The convex conjugate of a proper convex LSC function, is also proper, convex and LSC. Moreover,* $(f^\star)^\star = f$.

Fact 7. *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,* $x \in \partial f^\star(y)$.

Theorem 5. *[S28, Theorem 16.23] Let* $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *be proper, convex, and LSC. Then, the following statements are equivalent:*

$$\begin{aligned} &(x, y) \in \operatorname{gph} \partial f \\ &(y, x) \in \operatorname{gph} \partial f^\star \\ &x^\top y = f(x) + f^\star(y) \end{aligned} \tag{34}$$

Consider again the real-valued function $f(x) = \kappa|x|$ with $\kappa > 0$. Its convex conjugate is

$$f^\star(y) = \sup_{\xi \in \mathbb{R}} \left\{ \xi^\top y - \kappa|\xi| \right\}. \tag{35}$$

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:

$$\partial\Psi_{[-\kappa,\kappa]}(y) = \mathbf{N}_{[-\kappa,\kappa]}(y) = \begin{cases} (-\infty, 0] & \text{if } y = -\kappa \\ 0 & \text{if } y \in (-\kappa, \kappa) \\ [0, +\infty) & \text{if } y = \kappa \\ \emptyset & \text{otherwise} \end{cases}. \tag{36}$$

Verifying Fact 7 is also straightforward,

$$\mathbf{N}_{[-\kappa,\kappa]} = (\kappa\,\mathbf{sgn})^{-1}. \tag{37}$$

More generally, let $\|x\|_p$ be the p norm $\left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}}$, $1 \le p < +\infty$, 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

$$\mathcal{B}_{p^\star} = \{\xi \in \mathbb{R}^n \mid \|\xi\|_{p^\star} \le 1\} \tag{38}$$

is the unit ball in the $p^\star$ norm. Thus, by Facts 7 and 5,

$$\partial\kappa\|\cdot\|_p = \left(\partial\Psi_{\kappa\mathcal{B}_{p^\star}}\right)^{-1} = \left(\mathbf{N}_{\kappa\mathcal{B}_{p^\star}}\right)^{-1}. \tag{39}$$

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

$$\kappa\,\mathbf{Sgn}(x) = \mathbf{N}^{-1}_{\kappa\mathcal{B}_\infty}(x) \tag{40}$$

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

$$\mathbf{N}^{-1}_{\kappa\mathcal{B}_2}(x) = \kappa\partial\|x\|_2 = \begin{cases} \kappa x/\|x\|_2 & \text{if } x \neq 0 \\ \kappa\mathcal{B}_2 & \text{if } x = 0. \end{cases} \tag{41}$$

The following result, known as the chain rule of convex analysis, is used at several places in this article.

Fact 8. *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* $x \in \mathbb{R}^m$.

Sidebar (cont.)

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

$$\text{Proj}_M[\mathcal{C}; x] = \arg\min_{\xi \in \mathcal{C}} \frac{1}{2}\langle \xi - x, \xi - x\rangle_M. \tag{42}$$

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

$$\text{Proj}[\mathcal{B}_2, x] = \begin{cases} \frac{x}{\|x\|_2} & \text{if } \|x\|_2 \geq 1 \\ x & \text{otherwise} \end{cases}. \tag{43}$$

For $\mathcal{B}_\infty$, the projection is easily carried out component-wise:

$$(\text{Proj}[\mathcal{B}_\infty, x])_i = \min\{|x_i|, 1\}\,\mathbf{sgn}(x_i), \quad i = 1, \ldots, n. \tag{44}$$

Proposition 6. *Consider a vector $x \in \mathbb{R}^n$, a closed convex set : $\mathcal{C} \subseteq \mathbb{R}^n$, and a weighted inner product $\langle \cdot, \cdot \rangle_M$. The following statements are equivalent:*

$$y = \text{Proj}_M[\mathcal{C}; x], \tag{45a}$$

$$M(x - y) \in \mathbf{N}_\mathcal{C}(y). \tag{45b}$$

$$y = (M + \mathbf{N}_\mathcal{C})^{-1}(Mx) \tag{45c}$$

Proof. The proof can be found in [60, p.79] inside another proof. It is reproduced here for convenience. The optimization problem (42) can be equivalently written as $\text{Proj}_M[\mathcal{C}; x] = \arg\min_{\xi \in \mathbb{R}^n} \frac{1}{2}\langle \xi - x, \xi - x\rangle_M + \Psi_\mathcal{C}(\xi)$. By the convexity of the weighted norm and of the indicator function, it follows that $y = \text{Proj}_M[\mathcal{C}; x]$ if, and only if, $0 \in M(y - x) + \partial\Psi_\mathcal{C}(y)$, that is, if, and only if, $M(x - y) \in \mathbf{N}_\mathcal{C}(y)$ or equivalently, if and only if $Mx \in (M + \mathbf{N}_\mathcal{C})(y)$. Hence, $y \in (M + \mathbf{N}_\mathcal{C})^{-1}(Mx)$. □

In addition if $\mathcal{C}$ is a cone, then the statements (45) are equivalent to [S22, Corollary 23.5.4]

$$\mathcal{C}^\star \ni M(y - x) \perp y \in \mathcal{C}, \tag{46}$$

where

$$\mathcal{C}^\star = \{\eta \in \mathbb{R}^n \mid \langle \eta, \xi \rangle \geq 0 \text{ for all } \xi \in \mathcal{C}\} \tag{47}$$

is the *dual cone* to $\mathcal{C}$.
If $\mathcal{C}$ 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

$$\mathcal{C} = \{\xi \in \mathbb{R}^n \mid G\xi \leq b\}. \tag{48}$$

For instance, if $\mathcal{C} = \mathcal{B}_\infty$, then $G = [I_n, -I_n]^\top$ and $b = \mathbf{1}_{2n}$. For polyhedral sets, the projection (45a) can be computed solving a conventional quadratic program. The code below shows an implementation in Python 3 using the open source modeling language for convex optimization CVXPY [74].

```
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 $\mathcal{C} = \mathcal{B}_1$, it is possible to use the above code with $G \in \mathbb{R}^{2^n \times n}$ with the rows of G covering all possible vectors with entries in $\{-1, 1\}$ and $b = \mathbf{1}_n$. For large problems, a more efficient solution consists in splitting ξ into its positive and negative components, that is $\xi = \xi^+ - \xi^-$, where $\xi^+, \xi^- \in \mathbb{R}^n_+$.

Sidebar (cont.)

In such a case the code below solves the projection problem by setting

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

```
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
```

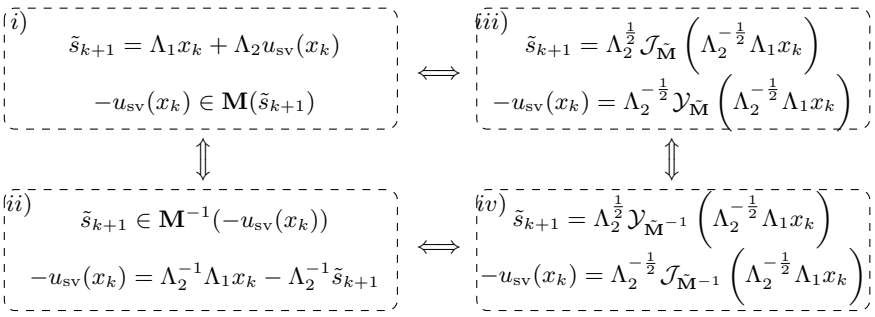

Figure 4: Implicit (left-hand side) and explicit (right-hand side) equivalent descriptions of the closed-loop sliding dynamics. Implicit expressions are useful for stability analysis, whereas the explicit ones are used for the numerical implementation of the controller.

To complete the proof let us show the equivalence between ii) and iv). To this end, let us proceed in a similar manner as above, *i.e.*, the following inclusions are equivalent:

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

Finally, the substitution of (83) into the sliding output yields

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

□

The controller $u_{\rm sv}(x_k)$ in (62), (67), is calculated with the help of Proposition 7 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 (40) were used to compute $\mathbf{M}^{-1}$ (see "Sidebar VI"). To write the explicit expression for $-u_{\rm sv}(x_k)$ we make use of the implicit expression *ii)* in the lower part of Figure 4 by pre-multiplying by $\tilde{K}^{-1}$. Thus, after rearranging terms, it follows that

$$\begin{aligned}
&\tilde{K}^{-1}\Lambda_2^{-1}\Lambda_1 x_k + \tilde{K}^{-1}u_{\rm sv}(x_k)\\
&\qquad\in (\tilde{K}\Lambda_2\tilde{K})^{-1}\,\mathbf{N}_{\mathcal{B}_\infty}(-\tilde{K}^{-1}u_{\rm sv}(x_k)). \qquad (85)
\end{aligned}$$

It thus follows from (45) that

$$\begin{aligned}
-u_{\rm sv}(x_k) &= \tilde{K}\left(\mathrm{I_d}+M^{-1}\,\mathbf{N}_{\mathcal{B}_\infty}\right)^{-1}\left(\tilde{K}^{-1}\varphi(x_k)\right)\\
&= \tilde{K}\,\mathrm{Proj}_M\left(\mathcal{B}_\infty;\varphi(x_k)\right), \qquad (86)
\end{aligned}$$

where

$$M = M^\top := \tilde{K}(\tilde{B}^\top\tilde{P}^{-1}\tilde{B})\tilde{K}, \qquad (87)$$

$$\begin{aligned}
\varphi(x_k) &:= \tilde{K}^{-1}(\tilde{B}^\top\tilde{P}^{-1}\tilde{B})^{-1}\tilde{B}^\top\left(\tilde{P}+\frac{1}{2}\tilde{B}\tilde{B}^\top\right)^{-1}\tilde{A}x_k\\
&= \tilde{K}^{-1}(\tilde{B}^\top\tilde{P}^{-1}\tilde{B})^{-\frac{1}{2}}\tilde{L}x_k. \qquad (88)
\end{aligned}$$

This extends the scalar case in (13). 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 (44) can be applied to obtain

$$\begin{aligned}
w_i &= \min\{\varphi(x_k)_i, 1\}\,\mathbf{sgn}(\varphi(x_k)_i) \quad i=1,\dots,n\\
u_{\rm sv}(x_k) &= -\gamma(\tilde{B}^\top\tilde{P}^{-1}\tilde{B})^{-1/2}w
\end{aligned} \qquad (89)$$

Otherwise, the controller can be computed as the solution of a quadratic program under constraints, similarly to (15),

$$\begin{aligned}
&\mathrm{Proj}_M(\mathcal{B}_\infty;\tilde{K}^{-1}\varphi(x_k))\\
&= \underset{\xi\in\mathcal{B}_\infty}{\arg\min}\,\frac{1}{2}(\xi-\tilde{K}^{-1}\varphi(x_k))^\top M(\xi-\tilde{K}^{-1}\varphi(x_k)), \qquad (90)
\end{aligned}$$

see "Sidebar VI" for $\mathcal{B}_\infty$, (45a) and the associated piece of code for the computation of the projection.

3.2 Discrete-Time Classical Approach

Let us pass to the discretization of the first controller in (17), (19). To start with, consider (42) and set $\tilde{s}_{k+1} = \sigma(x_k,u_k)$ as in (62), but with $\sigma(x,w)=\tilde{C}(\tilde{A}x+\tilde{B}w)$, so that $s_{k+1}=\sigma(x_k,u_k+\tilde{\delta}_k)=\tilde{C}x_{k+1}$. Let us impose the condition

$$\tilde{C}(\tilde{A}x_k+\tilde{B}u_{\rm lin}(x_k)) = \tilde{C}x_k = s_k \qquad (91)$$

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

$$u_{\rm lin}(x_k) = (\tilde{C}\tilde{B})^{-1}\tilde{C}(I_n-\tilde{A})x_k. \qquad (92)$$

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

$$s_{k+1} = s_k + v_k + \tilde{C}\tilde{B}\tilde{\delta}_k \qquad (93a)$$

$$\tilde{s}_{k+1} = s_k + v_k. \qquad (93b)$$

From (93b) it is natural to define the control action $v_k = u_{\rm sv}(x_k)\in -\tilde{K}\,\mathbf{Sgn}(\tilde{s}_{k+1})$. It gives

$$s_{k+1} \in s_k - \tilde{K}\,\mathbf{Sgn}(\tilde{s}_{k+1}) + \tilde{C}\tilde{B}\tilde{\delta}_k \qquad (94a)$$

$$\tilde{s}_{k+1} \in s_k - \tilde{K}\,\mathbf{Sgn}(\tilde{s}_{k+1}). \qquad (94b)$$

A generalized equation with unknown $\tilde{s}_{k+1}$ is obtained in (94b). The resemblance between equalities in (94b) and (70) is clear. Define $\tilde{s}'_k = \tilde{K}^{-\frac{1}{2}}\tilde{s}_k$ and note that (94b) is equivalent to

$$\tilde{s}'_{k+1} \in \tilde{K}^{-\frac{1}{2}}s_k - \tilde{K}^{\frac{1}{2}}\,\mathbf{Sgn}(\tilde{K}^{\frac{1}{2}}\tilde{s}'_{k+1}) = \tilde{K}^{-\frac{1}{2}}s_k - \partial f(\tilde{s}'_{k+1}) \qquad (95)$$

with $f(\cdot)=\|\cdot\|_1\circ\tilde{K}^{\frac{1}{2}}$. Once again, the chain rule has been used (see Fact 8 in "Sidebar VI"). Notice that the functions f in (95) and (22) share the same structure and are both proper, convex, and LSC. Thus, the expression (95) is also a generalized equation with unknown $\tilde{s}'_{k+1}$, rewritten equivalently as

$$\begin{aligned}
\tilde{s}'_{k+1} &= \mathrm{Prox}_f(\tilde{K}^{-\frac{1}{2}}s_k)\\
&= \mathrm{Prox}_f(\tilde{K}^{-\frac{1}{2}}(\tilde{s}_k+\tilde{C}\tilde{B}\tilde{\delta}_{k-1}))\\
&= \mathrm{Prox}_{\left\|\tilde{K}^{\frac{1}{2}}(\cdot)\right\|_1}(\tilde{s}'_k+\tilde{C}\tilde{B}\tilde{\delta}_{k-1}). \qquad (96)
\end{aligned}$$

This algorithm may be named a *robust* or *perturbed proximal-point algorithm*. Using (93) it is also inferred that

$$s_{k+1} = \tilde{K}^{\frac{1}{2}}\,\mathrm{Prox}_{\left\|\tilde{K}^{\frac{1}{2}}(\cdot)\right\|_1}(s_k)+\tilde{C}\tilde{B}\tilde{\delta}_k. \qquad (97)$$

Remark 5. *The above control strategy has been proposed in [21, 31]. For instance, equation (94b) is exactly [21, Equation (9)].*

3.2.1 Stability Analysis and Sliding Mode

Let us briefly recall the main stability properties of (94). It is proved in [21, Proposition 2] that $\{\tilde{s}_k\}_{k\in\mathbb{N}}$ solution of (94b) converges to zero in a finite number $k_{\min} < +\infty$ of steps (cf. Proposition 8 in "Sidebar VII" as well), while $\{s_k\}_{k\in\mathbb{N}}$ solution of (94a) 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 $s_k + u_{\rm sv}(x_k) = 0$ is equivalent to

$$u_{\rm sv}(x_k) = -s_k = -\tilde{C}\tilde{B}\tilde{\delta}_{k-1}. \tag{98}$$

As a fundamental result, the control (21) is approximated by (98): in discrete time, the implicit method allows us to design an input which compensates for the perturbation with a one-step delay, in a similar way to (39). Once again the set-valued controller does not depend on the gain $\tilde{K}$ during the sliding phase.

3.2.2 Control-Input Calculation

Using Proposition 6 in "Sidebar VI", equation (94), and mapping inversion (see "Sidebar III"), we can see that

$$\tilde{K}\left(\tilde{K}^{-1}s_k - (-\tilde{K}^{-1}u_{\rm sv}(x_k)\right) \in \mathbf{N}_{\mathcal{B}_\infty}\left(-\tilde{K}u_{\rm sv}(x_k)\right) \tag{99}$$

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

$$-u_{\rm sv}(x_k) = \tilde{K}\,\mathrm{Proj}_{\tilde{K}}\left(\mathcal{B}_\infty; \tilde{K}^{-1}s_k\right). \tag{100}$$

Again, the projection can be computed by solving a quadratic program or, similarly to (89), conveniently choose $\tilde{K} = \gamma I$ so that

$$(u_{\rm sv}(x_k))_i = -\gamma \min\left\{\frac{|(s_k)_i|}{\gamma}, 1\right\}\mathbf{sgn}((s_k)_i), \tag{101}$$

see "Sidebar VI" for $\mathcal{B}_\infty$, (45a) and the associated piece of code for the computation of the projection in a general setting. The controller in (101) is in the minimum-operator formalism [61, 48, 50, 62, 63].

Remark 6. ***Dead-beat controller.*** *The linear controllers in (17) and in (92) have different structures. Suppose that $u_{\rm lin}(x_k)$ is computed as $u_{\rm lin}(x_k) = -(\tilde{C}\tilde{B})^{-1}\tilde{C}\tilde{A}x_k$, which would be a naive implementation of (17). Then,*

$$\tilde{C}x_{k+1} = \tilde{C}\tilde{A}x_k + \tilde{C}\tilde{B}(u_{\rm lin}(x_k) + \tilde{\delta}_k) + v_k = v_k + \tilde{C}\tilde{B}\tilde{\delta}_k, \tag{102}$$

which is different from (93). Applying the above $u_{\rm sv}(x_k)$ yields $\tilde{s}_{k+1} \in -\tilde{K}\,\mathbf{Sgn}(\tilde{s}_{k+1})$, which is equivalent to $\tilde{s}'_{k+1} = (\mathrm{I_d} + \partial f)^{-1}(0)$. Since the operator $(\mathrm{I_d} + \partial f)^{-1}$ is single-valued and its graph contains $(0,0)$, it follows that $\tilde{s}_{k+1} = 0$. Therefore, this algorithm is a one-step dead-beat controller. The implicit algorithms introduced in [30, 31, 21] are not of the dead-beat type. It is noteworthy that a major discrepancy exists between (92) and the dead-beat input in the framework of discretization of continuous systems. Consider for instance the exact ZOH discretization with constant sampling time $h > 0$, where $\tilde{A} = e^{Ah}$, $\tilde{B} = e^{Ah}\int_0^h e^{-A\tau}Bd\tau$. Then, as $h \to 0$ we have $(\tilde{C}\tilde{B})^{-1}\tilde{C}(I_n - \tilde{A}) = \mathcal{O}(1)$ while $(\tilde{C}\tilde{B})^{-1}\tilde{C}\tilde{A} = \mathcal{O}(h^{-1})$. Clearly, the dead-beat input grows unbounded as $h \to 0$ and cannot be a good candidate for convergence towards its continuous-time counterpart. This shows that such dead-beat controller is well-suited to purely discrete-time systems, but not for discretized continuous-time systems.

3.3 Recapitulation

Allow us to summarize some of the connections between sliding-mode control, passivity, and optimization.

3.3.1 Link with Optimization

It is interesting to note that operators of the form $(\mathrm{I_d} + \mathbf{M})^{-1}$ are ubiquitous in the above discrete-time algorithms: they are the well-known resolvent mappings, very close to Yosida approximations (see (23) and (26) in "Sidebar V"). They have been exhibited in (10), (11), (73), (86), (96), and (100). They are a direct consequence of the presence of a maximal monotone mapping $\mathbf{M}$ on the right-hand side of the differential inclusions in (22), (36). These operators are resolvents of mappings of the form $\mathbf{M} = \partial f$ for some proper convex lower semi-continuous function f. This shows the strong connection between the implicit discretization and Optimization (see also (86)). 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 "Sidebar VIII", where the implicitly discretized scheme is reinterpreted as a one-step MPC. The explicit discretization (that is, choosing $u_{\rm sv}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(s_k)$) yields operators of the form $(\mathrm{I_d} - \partial f)$, which are neither monotone nor possess interesting properties, neither in terms of stability nor robustness.

The above analysis is closely related to the analysis of proximal-point algorithms (Definition 12 in "Sidebar VII"). However, a peculiarity of discrete sliding-mode control is the notion of a *robust* proximal-point algorithm, formulated in (57) in "Sidebar VII" and shown to converge in a finite number of steps, given sufficiently large gains. It is worth emphasizing that (67) has the form $u_{\rm sv}(x_k) \in -\partial\|\tilde{K}\tilde{s}_{k+1}\|_p$ 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 (86) by $\mathcal{B}_{p^*}$. For

Sidebar VII: Proximal-Point Algorithm and Proximal Mapping

Proximal-point algorithms and proximal operators are fundamental tools in Optimization. The big names and founding fathers of this field are Jean Jacques Moreau [S30, S31, S32], Kôsacu Yosida [S33], R. Tyrell Rockafellar [S18, 54], and Hedy Attouch [S34]. Let us recall that this article deals with finite-dimensional systems only. As seen in (11), (13), (74), (86), (96), and (97), 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 such algorithms in the perturbed scenarios that naturally arise in Control is apparently unknown in Optimization. Given a maximal monotone operator $\mathbf{M} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ (see "Sidebar V"), its set of zeros is denoted by $\operatorname{zero}\mathbf{M} = \mathbf{M}^{-1}(0) = \{\xi \in \mathbb{R}^n \mid 0 \in \mathbf{M}(\xi)\}$.

Definition 12. ***Proximal-Point Algorithm*** *[S28, p. 345]. Assume that $\mathbf{M} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is a maximal monotone operator satisfying $\operatorname{zero}\mathbf{M} \neq \emptyset$ and let $x_0 \in \mathbb{R}^n$ and $\gamma > 0$. The proximal-point algorithm associated with $\mathbf{M}$ is defined as*

$$x_{k+1} = \mathcal{J}_{\gamma\mathbf{M}}(x_k), \tag{49}$$

for $k \in \mathbb{N}$, where $\mathcal{J}_{\gamma\mathbf{M}}(\cdot)$ is the resolvent of $\mathbf{M}$.

Theorem 1 in [S35] establishes the convergence of (49) towards $\operatorname{zero}\mathbf{M}$. Consider a proper, lower semicontinuous, and convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. The operator

$$\operatorname{Prox}_f = (\mathrm{I_d} + \partial f)^{-1} \tag{50}$$

is called the *proximal mapping* [S28, Definition 12.23]. By definition, when $\mathbf{M} = \partial f$ the algorithm (49) is equivalent to any of the following iterations:

$$x_{k+1} = \operatorname{Prox}_{\gamma f}(x_k) \tag{51}$$

$$x_{k+1} - x_k \in -\gamma \partial f(x_{k+1}) \tag{52}$$

$$x_{k+1} = \operatorname*{arg\,min}_{\xi \in \mathbb{R}^n} \left\{ f(\xi) + \frac{1}{2\gamma} |\xi - x_k||^2 \right\}. \tag{53}$$

The form (51) gives the algorithm its name. It can be readily seen from (50) that

$$\operatorname{zero}\operatorname{Prox}_f = \partial f(0). \tag{54}$$

Also remark that, if $f = \Psi_{\mathcal{C}}$ (see Definition 10) with $\mathcal{C} \subseteq \mathbb{R}^n$ a closed convex nonempty set, then,

$$\operatorname{Prox}_f = \operatorname{Proj}(K; \cdot) = (\mathrm{I_d} + \mathbf{N}_K)^{-1}. \tag{55}$$

When implicitly or semi-implicitly discretized, several higher-order differentiators and controllers involve the resolvent of the subdifferential of the proper convex LSC function $f : \mathbb{R} \to \mathbb{R}$, $x \mapsto \frac{1}{2}\gamma x^2 + \Psi_{[-1,1]}(x)$, $\gamma > 0$, that is, $\partial f(x) = \gamma x + \mathbf{N}_{[-1,1]}(x)$, see "Sidebar VI". 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,

$$\mathcal{J}_{\partial f'}(y) = \begin{cases} -1 & \text{if } y \leq -1 - \gamma \\ \frac{1}{1+\gamma} y & \text{if } y \in [-1-\gamma, 1+\gamma] \\ 1 & \text{if } y \geq 1 + \gamma \end{cases}. \tag{56}$$

Remark 7. *It can be readily seen from (52) that the proximal-point algorithm corresponds to the backward (implicit) Euler discretization of the gradient system $\dot{x}(t) \in -\frac{\gamma}{h}\partial f(x(t))$, which is well-known to be more efficient and have better approximation properties than its forward (explicit) Euler counterpart [55].*

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

$$x_{k+1} = \mathcal{J}_{\gamma\mathbf{M}}(x_k + w_k). \tag{57}$$

The next Proposition is the discrete-time counterpart of Proposition 2.

Sidebar (cont.)

Proposition 8. Robust proximal-point algorithm. *Consider the perturbed proximal-point algorithm* (57) *with* $\gamma > 0$ *and* $\mathbf{M} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *a maximal monotone operator such that*

$$(\gamma\,\mathbf{M})^{-1}\left(\bigcup_{k\in\mathbb{N}} w_k\right) = \{0\}\,. \tag{58}$$

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

$$(\gamma\,\mathbf{M})^{-1}\left(\bigcup_{k\in\mathbb{N}} w_k + \varepsilon\mathcal{B}\right) = \{0\} \tag{59}$$

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

Proof. From the definition of Yosida approximation (26), it follows that $x_{k+1} = x_k + w_k - \gamma\mathcal{Y}_{\gamma\,\mathbf{M}}(x_k + w_k)$. Thus

$$\begin{aligned}\|x_{k+1}\|_2^2 &= x_{k+1}^\top\left(x_k + w_k - \gamma\mathcal{Y}_{\gamma\,\mathbf{M}}(x_k + w_k)\right)\\ &\leq \frac{1}{2}\|x_{k+1}\|_2^2 + \frac{1}{2}\|x_k\|_2^2\\ &\quad - \mathcal{J}_{\gamma\,\mathbf{M}}(x_k + w_k)^\top\left(\gamma\mathcal{Y}_{\gamma\,\mathbf{M}}(x_k + w_k) - w_k\right),\end{aligned} \tag{60}$$

where we used the fact that $\langle x, y\rangle \leq \frac{1}{2}\|x\|_2^2 + \frac{1}{2}\|y\|_2^2$. Setting the Lyapunov function candidate as $V_{k+1} = \frac{1}{2}\|x_{k+1}\|_2^2$, it is obtained

$$V_{k+1} - V_k \leq -\mathcal{J}_{\gamma\,\mathbf{M}}(x_k + w_k)^\top\left(\gamma\mathcal{Y}_{\gamma\,\mathbf{M}}(x_k + w_k) - w_k\right). \tag{61}$$

In what follows, it is shown that the right-hand side of (61) is negative definite. To that end, we point out that assumption (59) implies that for any $b \in \mathcal{B}$ and any $k \in \mathbb{N}$, $\varepsilon b - w_k \in \gamma\,\mathbf{M}(0)$. Thus, it follows from (28) and the maximal monotonicity of $\gamma\,\mathbf{M}$, that for all $k \in \mathbb{N}$

$$\langle\mathcal{J}_{\gamma\,\mathbf{M}}(x_k + w_k), \gamma\mathcal{Y}_{\gamma\,\mathbf{M}}(x_k + w_k) - \varepsilon b - w_k\rangle \geq 0. \tag{62}$$

After rearranging terms, we can see that

$$\begin{aligned}&\langle\mathcal{J}_{\gamma\,\mathbf{M}}(x_k + w_k), \gamma\mathcal{Y}_{\gamma\,\mathbf{M}}(x_k + w_k) - w_k\rangle \geq\\ &\qquad\qquad \varepsilon\sup_{b\in\mathcal{B}}\langle b, \mathcal{J}_{\gamma\,\mathbf{M}}(x_k + w_k)\rangle.\end{aligned} \tag{63}$$

That is,

$$\begin{aligned}\langle\mathcal{J}_{\gamma\,\mathbf{M}}(x_k + w_k), \gamma\mathcal{Y}_{\gamma\,\mathbf{M}}(x_k + w_k) - w_k\rangle &\geq \varepsilon\,\|\mathcal{J}_{\gamma\,\mathbf{M}}(x_k + w_k)\|\\ &\geq \varepsilon\,\|x_{k+1}\|\,.\end{aligned} \tag{64}$$

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 (57) is equivalent to

$$x_k + w_k \in (\mathrm{I_d} + \gamma\,\mathbf{M})(x_{k+1}), \tag{65}$$

whereas (59) 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 $b \in \mathcal{B}$,

$$w_k + x_k + \varepsilon_2 b \in (\mathrm{I_d} + \gamma\,\mathbf{M})(0). \tag{66}$$

Thus, (65)-(66) and the maximal monotonicity of $\mathrm{I_d} + \gamma\,\mathbf{M}$ imply that for all $b \in \mathcal{B}$

$$\varepsilon_2\langle b, x_{k+1}\rangle \geq 0. \tag{67}$$

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

Sidebar (cont.)

References

[S30] J.J. Moreau, Fonctions convexes duales et points proximaux dans un espace hilbertien, *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 255:2897-2899, 1962.

[S31] J.J. Moreau, Propriétés des applications "prox", *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 256:1069-1071, 1963.

[S32] J.J. Moreau, Proximité et dualité dans un espace hilbertien, *Bulletin de la Soc. Math. France*, tome 93:273-299, 1965.

[S33] K. Yosida, Functional Analysis, Springer Berlin, Heidelberg, 1974.

[S34] H. Attouch, G. Buttazzo and G. Michaille, Variational Analysis in Sobolev and BV Spaces, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

[S35] E.K. Ryu, W. Yin, Large-scale convex optimization via monotone operators, Cambridge University Press, 2023.

example, the implicit discretization of the unit control is

$$-u_{\mathrm{sv}}(x_k) = \tilde{K}\,\mathrm{Proj}_M\left(\mathcal{B}_2; \tilde{K}^{-1}\xi(x_k)\right). \tag{103}$$

Again, it is possible to choose $\tilde{K}$ such that $M = \gamma^2 I$ and use (43) to compute an explicit form for the projection.

3.3.2 Link with Passivity

Both control design methodologies described above follow the celebrated passivity plus zero-state detectability or observability results [68, 75, 67] (see [72] for the discrete-time version). Regarding the direct passivity-based design, Theorem 4 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

$$\begin{aligned} x_{k+1} &= \tilde{A}x_k + \tilde{B}u_{\mathrm{lin}}(x_k) + \tilde{B}(v_k + \tilde{\delta}_k) \\ s_{k+1} &= \tilde{C}x_{k+1} \end{aligned} \tag{104}$$

happens to be zero-state detectable (ZSD) [72, Definition 2.2]. In this setting, the discretization with an implicit scheme certainly 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 [72, Theorem 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 ZSD property is assumed *a priori* rather than shown like it was done in Theorem 4. 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, 71], although it is worth mentioning that the idea already appears in its early stages in [77, 78].

3.3.3 Closed-Loop Properties

It is well known that, in continuous time, a sliding-mode controller gives rise to a differential inclusion. 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 it 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 only be explained 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, of the implicitly-defined control law (67), since the condition $\sigma(x_k, u_k) = 0$ is attained in finite time and maintained there after, regardless of any bounded matched perturbation.

The discretization $u_{\mathrm{sv}}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_k)$ 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 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 $h \downarrow 0$ [43, 21]. When written explicitly, the implicit (or, 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. Mark, however, that when using the implicit discretization, correct parameters for the saturation functions are automatically provided by

Discretization with a backward-Euler scheme turns out to be necessary for assuring passivity.

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 h^2 of the sliding surface, $\|s_k\| = \mathcal{O}(h^2)$, then at the following step we only have $\|s_{k+1}\| = \mathcal{O}(h)$ 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, $\|s_{k+1}\| = \mathcal{O}(h^2)$ [21, Lemma 11].

The implicit discretization method results in a controlled system that is also robust with respect to uncertainty in the system parameters [42, 43]. With the appropriate choice of the maximal monotone operators, it is also possible to obtain robustness with respect to unmatched external perturbations as well [79, 44]. 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 (100)). In more complex scenarios, it is always possible to resort to the numerical machinery described in section "6".

3.3.4 On the Nature of Sliding Mode Control

Contrarily to a widely spread idea, sliding-mode control 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 3 in "Sidebar III") and the calculation of the controller with optimization tools, as detailed below. See Figure 8 below, for a typical illustration of discontinuous and selections of set-valued control inputs.

4 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 Arie Levant and co-authors [9, 8, 10], have witnessed considerable attention in the past twenty years. Roughly speaking, first-order schemes permit the designer to guarantee that a sliding variable (denoted s in section "2") 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 (or, 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 named an r-sliding surface and the r-sliding mode is defined by $\{x \in \mathbb{R}^n \mid s(x) = \dot{s}(x) = \cdots = s^{(r-1)}(x) = 0\}$. From a practical point of view, 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, 3) given a signal with 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-sliding mode control. 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 feedback equivalent to a passive system. Second, most higher-order sliding-mode control algorithms are not defined by maximal monotone differential inclusions. Finally, the main underlying structure is homogeneity [80], 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 super-twisting 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 paragraph about homogeneity-based schemes and on how proximal-point algorithms appear in this setting.

4.1 Super-Twisting Algorithm

The super-twisting is an order-2 algorithm. It has proven useful for control, differentiation and state observation. In the following, all three application fields are presented.

4.1.1 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

Sliding-mode control is <u>not</u> intrinsically related to infinitely-fast switching inputs (bang-bang-like controllers).

Sidebar VIII: Implicit Sliding-Mode and Model Predictive Control (MPC)

Implicit discrete-time sliding-mode control and differentiation require the solution of an optimization problem at each time-step. Model predictive control requires the solution of an optimization problem on longer time intervals. Besides this feature, there exists a closer and more formal link between implicit sliding modes and model predictive control. Consider an optimization problem over an horizon event of one step,

$$\begin{aligned} &\min_{(s_{k+1},u_k)} f(s_{k+1}-h\delta_k)+\frac{h}{2}\left\|u_k\right\|^2, \\ &\text{such that } s_{k+1}=s_k+hu_k+h\delta_k \end{aligned} \tag{68}$$

where the function $f:\mathbb{R}^m\to\mathbb{R}$ is proper, convex and LSC. In this problem, at time $t_k=kh$, s_k is treated as a known parameter, whereas δ_k is an unknown parameter and the target is to find the optimal controller value u_k and the optimal state value s_{k+1} that minimize the given cost function. The Lagrangian associated with (68) is given as,

$$\begin{aligned} L(s_{k+1},u_k,\lambda_{k+1}) = f(s_{k+1}-h\delta_k)+\frac{h}{2}\left\|u_k\right\|^2 \\ -\lambda_{k+1}^\top(s_{k+1}-s_k-hu_k-h\delta_k)\,. \end{aligned} \tag{69}$$

The Karush-Kuhn-Tucker conditions associated with (68) are

i) $0\in\partial_{s_{k+1}}L(s_{k+1}^*,u_k^*)$,

ii) $0=\nabla_{u_k}L(s_{k+1}^*,u_k^*)$, and

iii) $s_{k+1}^*=s_k+hu_k^*+h\delta_k$.

From KKT-i) one sees that $0\in\partial f(s_{k+1}^*-h\delta_k)-\lambda_{k+1}^*\Leftrightarrow\lambda_{k+1}^*\in\partial f(s_{k+1}^*-h\delta_k)$, whereas KKT-ii) yields $0=hu_k^*+h\lambda_{k+1}^*\Leftrightarrow\lambda_{k+1}^*=-u_k^*$. Hence, the combination of both relations together with condition KKT-iii) yields $h\lambda_{k+1}^*=-hu_k^*=s_k+h\delta_k-s_{k+1}^*\in h\partial f(s_{k+1}^*-h\delta_k)$. Consequently,

$$s_k\in(I+h\partial f)(s_{k+1}^*-h\delta_k)\Leftrightarrow s_{k+1}^*-h\delta_k=\mathrm{Prox}_{hf}(s_k)\,, \tag{70}$$

and the substitution of (70) back into KKT-iii) leads to

$$s_{k+1}^*-h\delta_k=s_k+hu_k^*=\mathrm{Prox}_{hf}(s_k), \tag{71}$$

equivalently,

$$u_k^*=-\frac{1}{h}\left(s_k-\mathrm{Prox}_{hf}(s_k)\right)\,. \tag{72}$$

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

$$\begin{aligned} s_{k+1}^*&=\tilde{s}_{k+1}+h\delta_k \\ \tilde{s}_{k+1}&=s_k+hu_k^* \\ -u_k^*&\in\partial f(\tilde{s}_{k+1}) \end{aligned}\quad, \tag{73}$$

which are the same equations as the implicit discretization with $f(\cdot)=\gamma\left\|\cdot\right\|$, $\gamma>0$.

Sidebar IX: Projected Dynamical Systems

Projected dynamical systems (PDS) make a class of nonsmooth systems which, as seen below, is closely related to sliding-mode systems. They were introduced in [S36, S37]. Recently, they have witnessed a strong interest from the automatic-control scientific community, see, for example, [S38, S39]. PDS may be expressed under different equivalent forms [S40, S41] (see definitions 8 and 9 in "Sidebar VI"). Some examples of such equivalent dynamics are:

$$\begin{aligned}
&\dot{x} \in -f(x) - g(t) - \mathbf{N}_{\mathcal{C}}(x) \\
&\text{with } \dot{x} = -f(x) - g(t) - \mathrm{Proj}[\mathbf{N}_{\mathcal{C}}(x); -f(x) - g(t)] && \text{(74a)}\\
&\dot{x} \in -f(x) - g(t) - \mathbf{N}_{\mathbf{T}_{\mathcal{C}}(x)}(\dot{x}) && \text{(74b)}\\
&\dot{x} = (\mathrm{I_d} + \mathbf{N}_{\mathbf{T}_{\mathcal{C}}(x)})^{-1}(-f(x) - g(t)) && \text{(74c)}\\
&\dot{x} = \mathrm{Proj}[\mathbf{T}_{\mathcal{C}}(x); -f(x) - g(t)] && \text{(74d)}
\end{aligned}$$

see [S40, S41] for details. The second condition in (74a) means that solutions are slow (with $-\dot{x}(t)$ of minimal norm inside $f(x) + g(t) + \mathbf{N}_{\mathcal{C}}(x)$). The resolvent of the normal cone to the tangent cone appears in (74c). Use was made of Fact 5 in "Sidebar VI", (50), and the comment just after (54) in "Sidebar VII" to obtain (74d) . The projection is a particular instance of Prox, see (55), showing that PDS belong to the class of proximal systems.

3.3.5 Discrete-time PDS

Let us now discretize (74a) as

$$x_{k+1} \in x_k - h(f(x_k) - g_k) - \mathbf{N}_{\mathcal{C}}(x_{k+1}) \tag{75}$$

$h > 0$. This is rewritten equivalently as

$$x_{k+1} = (\mathrm{I_d} + \mathbf{N}_{\mathcal{C}})^{-1}(x_k - h(f(x_k) + g_k)) \tag{76}$$

and as

$$x_{k+1} = \mathrm{Proj}[\mathcal{C}; x_k - h(f(x_k) + g_k)], \tag{77}$$

where the same tools as above have again been used to pass from (74c) to (74d). It is seen once again that resolvents are ubiquitous in PDS. The system (77) furnishes a convenient way to obtain solutions whenever the projection onto the set $\mathcal{C}$ can be computed. This is the case when $\mathcal{C}$ is nonempty convex polyhedral (48). Indeed, in this case the normal cone can be expressed as $\mathbf{N}_{\mathcal{C}} = \{\xi \in \mathbb{R}^n \mid \xi = G^\top \lambda,\ 0 \leq \lambda \perp -G\xi + b \geq 0\}$. In other words, the normal cone is generated by the outwards normals to the active constraints at x. Thus, using (75), the PDS is equivalently rewritten as a complementarity system

$$\begin{aligned}
x_{k+1} &= x_k - h(f(x_k) - g_k) - G^\top \lambda_{k+1} \\
0 \leq \lambda_{k+1} &\perp w_{k+1} = -Gx_{k+1} + b \geq 0.
\end{aligned} \tag{78}$$

The discrete-time PDS in (78) is a mixed linear complementarity problem [S42] with unknowns x_{k+1} and λ_{k+1}. It gives the linear complementarity problem (LCP)

$$\begin{aligned}
0 \leq \lambda_{k+1} \perp w_{k+1} = \\
- G(x_k - h(f(x_k) - g_k)) + GG^\top \lambda_{k+1} + b \geq 0,
\end{aligned} \tag{79}$$

Obviously, $GG^\top \succ 0$ if and only if, G has row-rank m (which implies that $m \leq n$), otherwise $GG^\top \succcurlyeq 0$ and it is also a copositive-plus matrix. Using [S42, Theorem 3.8.6], if $b \in (\mathrm{SOL}(GG^\top, 0))^\star$, where $\mathrm{SOL}(M, 0)$ denotes the set of solutions to $0 \leq \lambda \perp M\lambda \geq 0$ and $(\mathrm{SOL}(M, 0))^\star$ is its dual cone, then the LCP in (79) 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, Theorem 5.3.9] the LCP in (79) has a solution. Also if λ_1 and λ_2 are any two solutions, then $GG^\top(\lambda_1 - \lambda_2) = 0$ [S42, Theorem 3.1.7]. Therefore $\lambda_1 - \lambda_2 \in \mathrm{Ker}(GG^\top) = \mathrm{Ker}(G^\top)$ [65, Theorem 3.5.3]. It is easily deduced that, if

$$x^i_{k+1} = x^i_k + h(f(x^i_k) - g_k) - G^\top \lambda^i_{k+1}, \tag{80}$$

$i = 1, 2$, then $x^1_k = x^2_k$ implies $x^1_{k+1} = x^2_{k+1}$, $k \geq 0$. As expected, the projection onto $\mathcal{C}$ is unique. A quite interesting fact is that efficient algorithms can be used to solve the LCP in (79) [S42, 59], at each time-step $k \geq 1$.

Sidebar (cont.)

3.3.6 Sliding Modes in PDS

Some kind of sliding modes can occur in PDS when the single-valued vector field is such that the trajectory evolves persistently on the boundary of $\mathcal{C}$. In this case there exists an element of $\lambda(x,t) \in -\mathbf{N}_{\mathcal{C}}(x)$ (a selection, see Definition 3 in "Sidebar III") which "compensates" for the part of $-f(x) - g(t)$ which tends to "push" the trajectory outside $\mathcal{C}$ (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 [79, Equation (6)] can be transformed into a PDS after the classical state space transformation (see [S41, section 3.4]). It guarantees robust output tracking.

References

[S36] C. Henry, Differential equations with discontinuous right-hand side for planning procedures, *J. Econom. Theory*, 4:545-551, 1972.

[S37] C. Henry, An existence theorem for a class of differential equations with multivalued right- hand side, *J. Math. Anal. Appl.*, 41:179-186, 1973.

[S38] D. Gadjov, L. Pavel, A passivity-based approach to Nash equilibrium seeking over networks, *IEEE Transactions on Automatic Control*, 64(3):1077-1092, March 2019.

[S39] L. Pavel, Dissipativity theory in game theory: On the role of dissipativity and passivity in Nash equilibrium seeking, *IEEE Control Systems Magazine*, 42(3):150-164, June 2022.

[S40] B. Brogliato, A. Daniliidis, C. Lemaréchal, V. Acary, On the equivalence between complementarity systems, projected systems and differential inclusions, *Systems Control Letters*, 55:45-51, 2006

[S41] B. Brogliato, A. Tanwani, Dynamical systems coupled with monotone set-valued operators: Formalisms, applications, well-posedness, and stability, *SIAM Review*, 62(1):3-129, 2020.

[S42] R.W. Cottle, J.S. Pang, R.E. Stone, The Linear Complementarity Problem, Academic Press, Computer Science and Science Computing, 1992.

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

$$\begin{aligned} \dot{z}(t) &= -\gamma(z(t) + \gamma f(t)) \\ \hat{g}(t) &= z(t) + \gamma f(t) \end{aligned} \tag{105}$$

with $\gamma \gg 1$. It can be easily shown that, as γ approaches $+\infty$, the transfer function of the filter converges to the transfer function 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 Lipschitz constant $L > 0$, then a super-twisting differentiation algorithm can be used. The advantages of such scheme are: 1) exact differentiation with finite gains in the absence of noise, 2) limited noise amplification, 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

$$\zeta_\alpha(w) = \frac{1}{\alpha + 1} |w|^{\alpha+1}. \tag{106}$$

The convexity of ζ_α immediately guarantees the maximal monotonicity of $\lceil \cdot \rfloor^\alpha$, see Theorem 3 in "Sidebar V". Notice also that $\lceil \cdot \rfloor^\alpha$ is single-valued for $\alpha > 0$, but multivalued at the origin when $\alpha = 0$. The super-twisting differentiation algorithm [6] reads as:

$$\begin{aligned} \dot{z}_0(t) &= z_1(t) - \gamma_0 L^{\frac{1}{2}} \lceil s_0(t) \rfloor^{\frac{1}{2}} \\ \dot{z}_1(t) &\in -\gamma_1 L \, \mathbf{sgn}(s_0(t)) \end{aligned}, \tag{107}$$

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 $\sigma_0(t) := \frac{1}{L}(z_0 - f(t))$ and $\sigma_1(t) := \frac{1}{L}(z_1(t) - \dot{f}(t))$. In these coordinates, system (107) is rewritten in recursive form as [81]

$$\begin{aligned} \dot{\sigma}_0(t) &= \sigma_1(t) - \gamma_0 \lceil \sigma_0(t) \rfloor^{\frac{1}{2}} \\ \dot{\sigma}_1(t) &\in -\gamma_1 \, \mathbf{sgn}(\sigma_0(t)) - \ddot{f}(t) \end{aligned}. \tag{108}$$

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 [82]. Hence, the differentiator attains a 2-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 (107) is

$$s_{0,k+1} = s_{0,k} + h z_{1,k+1} - h\gamma_0 L^{\frac{1}{2}} \lceil s_{0,k+1} \rfloor^{\frac{1}{2}} - \Delta f_k \tag{109a}$$

$$z_{1,k+1} \in z_{1,k} - h\gamma_1 L \, \mathbf{sgn}(s_{0,k+1}), \tag{109b}$$

where $\Delta f_k := f_{k+1} - f_k$. Our purpose now is to derive an explicit representation of (109). Substituting (109b) into (109a) and solving for $s_{0,k+1}$ yields

$$s_{0,k+1} = (\mathrm{I_d} + h \, \mathbf{M}_h)^{-1}(s_{0,k} + h z_{1,k} - \Delta f_k), \tag{110}$$

where

$$\mathbf{M}_h(w) = \gamma_0 L^{\frac{1}{2}} \lceil w \rfloor^{\frac{1}{2}} + h\gamma_1 L \, \mathbf{sgn}(w) \tag{111}$$

is a maximal monotone operator (being the sum of two maximal monotone operators, both with domain $\mathbb{R}$). The iteration (110) is compactly written as the resolvent-based algorithm

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

(see (57) and Proposition 8 in "Sidebar VII"), where the resolvent is defined in (23) in "Sidebar V" (compare with the classical algorithm (96) and the general iteration (57)). It is interesting to note that choosing an exact discretization as in [83] and neglecting unmeasurable terms also yields a form like (112) [83, Equation (24)]. Together with (10), (73), and (96), this shows the ubiquity of resolvent-based algorithms in the implicit discretization of sliding-mode systems. The algorithms provided in [84, 83] also correspond to the resolvent in (112). 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 (112), we can readily solve for the state $z_{1,k+1}$ for all $h > 0$ in (109a),

$$z_{1,k+1} = \frac{1}{h}\left(s_{0,k+1} - s_{0,k} + h\gamma_0 L^{\frac{1}{2}} \lceil s_{0,k+1} \rfloor^{\frac{1}{2}} + \Delta f_k\right). \tag{113}$$

Therefore, the pair of equations (112), (113) forms a nonlinear difference equation with state vector $\begin{pmatrix} s_{0,k} & z_{1,k} \end{pmatrix}^\top$ and allows us to advance the discretized super-twisting differentiator from step k to step $k+1$. Notice in passing that, having computed the future state $z_{1,k+1}$ in (113), 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 (109b) holds.

Remark 8. *In real-time applications it is possible to substitute Δf_k by Δf_{k-1} in (109a), and hence in (113), to render the differentiator non-anticipative.*

Semi-implicit discretization is also possible. Let us replace (109) by

$$s_{0,k+1} \in s_{0,k} + h z_{1,k+1} - h\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} \, \mathbf{sgn}(s_{0,k+1}) - \Delta f_k \tag{114a}$$

$$z_{1,k+1} \in z_{1,k} - h\gamma_1 L \, \mathbf{sgn}(s_{0,k+1}), \tag{114b}$$

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

$$\begin{aligned} &s_{0,k+1} \\ &= (\mathrm{I_d} + (h\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} + h^2 \gamma_1 L) \, \mathbf{sgn})^{-1}(s_{0,k} + h z_{1,k} - \Delta f_k). \end{aligned} \tag{115}$$

Compared to (111), the maximal monotone operator is modified to

$$\mathbf{M}_{k,h}(w) = (\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} + h\gamma_1 L) \, \mathbf{sgn}(w). \tag{116}$$

Computing $z_{1,k+1}$ is somewhat more complicated than before. First let us 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 3 in "Sidebar III", Fact 7 and (37) in "Sidebar VI"). By inverting the operators in (114), the system in the new coordinates can be rewritten as

$$s_{0,k} + hz_{1,k+1} - h\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} \lambda_{k+1} - \Delta f_k \in \mathbf{N}_{[-1,1]}(\lambda_{k+1}) \tag{117a}$$

$$z_{1,k+1} = z_{1,k} - h\gamma_1 L \lambda_{k+1} \tag{117b}$$

and then solved for λ_{k+1}. To do so, first substitute (117b) in (117a) to obtain

$$\lambda_{k+1} \in \frac{1}{h^2\gamma_1 L} \Big(s_{0,k} + hz_{1,k} - h\gamma_0 L^{\frac{1}{2}} |s_{0,k}|^{\frac{1}{2}} \lambda_{k+1} - \Delta f_k - \mathbf{N}_{[-1,1]}(\lambda_{k+1}) \Big). \tag{118}$$

The inclusion is equivalent to

$$\lambda_{k+1} = \mathcal{J}_{\mathbf{M}_{k,h}} \left(\frac{s_{0,k} + hz_{1,k} - \Delta f_k}{h^2\gamma_1 L} \right), \tag{119}$$

$$\mathbf{M}_{k,h}(w) = \frac{\gamma_0}{h\gamma_1 L^{\frac{1}{2}}} |s_{0,k}|^{\frac{1}{2}} w + \mathbf{N}_{[-1,1]}(w). \tag{120}$$

Finally, $z_{1,k+1}$ is recovered from (117b) as

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

The resolvent $\mathcal{J}_{\mathbf{M}_{k,h}}(\cdot)$ can be calculated as indicated in "Sidebar VII", after (55). It happens that resolvents of this kind are basic tools which are to be found in all the higher-order algorithms in this section.

4.1.2 Controller

When the super-twisting algorithm is used for control (as opposed to differentiation), a perturbed integrator $\dot{x}_1(t) = u(t) + \varphi(t)$ is considered. This dynamics may stem from some transformation of the plant's dynamics, and 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 [85]. The proposed controller is given by

$$\begin{aligned} u(t) &= \nu(t) - \gamma_0 L^{\frac{1}{2}} \lceil x_1(t) \rfloor^{\frac{1}{2}} \\ \dot{\nu}(t) &\in -\gamma_1 L \, \mathbf{sgn}(x_1(t)) \end{aligned}. \tag{122}$$

It is apparent that u in (122) is a kind of nonlinear non-smooth set-valued proportional-integral input. The main interest in this controller stems from the fact that, contrarily 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

$$\begin{aligned} \dot{x}_1(t) &= x_2(t) - \gamma_0 L^{\frac{1}{2}} \lceil x_1(t) \rfloor^{\frac{1}{2}} \\ \dot{x}_2(t) &\in -\gamma_1 L \, \mathbf{sgn}(x_1(t)) + \delta(t) \end{aligned} \tag{123}$$

with $\delta(t) = \dot{\varphi}(t)$. The closed-loop dynamics (123) is the same as the differentiator's dynamics (108). Finite-time stability of the origin $x_1 = x_2 = 0$ is guaranteed under similar assumptions on the gains [82], hence $x_1 = \dot{x}_1 = 0$ defines the 2-sliding surface. Notice that, similarly to (21) and (39), 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 (12) and (13). A discrete-time model for the plant is

$$x_{1,k+1} = x_{1,k} + hu_k + h\bar{\varphi}_k \tag{124a}$$

$$\varphi_{k+1} = \varphi_k + h\bar{\delta}_k \tag{124b}$$

Here, $\bar{\varphi}_k$ and $\bar{\delta}_k$ are discretized versions of $\varphi(t)$ and $\delta(t)$ (see [34] for details). The implicit discretization can now be achieved by following the same steps as in (109). Let us introduce, for instance, a nominal virtual variable $\tilde{x}_{1,k} := x_{1,k} - h\bar{\varphi}_{k-1}$ as done in (62) and (93), see [34, Equations (8), (9)]. According to (124a), the nominal variable evolves as

$$\tilde{x}_{1,k+1} = x_{1,k} + hu_k, \tag{125}$$

while the nominal backward-Euler discretization of (122) is

$$\begin{aligned} u_k &\in -\gamma_0 L^{\frac{1}{2}} |\tilde{x}_{1,k+1}|^{\frac{1}{2}} \, \mathbf{sgn}(\tilde{x}_{1,k+1}) + \nu_{k+1} \\ \nu_{k+1} &= \nu_k - h\gamma_1 L \, \mathbf{sgn}(\tilde{x}_{1,k+1}). \end{aligned} \tag{126}$$

Substitution of (126) in (125) gives

$$\begin{aligned} \tilde{x}_{1,k+1} &= (\mathrm{I_d} + h\,\mathbf{M}_h)^{-1}(x_{1,k} + h\nu_k) \\ &= \mathcal{J}_{h\,\mathbf{M}_h}(x_{1,k} + h\nu_k), \end{aligned} \tag{127}$$

with $\mathbf{M}_h$ as in (111). The explicit expression for the control in (126) is [34, 86]

$$\begin{aligned} u(x_{1,k}, \nu_k) = &-\gamma_0 L^{\frac{1}{2}} \beta_k \, \mathbf{sgn}(x_{1,k} + h\nu_k) \\ &+ \nu_k - h\gamma_1 L \mathrm{Proj}\left([-1,1]; \frac{x_{1,k} + h\nu_k}{h^2\gamma_0 L} \right) \end{aligned} \tag{128}$$

with

$$\beta_k = \frac{-h\gamma_0 L^{\frac{1}{2}}}{2} + \sqrt{\frac{h^2\gamma_0^2 L}{4} + \rho_k} \tag{129}$$

$$\rho_k = \max\{0, |x_{1,k} + h\nu_k| - h^2\gamma_1 L\}, \tag{130}$$

while

$$\nu_{k+1} = \nu_k - h\gamma_1 \, \mathrm{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h\mu_k} \right). \tag{131}$$

Alternatively, we may rewrite the control as

$$u(x_{1,k}, \nu_k) = \frac{1}{h}(\mathrm{I_d} + h\,\mathbf{M}_h)^{-1}(x_{1,k} + h\nu_k) - \frac{x_{1,k}}{h}. \tag{132}$$

It is noteworthy that this boils down once again to the calculation of a resolvent and of a projection (the saturation

function). That is, equation (127) defines a resolvent-based algorithm similar to (112) and (114).

As above, a semi-implicit method can be applied [35]. It yields

$$\tilde{x}_{1,k+1} = (\mathrm{I_d} + h\,\mathbf{M}_{k,h})^{-1}(x_{1,k} + h\nu_k) \tag{133}$$

but now with

$$\mathbf{M}_{k,h}(w) = (\gamma_0 L^{\frac{1}{2}}|x_{1,k}|^{\frac{1}{2}} + h\gamma_1 L)\,\mathbf{sgn}(w), \tag{134}$$

similar to (116). Another choice is

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

Which alternative should be chosen is an open issue.

Note that, during the sliding motion, $\tilde{x}_{1,k+1} = \tilde{x}_{1,k} = 0$, and the inclusion

$$\nu_k + \bar{\varphi}_k \in \frac{1}{h}(\mathrm{I_d} + h\,\mathbf{M}_h)(0) = h\gamma_0 L[-1,1] \tag{136}$$

follows from (127). Using (128), (130) we obtain

$$u(x_{1,k},\nu_k) = -\frac{x_{1,k}}{h} = -\frac{\tilde{x}_{1,k} + h\bar{\varphi}_{k-1}}{h} = -\bar{\varphi}_{k-1}. \tag{137}$$

It is inferred that, similarly to first order controllers, see (98), the implicit super-twisting scheme compensates for the perturbation with one-step delay.

4.1.3 Multivariable super-twisting

The above extends to the multivariable super-twisting [87], with $x_1,\, x_2 \in \mathbb{R}^n$. One replaces (106) by

$$\zeta_\alpha(w) = \frac{1}{\alpha+1}\|w\|_2^{\alpha+1}, \tag{138}$$

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

$$\|x_1\|_2^{\frac{1}{2}}\partial\|x_1\|_2 = \|x_1\|_2^{\frac{1}{2}}\,\mathbf{N}_{\mathcal{B}_2}^{-1}(x_1) \tag{139}$$

(see (41) in "Sidebar VI"), and $\mathbf{sgn}(x_1)$ by $\partial\|x_1\|_2$ (or the subdifferential of any other norm), in (122). This yields the resolvent $\mathcal{J}_{h\,\mathbf{M}_h}$, to be computed at each step with

$$\mathbf{M}_h(w) = \alpha_1\|w\|_2^{\frac{1}{2}}\partial\|w\|_2 + h\alpha_2\partial\|w\|_2, \tag{140}$$

$\alpha_1 > 0,\ \alpha_2 > 0$. Note that $\mathbf{M}_h = \partial\left(\alpha_1\zeta_{1/2} + h\alpha_2\zeta_0\right)$. It follows from the convexity of ζ_α and Theorem 3 in "Sidebar V" that $\mathbf{M}_h$ is maximal monotone.

4.1.4 Generalized Observer

The equivalence between uniform observability and the ability to recover the complete state by differentiating the system output is well known. In view of (107), it is not surprising that the super-twisting 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 [88]

$$\begin{aligned} \dot{x}_1 &= f_1(x_1,u) + x_2 \\ \dot{x}_2 &= f_2(x_1,x_2,u) \\ y &= x_1 \end{aligned}, \tag{141}$$

from where it is clear that the remaining state x_2 can be recovered by differentiating the output y. Let us consider the class of set-valued super-twisting observers presented in [89, Equations (12.3)–(12.5)]

$$\dot{\hat{x}}_1 = f_1(\hat{x}_1,u) + \hat{x}_2 - l_1\gamma\phi_1(e_1) \tag{142a}$$

$$\dot{\hat{x}}_2 \in f_2(\hat{x}_1,\hat{x}_2,u) - l_2\gamma^2\phi_2(e_1) \tag{142b}$$

$$\phi_1(e_1) = \mu_1\lceil e_1\rfloor^{\frac{1}{2}} + \mu_2\lceil e_1\rfloor^q \tag{142c}$$

$$\phi_2(e_1) = \frac{\mu_1^2}{2}\,\mathbf{sgn}(e_1) + \mu_1\mu_2\lceil e_1\rfloor^{q-\frac{1}{2}} + \mu_2^2\lceil e_1\rfloor^{2q-1}, \tag{142d}$$

where $\mu_1 \geq 0,\ \mu_2 \geq 0,\ \mu_1\mu_2 \neq 0,\ q \geq \frac{1}{2},\ l_1 > 0,\ l_2 > 0,\ \gamma > 0$, and $e_1 = \hat{x}_1 - x_1$ is the state estimation error. An implicit discretization of (142) reads

$$\hat{x}_{1,k+1} = \hat{x}_{1,k} + hf_1(\hat{x}_{1,k+1},u_k) + h\hat{x}_{2,k+1} - hl_1\gamma\phi_1(e_{1,k+1}) \tag{143a}$$

$$\hat{x}_{2,k+1} \in \hat{x}_{2,k} + hf_2(\hat{x}_{1,k+1},\hat{x}_{2,k+1},u_k) - hl_2\gamma^2\phi_2(e_{1,k+1}) \tag{143b}$$

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 (143b) 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 (143a) yields

$$\begin{aligned} e_{1,k+1} \in\ & y_{k-1} - y_k + e_{1,k} + hf_1(e_{1,k+1},y_k,u_k) \\ & + h\Phi(e_{1,k+1},y_k,\hat{x}_{2,k},u_k,h\phi_2(e_{1,k+1})) \\ & - hl_1\gamma\phi_1(e_{1,k+1}). \end{aligned} \tag{144}$$

It is inferred that

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

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 $q > \frac{1}{2}$), the inclusion in (145) 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 below are necessary. The observed variable $\hat{x}_{2,k+1}$ is obtained from (143a) and (145). Of course, semi-implicit methods can also be applied so that the computations are easier and the assumptions may be relaxed.

4.2 Higher-Order Differentiators

It is sometimes necessary to compute higher-order derivatives of f. A naive approach is to cascade several differentiators of the form (107) (which may provide reasonably good results for the second derivative [90]). 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 super-twisting differentiator (AO-STD) has the dynamics in non-recursive form [7, 81]

$$\begin{aligned} &\dot{z}_i(t) = z_{i+1}(t) - \gamma_i L^{\frac{i+1}{r+1}} \lceil s_0(t) \rfloor^{\frac{r-i}{r+1}}, \quad i = 0, \dots, r-1 \\ &\dot{z}_r(t) \in -\gamma_r L \, \mathbf{sgn}(s_0(t)) \end{aligned} \tag{146}$$

with $s_0(t) = z_0(t) - f(t)$, which is the counterpart of (107), while the counterpart of (108) writes as the recursive form

$$\begin{aligned} &\dot{\sigma}_i(t) = \sigma_{i+1}(t) - \gamma_i \lceil \sigma_0(t) \rfloor^{\frac{r-i}{r+1}}, \quad i = 0, \dots, r-1 \\ &\dot{\sigma}_r(t) \in -\gamma_r \, \mathbf{sgn}(\sigma_0(t)) + f^{(r+1)}(t) \end{aligned}, \tag{147}$$

where $\sigma_i(t) := \frac{1}{L}(z_i(t) - f^{(i)}(t))$. Accuracy and robustness of these differentiators have been studied by exploiting the homogeneity of (147) [81, 49]. As shown in [84] and [83, 91], their discrete-time versions lend themselves to transformations which yield the same structures as in (127) or (112). In particular it is shown in [84, Table 2] that "classical" higher-order differentiators known as URED (uniform robust exact differentiator) [92], AO-STD as above [7, 81], HDD (homogeneous discrete-time differentiator), GHDD (generalized homogeneous discrete-time differentiator) [93], FDFF (first-order differentiator with first-order sliding-mode filtering) [5, 94], AO-FDFF (arbitrary-order differentiator with first-order sliding-mode filtering) [94, 95] yield, when implicitly or semi-implicitly discretized, the equation

$$s_{0,k+1} = \mathcal{J}_{h\,\mathbf{M}_h}(-b_k), \tag{148}$$

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 $b_k = b(s_{0,k}, z_{1,k}, \dots, z_{n,k}, \Delta f_k)$, $\Delta f_k = f_{k+1} - f_k$.

Let us provide the implicit URED operators, which consist of the super-twisting operators with additional higher-order terms [84, Table 2, Equation (39)]. Its defining operators are

$$\begin{aligned} \mathbf{M}_{\mathrm{sing},h}(w) &= \gamma_0 L^{\frac{1}{2}} \left(\lceil w \rfloor^{\frac{1}{2}} + \mu \lceil w \rfloor^{\frac{3}{2}} \right) + \frac{3}{2} h \mu^2 \gamma_1 L \lceil w \rfloor^2 \\ &\quad + 2 h \gamma_1 L \mu w \end{aligned} \tag{149a}$$

$$\mathbf{M}_{\mathrm{set},h}(w) = \frac{1}{2} h \gamma_1 L \, \mathbf{sgn}(w) \tag{149b}$$

and the argument is

$$b(s_{0,k}, z_{1,k}, \Delta f_k) = -s_{0,k} - h z_{1,k} + \Delta f_k, \tag{149c}$$

where λ_0, λ_1, μ are parameters. Similarly to the super-twisting differentiator, the URED has an additional state variable x_1, see, for example, [84, equation (6)]. The URED counterpart of (113) takes the form

$$\begin{aligned} z_{1,k+1} = \frac{1}{h} \Big(& s_{0,k+1} - s_{0,k} + \Delta f_k + h \gamma_0 L^{\frac{1}{2}} \lceil s_{0,k+1} \rfloor^{\frac{1}{2}} + \\ & h L^{\frac{1}{2}} \mu \lceil s_{0,k+1} \rfloor^{\frac{3}{2}} \Big). \end{aligned} \tag{150}$$

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

$$\begin{aligned} \mathbf{M}_{k,h}(w) = \Big(& \gamma_0 L^{\frac{1}{2}} \left(|s_{0,k}|^{\frac{1}{2}} + \mu |s_{0,k}|^{\frac{3}{2}} \right) + \frac{3}{2} h \mu^2 \gamma_1 L s_{0,k}^2 \\ & + h \gamma_L |s_{0,k}| + \frac{1}{2} h \gamma_1 L \Big) \, \mathbf{sgn}(w), \end{aligned} \tag{151}$$

which is the URED counterpart of (115). 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

$$\begin{aligned} & s_{0,k} + h x_{1,k} - \Delta f_k - h \gamma_0 L^{\frac{1}{2}} \left(|s_{0,k}|^{\frac{1}{2}} + \mu |s_{0,k}|^{\frac{3}{2}} \right) \lambda_{k+1} - \\ & h^2 \gamma_1 L \left(2\mu |s_{0,k}| + \frac{3}{2} \mu^2 s_{0,k}^2 + \frac{1}{2} \right) \lambda_{k+1} \in \mathbf{N}_{[-1,1]}(\lambda_{k+1}), \end{aligned} \tag{152}$$

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

$$\begin{aligned} \mathbf{M}_{k,h} = \frac{2}{h \lambda_1 L^{\frac{1}{2}}} \Big(& \gamma_0 \left(|s_{0,k}|^{\frac{1}{2}} + \mu |s_{0,k}|^{\frac{3}{2}} \right) \\ & + h \gamma_1 L^{\frac{1}{2}} \left(2\mu |s_{0,k}| + \frac{3}{2} \mu^2 s_{0,k}^2 + \frac{1}{2} \right) \Big) \, \mathrm{I_d} + \mathbf{N}_{[-1,1]}. \end{aligned} \tag{153}$$

Finally, we can recover

$$z_{1,k+1} = z_{1,k} + h \gamma_1 L \left(2\mu |s_{0,k}| + \frac{3}{2} \mu^2 s_{0,k}^2 + \frac{1}{2} \right) \lambda_{k+1}, \tag{154}$$

which is the URED counterpart of (121). Thus, (112), (113), and (115), (121), and (148), (150), and (151), (154) make nonlinear difference equations with state vector $(s_{0,k} \;\; z_{1,k})^\top$. Some of these may be named *higher-order* proximal-point algorithms since inserting, for example, (113) into (112) 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 [84, sections 5.1, 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

$$\mathbf{M}_{k,h}(w) = \sum_{l=0}^{2} h^l \gamma_l L^{\frac{l+1}{4}} \lceil w \rfloor^{\frac{3-l}{4}} + h^3 \gamma_3 L \, \mathbf{sgn}(w) \tag{155}$$

for the implicit method, and

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

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 super-twisting and the URED, yielding other higher-order proximal-point algorithms.

4.3 Relay polynomial controllers

As mentioned above, 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 such objective (see, *e.g.*, the *quasi-continuous* or the *nested* controllers [96]). The following algorithm is one of the simplest. Consider a perturbed chain of integrators

$$\begin{aligned} \dot{x}_i(t) &= x_{i+1}(t)\,, \quad i = 1, \dots, n-1, \\ \dot{x}_n(t) &= u(t) + \delta(t). \end{aligned} \tag{157}$$

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

$$\hat{u}(x) \in -\gamma_n \, \mathbf{sgn}\left(\lceil x_n \rfloor^n + \gamma_{n-1} \lceil x_{n-1} \rfloor^{\frac{n}{2}} + \dots + \gamma_1 x_1 \right) \tag{158}$$

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 [97, 76]. The forward-Euler discretization of the open-loop system is

$$\begin{aligned} x_{i,k+1} &= x_{i,k} + h x_{i+1,k}\,, \quad i = 1, \dots, n-1, \\ x_{n,k+1} &= x_{n,k} + h u_k + \delta_k. \end{aligned} \tag{159}$$

The backward-Euler discretization of (158) takes the implicit form $u_k \in -\gamma_n \, \mathbf{M}_{k,h}(u_k)$ with

$$\begin{aligned} \mathbf{M}_{k,h}(w) = \mathbf{sgn}\Big(\lceil x_{n,k} + hw \rfloor^n + \gamma_{n-1} \lceil x_{n-1,k} + h x_{n,k} \rfloor^{\frac{n}{2}} \\ + \dots + \gamma_1 (x_{1,k} + h x_{2,k}) \Big)\,. \end{aligned} \tag{160}$$

It is not difficult to see that, for every x_k, $\mathbf{M}_{k,h}$ is maximal monotone, hence a unique solution $u_k(x_k) = (\mathrm{I_d} + \gamma_n \, \mathbf{M}_{k,h})^{-1}(0)$ 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

$$u_k \in -\gamma_n \, \mathbf{sgn}(u_k - \bar{u}(x_k)), \tag{161}$$

where

$$\begin{aligned} \bar{u}(x_k) = -\frac{1}{h} \Big(\Big\lceil \gamma_{n-1} \lceil x_{n-1,k} + h x_{n,k} \rfloor^{\frac{\alpha}{2}} + \dots \\ + \gamma_1 \lceil x_{1,k} + h x_{2,k} \rfloor^{\frac{\alpha}{n}} \Big\rfloor^{\frac{1}{\alpha}} + x_{n,k} \Big)\,. \end{aligned} \tag{162}$$

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

$$u_k = \min\{|\bar{u}(x_k)|, \gamma_n\} \, \mathbf{sgn}(\bar{u}(x_k)), \tag{163}$$

which once again is a form of minimum-operator formalism as (13c). Consider now the ZOH discrete model of the chain of integrators,

$$x_{i,k+1} = x_{i,k} + \sum_{j=1}^{n-i} \frac{h^j}{j!} x_{i+j,k} + \frac{h^{n+1-i}}{(n+1-i)!} (u_k + \bar{\delta}_{i,k}), \tag{164}$$

where $i = 1, \dots n$ and $\bar{\delta}_i$ are (possibly unmatched) disturbances resulting from $\delta(t)$. Consider also the inclusion $u_k \in -\gamma_n \, \mathbf{M}_{k,h}(u_k)$, where

$$\begin{aligned} \mathbf{M}_{k,h}(w) = \mathbf{sgn}\Big(\lceil x_{n,k} + hw \rfloor^n \\ + \gamma_{n-1} \left\lceil x_{n-1,k} + h x_{n,k} + \frac{h^2}{2!} w \right\rfloor^{\frac{n}{2}} + \\ \dots + \gamma_1 \big(x_{1,k} + \sum_{j=1}^{n-1} \frac{h^j}{j!} x_{1+j,k} + \frac{h^n}{n!} w \big) \Big). \end{aligned} \tag{165}$$

For each fixed x_k, the operator $\mathbf{M}_{k,h}$ is again maximal monotone, so the resolvent $(\mathrm{I_d} + \mathbf{M}_{k,h})^{-1}$ has domain $\mathbb{R}$ and is single-valued. While an analytic expression is no longer available, it is still possible to compute it numerically by using the techniques described in Section "6".

4.4 Stability Analysis

The closed-loop systems (112), (113), or (114), or (148), (150), or its semi-implicit version in (151), (154), are higher-order proximal-point algorithms which do not fit within the robust proximal-point algorithm in (57) in "Sidebar VII". Boundedness of the closed-loop state with (128) is proved only in the absence of perturbations in [34]. As shown in [86], the algorithm is not robust with respect to an unbounded perturbation. A modification to (128) is proposed in [86] that enhances robustness. Some results concerning higher-order differentiators of the implicit AO-STD type can be found in [84]. Some properties of the implicit higher-order differentiators (sliding surface invariance, accuracy, finite convergence) are shown in [84, 91]. The stability of the semi-implicit super-twisting controller is analysed in [35], see also the semi-implicit discretization [31, Equation (60)] of the super-twisting velocity observer [98].

4.5 Homogeneity and maximal monotonicity

Homogeneity is a powerful tool for control design [99]. It plays an important role in all of the higher-order sliding-mode algorithms presented above. While homogeneity has no clear discrete-time counterpart, it is possible to construct consistent discretizations by keeping track of the maximal monotone operators that appear

Backward-Euler discretization of sliding-mode systems gives rise to new variants of proximal-point algorithms.

in the homogeneous vector fields. Consistent discretization of finite-time convergent systems have been presented in [19, Definitions 2.1, 2.2]. As an example, let us consider the systems analysed in [19, Section 6.2]. The first analysis concerns the discretization of a generalized homogeneous closed-loop system with homogeneity degree $d = 1$ (see [99] for definitions). After a suitable transformation into another homogeneous system, the consistent discrete-time dynamics

$$x_{k+1} = x_k - h\|x_k\|(I_n - \tilde{A})x_{k+1} \tag{166}$$

is obtained [19, Section 6.2, Equation (6.8)]. Similar to the discrete dynamics presented in the subsections above, this dynamics is given by the resolvent of a maximal monotone operator. Indeed, these are equivalent to

$$x_{k+1} = \left(\mathrm{I_d} + h\|x_k\|(I_n - \tilde{A})\right)^{-1}(x_k), \quad k \geq 0, \tag{167}$$

with $\tilde{A}^\top P + P\tilde{A} = 0$ for some $P = P^\top \succ 0$ [19, Definition 2.1]. Let us define $R = R^\top \succ 0$, $R^2 = P$, 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

$$\bar{x}_{k+1} = \bar{x}_k - h\|R^{-1}\bar{x}_k\|R(I_n - \tilde{A})R^{-1}\bar{x}_{k+1}, \tag{168}$$

which is equivalent to

$$\bar{x}_{k+1} = \left(\mathrm{I_d} + h\|R^{-1}\bar{x}_k\|R(I_n - \tilde{A})R^{-1}\right)^{-1}(\bar{x}_k), \quad k \geq 0. \tag{169}$$

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 $\bar{x}_{k+1} = \mathcal{J}_{\mathbf{M}_k}(\bar{x}_k)$, it may be named a *time-varying proximal-point* algorithm.

A second case, with homogeneity degree $d = -1$, is treated in [19, Section 6.2, Equation (6.4)]. It yields the consistent discretization

$$x_{k+1} \in x_k - h(I_n - \tilde{A})\tilde{F}(x_{k+1}), \tag{170}$$

which is equivalent to

$$x_{k+1} = \left(\mathrm{I_d} + h(I_n - \tilde{A})\tilde{F}\right)^{-1}(x_k) \tag{171}$$

with

$$\tilde{F}(x) = \begin{cases} \frac{x}{\|x\|_P} & \text{if } x \neq 0 \\ \mathcal{B}_P & \text{if } x = 0, \end{cases} \tag{172}$$

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

$$\begin{aligned} \bar{x}_{k+1} &\in \bar{x}_k - hR(I_n - \tilde{A})R^{-1}R\tilde{F}(x_{k+1}) \\ &= \bar{x}_k - hR(I_n - \tilde{A})R^{-1}\partial\|\bar{x}_{k+1}\|_2. \end{aligned} \tag{173}$$

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, let us proceed as in the proof of [19, Theorem 4.1]. For all $\bar{x}_k \neq 0$,

$$\begin{aligned} \bar{x}_k^\top \bar{x}_k &= \bar{x}_{k+1}^\top \bar{x}_{k+1} + 2h\bar{x}_{k+1}^\top \Lambda\partial\|\bar{x}_{k+1}\|_2 \\ &\quad + h^2(\Lambda\partial\|z_{k+1}\|_2)^\top(\Lambda\partial\|z_{k+1}\|_2) && (174) \\ &= \bar{x}_{k+1}^\top \bar{x}_{k+1} + 2h\frac{\bar{x}_{k+1}^\top \Lambda \bar{x}_{k+1}}{\|\bar{x}_{k+1}\|_2} \\ &\quad + h^2(\Lambda\partial\|\bar{x}_{k+1}\|_2)^\top \Lambda\partial\|\bar{x}_{k+1}\|_2 && (175) \\ &> \bar{x}_{k+1}^\top \bar{x}_{k+1} + h^2(\Lambda\partial\|\bar{x}_{k+1}\|_2)^\top \Lambda\partial\|\bar{x}_{k+1}\|_2 && (176) \\ &= \bar{x}_{k+1}^\top \bar{x}_{k+1} + h^2\frac{\bar{x}_{k+1}^\top \Lambda^\top \Lambda \bar{x}_{k+1}}{\|\bar{x}_{k+1}\|_2^2}. && (177) \end{aligned}$$

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

$$\inf_{\xi \neq 0} h^2\frac{\xi^\top \Lambda^\top \Lambda \xi}{\|\xi\|_2^2} > 0, \tag{178}$$

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 (171) nor inside $\bar{x}_{k+1} = (\mathrm{I_d} + h\Lambda\partial\|\cdot\|_2)^{-1}(\bar{x}_k)$ are maximal monotone in general, finite-time convergence of this algorithm is guaranteed. Moreover the calculation is made in [19, Section 6.2]. As a side note, mark that (173) is equivalent to

$$\bar{x}_{k+1} \in \bar{x}_k - h(\Lambda_{\mathrm{sym}} + \Lambda_{\mathrm{sk}})\partial\|\bar{x}_{k+1}\|_2, \tag{179}$$

with $\Lambda_{\mathrm{sym}} = \frac{1}{2}(\Lambda + \Lambda^\top)$, $\Lambda_{\mathrm{sk}} = \frac{1}{2}(\Lambda - \Lambda^\top)$. Therefore, Equation (173) is interpreted as a discrete-time Hamiltonian system

$$\bar{x}_{k+1} \in \bar{x}_k + h(J - R)\partial H(\bar{x}_{k+1}) \tag{180}$$

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

$$\dot{x}(t) \in (J - R)\partial\|x(t)\|_2. \tag{181}$$

Thus, *Hamiltonian proximal-point* algorithms with resolvents $\mathcal{J}_{h\,\mathbf{M}} = (\mathrm{I_d} - h(J - R)\partial\|\cdot\|)^{-1}$, $\mathbf{M} = -(J - R)\partial\|\cdot\|$ may be introduced. More details on the discretization of Hamiltonian systems can be found in [100].

5 FURTHER IMPLICIT ALGORITHMS

5.1 Other Classes of Set-Valued Algorithms

The implicit discretization for first-order sliding-mode controllers, super-twisting controller, observer and differentiator, and higher-order differentiators, has been studied in the foregoing sections. The implicit discretization of other classes of algorithms has been analysed in the literature: terminal sliding-mode control for first and second order systems [41, 101], twisting controller [102], proxy-based sliding-mode [32, 94, 103], homogeneous differentiators and observers [38, 39], super-twisting velocity observer [31], other types of differentiators [94, 95, 104, 105, 106, 107], the differentiator proposed in [5] discretized in [84], sliding-mode control of nonsmooth actuators [36], homogeneous systems and controllers [19, 108], nested controllers for second-order systems with unmatched perturbations [44], first-order controllers applied to Lagrangian nonlinear systems with parameter uncertainties [42] and to linear hyperbolic infinite dimensional systems [109], fixed-time or prescribed-time schemes [19, 110, 111, 101], adaptive sliding-mode algorithms [112, 113, 114], sliding-mode control with reaching law [115], fractional-order differentiators [116, 117], multivariable super-twisting [118], modified super-twisting using minimum-operator-based algorithms (similar to (13c)) [119].

The articles mentioned above study implicit or semi-implicit schemes. The solvability of the one-step nonsmooth problem (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 analyse the discrete solutions convergence towards the solutions of the continuous-time closed-loop. The analysis made in foregoing sections focuses on the first step, which is necessary for the implementation.

5.2 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 section "7"). Nevertheless, open issues have emerged which show that room exists for significant improvements. The implicit method has been studied in [102] for the twisting scheme and in [34] for the super-twisting one. However, the straightforward implicit discretization of the twisting scheme (*i.e.*, the implicit or backward-Euler emulation) does not guarantee finite-time convergence to the origin [120]: a modification of the emulated implicit method is proposed in [102] which secures it. As noticed in [86], the implicit emulation of the super-twisting scheme (127), (128) does not yield acceptable robustness (a fact also noticed experimentally in [121], see also [122] for the influence of the plant discretization). Again, a suitable modification of the implicit emulation allows us to enhance it. Also, implicit emulation discretization of some higher-order differentiators may still undergo numerical chattering, and suitable modifications have to be made to avoid it [123]. 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].

6 COMPUTATIONAL ISSUES

The analysis of closed-loop systems with implicit discrete-time controllers and the computation of output differentiators, relies on computing proximal maps to convex functions, or more generally, resolvents of maximal monotone operators (in order to solve the associated generalized equations). Figure 5 depicts the steps to follow in order to compute the resolvent of the sign multifunction. However, as seen in foregoing 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 influence significantly the closed-loop behavior in some applications [40, 83, 90, 124].

6.1 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

$$\begin{aligned} \zeta_{k+1} &= \varphi(x_k) + \rho\eta(x_k) \\ -\eta(x_k) &\in \mathbf{M}(\zeta_{k+1}) \end{aligned} \tag{182a}$$

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

$$\begin{aligned} -\eta(x_k) &= \frac{1}{\rho}\varphi(x_k) - \frac{1}{\rho}\zeta_{k+1} \\ \zeta_{k+1} &\in \mathbf{M}^{-1}(-\eta(x_k)) \end{aligned}, \tag{182b}$$

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 equation (71) has this form with $\zeta_{k+1} = \tilde{s}'_{k+1}$, $\eta(x_k) = u_{\mathrm{sv}}(x_k)$,

$$\mathbf{M} = \partial \left\|\cdot\right\|_1 \circ \tilde{K}(\tilde{B}^\top \tilde{P}^{-1} \tilde{B})^{\frac{1}{2}}, \tag{183}$$

and $\varphi(x_k) = \tilde{L}x_k$. Likewise, equation (95) with $\zeta_{k+1} = \tilde{s}'_{k+1}$, $\eta(x_k) = u_{\mathrm{sv}}(x_k)$, $\mathbf{M} = \partial \left\|\cdot\right\|_1 \circ \tilde{K}^{\frac{1}{2}}$, and $\varphi(x_k) = \tilde{K}^{-\frac{1}{2}} s_k$ belongs to this class. Equations (124)–(130) 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 (111), $\varphi(x_k) = x_{0,k} + h\nu_k$. Equations (148)–(149) are an instance of (182) 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}

Maximal monotone operators and their resolvents provide a powerful and unified tool for implicit algorithms computation.

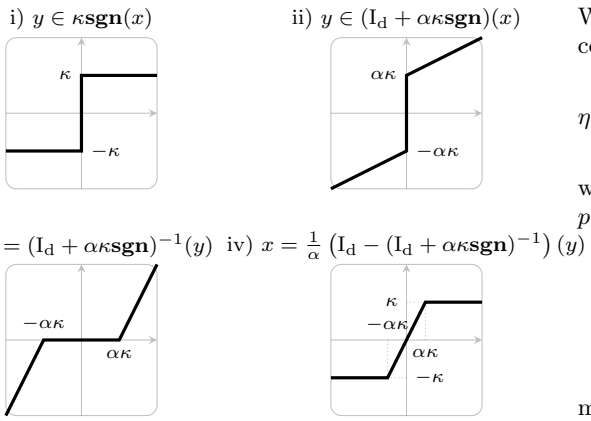

Figure 5: Graphical computation of resolvent and Yosida approximation (lower left and right figures, respectively) for the **sgn** multifunction in the scalar case. Note that, at each step, a maximal monotone operator is obtained.

for the generalized equation in (182a) 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 (182b) has the unknown $\eta(x_k)$ and can be interpreted as the dual, or reciprocal of (182a). From (182a) the *selection* $\eta(x_k)$ is explicitly given as

$$-\eta(x_k) = \frac{1}{\rho}\left(\mathrm{I_d} - \mathcal{J}_{\rho\,\mathbf{M}}\right)(\varphi(x_k)) = \mathcal{Y}_{\rho\,\mathbf{M}}(\varphi(x_k)) \tag{184}$$

(see (50) in "Sidebar VII"), whereas (182b) gives

$$-\eta(x_k) = \mathcal{J}_{\frac{1}{\rho}\mathbf{M}^{-1}}\left(\frac{1}{\rho}\varphi(x_k)\right). \tag{185}$$

The expressions (184)–(185) yield the same control law. The main difference relies on the proximal map to be computed. That is, for a certain $\mathbf{M}$, it could be easier to compute first the inverse map $\mathbf{M}^{-1}$ and then (185), while for other cases (184) might be a simpler choice. In any case, a convex optimization problem has to be solved at each time-step.

For $\mathbf{M} = \partial\left\|\cdot\right\|_p$, the calculation of the control inputs can be performed according to (39)–(41) in "Sidebar VI", so that

$$-\eta(x_k) = \mathrm{Proj}\left(\mathcal{B}_{p^\star}; \frac{1}{\rho}\varphi(x_k)\right). \tag{186}$$

When $p = 1$ then $p^\star = \infty$ and, according to (44), the control can be computed component-wise as

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

which is a minimum-operator formalism, whereas if $p = p^\star = 2$ then, according to (43), it is inferred that

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

For sliding-mode control, 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 non-singular, then it follows from [S28, Proposition 23.23] that

$$\begin{aligned} &\mathcal{J}_{\rho\,\mathbf{M}}(\varphi(x_k)) \\ &= \left(\mathrm{I_d} - R^\top \circ \left(RR^\top + (\partial\rho\|\cdot\|_p)^{-1}\right)^{-1} \circ R\right)(\varphi(x_k)) \\ &= \left(\mathrm{I_d} - R^\top \circ \left(RR^\top + \mathbf{N}_{\rho\mathcal{B}_{p^\star}}\right)^{-1} \circ R\right)(\varphi(x_k)) \\ &= \varphi(x_k) - R^\top\,\mathrm{Proj}^{RR^\top}\left(\rho\mathcal{B}_{p^\star}; R^{-\top}\varphi(x_k)\right) \end{aligned} \tag{187}$$

where we have made use of (39) to get the second equality and (45c) to get the last equality, both in the sidebar "Sidebar VI".

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

$$\mathcal{J}_{\frac{1}{\rho}\mathbf{M}^{-1}}\left(\frac{1}{\rho}\varphi(x_k)\right) = R^\top\theta, \tag{188}$$

where

$$\begin{aligned} \theta = \underset{\xi\in\mathbb{R}^n}{\arg\min}\Big\{ & g^\star(\xi) \\ & + \frac{\rho}{2}\left(\xi - \frac{1}{\rho}R^{-\top}\varphi(x_k)\right)RR^\top\left(\xi - \frac{1}{\rho}R^{-\top}\varphi(x_k)\right)\Big\}. \end{aligned} \tag{189}$$

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

$$\mathbf{M}(x) = \left(\sum_{i=0}^{N} a_i |x|^{\frac{p_i}{q_i}} \right) \mathbf{sgn}(x) + b_0, \tag{190}$$

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

$$\mathcal{J}_{\rho \mathbf{M}}(x) = \beta_{\mathbf{M}}(x, b_0, \rho)^C \, \mathbf{sgn}(x - \rho b_0), \tag{191}$$

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

$$\beta^C + \rho \sum_{i=1}^{N} a_i \beta^{r_i} - \max\{0, |x - \rho b_0| - \rho a_0\} \tag{192}$$

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

Proof. From the definition of resolvent in (23), it follows that $y = \mathcal{J}_{\rho \mathbf{M}}(x)$ if, and only if, $x \in y + \rho \mathbf{M}(y)$. That is,

$$x - \rho b_0 \in \left(|y| + \rho a_0 + \rho \sum_{i=1}^{N} a_i |y|^{\frac{p_i}{q_i}} \right) \mathbf{sgn}(y). \tag{193}$$

Since all the coefficients a_i are nonnegative, it follows from (193) 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 (193) it follows that $\mathbf{sgn}(x - \rho b_0) = \mathbf{sgn}(y)$, so that (193) becomes single-valued as

$$|y| + \rho \sum_{i=1}^{N} a_i |y|^{\frac{p_i}{q_i}} - (|x - \rho b_0| - \rho a_0) = 0. \tag{194}$$

Let $C > 0$ be the least common denominator of the set $\left\{ \frac{p_i}{q_i} \mid i = 1, \ldots N \right\}$. The change of variables $|y| = \beta^C$, where $\beta > 0$, transforms (194) into a root-finding problem for the polynomial

$$\beta^C + \rho \sum_{i=1}^{N} a_i \beta^{r_i} - (|x - \rho b_0| - \rho a_0), \tag{195}$$

where $r_i = C \frac{p_i}{q_i}$. Hence, in order to solve (194) let us look for the positive root of (195). Note that, the non-negativity of all a_i, $i \in \{1, \ldots, N\}$ together with Descartes' rule of signs [125, Theorem 2.23] imply the uniqueness of such positive root of (195) in the cases when $|x - \rho b_0| > \rho a_0$. The polynomial (192) includes both cases above. Indeed, if $|x - \rho b_0| \leq \rho a_0$, then the unique nonnegative root of (192) is $\beta = 0$, implying $y = 0$. Finally, it follows that

$$\mathcal{J}_{\rho \mathbf{M}}(x) = y = |y| \, \mathbf{sgn}(y) = \beta_{\mathbf{M}}(x, b_0, \rho)^C \, \mathbf{sgn}(x - \rho b_0). \tag{196}$$

The proof is complete. □

Instances of Proposition 9 have been independently developed in [84] and [83] in the context of higher-order differentiators, since (191) is the unique solution of the generalized equation

$$0 \in y + \sum_{i=0}^{N} a_i |y|^{\frac{p_i}{q_i}} \, \mathbf{sgn}(y) + b_0 - x. \tag{197}$$

6.2 Computation of proximal maps via splittings

For complex strategies such as higher-order sliding-mode control and differentiation, the associated monotone operator is composed as the addition of several simpler operators. In this situation, the splitting algorithms in "Sidebar X" can be used to compute the associated proximal map in an iterative way.

Example 1. *The splitting* (89) *can be used to compute the URED proximal map in* (148), (149) *as follows. Consider the optimization problem* (84) *with* $A = 1$ *and*

$$f(\xi) = \frac{2}{3} h \gamma_0 L^{\frac{1}{2}} |\xi|^{\frac{3}{2}} + \frac{1}{2} h^2 \gamma_1 L |\xi|, \tag{198a}$$

$$g(\xi) = \frac{1}{2} h^2 \mu^2 \gamma_1 L |\xi|^3, \tag{198b}$$

$$h(\xi) = \frac{2}{5} h \gamma_0 L^{\frac{1}{2}} \mu |\xi|^{\frac{5}{2}} + h^2 \gamma_1 L \mu \xi^2 + \frac{1}{2} (\xi + b_k)^2. \tag{198c}$$

With the given parameters, the solution of (84) *indeed gives the proximal map* $\mathcal{J}_{h \mathbf{M}_h}(-b_k)$, *where* $\mathbf{M}_h$ *is as in* (149). *Also, note that* $\nabla h(\cdot)$ *is Lipschitz continuous in any bounded set of* $\mathbb{R}$, *since*

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

Notice that, with such decomposition, each map appearing in (89) *is proximable. Indeed, it follows from Proposition 9 that*

$$\mathrm{Prox}_{\alpha_1 f}(x) = \mathcal{J}_{\alpha_1 \partial f}(x) = \beta_{\partial f}(x, \alpha_1)^2 \, \mathbf{sgn}(x), \tag{200}$$

where

$$\beta_{\partial f}(x, \alpha_1) = \frac{-\alpha_1 h \gamma_0 L^{\frac{1}{2}} + \sqrt{\left(\alpha_1 h \gamma_0 L^{\frac{1}{2}} \right)^2 + 4\rho(x)}}{2}, \tag{201}$$

where $\rho(x) = \max\left\{ 0, |x| - \frac{\alpha_1 h^2 \gamma_1 L}{2} \right\}$. *Now, in order to compute the proximal map associated with* $g^\star(\cdot)$ *in* (89), *we make use of Moreau's decomposition (see, e.g., [S28, Theorem 14.3] in "Sidebar V"),*

$$x = \mathrm{Prox}_{\alpha_2 g^\star}(x) + \alpha_2 \, \mathrm{Prox}_{\frac{1}{\alpha_2} g}\left(\frac{x}{\alpha_2} \right). \tag{202}$$

Thus, let us compute $\mathrm{Prox}_{\frac{1}{\alpha_2} g}\left(\frac{x}{\alpha_2} \right)$ *using once again Proposition 9, so that*

$$\mathrm{Prox}_{\frac{1}{\alpha_2} g}\left(\frac{x}{\alpha_2} \right) = \beta_{\partial g}\left(\frac{x}{\alpha_2}, \frac{1}{\alpha_2} \right) \mathbf{sgn}(x), \tag{203}$$

where

$$\beta_{\partial g}(x,\rho)=\frac{-1+\sqrt{1+6h^2\mu^2\gamma_1 L\rho|x|}}{3h^2\mu^2\gamma_1 L\rho}. \tag{204}$$

Finally, it follows from (202) *and* (203) *that*

$$\begin{aligned}\mathrm{Prox}_{\alpha_2 g^\star}(x)&=x-\alpha_2\,\mathrm{Prox}_{\frac{1}{\alpha_2}g}\left(\frac{x}{\alpha_2}\right)\\&=x-\alpha_2\beta_{\partial g}(x,\rho)\,\mathbf{sgn}(x).\end{aligned} \tag{205}$$

Therefore, by using the expressions (200) *and* (205) *in the iteration* (89) *and by selecting the parameters* α_1 *and* α_2 *sufficiently small such that* (90) *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 as those described in [126, S35] can be used.

Example 2. *The Douglas-Rachford splitting can also be used to compute the proximal map in* (148), (149). *This time a splitting of the following form is considered:*

$$f(\xi)=\frac{2}{5}h\gamma_0 L^{\frac{1}{2}}\mu|\xi|^{\frac{5}{2}}+\frac{2}{3}h\gamma_0 L^{\frac{1}{2}}|\xi|^{\frac{3}{2}}+\frac{1}{2}h^2\gamma_1 L|\xi|, \tag{206}$$

$$g(\xi)=\frac{1}{2}h^2\mu^2\gamma_1 L|\xi|^3+h^2\gamma_1 L\mu\xi^2+\frac{1}{2}(\xi+b_k)^2. \tag{207}$$

From Proposition 9, the expressions

$$\mathrm{Prox}_{\alpha f}(x)=\beta_{\partial f}(x,\alpha)^2\,\mathbf{sgn}(x), \tag{208}$$

$$\mathrm{Prox}_{\alpha g}(x)=\beta_{\partial g}(x,b_k,\alpha)\,\mathbf{sgn}(x-\alpha b_k), \tag{209}$$

are obtained, where $\beta_{\partial f}(x,\alpha)$ *is the unique positive root of the cubic polynomial*

$$\alpha h\lambda_0 L^{\frac{1}{2}}\mu\beta^3+\beta^2+\alpha\lambda_0 L^{\frac{1}{2}}\beta-\max\left\{0,|x|-\frac{\alpha h^2\lambda_1 L}{2}\right\}, \tag{210}$$

that is,

$$\begin{aligned}\beta_{\partial g}(x,b_k,\alpha)=&\frac{-2(1+h^2\mu\lambda_1 L)}{3h^2\mu^2\lambda_1 L}\\&+\frac{\sqrt{4(1+h^2\mu\lambda_1 L)^2+6h^2\mu^2\lambda_1 L|x-b_k|}}{3h^2\mu^2\lambda_1 L}.\end{aligned} \tag{211}$$

Hence, by taking the Douglas-Rachford iteration (94) *with* $\mathrm{Prox}_{\alpha f}$ *and* $\mathrm{Prox}_{\alpha g}$ *as in* (208)–(211), *we have that* ζ_j *converges to the resolvent* (148)–(149) *as* $j\uparrow+\infty$.

6.3 Relaxation

One of the main objectives of sliding-mode control concerns the finite-time stability of the sliding variable, whose discrete-time dynamics are of the form

$$s_{k+1}=s_k+\rho\,u_{\mathrm{sv}}(x_k)+\delta_k. \tag{212}$$

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 in considering a copy of the nominal behavior as in (182) One way of accelerating the convergence of $\tilde{s}_k$ towards the origin consists in setting the control law as

$$-u_{\mathrm{sv}}(x_k)\in\partial f(\tilde{s}_{k+1}) \tag{213}$$

$$\tilde{s}_{k+1}=s_k+\mu\,u_{\mathrm{sv}}(x_k), \tag{214}$$

where $\mu>0$, so that the closed-loop is given as

$$\tilde{s}_{k+1}=\mathrm{Prox}_{\mu f}(s_k) \tag{215a}$$

$$s_{k+1}=s_k+\frac{\rho}{\mu}(\tilde{s}_{k+1}-s_k)+\delta_k. \tag{215b}$$

The iteration (215) is known as a relaxed iteration in the literature of proximal-point algorithms [127]. In the cases when $\delta_k\equiv 0$, it has been shown that (215) converges towards a minimum of $f(\cdot)$ [S28, Corollary 5.16]. Moreover, for certain values of $\frac{\rho}{\mu}\in(0,2)$, the algorithm (215) has a faster convergence with respect to the case $\rho=\mu$, see, *e.g.*, [127].

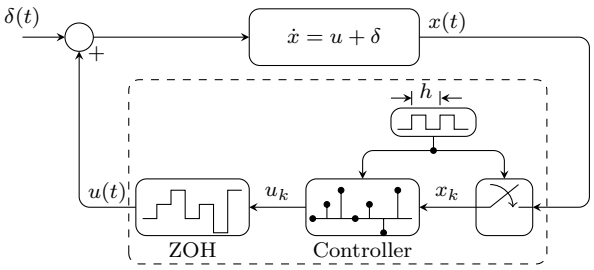

Figure 6: Closed-loop system. The state of the continuous-time dynamics is sampled every h seconds. Such information is used for computing the control input u_k which passes through a classical zero-order-hold (ZOH) mechanism to obtain the feedback input signal $u(t)=u_k$ for $kh<t\leq(k+1)h$.

6.4 Numerical Experiments

Taking advantage of the fact that the super-twisting controller can be calculated in the closed form (128), it is used as a benchmark to test algorithms which 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. Let us consider a perturbed integrator in feedback with a discrete-time controller as shown in Figure 6. The target amounts to driving the state $x_0(t)$ towards the origin in the presence of the persistent disturbance $\delta(t)=\cos(\pi t)\sin\left(5\sqrt{\frac{t}{3}}\right)$. To this end, let us consider the super-twisting algorithm, taking in to account the two main approaches for its discrete-time implementation, *i.e.*, emulation of the continuous-time controller and discrete-time design. For the emulation part, *i*) the explicit (forward) Euler discretization,

ii) the implicit (backward) Euler discretization in [34] and *iii*) the semi-implicit discretization in [35], are considered. In each case the control law takes the form

$$\begin{aligned} u_k &= -\gamma_0 L^{\frac{1}{2}} |x_{0,k}|^{\frac{1}{2}} \operatorname{\mathbf{sgn}}(x_{0,k}) + \nu_k \\ \nu_{k+1} &= \nu_k - h\gamma_1 L \operatorname{\mathbf{sgn}}(x_{0,k}), \end{aligned} \tag{216}$$

and

$$\begin{aligned} u_k &= -\gamma_0 L^{\frac{1}{2}} \beta_k \operatorname{\mathbf{sgn}}(x_{0,k} + h\nu_k) + \nu_k \\ &\quad - h\gamma_1 L \operatorname{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h^2\gamma_1 L}\right) \\ \beta_k &= \frac{-h\gamma_0 L^{\frac{1}{2}}}{2} + \sqrt{\frac{h^2\gamma_0^2 L}{4} + \max\{0, |x_{0,k} + h\nu_k| - h^2\gamma_1 L\}} \\ \nu_{k+1} &= \nu_k - h\gamma_1 L \operatorname{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h^2\gamma_1 L}\right), \end{aligned} \tag{217}$$

and

$$\begin{aligned} u_k &= \nu_k - \mu_k \operatorname{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h\mu_k}\right) \\ \nu_{k+1} &= \nu_k - h\gamma_1 \operatorname{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h\mu_k}\right) \\ \mu_k &= \gamma_0 L^{\frac{1}{2}} |x_{0,k}|^{\frac{1}{2}} + h\gamma_1 L. \end{aligned} \tag{218}$$

For the second approach, which relies on the direct design in discrete-time, the following controller, which is proposed in [86], is considered:

$$\begin{aligned} u_k &= -\gamma_0 \beta_k \operatorname{\mathbf{sgn}}(x_{1,k}) - 2h\gamma_1 \operatorname{Proj}\left([-1,1]; \frac{x_{1,k}}{h^2\gamma_1}\right) + \nu_k \\ \beta_k &= -\frac{h\gamma_0}{2} + \frac{1}{2}\sqrt{(h\gamma_0)^2 + 4\max\{0, |x_{1,k}| - h^2\gamma_1\}} \\ \nu_{k+1} &= \nu_k - h\gamma_1 \operatorname{Proj}\left([-1,1]; \frac{x_{1,k}}{h^2\gamma_1}\right) \end{aligned} \tag{219}$$

The controller (219) is in explicit form, though strongly motivated from the implicit controller obtained from the backward-Euler discretization (217). In addition, the splitting schemes (89) and (94) are considered, applied to the implicit super-twisting algorithm in (110). That is, for solving the minimization problem

$$\min_{\xi \in \mathbb{R}} \frac{2}{3} h\gamma_0 L^{\frac{1}{2}} |\xi|^{\frac{3}{2}} + h^2\gamma_1 L|\xi| + \frac{1}{2}\left(\xi - (x_{0,k} + h\nu_k)\right)^2. \tag{220}$$

As a last scheme to compare, the problem (220) is solved numerically at each time step by using the interior point solver *Clarabel* [128]. The listings 1 and 2 show the code implementation in Python 3 used for the Douglas-Rachford and Condat-Vũ splittings, respectively.

For the simulations, the initial condition $x_0(0) = 5$ is taken and several discretization steps $100\mu s \leq h \leq 1s$ are considered. For each simulation a final time of $t_f = 20s$ is chosen. Figure 7 displays the L_2-norm, 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

$$\operatorname{SVar}(\{\xi_k\}) = \sum_{k=0}^{N-2} |\xi_{k+1} - \xi_k|. \tag{221}$$

For large sampling times, $0.5s < h < 1s$, all the 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 (219) and (216) are less performing. Indeed, for the explicit controller (216), $\operatorname{SVar}_{[0,20]}(\{u_k\}) \approx 500$ units, regardless of the sampling time. For smaller sampling times, *i.e.*, $2ms < h < 20ms$, 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 (216) is also higher that in the other cases. Finally, when $h < 2ms$, 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 (216)). Such increment in the step-to-step variation, whilst maintaining the L_2-norm unchanged, is a sign of the presence of chattering in the input.

Listing 1: Douglas-Rachford splitting for the super-twisting controller (217)

```
import numpy as np

def bracket(x, a):
    return np.sign(x)*np.abs(x)**a

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)

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)

def sta_dgl_rfd(h_c, g0, g1, x_0_k, nu, x_0_tilde, w,
     N):
    r = x_0_k + h_c*nu[-1]
    z = w[-1]
    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)
    x_0_next_tilde = J_A1(z, h_c, g1)
    u_k = -(x_0_k - x_0_next_tilde)/h_c
    nu_next = u_k + g0*bracket(x_0_next_tilde, 0.5)
    x_0_tilde.append(x_0_next_tilde)
    nu.append(nu_next)
    w.append(z)
    return u_k
```

Listing 2: Condat-Vũ splitting for the super-twisting controller (217)

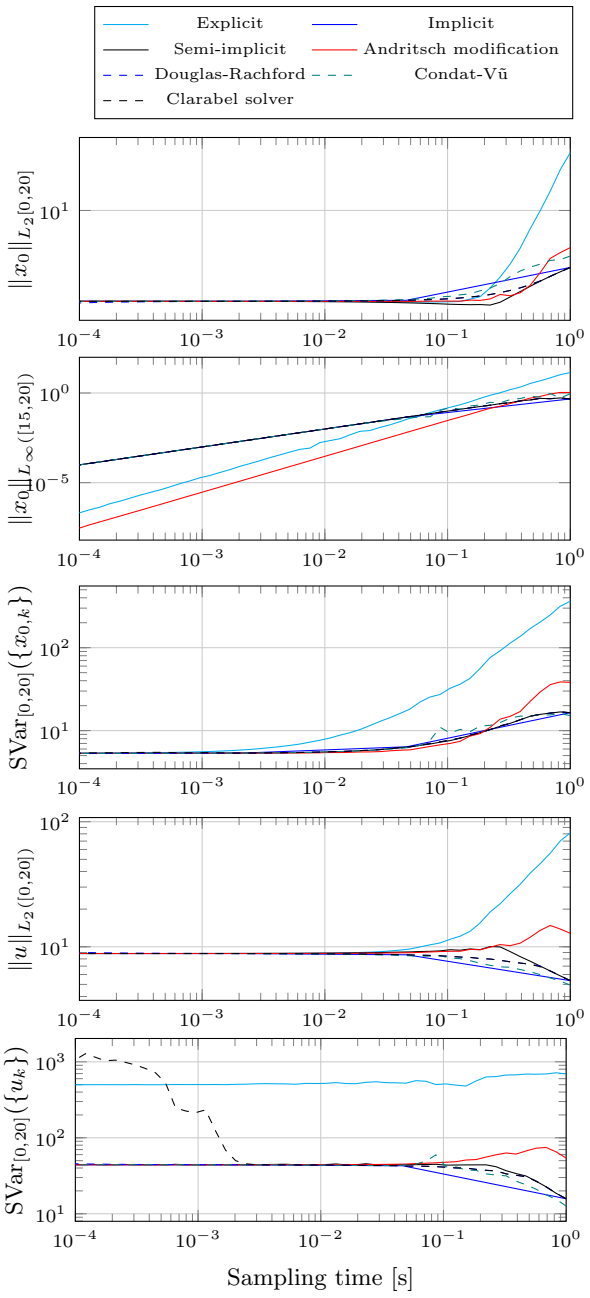

Figure 7: Norms and total variations of state and control signals for several discrete-time super-twisting controllers. In terms of the L_2-norm of the state x_0 and the control input u, the implicit controller (217) (and its associated splittings); the semi-implicit controller (218); and the numerical solver for (220), have a similar behavior for all sampling times. In terms of the step-to-step variation (221) of the control input sequence, the explicit controller shows the larger values. It is noteworthy that for sampling times $h < 2ms$, 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 (219) shows the best performance for $h \leq 200ms$.

```
import numpy as np

def bracket(x, a):
    return np.sign(x)*np.abs(x)**a

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)

def J_B1(x, a):
    return x if np.abs(x)<=a else a*np.sign(x)

def sta_cdt_vu(h_c, g0, g1, x_0_k, nu, x_0_tilde, mu,
     N, a1, a2):
    z=x_0_tilde[-1]
    m = mu[-1]
    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
    x_0_next_tilde = z
    u_k = -(x_0_k - x_0_next_tilde)/h_c
    nu_next = u_k + g0*bracket(x_0_next_tilde,0.5)
    x_0_tilde.append(x_0_next_tilde)
    nu.append(nu_next)
    mu.append(m)
    return u_k
```

7 EXPERIMENTAL RESULTS AND VALIDATIONS

7.1 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 (77), (98), (137)), moderate deterioration of the performance with increasing sampling times, significant chattering alleviation at both outputs and inputs, are validated experimentally, see, *e.g.*, [21, 33, 130, 131]. 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 [33, 130, 131], the twisting controller [130], several higher order and homogeneous differentiators have been implemented on an electropneumatic system [90, 124, 38] and compared. Homogeneous differentiators have been tested on an RLC circuit [132], differentiators have been implemented on DC-DC buck converters in [133]. Various discretizations of the classical first-order sliding-mode control [21], a switching input upgrading linear control [134], several higher-order differentiators [90, 124, 135] have been tested and compared in closed-loop on the inverted pendulum and the rotary inverted pendulum (including cascaded first-order differentiators). Set-valued admittance controllers [136, 107, 137, 138, 139], first-order differentiators [94],

Sidebar X: 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

$$0 \in \mathbf{M}_1(\xi) + \mathbf{M}_2(\xi). \tag{81}$$

Under constraint-qualification assumptions guaranteeing that $\mathbf{M}_1 + \mathbf{M}_2$ is maximal monotone, it follows from (23) in "Sidebar V" that problem (81) is equivalent to finding $\xi \in \mathbb{R}^n$ such that

$$\xi = \mathcal{J}_{\rho(\mathbf{M}_1 + \mathbf{M}_2)}(\xi) \tag{82}$$

for any $\rho > 0$. The formulation (82) allows for a search of a solution to (81) in an iterative way. Indeed, it follows from the contents of "Sidebar VII" that the fixed-point iteration

$$\xi_{k+1} = \mathcal{J}_{\rho(\mathbf{M}_1 + \mathbf{M}_2)}(\xi_k) \tag{83}$$

converges towards a solution of (82). However, an explicit expression for the resolvent map of the sum $\mathbf{M}_1 + \mathbf{M}_2$ in (83) may not be available. To solve such issue, proximal splitting schemes are applied to compute alternative iterations that converge to a solution of (81) and that only involve the individual resolvents of $\mathbf{M}_1(\cdot)$ and $\mathbf{M}_2(\cdot)$ at each iteration. In what follows, two popular splitting schemes are discussed: the Condat-Vũ and the Douglas-Rachford splittings.

6.5 Condat-Vũ splitting

The Condat-Vũ splitting was independently proposed in [S43] and [S44] for solving the optimization problem

$$\min_{\xi \in \mathbb{R}^n} f(\xi) + g(A\xi) + h(\xi), \tag{84}$$

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 (51) and (53) in "Sidebar VII"), 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 (127) and (148)). The splitting relies on the fact that $g(A\xi) = \sup_{\mu \in \mathbb{R}^m} \{\langle A\xi, \mu \rangle - g^\star(\mu)\}$ (Fenchel-Moreau theorem [S28, Theorem 13.32]) so that, as pointed out in [S43], the primal problem (84) can be rewritten as

$$\min_{\xi \in \mathbb{R}^n} \sup_{\mu \in \mathbb{R}^m} \mathcal{L}(\xi, \mu), \tag{85}$$

where $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is of the form

$$\mathcal{L}(\xi, \mu) = f(\xi) + h(\xi) + \langle \mu, A\xi \rangle - g^\star(\mu). \tag{86}$$

Thus, the saddle points of $\mathcal{L}(\cdot)$ are characterized by the following generalized equations:

$$0 \in \nabla h(\xi) + A^\top \mu + \partial f(\xi), \tag{87a}$$
$$0 \in -A\xi + \partial g^\star(\mu). \tag{87b}$$

Proceeding in a similar way as above, (87) is equivalent to the following fixed-point conditions:

$$\xi = \mathrm{Prox}_{\alpha_1 f}\left(\xi - \alpha_1(\nabla h(\xi) + A^\top \mu)\right), \tag{88a}$$
$$\mu = \mathrm{Prox}_{\alpha_2 g^\star}(\mu + \alpha_2 A\xi), \tag{88b}$$

for any $\alpha_1, \alpha_2 > 0$. It is well-known that, if $\xi^\star$ and $\mu^\star$ are solutions of the primal problem (84) and its associated dual problem, respectively, then they also satisfy (88), see *e.g.*, [S28, Proposition 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 total duality assumption, the following iteration was proposed in [S43, S44]:

$$\xi_{j+1} = \mathrm{Prox}_{\alpha_1 f}\left(\xi_j - \alpha_1(\nabla h(\xi_j) + A^\top \mu_j)\right), \tag{89a}$$
$$\mu_{j+1} = \mathrm{Prox}_{\alpha_2 g^\star}\left(\mu_j + \alpha_2 A(2\xi_{j+1} - \xi_j)\right). \tag{89b}$$

Sidebar (cont.)

Thus, if the gradient of $h(\cdot)$ is Lipschitz continuous with constant ℓ, and

$$\frac{\alpha_1 \ell}{2} + \alpha_1 \alpha_2 \lambda_{\max}(A^\top A) < 1, \tag{90}$$

then the iteration (89) converges towards solutions of (88), implying that the sequence $\{\xi_k\}_{k\in\mathbb{N}}$ converges to a solution of (84).

6.6 Douglas-Rachford splitting

Similar to the previous approach, the Douglas-Rachford splitting can be used to approximate proximal maps composed by the sum of two maximal monotone operators. Let us start considering the problem of finding $\xi \in \mathbb{R}^n$ such that

$$0 \in \partial f(\xi) + \partial g(\xi), \tag{91}$$

where, as before, $f(\cdot)$ and $g(\cdot)$ are proper, convex, LSC functions. Thus, as pointed out in [S35, p. 44] the following equivalence is true:

$$0 \in \partial f(\xi) + \partial g(\xi) \\ \Leftrightarrow z = (\mathcal{R}_{\alpha\partial f} \circ \mathcal{R}_{\alpha\partial g})(z), \text{ and } \xi = \operatorname{Prox}_{\alpha g}(z), \tag{92}$$

where $\mathcal{R}_{\alpha\partial f}(\cdot) = (2\operatorname{Prox}_{\alpha f} - \mathrm{I_d})(\cdot)$ is the reflected resolvent. Now, since the operator $\mathcal{R}_{\alpha\partial f} \circ \mathcal{R}_{\alpha\partial g}$ is simply non-expansive, a Picard iteration using (92) may not converge. To overcome such issue, the Douglas-Rachford splitting uses the following Krasnosel'skiĭ-Mann iteration

$$\begin{aligned} z_{j+1} &= \frac{1}{2} z_j + \frac{1}{2}(\mathcal{R}_{\alpha\partial f} \circ \mathcal{R}_{\alpha\partial g})(z_j) \\ &= \operatorname{Prox}_{\alpha f} \circ (2\operatorname{Prox}_{\alpha g} - \mathrm{I_d})(z_j) + (\mathrm{I_d} - \operatorname{Prox}_{\alpha g})(z_j), \end{aligned} \tag{93}$$

for which its convergence is guaranteed for any $\alpha > 0$ [129, Theorem 1]. Thus, the iteration

$$z_{j+1} = \operatorname{Prox}_{\alpha f} \circ (2\operatorname{Prox}_{\alpha g} - \mathrm{I_d})(z_j) + (\mathrm{I_d} - \operatorname{Prox}_{\alpha g})(z_j) \tag{94a}$$

$$\xi_{j+1} = \operatorname{Prox}_{\alpha g}(z_j) \tag{94b}$$

converges and ξ_{j+1} approaches a solution of (91) as $j \uparrow +\infty$.

References

[S43] L. Condat, A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158:460–479, 2013.

[S44] C.B. Vu, A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38:667–681,2013

have been tested on robotic systems. The super-twisting observer and sliding mode observers have been validated *in silico* on pancreas and diabetes control, and glucose regulation with insulyn infusion in [140, 141, 142]. A parabolic differentiator is implemented on a direct current servomotor equipped with an optical encoder in [106]. A set-valued anti-sway controller is applied on an overhead crane in [143]. A super-twisting observer is applied to a magneto-rheological clutch in [144, 145], while a super-twisting controller semi-implicitly discretized is applied to an electromechanical system with time-delay in [146]. Various controllers (super-twisting, first-order) are applied on a floating wind turbine in [147, 148, 149], with extensive comparisons with other types of controllers. The super-twisting controller is applied to functional electrical stimulation of index finger's metacarpophalangeal in [150]. A first-order sliding-mode set-valued input is designed for the stabilization of a nonsmooth model of an hydraulic actuator in [36] and applied to an industrial commercial excavator [151, 152, 153]. A toolbox for automatic gain-tuning of higher-order differentiators implemented with the backward-Euler method as those introduced in section "4.2" is proposed in [154, 84] (see also [155] for a toolbox dedicated to differentiators with the forward-Euler discretization).

7.2 Typical Experimental Results

Let us present a few experimental results which have been obtained in [120, 156, 21, 33, 130, 157, 131] on two setups: an electropneumatic system, and an inverted pendulum. Two controllers are tested: the classical first-order SMC and the twisting algorithm. They nicely illustrate some important properties of backward-Euler schemes. The aim is to illustrate the main specific features without presenting details which are available in the cited articles. The first property is digital chattering alleviation and magnitude decrease in the input, see Fig. 8 for first-order SMC applied to the electropneumatic setup [33] (see [158] for a video showing the setup). The explicit controller (Fig. 8 top) switches between the two extremes of the sign function at each sampling step: this is a high-frequency bang-bang signal. By 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 Fig. 8 bottom): we are therefore exactly in a situation where (98) (or (77) for the passivity-based first-order SMC) hold. Fig. 9 and 10 depict experiments on the same electropneumatic system, using the twisting algorithm for a trajectory tracking task [120, 130, 102]. It is seen in Fig. 9 that the explicit implementation performance deteriorates a lot when h increases, while the implicit one keeps a very good level of performance. In Fig. 10 the gain insensitivity property (as a consequence of (21) and (98)) is shown for three different sampling times h, and a large range of control gains (equivalent to K in (19)). Very similar results on gain insensitivity are obtained in [21] (first-order SMC and inverted pendulum) and [33] (first-order SMC and electropneumatic system). Fig. 11 and 12 show the evolution of the system's output for two different sampling times (first-order SMC, electropneumatic system [33]). Explicit and implicit controllers produce similar outputs for very small h (Fig. 11 and 12, top figures). 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 Fig. 9. This also demonstrates that both methods yield similar outputs for very small h, however the inputs are still quite different one from each other (they always look like in Fig. 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 [148, Fig. 7-10], where codimension $n \geq 2$ sliding surfaces are considered.

8 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 [31] 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 sliding-mode control 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 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 the authors' 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, are still largely open problems. The problem's complexity (which can stem from the sliding surface codi-

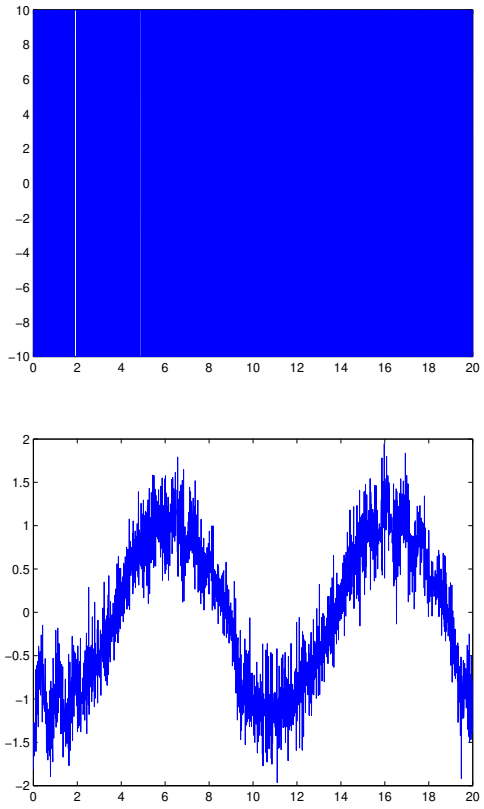

Figure 8: Typical control inputs for explicit (above) and implicit (below) first-order SMC with same gain and sampling period [156, Figures 4, 5], see also [33, 157]. See [158] for further illustration of chattering alleviation.

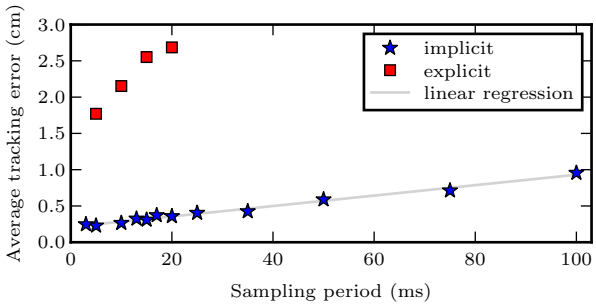

Figure 9: Twisting controller: Evolution of the average tracking error with respect to the sampling time for both implicit and explicit discretizations [120, Figure 4.3].

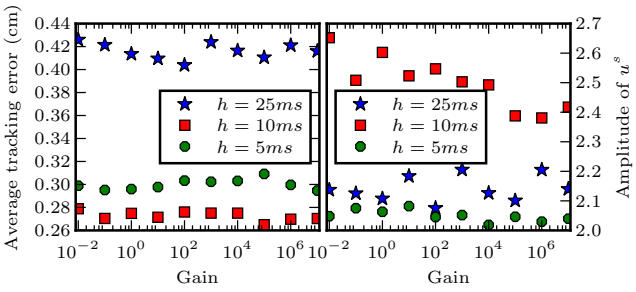

Figure 10: Twisting controller: Evolution of the average tracking error and the implicit control magnitude when the gain varies, for three sampling periods [120, Figure 4.8] (Gain= 10^5 in Fig. 9).

mension, or from the controller/differentiator's structure) also deserves future studies. Finally, it seems that there exists a gap between monotonicity (extensively used in this article) and homogeneity (extensively used recently in continuous-time sliding-mode systems design). Analyzing deeply the link between both deserves future attention as well. The choice of solvers for the one-step generalized equations also represents a source of investigations for real-time implementation.

9 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.

References

[1] J. André and P. Seibert. Über stückweise lineare Differentialgleichungen, die bei Regelungsproblemen auftreten I. *Archiv der Mathematik*, 7:148–156, 1956.

[2] J. André and P. Seibert. Über stückweise lineare Differentialgleichungen, die bei Regelungsproblemen auftreten II. *Archiv der Mathematik*, 7:157–164, 1956.

[3] Y. Z. Tsypkin. *Theory of Relay Control Systems*. Moscow: 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*, 18:1017–1028, 1957.

[5] J. J. Slotine, J. K. Hedrick, and E. A. Misawa. On sliding observers for nonlinear systems. *Journal of Dynamic Systems, Measurement, and Control*, 109(3):245–252, 1987.

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

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

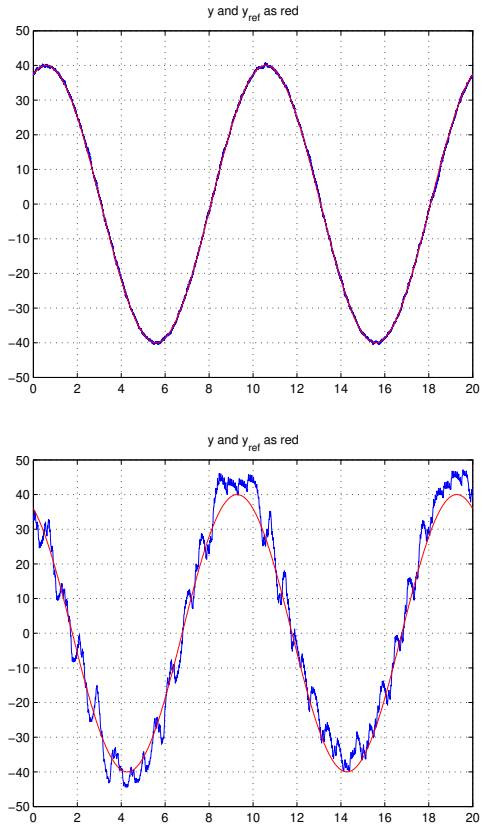

Figure 11: Explicit method: $h = 2$ms (top), $h = 15$ms (bottom), constant control gains [156, Figures 2, 3] and [33]. The signals y and y_{ref} are the position output and the desired output of the electropneumatic setup.

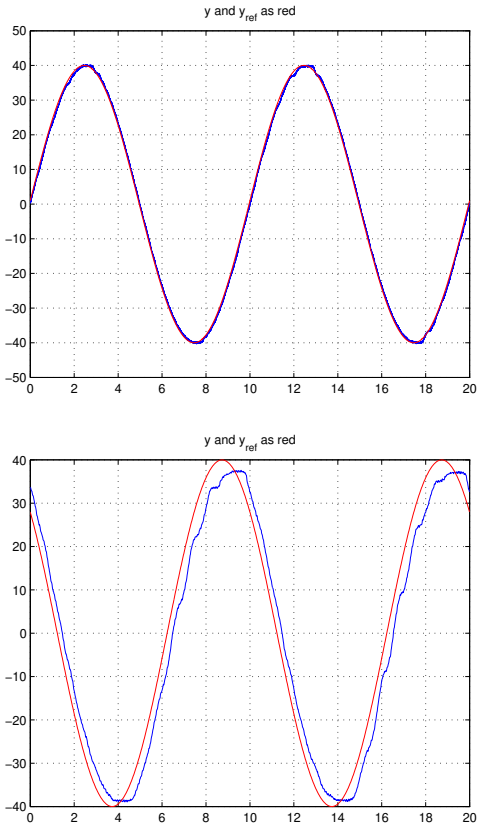

Figure 12: Implicit method: $h = 2$ms (top), $h = 15$ms (bottom), constant control gains [156, Figures 2, 3] and [33, 131]. The signals y and y_{ref} are the position output and the desired output of the electropneumatic setup.

[8] L.V. Levantovsky. Second order sliding algorithms: their realization. *Dynamics of Heterogeneous Systems*, 31(4):32–43, 1985. Institute for System Studies, Moscow (in Russian).

[9] S. V. Emelyanov, S. K. Korovin, and L.V. Levantovsky. Higher order sliding modes in the binary control systems. *Sov. Phys. Dokl.*, 31(4):291–293, 1986.

[10] A. Levant. Sliding order and sliding accuracy in sliding mode control. *International Journal of Control*, 58(6):1247–1263, 1993.

[11] H. Komurcugil, S. Biricik, S. Bayhan, and Z. Zhang. Sliding mode control: Overview of its applications in power converters. *IEEE Industrial Electronics Magazine*, 15(1):40–49, 2021.

[12] A. Pisano and E. Usai. Sliding mode control: A survey with applications in math. *Mathematics and Computers in Simulation*, 81(5):954–979, 2011.

[13] G. Bartolini, A. Pisano, E. Punta, and E. Usai. A survey of applications of second-order sliding mode control to mechanical systems. *International Journal of Control*, 76(9-10):875–892, 2003.

[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. Dynam. Control*, 9:363–378, 2021.

[15] Y. Mousavi, G. Bevan, I. Beklan Kucukdemiral, and A. Fekih. Sliding mode control of wind energy conversion systems: Trends and applications. *Renewable and Sustainable Energy Reviews*, 167:112734, 2022.

[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. *Annual Reviews in Control*, 49:173–196, 2020.

The crucial role played by passivity in the design of backward-Euler discretization of first-order sliding-mode algorithms is highlighted.

[17] A. F. Filippov. Differential equations with discontinuous right-hand side. *Matematicheskii Sbornik, Novaya Seriya*, 51(93)(1):99–128, 1960.

[18] A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Springer Science + Business Media, Dordrecht, Mathematics and Its Applications, 1988.

[19] A. Polyakov, D. Efimov, and B. Brogliato. Consistent discretization of finite-time and fixed-time stable systems. *SIAM Journal on Control and Optimization*, 57(1):78–103, 2019.

[20] A. Levant. On fixed and finite time stability in sliding mode control. In *Proceedings IEEE Conference on Decision and Control*, page 4260–4265, 2013.

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

[22] X. Yu and G. Chen. Discretization behaviors of equivalent control based sliding-mode control systems. *IEEE Transactions on Automatic Control*, 48(9):1641–1646, 2003.

[23] X. Yu, B. Wang, Z. Galias, and G. Chen. Discretisation effect on equivalent control-based multi- input sliding-mode control systems. *IEEE Transactions on Automatic Control*, 53(6):1563–1569, 2008.

[24] Y. Yan, Z. Galias, X. Yu, and C. Sun. Euler's discretization effect on a twisting algorithm based sliding mode control. *Automatica*, 6:203–208, 2016.

[25] Y. Yan, S. Yu, and X. Yu. Quantized super-twisting algorithm based sliding mode control. *Automatica*, 105:43–48, 2019.

[26] Y. Yan, X. Yu, and C. Sun. Discretization behaviors of a super-twisting algorithm based sliding mode control system. In *Proceedings of the International Workshop on Recent Advances in Sliding Modes (RASM)*, pages 1–5, Istanbul, Turkey, 2015.

[27] Z. Galias and X. Yu. Complex discretization behaviours of a simple sliding-mode control system. *IEEE Transactions on Circuits and Systems–II: Express Briefs*, 53(8):652–656, August 2006.

[28] Z. Galias and X. Yu. Euler's discretization of single input sliding-mode control systems. *IEEE Transactions on Automatic Control*, 52(9):1726–1730, September 2007.

[29] Z. Galias and X. Yu. Analysis of zero-order holder discretization of two-dimensional sliding-mode control systems. *IEEE Transactions on Circuits and Systems–II: Express Briefs*, 55(12):1269–1273, December 2008.

[30] V. Acary and B. Brogliato. Implicit Euler numerical scheme and chattering-free implementation of sliding mode systems. *Systems and Control Letters*, 59(5):284–293, 2010.

[31] V. Acary, B. Brogliato, and Y. Orlov. Chattering-free digital sliding-mode control with state observer and disturbance rejection. *IEEE Transactions on Automatic Control*, 57(5):1087–1101, May 2012.

[32] R. Kikuuwe, S. Yasukouchi, H. Fujimoto, and M. Yamamoto. Proxy-based sliding mode control: A safer extension of PID position control. *IEEE Transactions on Robotics*, 26(4):670–683, August 2010.

[33] 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, editors, *Recent Trends in Sliding Mode Control*, pages 207–235. IET Control, Robotics and Sensors Series, 2016.

[34] B. Brogliato, A. Polyakov, and D. Efimov. The implicit discretization of the supertwisting sliding-mode control algorithm. *IEEE Transactions on Automatic Control*, 65(8):3707–3713, August 2020.

[35] X. Xiong, G. Chen, Y. Lou, R. Huang, and S. Kamal. Discrete-time implementation of super-twisting control with semi-implicit Euler method. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(1):99–103, 2022.

[36] R. Kikuuwe, Y. Yamamato, and B. Brogliato. Implicit implementation of nonsmooth controllers to nonsmooth actuators. *IEEE Transactions on Automatic Control*, 67(9):4645–4657, 2022.

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

[38] 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 Journal of Emerging and Selected Topics in Industrial Electronics*, 2(3):227–236, 2021.

[39] 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*, 53(2):5135–5140, 2020.

[40] A. Winkler, G. Grabmair, and J. Reger. *On implementing the implicit discrete-time super-twisting observer on mechanical systems.* Int. Journal Robust and Nonlinear Control, 2023.

[41] 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 Transactions on Circuits and Systems–II: Express Briefs*, 68(7):2508–2512, 2021.

[42] F. Miranda-Villatoro, B. Brogliato, and F. Castanos. Multivalued robust tracking control of lagrange systems: continuous and discrete-time algorithms. *IEEE Transactions on Automatic Control*, 62(9):4436–4450, 2017.

[43] F. Miranda-Villatoro, B. Brogliato, and F. Castanos. Set-valued sliding-mode control of uncertain linear systems: continuous and discrete-time analysis. *SIAM Journal on Control and Optimization*, 56(3):1756–1793, 2018.

[44] F. Miranda-Villatoro, F. Castanos, and B. Brogliato. Continuous and discrete-time stability of a robust set-valued nested controller. *Automatica*, 107:406–417, September 2019.

[45] G. F. Franklin, J. D. Powell, and M. L. Workman. *Digital control of dynamic systems*, volume 3. Addison-wesley, Menlo Park, CA, 1998.

[46] K.J. Åström, P. Hagander, and J. Sternby. Zeros of sampled systems. *Automatica*, 20(1):31–38, 1984.

[47] S. Greenhalgh, V. Acary, and B. Brogliato. On preserving dissipativity properties of linear complementarity dynamical systems with the θ-method. *Numerische Mathematik*, 125:601–637, 2013.

[48] G. Golo and C. Milosavljevic. Robust discrete-time chattering free sliding mode control. *Systems and Control Letters*, 41:19–28, 2000.

[49] A. Levant and M. Livne. Globally convergent differentiators with variable gains. *International Journal of Control*, 91(9):1994–2008, 2018.

[50] B. Brogliato. Comments on "Finite-time stability of discrete autonomous systems [Automatica 122 (2020) 109282]". *Automatica*, 156:111206, October 2023.

[51] V. Utkin, J. Guldner, and J. Shi. *Sliding Mode Control in Electro-Mechanical Systems.* CRC Press, second edition, 2009.

[52] Y. Shtessel, C. Edwards, L. Fridman, and A. Levant. *Sliding Mode Control and Observation.* Birkhäuser, New York, USA, 2013.

[53] B. Wang, N-V. Truong, B. Brogliato, and S. Khoo. The oscillation behaviors in Euler discretized terminal sliding mode control sytems. In *2013 International Conference on Control, Automation and Information Sciences (ICCAIS)*, pages 201–205, 2013.

[54] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

[55] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, 2014.

[56] 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, Springer.* SpringerBriefs in Optimization. Springer, 2022.

[57] B. Baji and A. Cabot. An inertial proximal algorithm with dry friction: finite convergence results. *Set Valued Analysis*, 14(1):1–23, 2006.

[58] B. Brogliato. Dissipative dynamical systems with set-valued feedback loops: Well-posed set-valued Lur'e dynamical systems. *IEEE Control Systems Magazine*, 42(3):93–114, June 2022.

[59] V. Acary and B. Brogliato. *Numerical Simulation for Nonsmooth Dynamical Systems. Applications in Mechanics and Electronics*, volume 35 of *Lectures Notes in Applied and Computational Mechanics.* Springer-Verlag, Berlin-Heidelberg, 2008.

[60] F. Facchinei and J.S. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems*, volume I of *Springer Series in Operations Research.* Springer Verlag, New York, 2003.

[61] W. M Haddad and J. Lee. Finite-time stability of discrete autonomous systems. *Automatica*, 122:109282, 2020.

[62] P. Prasun, S. Kamal, S. Pandey, A. Bartoszewicz, and S. Ghosh. A minimum operator based discrete-time sliding mode control. *IEEE Transactions on Automatic Control*, pages 1–6, 2024.

[63] P. Prasun, S. Kamal, S. Ghosh, and T.N. Dinh. A minimum operator based discrete variable structure control. *European Journal of Control*, 76:100953, 2024.

[64] O. Hájek. Discontinuous differential equations I. *Journal of Differential Equations*, 32:149–170, 1979.

[65] D. S. Bernstein. *Scalar, Vector and Matrix Mathematics. Theory, Facts and Formulas.* Princeton University Press, expanded edition, 2018.

[66] D. S. Bernstein. Facing future challenges in feedback control of aerospace systems through scientific experimentation. *Journal of Guidance, Control and Dynamics*, 45(12):2202–2210, 2022.

[67] B. Brogliato, R. Lozano, B. Maschke, and O. Egeland. *Dissipative Systems Analysis and Control. Theory and Applications.* Springer Nature, 2020.

[68] C. I. Byrnes, A. Isidori, and J. C. Willems. Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems. *IEEE Transactions on Automatic Control*, 36(11):1228–1240, November 1991.

[69] B. Brogliato. Absolute stability and the Lagrange-Dirichlet theorem with monotone multivalued mappings. *Systems and Control Letters*, 51(5):343–353, 2004.

[70] G. R. Duan and H. H. Yu. *LMIs in Control Systems, Analysis, Design and Applications.* CRC Press, 2013.

[71] M. A. Estrada, J. A. Moreno, and L. Fridman. Sliding mode controllers design based on control lyapunov functions for uncertain lti systems. *IFAC-PapersOnLine*, 56(2):615–1620, 2023.

[72] W. Lin and C. I. Byrnes. Passivity and absolute stabilization of a class of discrete-time nonlinear systems. *Automatica*, 31(2):263–267, 1995.

[73] E. M. Navarro-López and E. Fossas-Colet. Feedback passivity of nonlinear discrete-time systems with direct input–output link. *Automatica*, 40(8):1423–1428, 2004.

[74] S. Diamond and S. Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[75] D. J. Hill and P. J. Moylan. Dissipative dynamical systems: basic input-output and state properties. *Journal of the Franklin Institute*, 30(5):327–357, 1980.

[76] E. Cruz-Zavala and J. A. Moreno. Homogeneous high order sliding mode design: A Lyapunov approach. *Automatica*, 80:232–238, 2017.

[77] S. Gutman. Uncertain dynamical systems–a Lyapunov min-max approach. *IEEE Transactions on Automatic Control*, 24(3):437–443, June 1979.

[78] G. Leitmann. Guaranteed asymptotic stability for some linear systems with bounded uncertainties. *Journal of Dynamic Systems, Measurement, and Control*, 101(3):212–216, 1979.

[79] F. A. Miranda and F. Castaños. Robust output regulation of linear passive systems using maximally monotone controls. In *54th IEEE Conference on Decision and Control (CDC)*, pages 6897–6902, Osaka, Japan, 2015.

[80] A. Levant. Homogeneity approach to high-order sliding mode design. *Automatica*, 41:823 – 830, 2005.

[81] A. Levant and M. Livne. Robust exact filtering differentiators. *European Journal of Control*, 55:33–44, 2020.

[82] J.A. Moreno and M. Osorio. Strict Lyapunov functions for the super-twisting algorithm. *IEEE Transactions on Automatic Control*, 57(4):1035–1040, 2012.

[83] 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. *International Journal of Robust Nonlinear Control*, 31(9):3606–3630, 2021.

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

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

[86] B. Andritsch, L. Watermann, S. Koch, M. Reichhartinger, J. Reger, and M. Horn. Modified implicit discretization of the super-twisting controller. *IEEE Transactions on Automatic Control*, 2024. doi: 10.1109/TAC.2024.3370494.

[87] I. Nagesh and C. Edwards. A multivariable super-twisting sliding mode approach. *Automatica*, 50(3):984–988, 2014.

[88] G. Besançon. *Nonlinear Observers and Applications*, volume 363 of *LNCIS*. Springer-Verlag, Berlin-Heidelberg, 2007.

[89] J. A. Moreno. On discontinuous observers for second order systems: Properties, analysis and design. In B. Bandyopadhyay et. al., editor, *Advances in Sliding Mode Control*, pages 243–265. Springer-Verlag Berlin Heidelberg, 2013.

[90] M. R. Mojallizadeh, B. Brogliato, A. Polyakov, S. Selvarajan, L. Michel, F. Plestan, M. Ghanes, J. P. Barbot, and Y. Aoustin. Discrete-time differentiators in closed-loop control systems: experiments on electro-pneumatic system and rotary inverted pendulum. Technical report, INRIA, February 2023. `{https://inria.hal.science/hal-03125960v2/document}`.

[91] 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. *Journal of the Franklin Institute*, 359:3951 – 3978, 2022.

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

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

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

[95] 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*, 18:3001–3014, 2020.

[96] A. Levant and A. Michael. Adjustment of high-order sliding-mode controllers. *International Journal of Robust and Nonlinear Control*, 19(15):1657–1672, 2009.

[97] S. Ding, A. Levant, and S. Li. Simple homogeneous sliding-mode controller. *Automatica*, 67:22–32, 2016.

[98] J. Davila, L. Fridman, and A. Levant. Second-order sliding mode observer for mechanical systems. *IEEE Trans. Autom. Control*, 50(11):1785–1789, November 2005.

[99] A. Polyakov. *Generalized Homogeneity in Systems and Control*. Communications and Control Engineering. Springer Nature, Switzerland AG, 2020.

[100] F. Castaños, F. Miranda Villatoro, and B. Brogliato. Multivalued Hamiltonian systems with multivalued dissipation: Analysis of the backward-Euler discretisation. Research report, INRIA, Grenoble, France, July 2024. `https://inria.hal.science/`.

[101] C. Wang, H. Xia, and S. Ren. An implicit discretization-based adaptive reaching law for discrete-time sliding mode control systems. *Journal of Vibration and Control*, 29(5-6):1117–1127, 2023.

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

[103] R. Kikuuwe. Sliding motion accuracy of proxy-based sliding mode control subjected to measurement noise and disturbance. *European Journal of Control*, 58:114–122, 2021.

[104] 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*, 7:30236–13024, 2019.

[105] S. Jin, Z. Lv, X. Xiong, and J. Yu. A chattering-free sliding mode filter enhanced by first order derivative feedforward. *IEEE Access*, 8:41175–41185, 2020.

[106] 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. *ASME. J. Dyn. Sys., Meas., Control*, 140(4):041005, 2018.

[107] S. Jin, R. Kikuuwe, and M. Yamamoto. Improving velocity feedback for position control by using a discrete-time sliding mode filtering with adaptive windowing. *Advanced Robotics*, 28(14):943–953, 2014.

[108] S. Jin, X. Xiong, D. Zhao, and C. Jin. Unified framework for implicit-Euler implementation of second-order sliding mode controllers. *Communications in Nonlinear Science and Numerical Simulation*, 111:106430, 2022.

[109] T. Liard, I. Balogoun, S. Marx, and F. Plestan. Boundary sliding mode control of a system of linear hyperbolic equations: A lyapunov approach. *Automatica*, 135:109964, 2022.

[110] B. Brogliato and A. Polyakov. Globally stable implicit Euler time-discretization of a nonlinear single-input sliding-mode control system. In 54th IEEE Conference on Decision and Control (CDC), pages 5426–5431, 2015.

[111] 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 Transactions on Circuits and Systems II: Express Briefs*, 68(12):3562–3566, December 2021.

[112] C. Hettiger, L. Watermann, K. Kumari, L. Eisenzopf, F. Weissenberger, M. Horn, S. Koch, J. Reger, and M. Reichhartinger. On discretization methods for indirect adaptive sliding mode control. In *IEEE 61st Conference on Decision and Control (CDC)*, pages 4930–4936, Mexico, 2022. Cancun.

[113] X. Xiong, S. Kamal, and S. Jin. Adaptive gains to super-twisting technique for sliding mode design. *Asian J Control*, 23:362–373, 2021.

[114] 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 & Applications*, 12:2347–2356, 2018.

[115] C. Wang, H. Xia, Y. Wang, and S. Ren. Discrete-time sliding mode control with adaptive reaching law via implicit Euler method. *International Journal of Control Automation and Systems*, 21:109–116, 2023.

[116] R. K. Sharma, X. Xiong, S. Kamal, and S. Ghosh. Discrete-time super-twisting fractional-order differentiator with implicit Euler method. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 68(4):1238–1242, April 2021.

[117] 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 Transactions on Circuits and Systems II: Express Briefs*, 69(6):2787–2791, June 2022.

[118] 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 Transactions on Circuits and Systems II: Express Briefs*, 69(11):4443–4447, November 2022.

[119] P. Prasun, S. Pandey, S. Kamal, S. Ghosh, and X. Xiong. A minimum operator based discrete-time super-twisting-like algorithm. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 71(1):286–290, 2024.

[120] O. Huber. *Analysis and implementation of discrete-time sliding mode control*. PhD thesis, Université Grenoble Alpes, May 2015. `https://inria.hal.science/tel-01194430v1/document`.

[121] A. Gonzalez-Garcia, H. Castañeda, and J. De León-Morales. Unmanned surface vehicle robust tracking control using an adaptive super-twisting controller. *Control Engineering Practice*, 149:105985, 2024.

[122] 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 Transactions on Circuits and Systems II: Express Briefs*, 69(8):3435–3439, August 2022.

[123] R. Seeber and S. Koch. Structural conditions for chattering avoidance in implicitly discretized sliding mode differentiators. *IEEE Control Systems Letters*, 7:2065–2070, 2023.

[124] M. R. Mojallizadeh, B. Brogliato, A. Polyakov, S. Selvarajan, L. Michel, F. Plestan, M. Ghanes, J. B. Barbot, and Y. Aoustin. A survey on the discrete-time differentiators in closed-loop control systems: Experiments on an electro-pneumatic system. *Control Engineering Practice*, 136:105546, 2023.

[125] O. Güler. *Foundations of Optimization,*. Graduate Texts in Mathematics 258. Springer Science+Business Media, 2010.

[126] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, editors, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 185–212. Springer, 2011.

[127] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi. Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists. *SIAM Review*, 65(2):375–435, 2023.

[128] P. Goulart and Y. Chen. Clarabel solver. Last accessed 23/11/2023.

[129] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal of Numerical Analysis*, 16(6):964–979, 1979.

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

[131] 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 input and output chattering suppression. *IEEE Transactions on Control Systems Technology*, 23(5):2071–2075, 2015.

[132] L. Michel, M. Ghanes, Y. Aoustin, and J. P. Barbot. A third order semi-implicit homogeneous differentiator: Experimental results. In *16th International Workshop on Variable Structure Systems (VSS)*, pages 77–82, Rio de Janeiro, Brazil, 2022.

[133] 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*, 15(14):5288, 2022.

[134] G. Perozzi, A. Polyakov, and B. Brogliato F. Miranda-Villatoro. Upgrading a linear controller to a sliding mode one: Theory and experiments. *Control Engineering Practice*, 123:105107, June 2022.

[135] M. R. Mojallizadeh and B. Brogliato. Effect of Euler explicit and implicit time discretizations on variable-structure differentiators. In T. R. Oliveira, L. Fridman, and L. Hsu, editors, *Sliding-Mode Control and Variable-Structure Systems: Studies in Systems Decision and Control*, pages 165–180. Springer, 2023.

[136] R. Mae and R. Kikuuwe. An admittance controller with a jerk limiter for position-controlled robots. *Journal of Robotics and Mechatronics*, 36(2):483–493, 2024.

[137] R. Kikuuwe. Torque-bounded admittance control realized by a set-valued algebraic feedback. *IEEE Transactions on Robotics*, 35(5):1136–1149, October 2019.

[138] X. Yuan, Y. Ding, X. Xiong, and Y. Lou. Torque-bounded admittance control with implicit Euler realization of set-valued operators. *IEEE/ASME Transactions on Mechatronics*, pages 1–10, 2023.

[139] M.T.S. Aung and R. Kikuuwe. Stability enhancement of admittance control with acceleration feedback and friction compensation. *Mechatronics*, 45:110–118, 2017.

[140] I. Sala-Mira, J. L. Díez, B. Ricarte, and J. Bondia. Sliding-mode disturbance observers for an artificial pancreas without meal announcement. *Journal of Process Control*, 78:68–77, 2019.

[141] S. Faccioli, I. Sala-Mira, J. L. Díez, A. Facchinetti, G. Sparacino, S. Del Favero, and J. Bondia. Super–twisting-based meal detector for type 1 diabetes management: Improvement and assessment in a real-life scenario. *Computer Methods and Programs in Biomedicine*, 219(10673):6, 2022.

[142] 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*, 9:143549–143563, 2021.

[143] R. Nishimoto and R. Kikuuwe. Position-commanding anti-sway controller for 2-d overhead cranes under velocity and acceleration constraints. *IEEE Access*, 11:35069–35079, 2023.

[144] G. Chen, X. Xiong, and Y. Lou. Multi-state modelling and observation of magneto-rheological clutch with rate-dependent hysteresis characteristic. *IEEE Robotics and Automation Letters*, 6(2):2445–2452, April 2021.

[145] G. Chen, X. Xiong, Y. Lou, and Z. Li. Modeling and observation of rate-dependent hysteresis and creep phenomena in magnetorheological clutch. *IEEE/ASME Transactions on Mechatronics*, 27(4):2053–2061, August 2022.

[146] 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 *IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pages 1–6, 2023.

[147] M.R. Mojallizadeh, F. Bonnefoy, V. Leroy, F. Plestan, S. Delacroix, J. Ohana, and B. Bouscasse. Control design for thrust generators with application to wind turbine wave-tank testing: A sliding-mode control approach with Euler backward time-discretization. *Control Engineering Practice*, 146:105894, 2024.

[148] M.R. Mojallizadeh, F. Bonnefoy, F. Plestan, M.A. Hamida, and J. Ohana. Euler implicit time-discretization of multivariable sliding-mode controllers. *ISA Transactions*, 147:140–152, 2024.

[149] 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 Engineering*, 309:118390, 2024.

[150] 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. *Applied Sciences*, 14(11), 2024.

[151] R. Kikuuwe, T. Okada, H. Yoshihara, T. Doi, T. Nanjo, and K. Yamashita. A nonsmooth quasi-static modeling approach for hydraulic actuators. *Transactions of ASME: Journal of Dynamic Systems, Measurement, and Control*, 143:12, December 2021.

[152] Y. Yamamoto, J. Qiu, Y. Munemasa, T. Doi, T. Nanjo, K. Yamashita, and R. Kikuuwe. A sliding-mode set-point position controller for hydraulic excavators. *IEEE Access*, 9:153735–153749, 2021.

[153] 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 Transactions on Automation Science and Engineering*, pages 1–17, 2024.

[154] M.R. Mojallizadeh. *Differentiation Toolbox*. Univ. Grenoble Alpes, INRIA, Grenoble, France, 2020. `https://github.com/Mojallizadeh/DifferentiationToolbox`.

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

[156] 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. Research Report hal-01087400, INRIA, Grenoble, France, November 2014. `https://inria.hal.science/hal-01087400v1/document`.

[157] 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 *2014 13th International Workshop on Variable Structure Systems (VSS)*, pages 1–6, 2014.

[158] O. Huber. `http://xhub.github.io/pages/videos.html`. 2016.