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

Félix Miranda-Villatoro, Fernando Castaños, Bernard Brogliato

When proximal-point algorithms meet set-valued systems

An Optimization point of view of discrete-time sliding modes

Félix Miranda-Villatoro[†] and Fernando Castaños[□] and Bernard Brogliato[†]

[†] Univ. Grenoble Alpes, INRIA, CNRS, LJK, Grenoble INP,
38000 Grenoble, France.

[□] Automatic Control Department, Cinvestav-IPN, 2508 Av. IPN, 07360,
Mexico City, Mexico

December 22, 2023

Sliding-mode control is a major field of Automatic Control, originating about 70 years ago in Germany [39], [40] and in the former Soviet Union [37], [38] during the development of relay systems. Later, sliding-mode differentiators were proposed [41], [44], [49]. Mathematical foundations were set by Filippov [47], [48] 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 concept in the Automatic Control field. Chattering phenomena have long been known as a major obstacle to the implementation of set-valued sliding-mode systems. It has recently become clear that the time-discretization of set-valued sliding-mode controllers is a major source of chattering when *explicit* methods are used [16]–[23], yielding high frequency bang-bang-like input signals which harm actuators, and possibly closed-loop instability [11], [12], [72]. This motivated the introduction of a new discretization approach, known as the *implicit* method [10], [13], [60].

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 Optimization, and that this close relationship allows 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 [10], [11], [13], [26], [30], [35], [58]–[62], [66], [67] and especially in [55]–[57], where *resolvents* and *Yosida approximations* were used for the first time in this context. When implicitly/semi-implicitly discretized, sliding-mode controllers and differentiators yield new kinds of proximal-

point algorithms, named *robust* proximal-point, and *higher-order* proximal-point, which belong to a class of nonlinear difference equations.

Control systems are mainly designed in a continuous-time setting, and their discretization comes in a second step. When discretization is carried out following an emulation procedure, 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 (see, *e.g.*, [65, Chapter 3]). It is well-known, however, that the discretization process can deteriorate the performance and modify the system's properties (zeroes [50], passivity [51], *etc*). Systems with sliding modes are particularly susceptible to the discretization method employed, some aspects of which are summarized in the flowchart of Figure S1. 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.

SUBGRADIENT FLOWS, PROXIMAL ALGORITHMS, AND ROBUST FINITE-TIME CONVERGENCE

In this section the intimate relation between optimization and the implicit discretization of set-valued sliding-mode systems is illustrated. To simplify the presentation we restrict ourselves 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 $b > 0$ constant and fixed. Suppose that the input is a feedback law $u(t) = u_{\rm sv}(x(t))$. The set-valued feedback

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

where $\mathbf{sgn}(\cdot)$ is the set-valued signum function (S18), achieves the finite-time convergence of the state towards the origin [8], [9]. Indeed, the closed-loop system belongs to the class of gradient systems

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

with $f(x) = -ax^2 + b\kappa|x|$. According to Proposition 4 in "Set-Valued Mappings and Differential Inclusions", the origin is finite-time stable. Due to the computational power and flexibility of digital electronic devices, it is now common to consider the discrete-time version of the control law (2) for its implementation in a digital computer. To that end, let us consider the exact discrete-time dynamics associated with (1), that is

$$x_{k+1} = \tilde{a}x_k + \tilde{b}u_k, \tag{4}$$

where $\tilde{a} = e^{ah}$, $\tilde{b} = \frac{(e^{ah}-1)b}{a}$, and $h > 0$ denotes the sampling time. Let us consider first the explicit (forward Euler) discretization of (2), that is, $u_k = u_{\rm sv}(x_k)$, where

$$-u_{\rm sv}(x_k) \in \kappa\, \mathbf{sgn}(x_k). \tag{5}$$

The closed-loop discrete-time dynamics becomes

$$x_{k+1} \in \tilde{a}x_k - \tilde{b}\kappa\, \mathbf{sgn}(x_k). \tag{6}$$

It is worth emphasizing that, at $x_k = 0$, the difference Equation (6) 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 a limit-cycle appears, creating *numerical* or *discretization chattering*. The amplitude of the limit cycle is proportional to the control gain κ and to h through $\tilde{b}$: 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 S2 and S3). The existence of limit cycles in explicitly discretized sliding-mode systems has been deeply analyzed in [16]–[23]. It is inferred that digital chattering is intrinsically linked to the explicit discretization (not to be confused with the explicit forms of the implicit discretization, as detailed below). Let us now consider the implicit discretization of (2),

$$-u_{\rm sv}(x_k) \in \kappa\, \mathbf{sgn}(x_{k+1}), \tag{7}$$

which results in

$$x_{k+1} = \tilde{a}x_k - \tilde{b}\kappa\, \mathbf{sgn}(x_{k+1}). \tag{8}$$

In this case the closed-loop is well-posed, as the solution is unique for any value of x_k. Indeed, (8) is equivalent to the following *generalized equation* in the unknown x_{k+1}:

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

In view of the maximal monotonicity of the sign multifunction (see "Maximal Monotone Operators"), the generalized equation (9) has a unique solution

$$x_{k+1} = \mathrm{Prox}_{\tilde{b}\kappa g}(\tilde{a}x_k), \tag{10}$$

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

$$\mathrm{Prox}_{\tilde{b}\kappa g}(\tilde{a}x_k) = \tilde{a}x_k - \tilde{b}\kappa\,\mathrm{sgn}(x_k)\min\left\{\frac{\tilde{a}}{\tilde{b}\kappa}|x_k|, 1\right\} = \begin{cases} 0 & \text{if } |x_k| \le \frac{\tilde{b}\kappa}{\tilde{a}} \\ \tilde{a}x_k - \tilde{b}\kappa\mathrm{sgn}(x_k) & \text{otherwise} \end{cases}. \tag{11}$$

The iteration (10) is an instance of the proximal-point algorithm (see Definition 10 in "Proximal-Point Algorithm and Proximal Mapping"), vastly studied in the literature of Convex Optimization [S23] and recognized as an effective discretization of the subgradient system (3) [S17]. It

is an explicit form of the algorithm implicitly defined by (8) and falls within the context of Krasnosel'skiĭ-Mann iterations [2]. Thus, its convergence is guaranteed by Krasnosel'skiĭ-Mann Theorem [2] for any proper, convex, lower semicontinuous (lsc) function $g(\cdot)$, whenever the set of minimizers of $g(\cdot)$ is nonempty [2, Theorem 1.1]. Moreover, since for $g(\cdot) = |\cdot|$ we have $0 \in \operatorname{int} \partial g(0)$, the origin of (8) is globally finite-time stable for any $h > 0$ [24] and, contrary to the explicit discretization (6), it exhibits no oscillations at all. Figure S3 shows the cobweb diagram of (8).

An explicit expression for the selection of the control law in (7) can be computed with the help of (4) as shown in the following proposition.

Proposition 1. *Consider the input implicitly defined by* (7) *together with the discrete-time system* (4)*. The following statements are equivalent:*

$$-u_{\mathrm{sv}}(x_k) \in \kappa\, \mathbf{sgn}(x_{k+1}). \tag{12a}$$

$$-u_{\mathrm{sv}}(x_k) = \mathrm{Proj}\left([-\kappa, \kappa]; \frac{\tilde{a}}{\tilde{b}} x_k\right). \tag{12b}$$

$$-u_{\mathrm{sv}}(x_k) = \kappa\, \mathbf{sgn}(x_k) \min\left\{\frac{\tilde{a}}{\tilde{b}\kappa}|x_k|, 1\right\}. \tag{12c}$$

Proof. The proof uses various results presented in "Convex Analysis Tools", "Set-Valued Mappings and Differential Inclusions", "Maximal Monotone Operators", and "Proximal-Point Algorithm and Proximal Mapping". It follows from (S25) and (4) that (12a) 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 5 in "Convex Analysis Tools" then establishes the equivalence between (12a) and (12b). The equivalence between (12b) and (12c) follows from (7) and (4), and it is proven in [25]. It may also be checked directly by enumeration. □

The expressions in (11) and (12) can also be written using complementarity problems and variational inequalities, using equivalent formulations of normal cones (see (S19) for the variational formulation, and "Projected Dynamical Systems" 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 [29, "Set-Valued Signum Function"]. The $\min\{\cdot,\cdot\}$ function in (12c) and in (11) is a *complementarity function* (or C-function) [S12], [S31]. The expression (12b) shows that $u_{\mathrm{sv}}(x_k)$ is the solution of the quadratic optimization problem

$$-u_{\mathrm{sv}}(x_k) = \underset{z\in[-\kappa,\kappa]}{\arg\min}\ \frac{1}{2}\left\|z - \frac{\tilde{a}}{\tilde{b}}x_k\right\|^2. \tag{13}$$

This suggests that implicit controllers can be implemented online by solving QPs with constraints. It is noteworthy that the explicit form (12c) has been studied in [1, Example 3.1] via Lyapunov methods, and that the connection with proximal maps via the implicit Euler method was shown in [25]. To close this section, it is also worth to emphasize that considering arbitrary regularizations of the set-valued sign function (*i.e.*, single-valued functions sufficiently close to the graph of the sign multifunction, like the saturation or the sigmoid functions) does 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 (see "Proximal-Point Algorithm and Proximal Mapping"), making the origin unstable and producing an oscillatory behavior.

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.

FIRST-ORDER SLIDING-MODE CONTROL: CONTINUOUS-TIME

Consider the perturbed linear dynamics

$$\dot{x}(t) = Ax(t) + B(u(t) + \delta(t)), \tag{14}$$

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 $B \in \mathbb{R}^{n\times m}$. We assume that $\mathrm{rank}(B) = m$ and that $\delta(t)$ is uniformly bounded.

The Classical Approach

In the classical approach to sliding-mode control, the robust stabilization of (14) 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 [8], [9]. To accomplish Step 2, first the time derivative $\dot{s}(t) = CAx(t) + CB(u(t) + \delta(t))$ is computed. By setting $u(t) = u_{\mathrm{lin}}(x(t)) + (CB)^{-1}v(t)$, where

$$u_{\mathrm{lin}}(x) = -(CB)^{-1}CAx \tag{15}$$

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

$$\dot{s}(t) = v(t) + CB\delta(t). \tag{16}$$

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

$$u_{\mathrm{sv}}(x, t) \in -K\,\mathbf{Sgn}(\sigma(x)), \quad K = K^\top > 0, \tag{17}$$

for all $t \in \mathbb{R}$. The term $u_{\mathrm{sv}}(x, t)$ is a selection of the set-valued signum function (see Definition 2 in "Set-Valued Mappings and Differential Inclusions"). With the controls (15) and (17), the sliding variable evolves according to the differential inclusion

$$\dot{s}(t) \in -K\,\mathbf{Sgn}(s(t)) + CB\delta(t). \tag{18}$$

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

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

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, *e.g.*, [3, Proposition 11.1.6]). Therefore, there exists some $t_{\min} < +\infty$ after which a sliding mode occurs: $s(t) = 0$ and therefore

$$u_{\mathrm{sv}}(x(t), t) = -CB\delta(t) \tag{19}$$

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* [28]. The exactness (equality) in (19) is obviously due to modelling idealization. However, as demonstrated by the implicit discrete-time analysis in (94) and by experimental results, it is the idealization of an *observed behaviour*. 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 (19) is far from being just a mathematical fuss.

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 (14) *passive with output* s *and input* $v + CB\delta$ *(see [4] 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 "Maximal Monotone Operators"). Following [S7] the change of variable $s' = K^{-\frac{1}{2}} s$ may be performed, transforming (18) 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

$$\dot{s}'(t) \in -\partial f(s'(t)) + K^{-\frac{1}{2}} CB\delta(t), \tag{20}$$

where $f(\cdot) = \| \cdot \|_1 \circ K^{\frac{1}{2}}(\cdot)$ and where the chain rule was used (see Theorem 4 in "Maximal Monotone Operators") to compute ∂f, the subdifferential of $f(\cdot)$ (see "Convex Analysis Tools"). The operator $\partial f(\cdot)$ is a maximal monotone operator, $f(\cdot)$ being a convex continuous proper function (see "Convex Analysis Tools"). The differential inclusion has the form (S4) so, if the eigenvalues of K are large enough, it satisfies the conditions of Proposition 4 in "Set-Valued Mappings and Differential Inclusions" 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 $P = P^\top > 0$ and [6]

$$AP + PA^\top < BB^\top. \tag{21}$$

Similarly to [56], [118], this inequality is exploited to construct the candidate storage function

$$V(x) = x^\top P^{-1} x. \tag{22}$$

Its time-derivative is

$$\dot{V}(x) = x^\top \left(P^{-1} A + A^\top P^{-1} \right) x + 2x^\top P^{-1} B(u + \delta) \tag{23}$$

(when clear from context, explicit dependence on t is omitted). According to the Kalman-Yakubovic-Popov Lemma [S16], the only candidate passive output of relative degree equal to one and associated with the storage function (22) is $s(t) = \sigma(x(t))$ with

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

It gives the energy balance

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

Since (21) can be equivalently written as $Q = P^{-1}BB^\top P^{-1} - P^{-1}A - A^\top P^{-1} > 0$, and since $x^\top P^{-1}BB^\top P^{-1}x = \frac{1}{4}s^\top s$, the energy balance is

$$\dot{V}(x) = s^\top \left(\frac{1}{4}s + u + \delta\right) - x^\top Q x. \tag{26}$$

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

$$u(t) = u_{\text{lin}}(x(t)) + v(t) \tag{27}$$

with

$$u_{\text{lin}}(x) = -\frac{1}{4}\sigma(x) \tag{28}$$

can be used, as it gives

$$\dot{V}(x) = s^\top (v + \delta) - x^\top Q x. \tag{29}$$

It can readily be seen that the system is passive with storage function (22), output (24), and new input $v + \delta$. Now let us set $v(t) = u_{\text{sv}}(x(t), t)$ with

$$u_{\text{sv}}(x, t) \in -K\,\mathbf{Sgn}(Ks), \quad K = K^\top > 0 \tag{30}$$

for all $t \in \mathbb{R}$. Substituting this control law in (29) gives

$$\dot{V}(x) = -s^\top K\left(\mathbf{Sgn}(Ks) - K^{-1}\delta\right) - x^\top Q x \leq -\|Ks\|_1(1 - \|K^{-1}\delta\|_\infty) - x^\top Q x, \tag{31}$$

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

$$\dot{s} \in 2B^\top P^{-1}\left(\left(A - \frac{1}{2}BB^\top P^{-1}\right)x - BK\,\mathbf{Sgn}(Ks) + B\delta\right). \tag{32}$$

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

$$\dot{V}_s(s) = -2s^\top \left(K\,\mathbf{Sgn}(Ks) - \delta - Lx\right), \tag{33}$$

where $L = (B^\top P^{-1}B)^{-1}B^\top P^{-1}\left(A - \frac{1}{2}BB^\top P^{-1}\right)$. In (32), 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 (32) can be analyzed within the framework of differential inclusions with maximal monotone right-hand sides, similarly to (20). To reveal the maximal monotone structure in (32), let us perform the change of variables $s' = \frac{1}{2}(B^\top P^{-1}B)^{-\frac{1}{2}}s$. We obtain

$$\dot{s}' \in -\partial g(s') + (B^\top P^{-1}B)^{\frac{1}{2}}(\delta + Lx) \tag{34}$$

with $g(\cdot) = \|\cdot\|_1 \circ K(B^\top P^{-1}B)^{\frac{1}{2}}(\cdot)$ a proper, convex, continuous function (here, we have used the fact that $\mathbf{Sgn}(2x) = \mathbf{Sgn}(x)$). Again, the differential inclusion (34) has the structure (S4) and, if the eigenvalues of K are large enough, it satisfies the conditions of Proposition 4 in "Set-Valued Mappings and Differential Inclusions", 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 2 in "Set-Valued Mappings and Differential Inclusions") which compensates exactly for the disturbance $(B^\top P^{-1}B)^{\frac{1}{2}}(\delta(t) + Lx(t))$ in (34). Equivalently, consider (30) 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 (32), *i.e.*, similarly to (19):

$$u_{\rm sv}(x(t),t) = -(B^\top P^{-1}B)^{-1}B^\top P^{-1}\left(A - \frac{1}{2}BB^\top P^{-1}\right)x(t) - \delta(t) \tag{35}$$

Let us stress once again that neither (19) nor (35) mean that the disturbance $\delta(t)$ is known. Finally, mark that the well-posedness of (34) holds under mild measurability conditions on the perturbation, and from the analysis of the closed-loop dynamics

$$\dot{x} \in Ax + B\left(-\frac{1}{4}\sigma(x) - K\mathbf{Sgn}(\sigma(x))\right) + B\delta, \tag{36}$$

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 2 (see "Set-Valued Mappings and Differential Inclusions") that (36) has AC solutions.

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 by 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 (32) that $u_{\rm sv}$ in (30) is selected as

$$u_{\rm sv}(x,t) = -Lx - \delta(t). \tag{37}$$

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 (15) and (17) (see (19)). Incidentally, the differential inclusions in (20) and (34) possess incremental passivity properties. Indeed, they have the generic form $\dot{s}'(t) \in -\mathbf{M}(s'(t)) + Nu(t)$ with $\mathbf{M}(\cdot)$ a maximal monotone mapping. Defining the output as $y = N^\top s'$ yields incremental passivity with storage function $V(s'_1, s'_2) = \frac{1}{2}(s'_1 - s'_2)^\top(s'_1 - s'_2)$ [S16].

It is worth stressing that (30) 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, we have

$$\partial \|x\|_p = \frac{1}{\|x\|_p^{p-1}} \begin{pmatrix} |x_1|^{p-1}\,\mathbf{sgn}(x_1) \\ \vdots \\ |x_n|^{p-1}\,\mathbf{sgn}(x_n) \end{pmatrix} \tag{38}$$

if $x \neq 0$ and $\partial \|x\|_p = \mathcal{B}_{p^*}$ if $x = 0$. For example, the norm $p = 2$ yields the *unit control*

$$u_{\rm sv}(x,t) \in \begin{cases} -K\frac{Ks}{\|Ks\|_2} & \text{if } s \neq 0 \\ -K\mathcal{B}_2 & \text{otherwise} \end{cases}. \tag{39}$$

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 S1 and Theorem 2 in "Set-Valued Mappings and Differential Inclusions").

FIRST-ORDER SLIDING-MODE CONTROL: DISCRETE-TIME

Consider now a sampled-data model of (14),

$$x_{k+1} = \tilde{A}x_k + \tilde{B}(u_k + \tilde{\delta}_k), \tag{40}$$

where $\tilde{A} \in \mathbb{R}^{n\times n}$, $\tilde{B} \in \mathbb{R}^{n\times m}$, and we assume that $\operatorname{rank}\tilde{B} = m$. In the spirit of Section "A Passivity-Based Approach", an output s 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 $\{0, 0, \ldots, 0\}$ [7], [15], which ultimately results in the necessity for implicitly defined control laws.

A Discrete-Time Passivity-Based Approach

The discrete-time counterpart of (21) is

$$\tilde{A}\tilde{P}\tilde{A}^\top - \tilde{P} < \tilde{B}\tilde{B}^\top\,. \tag{41}$$

That is, the pair $(\tilde{A}, \tilde{B})$ is stabilizable if, and only if, inequality (41) holds for some $\tilde{P} = \tilde{P}^\top > 0$ [6]. The forward difference $V(x_{k+1}) - V(x_k)$ of the storage function $V(x_k) = x_k^\top \tilde{P} x_k$ is

$$\Delta V(x_k) = \left(\tilde{A}x_k + \tilde{B}(u_k + \tilde{\delta}_k)\right)^\top \tilde{P}^{-1} \left(\tilde{A}x_k + \tilde{B}(u_k + \tilde{\delta}_k)\right) - x_k^\top \tilde{P}^{-1} x_k. \tag{42}$$

By isolating the terms that do not depend on the inputs, the counterpart of (23) is obtained as

$$\Delta V(x_k) = x_k^\top \left(\tilde{A}^\top \tilde{P}^{-1} \tilde{A} - \tilde{P}^{-1}\right) x_k + 2(u_k + \tilde{\delta}_k)^\top \tilde{B}^\top \tilde{P}^{-1} \left(\tilde{A} x_k + \tilde{B}(u_k + \tilde{\delta}_k)\right) - (u_k + \tilde{\delta}_k)^\top \tilde{B}^\top \tilde{P}^{-1} \tilde{B}(u_k + \tilde{\delta}_k). \quad (43)$$

Passivity Analysis

The candidate passive output

$$s_{k+1} := \sigma(x_k, u_k + \tilde{\delta}_k) \quad (44)$$

with

$$\sigma(x, w) = 2\tilde{B}^\top \tilde{P}^{-1} \left(\tilde{A} x + \tilde{B} w\right) \quad (45)$$

is unveiled and the energy balance becomes

$$\Delta V(x_k) = x_k^\top \left(\tilde{A}^\top \tilde{P}^{-1} \tilde{A} - \tilde{P}^{-1}\right) x_k + s_{k+1}^\top (u_k + \tilde{\delta}_k) - (u_k + \tilde{\delta}_k)^\top \tilde{B}^\top \tilde{P}^{-1} \tilde{B}(u_k + \tilde{\delta}_k). \quad (46)$$

Remark 2. *It is clear that* (44)*,* (45) *have an implicit (backward Euler) flavour since*

$$s_{k+1} = 2\tilde{B}^\top \tilde{P}^{-1} x_{k+1}, \quad (47)$$

and that (45) is the counterpart of (24). 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.

Remark 3. *Neither* $s_{k+1} = 2B^\top P^{-1} x_k$ *nor even* $s_{k+1} = 2B^\top P^{-1} x_{k+1}$ *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 (28), which can be discretized in various ways (by emulation,* $u_{\text{lin}}(x_k) = -\frac{1}{4} B^\top P^{-1} x_k$*, or directly in discrete time, as in (52) and (53)). Not all discretized controllers share similar closed-loop properties [11]. The same comments hold for the case of the classical first-order control in (87) and (88) below. See also Section "Limitations and Modifications of Implicit Algorithms".*

The following results are inspired by [6, Chapter 6].

Lemma 1. *The inequality* (41) *implies that*

$$\tilde{A}^\top \left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top\right)^{-1} \tilde{P} \left(\tilde{A} + \frac{1}{2}\tilde{B}\tilde{B}^\top\right)^{-1} \tilde{A} - \tilde{P}^{-1} < 0. \quad (48)$$

Proof. Using Schur complements, we can see that

$$(41) \iff \begin{pmatrix} \tilde{P} & \tilde{P}\tilde{A}^\top \\ \tilde{A}\tilde{P} & \tilde{P} + \tilde{B}\tilde{B}^\top \end{pmatrix} > 0 \Longrightarrow \begin{pmatrix} \tilde{P} & \tilde{P}\tilde{A}^\top \\ \tilde{A}\tilde{P} & \tilde{P} + \tilde{B}\tilde{B}^\top + \frac{1}{4}\tilde{B}\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{B}^\top \end{pmatrix} > 0, \tag{49}$$

that is, $\tilde{A}^\top \left(\tilde{P} + \tilde{B}\tilde{B}^\top + \frac{1}{4}\tilde{B}\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{B}^\top\right)^{-1}\tilde{A} - \tilde{P}^{-1} < 0$. Inequality (48) is recovered by noting that $\tilde{P} + \tilde{B}\tilde{B}^\top + \frac{1}{4}\tilde{B}\tilde{B}^\top\tilde{P}^{-1}\tilde{B}\tilde{B}^\top = \left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top\right)\tilde{P}^{-1}\left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top\right)$. □

Theorem 1. *Suppose that system* (40) *is stabilizable. Consider the output* (44) *with* $\tilde{P} = \tilde{P}^\top > 0$ *a solution of* (41)*. The control*

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

with

$$u_{\text{lin}}(x_k) = -\frac{1}{4}\sigma(x_k, u_{\text{lin}}(x_k)) \tag{51}$$

renders the system passive with storage function (22)*, output* s_{k+1} (44)*, and new input* $v_k + \tilde{\delta}_k$*.*

Notice that (51) is the counterpart of (28).

Proof. According to (45), the control (51) is implicitly defined by the equation

$$u_{\text{lin}}(x_k) = -\frac{1}{2}\tilde{B}^\top\tilde{P}^{-1}\left(\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k)\right). \tag{52}$$

It can be solved explicitly as

$$u_{\text{lin}}(x_k) = -\left(2I + \tilde{B}^\top\tilde{P}^{-1}\tilde{B}\right)^{-1}\tilde{B}^\top\tilde{P}^{-1}\tilde{A}x_k\,, \tag{53}$$

which gives

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

By the Matrix Inversion Lemma [3, Corollary 3.9.8],

$$\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k) = \tilde{P}\left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top\right)^{-1}\tilde{A}x_k. \tag{54}$$

It now follows from Lemma 1 that

$$-x_k^\top\tilde{Q}x_k = \left(\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k)\right)^\top\tilde{P}^{-1}\left(\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k)\right) - x_k^\top\tilde{P}^{-1}x_k \tag{55}$$

for some positive definite matrix $\tilde{Q}$. The dissipation equality (46) can be written as

$$\Delta V(x_k) = s_{k+1}^\top(v_k + \tilde{\delta}_k) - (v_k + \tilde{\delta}_k)^\top\tilde{B}^\top\tilde{P}^{-1}\tilde{B}(v_k + \tilde{\delta}_k) - x_k^\top\tilde{Q}x_k, \tag{56}$$

which is the counterpart of (29). This shows passivity with the required ingredients. □

Stability Analysis

It follows from (56) that the origin is exponentially stable if v_k is such that

$$s_{k+1}^\top (v_k + \tilde{\delta}_k) \leq 0. \tag{57}$$

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

$$v_k = u_{\rm sv}(x_k) \tag{58}$$

with $u_{\rm sv}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(\tilde{K} s_{k+1})$. Such control does satisfy (57) but, unfortunately, the passive output (44) is not available to the controller, because it depends directly on the unknown perturbation. To avoid this problem let us follow [10] and define the nominal output

$$\tilde{s}_{k+1} := \sigma(x_k, u_k) \tag{59}$$

instead of (44). Since $s_{k+1} = \tilde{s}_{k+1} + 2\tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k$, the forward difference (56) becomes

$$\Delta V(x_k) = \left(\tilde{s}_{k+1} + 2\tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k\right)^\top \left(v_k + \tilde{\delta}_k\right) - (v_k + \tilde{\delta}_k)^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}(v_k + \tilde{\delta}_k) - x_k^\top \tilde{Q} x_k. \tag{60}$$

By developing the right-hand side it is obtained:

$$\begin{aligned}\Delta V(x_k) = \tilde{s}_{k+1}^\top \left(v_k + \tilde{\delta}_k\right) + 2\delta_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B} v_k + 2\delta_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k \\ - v_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B} v_k - 2v_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k - \tilde{\delta}_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k - x_k^\top \tilde{Q} x_k,\end{aligned} \tag{61}$$

which can be simplified as

$$\Delta V(x_k) = \tilde{s}_{k+1}^\top (v_k + \delta_k) + \tilde{\delta}_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k - v_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B} v_k - x_k^\top \tilde{Q} x_k. \tag{62}$$

Define the set-valued control

$$u_{\rm sv}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1}), \quad \tilde{K} = \tilde{K}^\top > 0, \tag{63}$$

as a discrete-time counterpart of (30). It gives the inequality

$$\Delta V(x_k) \leq -\|\tilde{K}\tilde{s}_{k+1}\|_1 (1 - \|\tilde{K}^{-1}\tilde{\delta}_k\|_\infty) - v_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B} v_k - x_k^\top \tilde{Q} x_k + \tilde{\delta}_k^\top \tilde{B}^\top \tilde{P}^{-1}\tilde{B}\tilde{\delta}_k. \tag{64}$$

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 $\tilde{\delta}_k$.

Closed-loop Dynamics

The virtual output satisfies

$$\tilde{s}_{k+1} \in 2\tilde{B}^\top \tilde{P}^{-1} \left(\tilde{A}x_k + \tilde{B}u_{\mathrm{lin}}(x_k) - \tilde{B}\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1}) \right) \tag{65}$$

or, using (54),

$$\tilde{s}_{k+1} \in 2\tilde{B}^\top \left(\left(\tilde{P} + \frac{1}{2}\tilde{B}\tilde{B}^\top \right)^{-1} \tilde{A}x_k - \tilde{P}^{-1}\tilde{B}\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1}) \right). \tag{66}$$

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

$$\tilde{s}'_{k+1} \in -(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{\frac{1}{2}} \tilde{K}\,\mathbf{Sgn}\left(\tilde{K}(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{\frac{1}{2}} \tilde{s}'_{k+1} \right) + \tilde{L}x_k. \tag{67}$$

The generalized equation (67) shares the same structure as the differential inclusions (20) and (34). Namely, using once again the chain rule of Convex Analysis (see Fact 8 in "Convex Analysis Tools") it is obtained:

$$\tilde{s}'_{k+1} \in -\partial\tilde{g}(\tilde{s}'_{k+1}) + \tilde{L}x_k \tag{68}$$

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 (34)). This inclusion, which is of the implicit type since $\tilde{s}'_{k+1}$ appears in both sides, is equivalently rewritten as the equality

$$\tilde{s}'_{k+1} = (\mathrm{I_d} + \partial\tilde{g})^{-1}(\tilde{L}x_k). \tag{69}$$

Let us stop and explain the meaning of the right-hand side of (69) using the material in the "Proximal-Point Algorithm and Proximal Mapping", "Maximal Monotone Operators", and "Convex Analysis Tools" 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 (69) is in fact a proximal operator,

$$\tilde{s}'_{k+1} = \mathrm{Prox}_{\tilde{g}}(\tilde{L}x_k) \tag{70}$$

(see (S37) in "Proximal-Point Algorithm and Proximal Mapping").

Proposition 2. *Consider the virtual sliding variable* (70)*. We have* $\tilde{s}'_{k+1} = 0$ *if, and only if,*

$$(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-\frac{1}{2}} \tilde{K}^{-1}\tilde{L}x_k \in \mathcal{B}_\infty, \tag{71}$$

where the unit ball $\mathcal{B}_\infty$ *is defined in (S26).*

Proof. The claim follows from (S39) in "Proximal-Point Algorithm and Proximal Mapping". Indeed $\mathrm{zero}\,\mathrm{Prox}_{\tilde{g}} = \partial\tilde{g}(0) = \tilde{K}(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{\frac{1}{2}}\mathcal{B}_\infty$, where the equality follows from Fact 8 and

from (S28) in "Convex Analysis Tools", and the sets of zeroes is defined in "Proximal-Point Algorithm and Proximal Mapping". Thus, it follows from (71) that $\tilde{L}x_k \in \partial\tilde{g}(0)$ and the proof is complete. □

It is clear that, if the eigenvalues of $\tilde{K}$ are chosen sufficiently large, then the inclusion (71) 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 (65) is the counterpart of the differential inclusion (32), while (67) or (68) are the counterparts of (34). The equality in (69) is the explicit form of the implicit algorithm (68).*

Control Calculation and Input/Sliding Variable Duality

The way the set-valued part of the controller $u_{\rm sv}(x_k)$ in (63) is calculated needs to be clarified. Let us notice first that

$$u_{\rm sv}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_{k+1}) \quad \Longleftrightarrow$$
$$\tilde{K}\tilde{s}_{k+1} \in \mathbf{Sgn}^{-1}(-\tilde{K}^{-1}u_{\rm sv}(x_k)) = \mathbf{N}_{\mathcal{B}_\infty}(-\tilde{K}^{-1}u_{\rm sv}(x_k)), \quad (72)$$

where the inversion of set-valued mappings and (S28) were used (see "Convex Analysis Tools"). Now, Equation (66) is equivalently rewritten as

$$2\tilde{K}\tilde{B}^\top\left(\tilde{P}+\frac{1}{2}\tilde{B}\tilde{B}^\top\right)^{-1}\tilde{A}x_k + 2\tilde{K}\tilde{B}^\top\tilde{P}^{-1}\tilde{B}u_{\rm sv}(x_k) \in \mathbf{N}_{\mathcal{B}_\infty}\left(-\tilde{K}^{-1}u_{\rm sv}(x_k)\right), \quad (73)$$

which in turn is equivalent to

$$2\tilde{K}\tilde{B}^\top P^{-1}\tilde{B}\tilde{K}\left(\tilde{K}^{-1}(\tilde{B}^\top P^{-1}\tilde{B})^{-\frac{1}{2}}\tilde{L}x_k - (-\tilde{K}^{-1}u_{\rm sv}(x_k)\right) \in \mathbf{N}_{\mathcal{B}_\infty}\left(-\tilde{K}^{-1}u_{\rm sv}(x_k)\right). \quad (74)$$

The generalized equation (74) with unknown $u_{\rm sv}(x_k)$ can be interpreted as the dual of the generalized Equation (66) with unknown $\tilde{s}_{k+1}$. Also, it is clear from (72) that

$$\langle -v_k, \tilde{s}_{k+1}\rangle_{\tilde{K}} \geq 0. \quad (75)$$

In this regard, these variables are dual (or reciprocal). To solve (74) use is made of (S33) in "Convex Analysis Tools" so that

$$-u_{\rm sv}(x_k) = \tilde{K}\left(\mathrm{I_d} + M^{-1}\,\mathbf{N}_{\mathcal{B}_\infty}\right)^{-1}\left(\tilde{K}^{-1}\xi(x_k)\right) = \tilde{K}\,\mathrm{Proj}_M\left(\mathcal{B}_\infty;\xi(x_k)\right), \quad (76)$$

where

$$M = M^\top := 2\tilde{K}(\tilde{B}^\top\tilde{P}^{-1}\tilde{B})\tilde{K}, \quad (77)$$

$$\xi(x_k) := \tilde{K}^{-1}(\tilde{B}^\top\tilde{P}^{-1}\tilde{B})^{-1}\tilde{B}^\top\left(\tilde{P}+\frac{1}{2}\tilde{B}\tilde{B}^\top\right)^{-1}\tilde{A}x_k. \quad (78)$$

This extends the scalar case in (12). 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 $M = 2\gamma^2 I$ and (S32) can be applied to obtain

$$\begin{aligned} w_i &= \min\{\xi(x_k)_i, 1\}\,\mathbf{sgn}(\xi_k)_i \quad i = 1, \dots, n \\ u_{\mathrm{sv}}(x_k) &= -\gamma(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-1/2} w \end{aligned} . \tag{79}$$

Otherwise, the controller can be computed as the solution of a quadratic program under constraints, similarly to (13),

$$\mathrm{Proj}_M(\mathcal{B}_\infty; \tilde{K}^{-1}\xi(x_k)) = \underset{z \in \mathcal{B}_\infty}{\arg\min} \frac{1}{2}(z - \tilde{K}^{-1}\xi(x_k))^\top M (z - \tilde{K}^{-1}\xi(x_k)), \tag{80}$$

see "Convex Analysis Tools" for $\mathcal{B}_\infty$, (S33a) and the associated piece of code for the computation of the projection. Note that (67) can be written as the generalized equation

$$\begin{aligned} \tilde{s}'_{k+1} &= \tilde{L}x_k + u'_{\mathrm{sv}}(x_k) \\ u'_{\mathrm{sv}}(x_k) &\in -\partial\tilde{g}(\tilde{s}'_{k+1}). \end{aligned} \tag{81}$$

Hence, yet another alternative is to write

$$\tilde{L}x_k + u'_{\mathrm{sv}}(x_k) = \tilde{s}'_{k+1} \in -\partial\tilde{g}^\star(u'_{\mathrm{sv}}(x_k)), \tag{82}$$

where $\tilde{g}^\star(\cdot)$ is the conjugate function to $\tilde{g}(\cdot)$ (see Definition 9 in "Convex Analysis Tools"). After rearranging terms, it follows from the definition of proximal map that

$$u'_{\mathrm{sv}}(x_k) = -\operatorname{Prox}_{\tilde{g}^\star}(\tilde{L}x_k) \tag{83}$$

Notice that (83) is the "dual" version of (70). Thus, the input selection u_{sv} can be given as

$$u_{\mathrm{sv}}(x_k) = (\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-\frac{1}{2}} u'_{\mathrm{sv}}(x_k). \tag{84}$$

Controller During The Sliding Mode

The controller during the continuous-time sliding mode satisfies (37). Its discrete-time counterpart is:

$$\begin{aligned} u_{\mathrm{sv}}(x_k) &= -(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-\frac{1}{2}}\tilde{L}x_k \\ &= -(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{-1}\tilde{B}\tilde{P}^{-1}(\tilde{A} - I_n)x_k - \tilde{\delta}_{k-1}. \end{aligned} \tag{85}$$

The first equality stems from (81) and (84), while the second equality is proved as follows. We have

$$\begin{aligned} \tilde{s}_{k+1} &= s_k + 2\tilde{B}^\top \tilde{P}^{-1}\tilde{B}u_{\mathrm{sv}}(x_k) + 2\tilde{B}^\top \tilde{P}^{-1}(\tilde{A} - I_n)x_k \\ &= \tilde{s}_k + 2\tilde{B}^\top \tilde{P}^{-1}\tilde{B}(u_{\mathrm{sv}}(x_k) + \tilde{\delta}_{k-1}) + 2\tilde{B}^\top \tilde{P}^{-1}(\tilde{A} - I_n)x_k. \end{aligned} \tag{86}$$

The result follows by setting $\tilde{s}_{k+1} = \tilde{s}_k = 0$. This proves that the exact compensation in continuous-time 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.

Discrete-Time Classical Approach

Let us pass to the discretization of the first controller in (15), (17). To start with, consider (40) and set $\tilde{s}_{k+1} = \sigma(x_k, u_k)$ as in (59), 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

$$\tilde{C}(\tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k)) = \tilde{C}x_k = s_k \tag{87}$$

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

$$u_{\text{lin}}(x_k) = (\tilde{C}\tilde{B})^{-1}\tilde{C}(I_n - \tilde{A})x_k. \tag{88}$$

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

$$s_{k+1} = s_k + v_k + \tilde{C}\tilde{B}\tilde{\delta}_k \tag{89a}$$

$$\tilde{s}_{k+1} = s_k + v_k. \tag{89b}$$

From (89b) it is natural to set: $v_k = u_{\text{sv}}(x_k) = -\tilde{K}\,\mathbf{Sgn}(\tilde{s}_{k+1})$. It gives

$$s_{k+1} \in s_k - \tilde{K}\,\mathbf{Sgn}(\tilde{s}_{k+1}) + \tilde{C}\tilde{B}\tilde{\delta}_k \tag{90a}$$

$$\tilde{s}_{k+1} \in s_k - \tilde{K}\,\mathbf{Sgn}(\tilde{s}_{k+1}) \tag{90b}$$

A generalized equation with unknown $\tilde{s}_{k+1}$ is obtained in (90b). The resemblance between equalities in (90b) and (66) is clear. Define $\tilde{s}'_k = \tilde{K}^{-\frac{1}{2}}\tilde{s}_k$ and note that (90b) is equivalent to

$$\tilde{s}'_{k+1} \in \tilde{K}^{-\frac{1}{2}}s_k - \tilde{K}^{\frac{1}{2}}\,\mathbf{Sgn}(\tilde{K}^{\frac{1}{2}}\tilde{s}'_{k+1}) = \tilde{K}^{-\frac{1}{2}}s_k - \partial f(\tilde{s}'_{k+1}) \tag{91}$$

with $f(\cdot) = \|\cdot\|_1 \circ \tilde{K}^{\frac{1}{2}}$. Once again, the chain rule has been used (see Fact 8 in "Convex Analysis Tools"). Notice that the functions $f(\cdot)$ in (91) and (20) share the same structure and are both proper, convex, and lsc. Thus, the expression (91) is also a generalized equation with unknown $\tilde{s}'_{k+1}$, rewritten equivalently as

$$\tilde{s}'_{k+1} = \text{Prox}_f(\tilde{K}^{-\frac{1}{2}}s_k) = \text{Prox}_f(\tilde{K}^{-\frac{1}{2}}(\tilde{s}_k + \tilde{C}\tilde{B}\tilde{\delta}_{k-1})) = \text{Prox}_{\left\|\tilde{K}^{\frac{1}{2}}(\cdot)\right\|_1}(\tilde{s}'_k + \tilde{C}\tilde{B}\tilde{\delta}_{k-1}). \tag{92}$$

This algorithm may be named a *robust* or *perturbed proximal-point algorithm*. Using (89) it is also inferred that

$$s_{k+1} = \tilde{K}^{\frac{1}{2}}\,\text{Prox}_{\left\|\tilde{K}^{\frac{1}{2}}(\cdot)\right\|_1}(s_k) + \tilde{C}\tilde{B}\tilde{\delta}_k. \tag{93}$$

Remark 5. *The above control strategy has been proposed in [10], [11]. For instance, Equation (90b) is exactly [11, Equation (9)].*

Stability Analysis and Sliding Mode

Let us briefly recall the main stability properties of (90). It is proved in [11, Proposition 2] that $\{\tilde{s}_k\}_{k\in\mathbb{N}}$ solution of (90b) converges to zero in a finite number of steps $k_{\min} < +\infty$ (cf. Proposition 6 in "Proximal-Point Algorithm and Proximal Mapping" as well), while $\{s_k\}_{k\in\mathbb{N}}$ solution of (90a) is bounded. During the sliding mode ($\tilde{s}_k = 0$ for $k \geq k_{\min}$), we have $s_{k+1} = \tilde{C}\tilde{B}\tilde{\delta}_k$ since

$$s_k + u_{\mathrm{sv}}(x_k) = 0 \Longleftrightarrow u_{\mathrm{sv}}(x_k) = -s_k = -\tilde{C}\tilde{B}\tilde{\delta}_{k-1}. \tag{94}$$

As a fundamental result, the control (19) is approximated by (94): in discrete time, the implicit method allows to design an input which compensates for the perturbation with a one-step delay, in a similar way to (37) and (85). Once again the set-valued controller does not depend on the gain $\tilde{K}$ during the sliding phase.

Control-Input Calculation

Let us proceed as in (72) through (76) to compute the set-valued controller $u_{\mathrm{sv}}(x_k)$. Using again Proposition 5, Equation (90), and mapping inversion (see "Set-Valued Mappings and Differential Inclusions"), it follows that

$$\begin{gathered}\tilde{K}\left(\tilde{K}^{-1}s_k - (-\tilde{K}^{-1}u_{\mathrm{sv}}(x_k)\right) \in \mathbf{N}_{\mathcal{B}_\infty}\left(-\tilde{K}u_{\mathrm{sv}}(x_k)\right) \Longleftrightarrow \\ -u_{\mathrm{sv}}(x_k) = \tilde{K}(\mathrm{I_d} + \tilde{K}^{-1}\,\mathbf{N}_{\mathcal{B}_\infty})^{-1}(\tilde{K}^{-1}s_k) \Longleftrightarrow -u_{\mathrm{sv}}(x_k) = \tilde{K}\,\mathrm{Proj}_{\tilde{K}}\left(\mathcal{B}_\infty; \tilde{K}^{-1}s_k\right).\end{gathered} \tag{95}$$

Again, the projection can be computed by solving a quadratic program or, similarly to (79), conveniently choose $\tilde{K} = \gamma I$ so that

$$(u_{\mathrm{sv}}(x_k))_i = -\gamma \min\left\{\frac{|(s_k)_i|}{\gamma}, 1\right\} \mathbf{sgn}((s_k)_i). \tag{96}$$

See "Convex Analysis Tools" for $\mathcal{B}_\infty$, (S33a) and the associated piece of code for the computation of the projection in a general setting.

Remark 6 (Dead-beat controller)**.** *The linear controllers in (15) and in (88) have different structures. Suppose that $u_{\mathrm{lin}}(x_k)$ is computed as $u_{\mathrm{lin}}(x_k) = -(\tilde{C}\tilde{B})^{-1}\tilde{C}\tilde{A}x_k$, which would be a naive implementation of (15). Then,*

$$\tilde{C}x_{k+1} = \tilde{C}\tilde{A}x_k + \tilde{C}\tilde{B}(u_{\mathrm{lin}}(x_k) + \tilde{\delta}_k) + v_k = v_k + \tilde{C}\tilde{B}\tilde{\delta}_k, \tag{97}$$

which is different from (89). Applying the above $u_{\mathrm{sv}}(x_k)$ yields $\tilde{s}_{k+1} \in -\tilde{K}\,\mathbf{Sgn}(\tilde{s}_{k+1})$, which is equivalent to $\tilde{s}'_{k+1} = (\mathrm{I_d} + \partial f)^{-1}(0)$ and, since the operator $(\mathrm{I_d} + \partial f)^{-1}(\cdot)$ is single-valued and its graph contains $(0, 0)$, we have $\tilde{s}_{k+1} = 0$. Therefore, this algorithm is a one-step dead-beat controller.

The implicit algorithms introduced in [10], [11], [13] are not of the dead-beat type. It is noteworthy that a major discrepancy exists between (88) 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 $\tilde{A} = e^{Ah}$, $\tilde{B} = e^{Ah} \int_0^h e^{-A\tau} B d\tau$. 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.

IMPLICIT DISCRETE-TIME ALGORITHMS: RECAPITULATION

Link with Optimization

It is interesting to note that operators of the form $(\mathrm{I_d} + \mathbf{M})^{-1}(\cdot)$ are ubiquitous in the above discrete-time algorithms: they are the well-known resolvent mappings, very close to Yosida approximations (see (S11) and (S14) in "Maximal Monotone Operators"). They have been exhibited in (9), (10), (69), (76), (92), and (95). They are a direct consequence of the presence of a maximal monotone mapping $\mathbf{M}(\cdot)$ in the right-hand side of the differential inclusions in (20), (34). These operators are resolvents of mappings of the form $\mathbf{M}(\cdot) = \partial f(\cdot)$ for some proper convex lower semicontinuous function $f(\cdot)$. This shows the strong connection between the implicit discretization and Optimization (see also (76)). Thus, a common feature between model-predictive control (MPC) and set-valued sliding-mode implicit discretization is that an optimization problem has to be solved to compute the controller. This is formalized in "Implicit Sliding-Mode and Model Predictive Control", where the implicitly discretized scheme is reinterpreted as a one-step MPC. The explicit discretization (*i.e.*, choosing $u_{\mathrm{sv}}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(s_k)$) yields operators of the form $(\mathrm{I_d} - \partial f)(\cdot)$, which are neither monotone nor possess interesting properties, neither in terms of stability nor robustness.

Clearly, the above analysis is closely related to the analysis of proximal-point algorithms (Definition 10 in "Proximal-Point Algorithm and Proximal Mapping"). However, a peculiarity of discrete sliding-mode control is the notion of a *robust* proximal-point algorithm, formulated in (S41) in "Proximal-Point Algorithm and Proximal Mapping" and shown to converge in a finite number of steps, given sufficiently large gains. It is worth emphasizing that (63) has the form $u_{\mathrm{sv}}(x_k) \in -\partial \|\tilde{K}\tilde{s}_{k+1}\|_p$ 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 (76) by $\mathcal{B}_{p^*}$. For example, the implicit discretization of the unit control is

$$-u_{\mathrm{sv}}(x_k) = \tilde{K}\,\mathrm{Proj}_M\left(\mathcal{B}_2; \tilde{K}^{-1}\xi(x_k)\right). \tag{98}$$

Again, we can choose $\tilde{K}$ such that $M = 2\gamma^2 I$ and use (S31) to compute an explicit form for the projection.

Link with Passivity

Both control design methodologies described above follow the celebrated passivity plus zero-state detectability or observability results [4], [14] (see [15] for the discrete-time version). Regarding the direct passivity-based design, Theorem 1 shows that we can choose a triplet $(V(x_k), s_{k+1}, v_k + \tilde{\delta}_k)$ 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 $u_{\text{lin}}(x_k)$, so that

$$\begin{aligned} x_{k+1} &= \tilde{A}x_k + \tilde{B}u_{\text{lin}}(x_k) + \tilde{B}(v_k + \tilde{\delta}_k) \\ s_{k+1} &= \tilde{C}x_{k+1} \end{aligned} \tag{99}$$

happens to be zero-state detectable (ZSD) [15, 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 [15, 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 1. A notion that is closely related to feedback passivity is that of control Lyapunov functions. The connection between sliding variables and control-Lyapunov functions has been explicitly recognized and exploited, *e.g.*, in [94], [118], although it is worth mentioning that the idea already appears in its early stages in [92], [93].

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 $\{x \in \mathbb{R}^n \mid \sigma(x) = 0\}$ 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, *e.g.*, of the implicitly-defined control law (63), as 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 $u_{\text{sv}}(x_k) \in -\tilde{K}\,\mathbf{Sgn}(\tilde{K}\tilde{s}_k)$ produces discretization 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, no matter how large

or small h is. At the same time, implicit discretizations are consistent, in the sense that discrete-time solutions and inputs converge to their continuous-time counterparts as $h \downarrow 0$ [11], [56]. When written explicitly, the implicit 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 the method: no trial-and-error tuning 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 exists the neighborhood [11, Lemma 10]. If, on the other hand, the discretization is implicit, we remain inside a neighborhood of the same order, $\|s_{k+1}\| = \mathcal{O}(h^2)$ [11, Lemma 11]..

The implicit discretization method results in a controlled system that is also robust with respect to uncertainty in the system parameters [55], [56]. With the appropriate choice of the maximal monotone operators, it is also possible to obtain robustness with respect to unmatched external perturbations as well [57], [S32]. 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 (*e.g.*, as in (95)). In more complex scenarios, it is always possible to resort to the numerical machinery described in section "COMPUTATIONAL ISSUES".

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 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 2 in "Set-Valued Mappings and Differential Inclusions") and calculation of the controller with optimization tools, as detailed below.

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.

Super-Twisting Algorithm

Differentiator

The super-twisting differentiation algorithm [49] reads as follows:

$$\begin{cases} \dot{x}_0 = x_1 - \gamma_0 L^{\frac{1}{2}} |\sigma_0|^{\frac{1}{2}} \operatorname{\mathbf{sgn}}(\sigma_0) \\ \dot{x}_1 \in -\gamma_1 L \operatorname{\mathbf{sgn}}(\sigma_0) \end{cases}, \tag{100}$$

where $\sigma_0(t) = x_0(t) - f(t)$, $f(t)$ is the signal to be differentiated, and γ_0, γ_1 are positive gains. On the sliding surface $\sigma_0(t) \equiv 0$ we have $x_1(t) \equiv \dot{f}(t)$. It is assumed that $\dot{f}(\cdot)$ is Lipschitz continuous with Lipschitz constant $L > 0$. Following [29, Example 7], system (100) is interpreted as the feedback interconnection of a linear invariant system and a maximal monotone mapping (a set-valued static nonlinearity). Let $x = (x_0 \;\; x_1)^\top$ be the state and let

$$A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} \gamma_0 L^{\frac{1}{2}} & 0 \\ 0 & \gamma_1 L \end{pmatrix}, \quad C = \mathbf{e} \cdot (1 \quad 0) = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}, \quad D = -\mathbf{e}, \tag{101}$$

where $\mathbf{e} = (1 \;\; 1)^\top$. Then, (100) is equivalently rewritten as the set-valued Lur'e system (or, a nonlinear relay system [S29, section 2.6.2]):

$$\begin{cases} \dot{x} = Ax + B\lambda \\ w = Cx + Df(t) \\ \lambda \in -\mathbf{M}(w) \end{cases} \iff \begin{cases} \dot{\bar{x}} = A\bar{x} + B\lambda + \bar{B}\dot{f}(t) \\ w = C\bar{x} \\ \lambda \in -\mathbf{M}(w) \end{cases}, \tag{102}$$

where $\bar{B} = (-1 \;\; 0)^\top$, $\mathbf{M}(w) = (|w_1|^{\frac{1}{2}} \operatorname{\mathbf{sgn}}(w_1), \;\; \operatorname{\mathbf{sgn}}(w_2))^\top$, $w_1 = w_2 = \sigma_0$, $\bar{x} = (\sigma_0 \;\; x_1)^\top$. The mapping $\mathbf{M}(\cdot)$ is maximal monotone. However, the triplet (A, B, C) is not passive, which makes the study of the implicit discretization of (102) more difficult.

The implicit Euler discretization of (100) is

$$\sigma_{0,k+1} = \sigma_{0,k} + hx_{1,k+1} - h\gamma_0 L^{\frac{1}{2}} |\sigma_{0,k+1}|^{\frac{1}{2}} \operatorname{\mathbf{sgn}}(\sigma_{0,k+1}) - \Delta f_k \tag{103a}$$

$$x_{1,k+1} \in x_{1,k} - h\gamma_1 L \operatorname{\mathbf{sgn}}(\sigma_{0,k+1}), \tag{103b}$$

where $\Delta f_k = f_{k+1} - f_k$. Substituting (103b) into (103a) and solving for $\sigma_{0,k+1}$ yields

$$\sigma_{0,k+1} = (\mathrm{I_d} + h\,\mathbf{M}_h)^{-1}(\sigma_{0,k} + hx_{1,k} - \Delta f_k), \tag{104}$$

where

$$\mathbf{M}_h(x) := \gamma_0 |x|^{\frac{1}{2}} \operatorname{\mathbf{sgn}}(x) + h\gamma_1 L \operatorname{\mathbf{sgn}}(x) \tag{105}$$

is a maximal monotone operator (being the sum of two maximal monotone operators, both with domain $\mathbb{R}$). The iteration (104) is written compactly as the robust proximal-point algorithm

$$\sigma_{0,k+1} = \mathcal{J}_{h\,\mathbf{M}_h}(\sigma_{0,k} + hx_{1,k} - \Delta f_k) \tag{106}$$

(see (S41) and Proposition 6 in "Proximal-Point Algorithm and Proximal Mapping"), where the resolvent is defined in (S11) in "Maximal Monotone Operators" (compare with the classical algorithm (92) and the general iteration (S41)). It is noteworthy that the implicit discretization of the super-twisting controller shares the same closed-loop structure [30, equations (8),(9)]. It is also interesting to note that choosing an exact discretization as in [31] and neglecting unmeasurable terms also yields a form like (106) [31, Equation (24)]. Together with (9), (69), and (92), this shows the ubiquity of robust proximal-point algorithms in the implicit discretization of sliding-mode systems. The algorithms provided in [31], [33] also correspond to the resolvent in (106). As seen later in this article, there is no unique way to compute numerically the resolvents.

With an explicit expression for $\sigma_{0,k+1}$ available in (106), we can readily solve for the state $x_{1,k+1}$ for all $h > 0$ in (103a):

$$x_{1,k+1} = \frac{1}{h}\left(\sigma_{0,k+1} - \sigma_{0,k} + h\gamma_0 L^{\frac{1}{2}}|\sigma_{0,k+1}|^{\frac{1}{2}}\,\mathbf{sgn}(\sigma_{0,k+1}) + \Delta f_k\right). \tag{107}$$

Therefore, the pair of equations (106), (107) forms a nonlinear difference equation with state vector $(\sigma_{0,k}, x_{1,k})^\top$ and allows to advance the discretized super-twisting differentiator from step k to step $k+1$. Notice in passing that, having computed the future state $x_{1,k+1}$ in (107), we can use $x_{1,k+1} = x_{1,k} - h\gamma_1 L\xi_{k+1}$ to compute a selection $\xi_{k+1} \in \mathbf{sgn}(\sigma_{0,k+1})$ such that (103b) holds.

Remark 7. *In real-time applications it is possible to substitute Δf_k by Δf_{k-1} to render the differentiator non-anticipative.*

Semi-implicit discretization is also possible. Let us replace (103) by:

$$\sigma_{0,k+1} \in \sigma_{0,k} + hx_{1,k+1} - h\gamma_0 L^{\frac{1}{2}}|\sigma_{0,k}|^{\frac{1}{2}}\,\mathbf{sgn}(\sigma_{0,k+1}) - \Delta f_k \tag{108a}$$

$$x_{1,k+1} \in x_{1,k} - h\gamma_1 L\,\mathbf{sgn}(\sigma_{0,k+1}), \tag{108b}$$

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

$$\sigma_{0,k+1} = (\mathrm{I_d} + (h\gamma_0 L^{\frac{1}{2}}|\sigma_{0,k}|^{\frac{1}{2}} + h^2\gamma_1 L)\,\mathbf{sgn})^{-1}(\sigma_{0,k} + hx_{1,k} - \Delta f_k). \tag{109}$$

Compared to (106), the maximal monotone operator is modified to

$$\mathbf{M}'_h(x) := (\gamma_0 L^{\frac{1}{2}}|\sigma_{0,k}|^{\frac{1}{2}} + h\gamma_1 L)\,\mathbf{sgn}(x). \tag{110}$$

Computing $x_{1,k+1}$ is somewhat more complicated than before. First let us define the selection $\xi_{k+1} \in \mathbf{sgn}(\sigma_{0,k+1})$ and recall that it holds if, and only if, $\sigma_{0,k+1} \in \mathbf{N}_{[-1,1]}(\xi_{k+1})$ (see Definition 2 in "Set-Valued Mappings and Differential Inclusions", Fact 7 and (S25) in "Convex Analysis

Tools"). By inverting the operators in (108), the system in the new coordinates can be rewritten as

$$\sigma_{0,k} + hx_{1,k+1} - h\gamma_0 L^{\frac{1}{2}} |\sigma_{0,k}|^{\frac{1}{2}} \xi_{k+1} - \Delta f_k \in \mathbf{N}_{[-1,1]}(\xi_{k+1}) \tag{111a}$$

$$x_{1,k+1} = x_{1,k} - h\gamma_1 L \xi_{k+1}, \tag{111b}$$

and then solve for ξ_{k+1}. To do so, first substitute (111b) in (111a) to obtain

$$\xi_{k+1} \in \frac{1}{h^2\gamma_1 L} \left[\sigma_{0,k} + hx_{1,k} - h\gamma_0 L^{\frac{1}{2}} |\sigma_{0,k}|^{\frac{1}{2}} \xi_{k+1} - \Delta f_k - \mathbf{N}_{[-1,1]}(\xi_{k+1}) \right]. \tag{112}$$

The inclusion is equivalent to

$$\xi_{k+1} = \mathcal{J}_{\mathbf{M}'_{1,h}} \left(\frac{\sigma_{0,k} + hx_{1,k} - \Delta f_k}{h^2\gamma_1 L} \right), \quad \mathbf{M}'_{1,h}(\xi_{k+1}) = \frac{\gamma_0}{h\gamma_1 L^{\frac{1}{2}}} |\sigma_{0,k}|^{\frac{1}{2}} \xi_{k+1} + \mathbf{N}_{[-1,1]}(\xi_{k+1}) \tag{113}$$

Finally, we can recover $x_{1,k+1}$ from (111b):

$$x_{1,k+1} = x_{1,k} - h\gamma_1 L \, \mathcal{J}_{\mathbf{M}'_{1,h}} \left(\frac{\sigma_{0,k} + hx_{1,k} - \Delta f_k}{h^2\gamma_1 L} \right). \tag{114}$$

The resolvent $\mathcal{J}_{\mathbf{M}'_{1,h}}(\cdot)$ can be calculated as indicated in "Proximal-Point Algorithm and Proximal Mapping", after (S40). It happens that resolvents of this kind are basic tools which are to be found in all the higher-order algorithms in this section.

Controller

When the super-twisting algorithm is used for control (as opposed to differentiation), a perturbed integrator $\dot{x}_0(t) = u(t) + \varphi(t)$ is considered. The proposed controller is given by

$$\begin{cases} u(t) = \nu(t) - \gamma_0 L^{\frac{1}{2}} |x_0(t)|^{\frac{1}{2}} \, \mathbf{sgn}(x_0(t)) \\ \dot{\nu}(t) \in -\gamma_1 L \, \mathbf{sgn}(x_0(t)) \end{cases}. \tag{115}$$

By making the time-varying change of variable $x_1(t) = \nu(t) + \varphi(t)$ the next closed-loop system is obtained:

$$\begin{cases} \dot{x}_0(t) = x_1(t) - \gamma_0 L^{\frac{1}{2}} |x_0(t)|^{\frac{1}{2}} \, \mathbf{sgn}(x_0(t)) \\ \dot{x}_1(t) \in -\gamma_1 L \, \mathbf{sgn}(x_0(t)) + \delta(t) \end{cases} \tag{116}$$

with $\delta(t) = \dot{\varphi}(t)$. Notice that, similarly to (19) and (37), the controller compensates exactly for the perturbation on the sliding surface $(x_0, x_1) = (0, 0)$, where $u(t) = \nu(t) = -\varphi(t)$. The control input has to be computed at each time step, so it is interesting to make the link with the explicit expressions (11) and (12). A discrete-time model for the plant is

$$\begin{cases} x_{0,k+1} = x_{0,k} + hu_k + h\bar{\varphi}_k \\ \varphi_{k+1} = \varphi_k + h\bar{\delta}_k \end{cases}. \tag{117}$$

Here, $\bar{\varphi}_k$ and $\bar{\delta}_k$ are discretized versions of $\varphi(t)$ and $\delta(t)$ (see [30] for details). The implicit discretization can now be achieved by following the same steps as in (103). Let us introduce, for instance, an unperturbed virtual variable $\tilde{x}_{0,k} := x_{0,k} - h\bar{\varphi}_{k-1}$ as done in (59) and (89), see [30, Equations (8), (9)]. Let us introduce the virtual variable in the first line of (117) to obtain its dynamics. These are [30], [34]:

$$\tilde{x}_{0,k+1} = (\mathrm{I_d} + h\,\mathbf{M}_h)^{-1}(x_{0,k} + h\nu_k) = \mathcal{J}_{h\,\mathbf{M}_h}(\tilde{x}_{0,k} + h\nu_k + h\bar{\varphi}_{k-1}) \tag{118a}$$

$$u_k(x_{0,k}, \nu_k) = -\gamma_0 L^{\frac{1}{2}} \beta_k\, \mathbf{sgn}(x_{0,k} + h\nu_k) - h\gamma_1 L \mathrm{Proj}\left([-1,1]; \frac{x_{0,k} + h\nu_k}{h^2\gamma_1 L}\right) + \nu_k \tag{118b}$$

$$\beta_k = \frac{-h\gamma_0 L^{\frac{1}{2}}}{2} + \sqrt{\frac{h^2\gamma_0^2 L}{4} + \max\{0, |x_{0,k} + h\nu_k| - h^2\gamma_1 L\}} \tag{118c}$$

with $\mathbf{M}_h(\cdot)$ is as in (105) and $u_k = \frac{1}{h}(\mathrm{I_d} + h\,\mathbf{M}_h)^{-1}(x_{0,k} + h\nu_k) - \frac{x_{0,k}}{h}$. It is noteworthy that this boils down once again to the calculation of a resolvent and of a projection (the saturation function). The algorithm in (118a) is a robust proximal-point algorithm similar to (92), (106), and (108), an instance of (S41) indeed. Compare (118b) with (12c), and (118a) with (106).

Similarly as above, a semi-implicit method can be applied [35], yielding $\tilde{x}_{0,k+1} = (\mathrm{I_d} + h\,\mathbf{M}'_h)^{-1}(x_{0,k} + h\nu_k)$ with $\mathbf{M}'_h(x) := (\gamma_0 L^{\frac{1}{2}}|x_{0,k}|^{\frac{1}{2}} + h\gamma_1 L)\,\mathbf{sgn}(x)$ as in (110). Notice that another choice is $\mathbf{M}''_h(x) = (\gamma_0 L^{\frac{1}{2}}|\tilde{x}_{0,k}|^{\frac{1}{2}} + h\gamma_1 L)\,\mathbf{sgn}(x)$. Which alternative should be chosen is an open issue. When $\tilde{x}_{0,k+1} = \tilde{x}_{0,k} = 0$, we obtain from (118a): $\nu_k + \bar{\varphi}_k \in \frac{1}{h}(\mathrm{I_d} + h\,\mathbf{M}_h)(0) = h\gamma_1 L[-1,1]$, and using (118b) (118c):

$$u_k = -\frac{x_{0,k}}{h} = -\frac{\tilde{x}_{0,k} + h\bar{\varphi}_{k-1}}{h} = -\bar{\varphi}_{k-1}. \tag{119}$$

It is inferred that, similarly to first order controllers, see (85) and (94), the implicit super-twisting scheme compensates for the perturbation with one-step delay.

Multivariable super-twisting

The above extends to the multivariable super-twisting [116], with $x_0,\ x_1 \in \mathbb{R}^n$. One replaces $|x_0|^{\frac{1}{2}}\,\mathbf{sgn}(x_0)$ by $\frac{x_0}{||x_0||^{\frac{1}{2}}} = ||x_0||^{\frac{1}{2}}\partial||x_0||_2 = ||x_0||^{\frac{1}{2}}\,\mathbf{N}^{-1}_{\mathcal{B}_2}(x_0)$ (see (S29) in "Convex Analysis Tools"), and $\mathbf{sgn}(x_0)$ by $\partial||x_0||_2$ (or using another norm), in (115). This yields the resolvent $\mathcal{J}_{h\,\mathbf{M}_h}(\cdot)$ to be computed at each step, with $\mathbf{M}_h(x) = \alpha_1||x||^{\frac{1}{2}}\partial||x||_2 + h\alpha_2\partial||x||_2$, $\alpha_1 > 0$, $\alpha_2 > 0$. It follows from (S29) and Fact 4 in "Convex Analysis Tools", or Theorem 4 in "Maximal Monotone Operators" , that $\mathbf{M}_h(\cdot)$ is maximal monotone.

Generalized Observer

Let us consider the class of set-valued super-twisting observers presented in [115, Equations (12.3)–(12.5)]:

$$\begin{cases} \dot{\hat{x}}_1 = f_1(\hat{x}_1, u) + \hat{x}_2 - l_1\gamma\phi_1(e_1) \\ \dot{\hat{x}}_2 \in f_2(\hat{x}_1, \hat{x}_2, u) - l_2\gamma^2\phi_2(e_1) \\ \phi_1(e_1) = \mu_1\sqrt{|e_1|}\,\mathbf{sgn}(e_1) + \mu_2|e_1|^q\,\mathbf{sgn}(e_1) \\ \phi_2(e_1) = \frac{\mu_1^2}{2}\,\mathbf{sgn}(e_1) + \mu_1\mu_2|e_1|^{q-\frac{1}{2}}\,\mathbf{sgn}(e_1) + \mu_2^2|e_1|^{2q-1}\,\mathbf{sgn}(e_1), \end{cases} \tag{120}$$

where $\mu_1 \geq 0$, $\mu_2 \geq 0$, $\mu_1\mu_2 \neq 0$, $q \geq \frac{1}{2}$, $l_1 > 0$, $l_2 > 0$, $\gamma > 0$, $f_1(\cdot)$ and $f_2(\cdot)$ are plant nonlinearities, $e_1 = \hat{x}_1 - x_1$ is the state estimation error, $y = x_1$ is the plant's output. An implicit discretization of (120) reads as:

$$\hat{x}_{1,k+1} = \hat{x}_{1,k} + hf_1(\hat{x}_{1,k+1}, u_k) + h\hat{x}_{2,k+1} - hl_1\gamma\phi_1(e_{1,k+1}) \tag{121a}$$

$$\hat{x}_{2,k+1} \in \hat{x}_{2,k} + hf_2(\hat{x}_{1,k+1}, \hat{x}_{2,k+1}, u_k) - hl_2\gamma^2\phi_2(e_{1,k+1}) \tag{121b}$$

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 $e_{1,k+1}$, we assume that (121b) 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 $\Phi(\cdot)$. Inserting the latter in (121a) yields:

$$e_{1,k+1} \in y_{k-1} - y_k + e_{1,k} + hf_1(e_{1,k+1}, y_k, u_k) + h\Phi(e_{1,k+1}, y_k, \hat{x}_{2,k}, u_k, h\phi_2(e_{1,k+1})) - hl_1\gamma\phi_1(e_{1,k+1}). \tag{122}$$

It is inferred that:

$$e_{1,k+1} \in (\mathrm{I_d} - hf_1(\cdot, y_k, u_k) + h\Phi(\cdot, y_k, \hat{x}_{2,k}, u_k, h\phi_2(\cdot)) + hl_1\gamma\phi_1(\cdot))^{-1}(y_{k-1} - y_k + e_{1,k}) \tag{123}$$

Then, provided that the operators

$$x \mapsto -f_1(x, y_k, u_k) + l_1\gamma\phi_1(x) \quad \text{and} \quad x \mapsto \Phi(x, y_k, \hat{x}_{2,k}, u_k, h\phi_2(x)) \tag{124}$$

are maximal monotone (notice that $\phi_1(\cdot)$ and $\phi_2(\cdot)$ are strongly monotone for $q > \frac{1}{2}$), the inclusion in (123) 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 (121a) and (123). Of course semi-implicit methods can also be applied so that the computations are easier.

Higher-Order Differentiators

As shown in [33] and [31], [32], higher-order differentiators lend themselves to transformations which yield the same structures as in (118a) (and consequently as in (106)). In particular it is shown in [33, Table 2] that "classical" higher-order differentiators known as URED (uniform

robust exact differentiator) [46], AO-STD (arbitrary-order super-twisting differentiator) [44], [45], HDD (homogeneous discrete-time differentiator), GHDD (generalized homogeneous discrete-time differentiator) [54], FDFF (first-order differentiator with first-order sliding-mode filtering) [41], [42], AO-FDFF (arbitrary-order differentiator with first-order sliding-mode filtering) [42], [43] yield, when implicitly or semi-implicitly discretized:

$$\sigma_{0,k+1} = \mathcal{J}_{h\,\mathbf{M}_h}(-b_k), \tag{125}$$

where $\mathbf{M}_h(\cdot) := \mathbf{M}_{\text{sing}}(\cdot) + \mathbf{M}_{\text{set},h}(\cdot)$, $\mathbf{M}_{\text{set},h}(\cdot)$ is set-valued maximal monotone, $\mathbf{M}_{\text{sing},h}(\cdot)$ is single-valued monotone, and $b_k = b(\sigma_{0,k}, x_{1,k}, \dots, x_{n,k}, \Delta f_k)$.

Let us provide the implicit URED operators, which consist of the super-twisting operators with additional higher-order terms [33, Table 2, Equation (39)]. Its defining operators are

$$\begin{aligned}\mathbf{M}_{\text{sing},h}(\sigma_{0,k+1}) := {} & (\lambda_0 L^{\frac{1}{2}}|\sigma_{0,k+1}|^{\frac{1}{2}} + \lambda_0 L^{\frac{1}{2}}\mu|\sigma_{0,k+1}|^{\frac{3}{2}} + \frac{3}{2}h\mu^2\lambda_1 L\sigma_{0,k+1}^2)\,\mathbf{sgn}(\sigma_{0,k+1}) \\ & + 2h\lambda_1 L\mu\sigma_{0,k+1} \end{aligned} \tag{126a}$$

$$\mathbf{M}_{\text{set},h}(\sigma_{0,k+1}) := \frac{1}{2}h\lambda_1 L\,\mathbf{sgn}(\sigma_{0,k+1}) \tag{126b}$$

and the argument is

$$b(\sigma_{0,k}, x_{1,k}, \Delta f_k) = -\sigma_{0,k} - hx_{1,k} + \Delta f_k, \tag{126c}$$

where λ_0, λ_1, μ are parameters. Similarly to the super-twisting differentiator, the URED has an additional state variable x_1, see, *e.g.*, [33, equation (6)]. We obtain the URED counterpart of (107):

$$x_{1,k+1} = \frac{1}{h}\left(\sigma_{0,k+1} - \sigma_{0,k} + \Delta f_k + h\lambda_0 L^{\frac{1}{2}}|\sigma_{0,k+1}|^{\frac{1}{2}}\,\mathbf{sgn}(\sigma_{0,k+1}) + hL^{\frac{1}{2}}\mu|\sigma_{0,k+1}|^{\frac{3}{2}}\,\mathbf{sgn}(\sigma_{0,k+1})\right). \tag{127}$$

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

$$\begin{aligned}\mathbf{M}'_h(\sigma_{0,k+1}) := \Big(\lambda_0 L^{\frac{1}{2}}|\sigma_{0,k}|^{\frac{1}{2}} + \lambda_0 L^{\frac{1}{2}}\mu|\sigma_{0,k}|^{\frac{3}{2}} + \frac{3}{2}h\mu^2\lambda_1 L\sigma_{0,k}^2 \\ + h\lambda_1 L|\sigma_{0,k}| + \frac{1}{2}h\lambda_1 L\Big)\,\mathbf{sgn}(\sigma_{0,k+1}), \end{aligned} \tag{128}$$

which is the URED counterpart of (109). Let $\xi_{k+1} \in \mathbf{sgn}(\sigma_{0,k+1})$ be a selection of the set-valued sign function. Solving for $\sigma_{0,k+1} \in \mathbf{N}_{[-1,1]}(\xi_{k+1})$ gives

$$\begin{aligned}\sigma_{0,k} + hx_{1,k} - \Delta f_k - h\lambda_0 L^{\frac{1}{2}}\left(|\sigma_{0,k}|^{\frac{1}{2}} + \mu|\sigma_{0,k}|^{\frac{3}{2}}\right)\xi_{k+1} - \\ h^2\lambda_1 L\left(2\mu|\sigma_{0,k}| + \frac{3}{2}\mu^2\sigma_{0,k}^2 + \frac{1}{2}\right)\xi_{k+1} \in \mathbf{N}_{[-1,1]}(\xi_{k+1}), \end{aligned} \tag{129}$$

so that $\xi_{k+1} = \mathcal{J}_{\mathbf{M}'_{1,h}}\left(-\frac{2b_k}{h^2\lambda_1 L}\right)$ with

$$\mathbf{M}'_{1,h} := \frac{2}{h\lambda_1 L^{\frac{1}{2}}}\left(\lambda_0\left(|\sigma_{0,k}|^{\frac{1}{2}} + \mu|\sigma_{0,k}|^{\frac{3}{2}}\right) + h\lambda_1 L^{\frac{1}{2}}\left(2\mu|\sigma_{0,k}| + \frac{3}{2}\mu^2\sigma_{0,k}^2\right)\right)\mathrm{I_d} + \mathbf{N}_{[-1,1]}\,. \tag{130}$$

Finally, we can recover

$$x_{1,k+1} = x_{1,k} + h\lambda_1 L\left(2\mu|\sigma_{0,k}| + \frac{3}{2}\mu^2\sigma_{0,k}^2 + \frac{1}{2}\right)\xi_{k+1}. \tag{131}$$

which is the URED counterpart of (114). Thus (106), (107), and (109), (114), and (125), (127), and (128), (131) make nonlinear difference equations with state vector $(\sigma_{0,k}, x_{1,k})^\top$. We may name them *higher-order* proximal-point algorithms, since inserting, *e.g.*, (107) into (106) yields a difference equation with $\sigma_{0,k}$ and $\sigma_{0,k-1}$ 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 [33, sections 5.1, 5.8]. For instance, the AO-STD of order 3 yields $\sigma_{0,k+1} = \mathcal{J}_{h\,\mathbf{M}_h}(-b_k)$ with $\mathbf{M}_h(\sigma_{0,k+1}) := \sum_{l=0}^{2} h^l\lambda_1 L^{\frac{l+1}{4}}|\sigma_{0,k+1}|^{\frac{3-l}{4}}\,\mathbf{sgn}(\sigma_{0,k+1}) + h^3\lambda_1 L\,\mathbf{sgn}(\sigma_{0,k+1})$ for the implicit method, and $\mathbf{M}_h(\sigma_{0,k+1}) := \left(\sum_{l=0}^{2} h^l\lambda_1 L^{\frac{l+1}{4}}|\sigma_{0,k}|^{\frac{3-l}{4}} + h^3\lambda_1 L\right)\mathbf{sgn}(\sigma_{0,k+1})$ for the semi-implicit method. The remaining state variables $x_{1,k+1}$, $x_{2,k+1}$, $x_{3,k+1}$ (which are the higher-order derivatives of $f(\cdot)$) are calculated similarly as for the super-twisting and the URED, yielding other higher-order proximal-point algorithms.

Relay polynomial controllers

Consider a perturbed chain of integrators

$$\begin{aligned} \dot{x}_i(t) &= x_{i+1}(t)\,, \quad i = 1, \ldots, n-1, \\ \dot{x}_n(t) &= u(t) + \delta(t). \end{aligned} \tag{132}$$

Let $\lceil\xi\rfloor^\alpha := |\xi|^\alpha\,\mathbf{sgn}(\xi)$, $\xi \in \mathbb{R}$. The *relay polynomial controller*

$$u(x) \in -\gamma_n\,\mathbf{sgn}\left(\lceil x_n\rfloor^n + \gamma_{n-1}\lceil x_{n-1}\rfloor^{\frac{n}{2}} + \cdots + \gamma_1 x_1\right) \tag{133}$$

establishes an nth-order 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 [94], [120]. The forward Euler discretization of the open-loop system is

$$\begin{aligned} x_{i,k+1} &= x_{i,k} + hx_{i+1,k}\,, \quad i = 1, \ldots, n-1, \\ x_{n,k+1} &= x_{n,k} + hu_k + \delta_k. \end{aligned} \tag{134}$$

The backward Euler discretization of (133) takes the implicit form $u_k \in -\gamma_n\,\mathbf{M}_k(u_k)$ with

$$\mathbf{M}_k(u_k) = \mathbf{sgn}\left(\lceil x_{n,k} + hu_k\rfloor^n + \gamma_{n-1}\lceil x_{n-1,k} + hx_{n,k}\rfloor^{\frac{n}{2}} + \cdots + \gamma_1(x_{1,k} + hx_{2,k})\right). \tag{135}$$

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

$$u_k \in -\gamma_n \mathbf{M}_k(u_k) \quad \Longleftrightarrow \quad u_k \in -\gamma_n \mathbf{sgn}(u_k - \bar{u}_k(x_k)), \tag{136}$$

where

$$\bar{u}_k(x_k) = -\frac{1}{h}\left(\left\lceil \gamma_{n-1} \lceil x_{n-1,k} + h x_{n,k} \rfloor^{\frac{\alpha}{2}} + \cdots + \gamma_1 \lceil x_{1,k} + h x_{2,k} \rfloor^{\frac{\alpha}{n}} \right\rfloor^{\frac{1}{\alpha}} + x_{n,k} \right). \tag{137}$$

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

$$u_k = \min\{|\bar{u}_k(x_k)|, \gamma_n\} \mathbf{sgn}(\bar{u}_k(x_k)). \tag{138}$$

Consider now the ZOH discrete model of the chain of integrators,

$$x_{i,k+1} = x_{i,k} + \sum_{j=1}^{n-i} \frac{h^j}{j!} x_{i+j,k} + \frac{h^{n+1-i}}{(n+1-i)!}(u_k + \bar{\delta}_{i,k}), \quad i = 1, \ldots, n, \tag{139}$$

where $\bar{\delta}_i$ are (possibly unmatched) disturbances resulting from $\delta(t)$. The inclusion $u_k \in -\gamma_n \mathbf{M}'_k(u_k)$, where:

$$\mathbf{M}'_k(u_k) = \mathbf{sgn}\Bigg(\lceil x_{n,k} + h u_k \rfloor^n + \gamma_{n-1} \left\lceil x_{n-1,k} + h x_{n,k} + \frac{h^2}{2!} u_k \right\rfloor^{\frac{n}{2}} + \cdots + \gamma_1 \Big(x_{1,k} + \sum_{j=1}^{n-1} \frac{h^j}{j!} x_{1+j,k} + \frac{h^n}{n!} u_k \Big) \Bigg). \tag{140}$$

For each fixed x_k, the operator $\mathbf{M}'_k(\cdot)$ is again maximally monotone, so the resolvent $(\mathrm{I_d} + \mathbf{M}'_k)^{-1}(\cdot)$ 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 "COMPUTATIONAL ISSUES".

Stability Analysis

The complete systems (106), (107), or (108), or (125), (127), or its semi-implicit version in (128), (131), are higher-order proximal-point algorithms which do not fit with the robust proximal-point algorithm in (S41) in "Proximal-Point Algorithm and Proximal Mapping". Boundedness of the closed-loop state with (118b) is proved only in the absence of perturbations in [30]. As shown in [34], the algorithm is not robust with respect to an unbounded perturbation. A modification to (118b) is proposed in [34] that enhances robustness. Some results concerning higher-order differentiators of the implicit AO-STD type can be found in [33]. As alluded to above, the main analysis difficulty stems from the fact that the implicit discretization of (102) involves the operator

$(hC(\mathrm{I_d} - hA)^{-1}B + \mathbf{M}^{-1})^{-1}(\cdot)$. This operator does not possess good properties because CB is not symmetric positive definite (it would be if passivity of (A, B, C) held). Some properties of the implicit higher order differentiators (sliding surface invariance, accuracy, finite convergence) are shown in [32], [33]. The stability of the semi-implicit super-twisting controller is analysed in [35], see also the semi-implicit discretization [10, Equation (60)] of the super-twisting velocity observer [36].

Homogeneity and maximal monotonicity

Homogeneity is an important tool for control design [121]. It has been used in [72] to design consistent discretizations of fixed-time convergent systems. Let us consider the systems analysed in [72, Section 6.2]. The first analysis concerns the discretization of a generalized homogeneous closed-loop system, with homogeneity degree $\nu = 1$. After a suitable transformation into another homogeneous system, the following consistent discrete-time dynamics [72, Definition 2.1] is obtained:

$$y_{k+1} = y_k - h||y_k||(I_n - \tilde{A})y_{k+1} \Leftrightarrow y_{k+1} = \Big(\mathrm{I_d} + h||y_k||(I_n - \tilde{A})\Big)^{-1}(y_k), \quad k \geq 0, \tag{141}$$

with $\tilde{A}^\top P + P\tilde{A} = 0$ for some $P = P^\top \succ 0$. Let us define $R = R^\top \succ 0$, $R^2 = P$, then we obtain $R^{-1}(\tilde{A} - I_n)^\top R + R(\tilde{A} - I_n)R^{-1} = -2I_n \prec 0$. It follows that $R(I_n - \tilde{A})R^{-1} \succ 0$. Let $z_k = Ry_k$, we obtain equivalently:

$$\begin{array}{rl} & z_{k+1} = z_k - h||R^{-1}z_k||\ R(I_n - \tilde{A})R^{-1}z_{k+1} \\ \Leftrightarrow \quad & z_{k+1} = \Big(\mathrm{I_d} + h||R^{-1}z_k||\ R(I_n - \tilde{A})R^{-1}\Big)^{-1}(z_k), \quad k \geq 0. \end{array} \tag{142}$$

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

A second case with $\nu = -1$ is treated in [72, Section 6.2], which yields the consistent discretization:

$$y_{k+1} \in y_k - h(I_n - \tilde{A})\tilde{F}(y_{k+1}) \Leftrightarrow y_{k+1} = \Big(\mathrm{I_d} + h(I_n - \tilde{A})\tilde{F}(\cdot)\Big)^{-1}(y_k), \quad k \geq 0, \tag{143}$$

where $\tilde{F}(y) = \left\{ \begin{array}{ll} \frac{y}{||y||_P} & \text{if} \quad y \neq 0 \\ \mathcal{B}_P & \text{if} \quad y = 0 \end{array} \right.$, $||y||_P = \sqrt{y^\top P y}$. Proceeding as above with $z_k = Ry_k$, it is obtained:

$$\begin{array}{rl} z_{k+1} \in & z_k - hR(I_n - \tilde{A})R^{-1}R\tilde{F}(y_{k+1}) \\ & = z_k - hR(I_n - \tilde{A})R^{-1}\left\{ \begin{array}{ll} \frac{z_{k+1}}{||z_{k+1}||_2} & \text{if} \quad z_{k+1} \neq 0 \\ \mathcal{B}_2 & \text{if} \quad z_{k+1} = 0 \end{array} \right. \\ & = z_k - hR(I_n - \tilde{A})R^{-1}\partial||z_{k+1}||_2, \quad k \geq 0. \end{array} \tag{144}$$

For the same reasons as in the first case we have $\Lambda := R(I_n - \tilde{A})R^{-1} \succ 0$, however $\Lambda \neq \Lambda^\top$ in general. Now, let us proceed as in the proof of [72, Theorem 4.1] to obtain for all $z_k \neq 0$:

$$\begin{aligned} z_k^\top z_k &= z_{k+1}^\top z_{k+1} + 2h z_{k+1}^\top \Lambda \partial ||z_{k+1}||_2 + h^2 (\Lambda \partial ||z_{k+1}||_2)^\top (\Lambda \partial ||z_{k+1}||_2) \\ &= z_{k+1}^\top z_{k+1} + 2h \frac{z_{k+1}^\top \Lambda z_{k+1}}{||z_{k+1}||_2} + h^2 (\Lambda \partial ||z_{k+1}||_2)^\top \Lambda \partial ||z_{k+1}||_2 \\ &> z_{k+1}^\top z_{k+1} + h^2 (\Lambda \partial ||z_{k+1}||_2)^\top \Lambda \partial ||z_{k+1}||_2 \\ &= z_{k+1}^\top z_{k+1} + h^2 \frac{z_{k+1}^\top \Lambda^\top \Lambda z_{k+1}}{||z_{k+1}||_2^2}. \end{aligned} \tag{145}$$

Convergence in a finite number of steps is inferred from the fact that $\inf_{z_{k+1} \in \mathbb{R}^n \setminus \{0\}} h^2 \frac{z_{k+1}^\top \Lambda^\top \Lambda z_{k+1}}{||z_{k+1}||_2^2} > 0$ for all $h > 0$, since $\Lambda^\top \Lambda \succ 0$. Thus, in spite of the fact that neither the operator inside (143) nor inside $z_{k+1} = (\mathrm{I_d} + h\Lambda\, \partial ||\cdot||_2)^{-1}(z_k)$ are maximal monotone in general, finite-time convergence of this algorithm is guaranteed. Moreover the calculation is made in [72, Section 6.2]. As a side note, mark that (144) is equivalent to:

$$z_{k+1} \in z_k - h(\Lambda_{sym} + \Lambda_{sk})\, \partial ||z_{k+1}||_2, \tag{146}$$

with $\Lambda_{sym} := \frac{1}{2}(\Lambda + \Lambda^\top)$, $\Lambda_{sk} := \frac{1}{2}(\Lambda - \Lambda^\top)$. Therefore (144) is interpreted as a discrete-time Hamiltonian system $z_{k+1} \in z_k + h(J - R)\, \partial H(z_{k+1})$, with $J = -\Lambda_{sk}$, $R = \Lambda_{sym} \succ 0$, Hamiltonian function $H(z) = ||z||_2$. This is the implicit discretization of the Hamiltonian dynamics: $\dot{z} \in (J - R)\partial ||z||_2$. We may introduce *Hamiltonian proximal-point* algorithms with resolvents $\mathcal{J}_{h\,\mathbf{M}} := (\mathrm{I_d} - h(J - R)\partial ||\cdot||)^{-1}$, $\mathbf{M}(\cdot) = -(J - R)\partial ||\cdot||$.

FURTHER IMPLICIT ALGORITHMS

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 [67], [68], twisting controller [64], proxy-based sliding-mode [42], [60], [90], homogeneous differentiators and observers [61], [62], super-twisting velocity observer [10], other types of differentiators [42], [43], [78], [82], [86], [88], the differentiator proposed in [41] discretized in [33], sliding-mode control of nonsmooth actuators [58], homogeneous systems and controllers [72], [79], nested controllers for second-order systems with unmatched perturbations [57], first-order controllers applied to Lagrangian nonlinear systems with parameter uncertainties [55] and to linear hyperbolic infinite dimensional systems [117], fixed-time or prescribed-time schemes [68], [71]–[73], adaptive sliding-mode algorithms [75], [83], [87], sliding-mode control with reaching law [76], fractional-order differentiators [80], [84], multivariable super-twisting [85].

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 analyse 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, necessary for the implementation.

Limitations and Modifications of Implicit Algorithms

The implicit method has been studied in [64] for the twisting scheme and in [30] for the super-twisting. However, the straightforward implicit discretization of the twisting scheme (*i.e.*, the implicit emulation) does not guarantee finite-time convergence [64], [69]. As noticed in [34], the implicit emulation of the super-twisting scheme (118a), (118b) does not yield acceptable robustness (see also [81] for the influence of the plant discretization). Also, implicit emulation discretization of some higher-order differentiators may still undergo numerical chattering, and suitable modifications have to be made to avoid it [77]. 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 [72].

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). 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 [31], [66], [96], [97].

Solving Generalized Equations with Proximal Operators

The computation of the variable of insterest (control input or differentiator output) is reduced to solving either one of the following generalized equations:

$$i)\begin{cases}\zeta_{k+1}=\xi(x_k)+\rho\eta(x_k)\\ -\eta(x_k)\in\mathbf{M}(\zeta_{k+1})\end{cases}\qquad ii)\begin{cases}-\eta(x_k)=\frac{1}{\rho}\xi(x_k)-\frac{1}{\rho}\zeta_{k+1}\\ \zeta_{k+1}\in\mathbf{M}^{-1}(-\eta(x_k))\end{cases}, \tag{147}$$

where $\xi:\mathbb{R}^n\to\mathbb{R}^m$ is a known function of the state, $\rho>0$, and $\mathbf{M}:\mathbb{R}^m\rightrightarrows\mathbb{R}^m$ is maximal monotone. Equation (67) has this form with $\zeta_{k+1}=\tilde{s}'_{k+1}$, $\eta(x_k)=u_{\rm sv}(x_k)$, $\mathbf{M}=\partial\left\|\cdot\right\|_1\circ$

$\tilde{K}(\tilde{B}^\top \tilde{P}^{-1}\tilde{B})^{\frac{1}{2}}$, $\xi(x_k) = \tilde{L}x_k$. Likewise, Equation (91) with $\zeta_{k+1} = \tilde{s}'_{k+1}$, $\eta(x_k) = u_{\text{sv}}(x_k)$, $\mathbf{M} = \partial \left\|\cdot\right\|_1 \circ \tilde{K}^{\frac{1}{2}}$, $\xi(x_k) = \tilde{K}^{-\frac{1}{2}} s_k$, and Equations (117)–(118c) with $\zeta_{k+1} = \tilde{x}_{0,k+1}$, $\eta(x_k) = u_k(x_{0,k}, \nu_k)$, $\mathbf{M}(\cdot)$ as in (105), $\xi(x_k) = x_{0,k} + h\nu_k$ belong to this class. Also, Equations (125)–(126) are an instance of (147) with $\zeta_{k+1} = \sigma_{0,k+1}$, $\mathbf{M}(\cdot) := \mathbf{M}_{\text{sing}}(\cdot) + \mathbf{M}_{\text{set},h}(\cdot)$, $\xi(x_k) = -b_k$. The unknown ζ_{k+1} for the generalized equation in (147-i) 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 (147-ii) has the unknown $\eta(x_k)$ and can be interpreted as the dual, or reciprocal of (147-i). From (147-i) the *selection* $\eta(x_k)$ is explicitly given as

$$-\eta(x_k) = \frac{1}{\rho}\left(\mathrm{I_d} - \mathcal{J}_{\rho\mathbf{M}}\right)(\xi(x_k)) = \mathcal{Y}_{\rho\mathbf{M}}(\xi(x_k)) \tag{148}$$

(see (S37) in "Proximal-Point Algorithm and Proximal Mapping"), whereas (147-ii) gives

$$-\eta(x_k) = \mathcal{J}_{\frac{1}{\rho}\mathbf{M}^{-1}}\left(\frac{1}{\rho}\xi(x_k)\right). \tag{149}$$

The expressions (148)–(149) yield the same control law. The main difference relies on the proximal map to be computed. That is, for a certain $\mathbf{M}(\cdot)$, it could be easier to compute first the inverse map $\mathbf{M}^{-1}(\cdot)$ and then (149), while for other cases (148) might be a simpler choice. In any case, a convex optimization problem has to be solved at each time-step.

For $\mathbf{M}(\cdot) = \partial\left\|\cdot\right\|_p$, the calculation of the control inputs can be performed according to (S27)–(S29) in "Convex Analysis Tools", so that

$$-\eta(x_k) = \mathrm{Proj}\left(\mathcal{B}_q; \frac{1}{\rho}\xi(x_k)\right). \tag{150}$$

When $p = 1$ then $q = \infty$ and, according to (S32) the control can be computed component-wise as

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

whereas if $p = q = 2$ then, according to (S31), it is inferred that

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

For sliding-mode control, it is also common to find maps $f(\cdot)$ 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 $\mathbf{M}(\cdot) = \partial(\left\|\cdot\right\|_p \circ R)(\cdot)$, where R is non-singular, then it follows

from the definition of conjugate function (see Definition 9 in "Convex Analysis Tools") that $\mathbf{M}^{-1}(\cdot) = \partial(\Psi_{\mathcal{B}_q} \circ R^{-\top})(\cdot)$ and the proximal map in (149) becomes

$$\begin{aligned}
\mathcal{J}_{\frac{1}{\rho}\mathbf{M}^{-1}}\left(\frac{1}{\rho}\xi(x_k)\right) &= \mathrm{Prox}_{\Psi_{\mathcal{B}_q}\circ R^{-\top}}\left(\frac{1}{\rho}\xi(x_k)\right) \\
&= \underset{w\in\mathbb{R}^m}{\arg\min}\left\{\Psi_{\mathcal{B}_q}\left(R^{-\top}w\right) + \frac{1}{2}\left\|w - \frac{1}{\rho}\xi(x_k)\right\|_2^2\right\} \\
&= R^\top\left(\underset{\tilde{w}\in\mathbb{R}^m}{\arg\min}\left\{\Psi_{\mathcal{B}_q}\left(\tilde{w}\right) + \frac{1}{2}\left\|R^\top\left(\tilde{w} - \frac{1}{\rho}R^{-\top}\xi(x_k)\right)\right\|_2^2\right\}\right) \\
&= R^\top \mathrm{Proj}_{RR^\top}\left(\mathcal{B}_q; \frac{1}{\rho}R^{-\top}\xi(x_k)\right). \qquad (151)
\end{aligned}$$

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

$$\mathcal{J}_{\frac{1}{\rho}\mathbf{M}^{-1}}\left(\frac{1}{\rho}\xi(x_k)\right) = R^\top\eta, \qquad (152)$$

where

$$\eta = \underset{w}{\arg\min}\left\{g^\star(w) + \frac{\rho}{2}\left(w - \frac{1}{\rho}R^{-\top}\xi(x_k)\right)RR^\top\left(w - \frac{1}{\rho}R^{-\top}\xi(x_k)\right)\right\}. \qquad (153)$$

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

$$\mathbf{M}(\zeta) = \left(\sum_{i=0}^{N} a_i|\zeta|^{\frac{p_i}{q_i}}\right)\mathbf{sgn}(\zeta) + b_0, \qquad (154)$$

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* $\mathcal{J}_{\rho\mathbf{M}}(\cdot)$ *has the form*

$$\mathcal{J}_{\rho\mathbf{M}}(v) = \beta_{\mathbf{M}}(v, b_0, \rho)^C\,\mathbf{sgn}(v - \rho b_0), \qquad (155)$$

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*

$$\beta^C + \rho\sum_{i=1}^{N} a_i\beta^{r_i} - \max\{0, |v - \rho b_0| - \rho a_0\} \qquad (156)$$

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

Proof. From the definition of resolvent in (S11) it follows that

$$\begin{aligned} w = \mathcal{J}_{\rho\mathbf{M}}(v) &\Leftrightarrow v \in w + \rho\,\mathbf{M}(w) \\ &\Leftrightarrow v - \rho b_0 \in \left(|w| + \rho a_0 + \rho \sum_{i=1}^{N} a_i |w|^{\frac{p_i}{q_i}} \right) \mathbf{sgn}(w). \end{aligned} \tag{157}$$

Since all the coefficients a_i are nonnegative, it follows from (157) that $w = 0$ if, and only if, $|v - \rho b_0| \leq \rho a_0$. Hence, if $|v - \rho b_0| > \rho a_0$, then $w \neq 0$ and from (157) it follows that $\mathbf{sgn}(v - \rho b_0) = \mathbf{sgn}(w)$, so that (157) becomes single-valued as

$$|w| + \rho \sum_{i=1}^{N} a_i |w|^{\frac{p_i}{q_i}} - (|v - \rho b_0| - \rho a_0) = 0. \tag{158}$$

Let $C > 0$ be the least common denominator of the set $\left\{ \frac{p_i}{q_i} \mid i = 1, \dots N \right\}$. The change of variables $|w| = \beta^C$, where $\beta > 0$, transforms (158) into a root-finding problem for the polynomial

$$\beta^C + \rho \sum_{i=1}^{N} a_i \beta^{r_i} - (|v - \rho b_0| - \rho a_0), \tag{159}$$

where $r_i = C\frac{p_i}{q_i}$. Hence, in order to solve (158) we look for the positive root of (159). Note that, the non-negativity of all a_i, $i \in \{1, \dots, N\}$ together with Descartes' rule of signs [113, Theorem 2.23] imply the uniqueness of such positive root of (159) in the cases when $|v - \rho b_0| > \rho a_0$. It is clear that, the polynomial (156) includes both cases above. Indeed, if $|v - \rho b_0| \leq \rho a_0$, then the unique nonnegative root of (156) is $\beta = 0$, implying $w = 0$. Finally, it follows that

$$\mathcal{J}_{\rho\mathbf{M}}(v) = w = |w|\,\mathbf{sgn}(w) = \beta_{\mathbf{M}}(v, b_0, \rho)^C\,\mathbf{sgn}(v - \rho b_0). \tag{160}$$

The proof is complete. □

Instances of Proposition 3 have been independently developed in [33] and [31] in the context of higher-order differentiators, since (155) is the unique solution of the generalized equation

$$0 \in \zeta + \sum_{i=0}^{N} a_i |\zeta|^{\frac{p_i}{q_i}}\,\mathbf{sgn}(\zeta) + b_0 - v. \tag{161}$$

Splitting algorithms

For more complex strategies such as higher-order sliding-mode control and differentiation, the associated function $f(\cdot)$ is given as the addition of several simpler functions. In this situation, splitting algorithms [S14] can be used to compute the associated proximal map in an iterative way. Consider the following problem. Given two maximal monotone operators $\mathbf{M}_1, \mathbf{M}_2 : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, find $\zeta \in \mathbb{R}^n$ such that

$$0 \in \mathbf{M}_1(\zeta) + \mathbf{M}_2(\zeta). \tag{162}$$

Under constraint-qualification assumptions guaranteeing that $\mathbf{M}_1 + \mathbf{M}_2$ is maximal monotone, it follows from (S11) in "Maximal Monotone Operators" that problem (162) is equivalent to finding $\zeta \in \mathbb{R}^n$ such that

$$\zeta = \mathcal{J}_{\rho(\mathbf{M}_1 + \mathbf{M}_2)}(\zeta) \tag{163}$$

for any $\rho > 0$. The formulation (163) allows for a search of a solution to (162) in an iterative way. Indeed, it follows from the contents of "Proximal-Point Algorithm and Proximal Mapping" that the fixed-point iteration

$$\zeta_{k+1} = \mathcal{J}_{\rho(\mathbf{M}_1 + \mathbf{M}_2)}(\zeta_k) \tag{164}$$

converges towards a solution of (163). However, an explicit expression for the resolvent map of the sum $\mathbf{M}_1 + \mathbf{M}_2$ in (164) may not be available. To solve such issue, splitting schemes allow us to compute alternative iterations that also converge to a solution of (162) and that only involve the individual resolvents of $\mathbf{M}_1(\cdot)$ and $\mathbf{M}_2(\cdot)$ at each iteration. In what follows we discuss two popular splitting schemes: the Condat-Vũ and the Douglas-Rachford splittings.

Condat-Vũ splitting

The Condat-Vũ splitting was independently proposed in [108] and [109] for solving the optimization problem

$$\min_{\zeta \in \mathbb{R}^n} f(\zeta) + g(A\zeta) + h(\zeta), \tag{165}$$

where $f(\cdot)$, $g(\cdot)$, and $h(\cdot)$ are proper, convex, lsc functions. The function $h(\cdot)$ 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 (S38a) and (S38c) in "Proximal-Point Algorithm and Proximal Mapping"), 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 (118a) and (125)). The splitting relies on the fact that $g(A\zeta) = \sup_{\mu \in \mathbb{R}^m} \{\langle Ax, \mu \rangle - g^\star(\mu)\}$ (Fenchel-Moreau theorem [S13, Theorem 13.32]) so that, as pointed out in [108], the primal problem (165) can be rewritten as

$$\min_{\zeta \in \mathbb{R}^n} \sup_{\mu \in \mathbb{R}^m} \mathcal{L}(\zeta, \mu), \tag{166}$$

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

$$\mathcal{L}(\zeta, \mu) = f(\zeta) + h(\zeta) + \langle \mu, A\zeta \rangle - g^\star(\mu). \tag{167}$$

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

$$0 \in \nabla h(\zeta) + A^\top \mu + \partial f(\zeta), \tag{168a}$$

$$0 \in -A\zeta + \partial g^\star(\mu). \tag{168b}$$

Proceeding in a similar way as above, (168) is equivalent to the following fixed-point conditions:

$$\zeta = \mathrm{Prox}_{\alpha_1 f}\left(\zeta - \alpha_1(\nabla h(\zeta) + A^\top \mu)\right), \tag{169a}$$

$$\mu = \mathrm{Prox}_{\alpha_2 g^\star}(\mu + \alpha_2 A\zeta), \tag{169b}$$

for any $\alpha_1, \alpha_2 > 0$. It is well-known that, if $\zeta^\star$ and $\mu^\star$ are solutions of the primal problem (165) and its associated dual problem, respectively, then they also satisfy (169), see *e.g.*, [S13, 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 [108], [109]:

$$\zeta_{j+1} = \mathrm{Prox}_{\alpha_1 f}\left(\zeta_j - \alpha_1(\nabla h(\zeta_j) + A^\top \mu_j)\right), \tag{170a}$$

$$\mu_{j+1} = \mathrm{Prox}_{\alpha_2 g^\star}\left(\mu_j + \alpha_2 A(2\zeta_{j+1} - \zeta_j)\right). \tag{170b}$$

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

$$\frac{\alpha_1 \ell}{2} + \alpha_1 \alpha_2 \lambda_{\max}(A^\top A) < 1, \tag{171}$$

then the iteration (170) converges towards solutions of (169), implying that the sequence $\{\zeta_k\}_{k\in\mathbb{N}}$ converges to a solution of (165).

Example 1. *The splitting* (170) *can be used to compute the URED proximal map in* (125), (126) *as follows. Consider the optimization problem* (165) *with* $A = 1$ *and*

$$f(\zeta) = \frac{2}{3} h\lambda_0 L^{\frac{1}{2}} |\zeta|^{\frac{3}{2}} + \frac{1}{2} h^2 \lambda_1 L |\zeta|, \tag{172a}$$

$$g(\zeta) = \frac{1}{2} h^2 \mu^2 \lambda_1 L |\zeta|^3, \tag{172b}$$

$$h(\zeta) = \frac{2}{5} h\lambda_0 L^{\frac{1}{2}} \mu |\zeta|^{\frac{5}{2}} + h^2 \lambda_1 L \mu \zeta^2 + \frac{1}{2}(\zeta + b_k)^2. \tag{172c}$$

It is clear that, with the given parameters, the solution of (165) *indeed gives the proximal map* $\mathcal{J}_{h\,\mathbf{M}_h}(-b_k)$*, where* $\mathbf{M}_h(\cdot)$ *is as in* (126)*. Also, note that* $\nabla h(\cdot)$ *is Lipschitz continuous in any bounded set of* $\mathbb{R}$*, since*

$$\frac{\partial^2}{\partial \zeta^2} h(\zeta) = \frac{3}{2} h\lambda_0 L^{\frac{1}{2}} \mu |\zeta|^{\frac{1}{2}} + (1 + 2h^2 \lambda_1 L \mu). \tag{173}$$

Notice that, with such decomposition, each map appearing in (170) *is proximable. Indeed, it follows from Proposition 3 that*

$$\mathrm{Prox}_{\alpha_1 f}(v) = \mathcal{J}_{\alpha_1 \partial f}(v) = \beta_{\partial f}(v, \alpha_1)^2\, \mathbf{sgn}(v), \tag{174}$$

where

$$\beta_{\partial f}(v,\alpha_1)=\frac{-\alpha_1 h\lambda_0 L^{\frac{1}{2}}+\sqrt{\left(\alpha_1 h\lambda_0 L^{\frac{1}{2}}\right)^2+4\max\left\{0,|v|-\frac{\alpha_1 h^2\lambda_1 L}{2}\right\}}}{2}. \tag{175}$$

Now, in order to compute the proximal map associated with $g^\star(\cdot)$ *in* (170)*, we make use of Moreau's decomposition (see, e.g., [S13, Theorem 14.3]),*

$$v=\operatorname{Prox}_{\alpha_2 g^\star}(v)+\alpha_2\operatorname{Prox}_{\frac{1}{\alpha_2}g}\left(\frac{v}{\alpha_2}\right). \tag{176}$$

Thus, we compute $\operatorname{Prox}_{\frac{1}{\alpha_2}g}\left(\frac{v}{\alpha_2}\right)$ *using once again Proposition 3, so that*

$$\operatorname{Prox}_{\frac{1}{\alpha_2}g}\left(\frac{v}{\alpha_2}\right)=\beta_{\partial g}\left(\frac{v}{\alpha_2},\frac{1}{\alpha_2}\right)\mathbf{sgn}(v), \tag{177}$$

where

$$\beta_{\partial g}(\tilde{v},\rho)=\frac{-1+\sqrt{1+6h^2\mu^2\lambda_1 L\rho|\tilde{v}|}}{3h^2\mu^2\lambda_1 L\rho}. \tag{178}$$

Finally, it follows from (176) *and* (177) *that*

$$\operatorname{Prox}_{\alpha_2 g^\star}(v)=v-\alpha_2\operatorname{Prox}_{\frac{1}{\alpha_2}g}\left(\frac{v}{\alpha_2}\right)=v-\alpha_2\beta_{\partial g}(v,\rho)\,\mathbf{sgn}(v). \tag{179}$$

Therefore, by using the expressions (174) *and* (179) *in the iteration* (170) *and by selecting the parameters* α_1 *and* α_2 *sufficiently small such that* (171) *holds, we have that* $\zeta_{j+1}\to\mathcal{J}_{h\,\mathbf{M}_h}(-b_k)$ *as* $j\uparrow+\infty$.

It is worth mentioning that, for problems with more than two maximal monotone terms, parallel algorithms as those described in [110], [S15] can be used.

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. We start considering the problem of finding $\zeta\in\mathbb{R}^n$ such that

$$0\in\partial f(\zeta)+\partial g(\zeta), \tag{180}$$

where, as before, $f(\cdot)$ and $g(\cdot)$ are proper, convex, lsc functions. Thus, as pointed out in [S15, p. 44] the following equivalence is true:

$$0\in\partial f(\zeta)+\partial g(\zeta)\Leftrightarrow z=(\mathcal{R}_{\alpha\partial f}\circ\mathcal{R}_{\alpha\partial g})(z),\ \text{and}\ \zeta=\operatorname{Prox}_{\alpha g}(z), \tag{181}$$

where $\mathcal{R}_{\alpha\partial f}(\cdot) = (2\,\mathrm{Prox}_{\alpha f} - \mathrm{I_d})(\cdot)$ 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 (181) may not converge. To overcome such issue, the Douglas-Rachford splitting uses the following Krasnosel'skiĭ-Mann iteration

$$z_{j+1} = \frac{1}{2}z_j + \frac{1}{2}(\mathcal{R}_{\alpha\partial f} \circ \mathcal{R}_{\alpha\partial g})(z_j) = \mathrm{Prox}_{\alpha f} \circ (2\,\mathrm{Prox}_{\alpha g} - \mathrm{I_d})(z_j) + (\mathrm{I_d} - \mathrm{Prox}_{\alpha g})(z_j),$$

for which its convergence is guaranteed for any $\alpha > 0$ [52, Theorem 1]. Thus, the iteration

$$z_{j+1} = \mathrm{Prox}_{\alpha f} \circ (2\,\mathrm{Prox}_{\alpha g} - \mathrm{I_d})(z_j) + (\mathrm{I_d} - \mathrm{Prox}_{\alpha g})(z_j) \tag{182a}$$

$$\zeta_{j+1} = \mathrm{Prox}_{\alpha g}(z_j) \tag{182b}$$

converges and ζ_{j+1} approaches a solution of (180) as $j \uparrow +\infty$.

Example 2. *The Douglas-Rachford splitting can also be used to compute the proximal map in* (125)*,* (126)*. This time we consider a splitting of the following form:*

$$f(\zeta) = \frac{2}{5}h\lambda_0 L^{\frac{1}{2}}\mu|\zeta|^{\frac{5}{2}} + \frac{2}{3}h\lambda_0 L^{\frac{1}{2}}|\zeta|^{\frac{3}{2}} + \frac{1}{2}h^2\lambda_1 L|\zeta|, \tag{183}$$

$$g(\zeta) = \frac{1}{2}h^2\mu^2\lambda_1 L|\zeta|^3 + h^2\lambda_1 L\mu\zeta^2 + \frac{1}{2}(\zeta + b_k)^2. \tag{184}$$

From Proposition 3 we obtain the expressions

$$\mathrm{Prox}_{\alpha f}(v) = \beta_{\partial f}(v,\alpha)^2\,\mathbf{sgn}(v), \tag{185}$$

$$\mathrm{Prox}_{\alpha g}(v) = \beta_{\partial g}(v, b_k, \alpha)\,\mathbf{sgn}(v - \alpha b_k), \tag{186}$$

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

$$\alpha h\lambda_0 L^{\frac{1}{2}}\mu\beta^3 + \beta^2 + \alpha\lambda_0 L^{\frac{1}{2}}\beta - \max\left\{0, |v| - \frac{\alpha h^2\lambda_1 L}{2}\right\}, \tag{187}$$

that is,

$$\beta_{\partial g}(v, b_k, \alpha) = \frac{-2(1 + h^2\mu\lambda_1 L) + \sqrt{4(1 + h^2\mu\lambda_1 L)^2 + 6h^2\mu^2\lambda_1 L|v - b_k|}}{3h^2\mu^2\lambda_1 L}. \tag{188}$$

Hence, by taking the Douglas-Rachford iteration (182) *with* $\mathrm{Prox}_{\alpha f}$ *and* $\mathrm{Prox}_{\alpha g}$ *as in* (185)–(188)*, we have that ζ_j converges to the resolvent* (125)–(126) *as $j \uparrow +\infty$.*

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

$$s_{k+1} = s_k + \rho\, u_{\mathrm{sv}}(x_k) + \delta_k. \tag{189}$$

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 (147) One way of accelerating the convergence of $\tilde{s}_k$ towards the origin consists in setting the control law as

$$-u_{\mathrm{sv}}(x_k) \in \partial f(\tilde{s}_{k+1}) \tag{190}$$

$$\tilde{s}_{k+1} = s_k + \mu\, u_{\mathrm{sv}}(x_k), \tag{191}$$

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

$$\tilde{s}_{k+1} = \mathrm{Prox}_{\mu f}(s_k) \tag{192a}$$

$$s_{k+1} = s_k + \frac{\rho}{\mu}(\tilde{s}_{k+1} - s_k) + \delta_k. \tag{192b}$$

The iteration (192) is known as a relaxed iteration in the literature of proximal-point algorithms [S14]. In the cases when $\delta_k \equiv 0$, it has been shown that (192) converges towards a minimum of $f(\cdot)$, [S13, Corollary 5.16]. Moreover, for certain values of $\frac{\rho}{\mu} \in (0, 2)$, the algorithm (192) has a faster convergence with respect to the case $\rho = \mu$, see, *e.g.*, [S14].

Numerical Experiments

Taking advantage of the fact that the super-twisting controller can be calculated in the closed form (118b), 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 S5. 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 [30] and $iii)$ the semi-implicit discretization in [35], are considered. In each case the control law takes the form

$$i) \begin{cases} u_k = -\gamma_0 L^{\frac{1}{2}} |x_{0,k}|\, \mathbf{sgn}(x_{0,k}) + \nu_k \\ \nu_{k+1} = \nu_k - h\gamma_1 L\, \mathbf{sgn}(x_{0,k}) \end{cases}, \tag{193}$$

$$ii) \begin{cases} u_k = -\gamma_0 L^{\frac{1}{2}} \beta_k\, \mathbf{sgn}(x_{0,k} + h\nu_k) - h\gamma_1 L\, \mathrm{Proj}\left([-1,1]; \frac{x_{0,k}+h\nu_k}{h^2\gamma_1 L}\right) + \nu_k \\ \beta_k = \frac{-h\gamma_0 L^{\frac{1}{2}}}{2} + \sqrt{\frac{h^2\gamma_0^2 L}{4} + \max\{0, |x_{0,k} + h\nu_k| - h^2\gamma_1 L\}} \\ \nu_{k+1} = \nu_k - h\gamma_1 L\, \mathrm{Proj}\left([-1,1]; \frac{x_{0,k}+h\nu_k}{h^2\gamma_1 L}\right) \end{cases}, \tag{194}$$

and

$$iii)\begin{cases} u_k = -(\gamma_0 L^{\frac{1}{2}}|x_{0,k}|^{\frac{1}{2}} + h\gamma_1 L)\,\mathrm{Proj}\left([-1,1];\frac{x_{0,k}+h\nu_k}{h(\gamma_0 L^{\frac{1}{2}}|x_{0,k}|^{\frac{1}{2}}+h\gamma_1 L)}\right)+\nu_k \\ \nu_{k+1} = \nu_k - h\gamma_1\,\mathrm{Proj}\left([-1,1];\frac{x_{0,k}+h\nu_k}{h(\gamma_0 L^{\frac{1}{2}}|x_{0,k}|^{\frac{1}{2}}+h\gamma_1 L)}\right)\end{cases}. \tag{195}$$

For the second approach, which relies on the direct design in discrete-time, we consider the following controller, recently proposed in [34]

$$\begin{aligned} u_k &= -\gamma_0\beta_k\,\mathbf{sgn}(x_{1,k}) - 2h\gamma_1\,\mathrm{Proj}\left([-1,1];\frac{x_{1,k}}{h^2\gamma_1}\right)+\nu_k \\ \beta_k &= -\frac{h\gamma_0}{2}+\frac{1}{2}\sqrt{(h\gamma_0)^2+4\max\{0,|x_{1,k}|-h^2\gamma_1\}} \\ \nu_{k+1} &= \nu_k - h\gamma_1\,\mathrm{Proj}\left([-1,1];\frac{x_{1,k}}{h^2\gamma_1}\right)\end{aligned}. \tag{196}$$

The controller (196) is in explicit form, though strongly motivated from the implicit controller obtained from the backward Euler discretization (194). In addition, the splitting schemes (170) and (182) are considered, applied to the implicit super-twisting algorithm in (118). That is, for solving the minimization problem

$$\min_{w\in\mathbb{R}} \frac{2}{3}h\gamma_0 L^{\frac{1}{2}}|w|^{\frac{3}{2}} + h^2\gamma_1 L|w| + \frac{1}{2}\left(w-(x_{0,k}+h\nu_k)\right)^2. \tag{197}$$

As a last scheme to compare, the problem (197) is solved numerically at each time step by using the interior point solver *Clarabel* [114]. 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 S6 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 $\{\zeta_k\}$ of N samples as

$$\mathrm{SVar}(\{\zeta_k\}) = \sum_{k=0}^{N-2}|\zeta_{k+1}-\zeta_k|. \tag{198}$$

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 (196) and (193) are less performing. Indeed, for the explicit controller (193), $\mathrm{SVar}_{[0,20]}(\{u_k\}) \approx 500$ 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 (193) 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

(193)). 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 (194)

```
import numpy as np

def J_A0(alpha, v, h_c, g0):
    beta_A = (-alpha*h_c*g0 + np.sqrt((alpha*h_c*g0)**2 + 4.0*np.abs(v)))/2.0
    return (beta_A**2)*np.sign(v)

def J_A1(v, h_c, g1):
    return 0 if np.abs(v)<=(h_c**2)*g1 else v-(h_c**2)*g1*np.sign(v)

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*np.sign(x_0_next_tilde)*np.abs(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 (194)

```
import numpy as np

def J_B0(v, h_c, g0):
    beta = (-h_c*g0 + np.sqrt((h_c*g0)**2 + 4.0*np.abs(v)))/2.0
    return (beta**2)*np.sign(v)

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

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*np.sign(x_0_next_tilde)*np.abs(x_0_next_tilde)**0.5
    x_0_tilde.append(x_0_next_tilde)
    nu.append(nu_next)
    mu.append(m)
    return u_k
```

EXPERIMENTAL RESULTS AND VALIDATIONS

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 (94), (85), (119)), moderate deterioration of the performance with increasing sampling times, significant chattering alleviation at both outputs and inputs, are validated experimentally. The classical first-order sliding-mode controller [26], [27], [70], the twisting controller [70], several higher order and homogeneous differentiators have been implemented on an electropneumatic system [61], [96], [97] and compared. Homogeneous differentiators have been tested on an RLC circuit [63], differentiators have been implemented on DC-DC buck converters in [99]. Various discretizations of the classical first-order sliding-mode control [11], a switching input upgrading linear control [95], several higher-order differentiators [96]–[98] 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 [88], [89], [100], first order differentiators [42], 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 [101]–[103]. A parabolic differentiator is implemented on a direct current servomotor equipped with an optical encoder in [86]. A set-valued anti-sway controller is applied on an overhead crane in [104]. A super-twisting observer is applied to a magneto-rheological clutch in [91], [112], while a super-twisting controller semi-implicitly discretized is applied to an electromechanical system with time-delay in [111]. A first-order sliding-mode set-valued input is designed for the stabilization of a nonsmooth model of an hydraulic actuator in [58] and applied to an industrial commercial excavator [105]–[107].

Typical experimental results, presented in [70], are shown in [122] and in Figure S7. It is visible in this figure that the explicit controller behaves like a high-frequency bang-bang controller, while the implicit one has much smaller magnitude and reproduces the disturbance. More experimental

results showing gain insensitivity, sampling time influence, and comparisons between explicit, implicit and saturation controllers, are available in [26], [69].

CONCLUSIONS AND PERSPECTIVES

The implicit discretization of set-valued sliding mode systems has received significant attention during the last decade, since it was shown in [10] 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. A robust, a time-varying, a Hamiltonian and a higher-order versions of the proximal-point algorithm are introduced. Computational issues, which are crucial for real-time implementations, are analysed. 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 complexity (which can stem from the sliding surface codimension, 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 sliding-mode systems design). Analysing 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.

Sidebar:Article Summary

Sidebar:Set-Valued Mappings and Differential Inclusions

The material below is taken from [S1], [S5], [S11]. 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

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

The inverse mapping (set-valued or single-valued) of $\mathbf{F}(\cdot)$ is defined as $\mathbf{F}^{-1} : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ with $\mathbf{F}^{-1}(y) = \{x \in \mathbb{R}^n \mid y \in \mathbf{F}(x)\}$. 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 is given by

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

and its *graph* by

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

Semicontinuity properties (inner, upper, outer) are fundamental for set-valued maps analysis.

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 $\mathbf{F}(\cdot)$ is osc at x, then $\mathbf{F}(x)$ is a closed set (hence, an osc set-valued map $\mathbf{F}$ is closed valued). Moreover, $\mathbf{F}(\cdot)$ is osc if and only if $\operatorname{gph} \mathbf{F}$ is closed.*

Theorem 2. *[S1, 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*

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

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

Definition 2. *[S2, 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.

In the time-varying case with $F(x, t)$ 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.

We are particularly interested in differential inclusions of the form

$$\dot{s}(t) \in -\partial f(s(t)) + \delta(t), \tag{S4}$$

where $\partial f(\cdot)$ is the subdifferential of $f(\cdot)$ in the sense of convex analysis (see "Convex Analysis Tools"). Indeed they appear in (20) and (34).

Proposition 4. *Consider the perturbed subgradient system* (S4) *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*

$$\delta(t) + \varepsilon \mathcal{B}_2 \subseteq \partial f(0) \tag{S5}$$

for all $t \geq 0$ *(see* (S26) *in "Convex Analysis Tools"). The origin is finite-time stable.*

Note that, for $\delta(t) \not\equiv 0$, the condition (S5) implies that $\partial f(\cdot)$ is not a singleton at $s = 0$ and that $\delta(\cdot)$ is uniformly bounded.

Proof. The differential inclusion (S4) belongs to the class of differential inclusions of the form $\dot{s}(t) \in -\mathbf{M}(s(t), t)$, where the operator $\mathbf{M}(s, t)$ is maximal monotone for each t. Under mild conditions on $\delta(t)$, the differential inclusion possesses unique absolutely continuous solutions on $\mathbb{R}_+$, for any initial condition [5, Theorem 3.4].

Consider the Lyapunov function candidate $V(s) = \frac{1}{2}\|s\|^2$. The set-valued derivative [S3] of V along trajectories of (S4) 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 $f(\cdot)$ is proper, convex and lsc, we have

$$s^\top \vartheta = f(s) + f^\star(\vartheta), \tag{S6}$$

where $f^\star(\cdot)$ denotes the conjugate function to $f(\cdot)$ (see Definition 9 and Theorem 5 in "Convex Analysis Tools"). It follows from (S5) that $\delta(t) + \varepsilon\rho \in \partial f(0)$ for any $\rho \in \mathcal{B}_2$. By the definition of subdifferential (see Definition 5 in "Convex Analysis Tools"),

$$\delta^\top s \leq -\varepsilon \rho^\top s + f(s) - f(0) \tag{S7}$$

for any $s \in \mathbb{R}^n$ and for any $\rho \in \mathcal{B}_2$. Since $\partial f(s)$ is compact for any $s \in \mathbb{R}^m$ [S10, Theorem 23.4], $\max \dot{\tilde{V}}(s)$ is well defined. Combining (S6) and (S7), it follows that

$$\max \dot{\tilde{V}}(s) \leq -\varepsilon \rho^\top s - f^\star(\vartheta^\star) - f(0) \tag{S8}$$

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

$$\max \dot{\tilde{V}}(s) \leq -\varepsilon \|s\| = -\varepsilon \sqrt{2V(s)}. \tag{S9}$$

For $s = 0$ the inequality follows trivially.

□

References

[S1] K. Deimling, Multivalued Differential Equations, De Gruyter Series in Nonlinear Analysis and Applications, Berlin, New York, 1992.

[S2] G.V. Smirnov, Introduction to the Theory of Differential Inclusions, Graduate Studies in Mathematics vol. 41, American Mathematical Society, 2002.

[S3] A. Bacciotti and F. Ceragioli, Stability and stabilization of discontinuous systems and non-smooth Lyapunov function, *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 4, pp.361-376, 1999.

Sidebar:Maximal Monotone Operators

Since their introduction in 1960 by Minty [S4], 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 3 (Minty-Browder monotonicity)**.** *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* $(x_k, y_k) \in \operatorname{gph} \mathbf{M}$*,* $k = 1, 2$*,*

$$\langle x_1 - x_2, y_1 - y_2 \rangle \geq 0 \ \ (\textit{resp.} \ \geq \alpha ||x_1 - x_2||^2, \ \alpha > 0). \tag{S10}$$

The operator $\mathbf{M}(\cdot)$ *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 (S10). 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 $\langle x, y \rangle_M = x^\top M y$ for some $M = M^\top \succ 0$.

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.

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

Hence, 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 $\mathbf{M}_k : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$, $k = 1, 2$, and let $L \in \mathbb{R}^{n \times m}$ and $\gamma \geq 0$. Then:

1) $\mathbf{M}_k^{-1}$ and $\gamma \, \mathbf{M}_k$ 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) $\mathbf{T}_k(x) = L^\top \mathbf{M}_k(Lx)$ are maximal monotone whenever $\operatorname{rge}(L) \cap \operatorname{rint}(\operatorname{dom} \mathbf{M}_k) \neq \emptyset$.
4) The mapping $x \mapsto -Ax + B(D + \mathbf{M}^{-1})^{-1}(Cx)$ is maximal monotone whenever $\operatorname{Im}(C) \cap \operatorname{rint}(\operatorname{Im}(D + \mathbf{M}^{-1})) \neq \emptyset$, $\mathbf{M}(\cdot)$ is maximal monotone, and the quadruple (A, B, C, D) is passive.

Items 1, 2 and 3 are classical [S5]. Inverse mappings are defined in "Set-Valued Mappings and Differential Inclusions". Item 4 follows from [S6], building upon results in [S7], [S8] (there, monotonicity is understood with the weighted inner product defined by the solution $P = P^\top \succ$

0 of the passivity LMI associated with the quadruple). It means that the negative feedback interconnection of a passive linear invariant system with a maximal monotone static nonlinearity $\mathrm{M}(\cdot)$ preserves the maximal monotonicity. This is a kind of extension of [S5, Example 12.45 (b)]. One of the most important properties of maximal monotonicity is stated in the following theorem.

Theorem 3 (Minty's theorem)**.** *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 $\mathbf{M}(\cdot)$ we construct

$$\mathcal{J}_{\gamma \mathbf{M}} := (\mathrm{I_d} + \gamma \mathbf{M})^{-1}, \tag{S11}$$

the *resolvent* of $\mathbf{M}(\cdot)$ of index $\gamma > 0$. Thus, an alternative statement of Theorem 3 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

$$y \in x + \gamma \mathbf{M}(x) \tag{S12}$$

always admits a solution $x \in \operatorname{dom} \mathbf{M}$. It can be shown that $\mathcal{J}_{\gamma \mathbf{M}}$ is single-valued [S13, Corollary 23.10]. In other words, that the solutions to (S12) are unique.

The archetypical example of a maximal monotone map is the convex subdifferential of a proper convex lsc function (see Definitions 4 and 5 in "Convex Analysis Tools").

Theorem 4. *[S13, Theorem 20.40] Let* $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ *be a proper, convex, lower semicontinuous function. Then,* $\partial f : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *is maximal monotone.*

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

$$y \in x + \gamma \partial f(x) \tag{S13}$$

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,

$$\mathcal{Y}_{\gamma \mathbf{M}} := \frac{1}{\gamma}(\mathrm{I_d} - \mathcal{J}_{\gamma \mathbf{M}}), \tag{S14}$$

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

$$\mathcal{Y}_{\gamma \mathbf{M}} = (\gamma \, \mathrm{I_d} + \mathbf{M}^{-1})^{-1}, \tag{S15}$$

i.e., $\mathcal{Y}_{\gamma \mathbf{M}}(y) = \mathcal{J}_{\frac{1}{\gamma} \mathbf{M}^{-1}}(\frac{1}{\gamma} y)$ [S5, Lemma 12.14]. Moreover, for any $y \in \mathbb{R}^n$, we have [S9, Theorem 2, Chapter 3]

$$(\mathcal{J}_{\gamma \mathbf{M}}(y), \mathcal{Y}_{\gamma \mathbf{M}}(y)) \in \operatorname{gph} \mathbf{M} \, . \tag{S16}$$

References

[S4] G. J. Minty, "Monotone Networks", Proc. R. Soc. Lond. A 257, pp.194–212 http://doi.org/10.1098/rspa.1960.0144, 1960.

[S5] R.T. Rockafellar and R.J.B. Wets, Variational Analysis, Springer-Verlag Berlin Heidelberg, Grundlehren der mathematischen Wissenschaften, vol.317, 1998.

[S6] M.K. Camlibel, J.M. Schumacher, Linear passive systems and maximal monotone mappings, *Mathematical Programming*, Ser. B, vol.157, no 2, pp.397–420, 2016.

[S7] B. Brogliato, Absolute stability and the Lagrange-Dirichlet theorem with monotone multivalued mappings, *Systems and Control Letters*, vol. 51, no 5, pp. 343-353, 2004.

[S8] 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, vol.74, no 1, pp.195-212, 2011.

[S9] J.-P. Aubin, A. Cellina, *Differential Inclusions: Set-Valued Maps and Viability Theory*, Springer Berlin, Heidelberg, 2012.

Sidebar:Convex Analysis Tools

The following results and definitions are taken from classical references [S5], [S10]–[S13]. 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 4. *A proper convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is* lower semicontinuous *(lsc) if its epigraph is closed.*

Definition 5. *Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, convex, lower semicontinuous function. The convex* subdifferential *of $f(\cdot)$ at the point x is given by*

$$\partial f(x) := \{w \in \mathbb{R}^n \mid \langle w, v - x\rangle \leq f(v) - f(x) \text{ for all } v \in \operatorname{dom} f\}. \tag{S17}$$

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

$$\mathbf{sgn}(x) = \begin{cases} -1 & \text{if } x < 0 \\ [-1, 1] & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}. \tag{S18}$$

Definition 6. *Let $K \subseteq \mathbb{R}^n$ be a closed convex set. Its* normal cone *is the set-valued mapping $\mathbf{N}_K : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ defined by*

$$\mathbf{N}_K(x) = \{\zeta \in \mathbb{R}^n \mid \langle \zeta, z - x\rangle \leq 0 \text{ for all } z \in K\} \tag{S19}$$

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

The normal cone to a closed convex set is convex. Suppose that K is finitely represented, *i.e.*, $K = \{w \in \mathbb{R}^n \mid g(w) \geq 0\}$ for some differentiable function $g : \mathbb{R}^n \to \mathbb{R}^m$. Then, $\mathbf{N}_K(x)$ is generated by the outward normals $\nabla g_i(x)$ to K at the active constraints $g_i(x) = 0$, $i \in \{1, \ldots, m\}$. In other words, $\mathbf{N}_K(x) = \{\zeta \in \mathbb{R}^n \mid \zeta = -\nabla g(x)\lambda,\ 0 \leq \lambda \perp g(x) \geq 0\}$. It is noteworthy that weighted inner products can be used in the normal-cone definition. For instance, $\langle \zeta, z - x\rangle_M$ with $M = M^\top \succ 0$ can be used in (S19), in which case the normals are calculated as $M^{-1}\nabla g_i(x)$.

Definition 7. *Let $K \subseteq \mathbb{R}^n$ be a closed convex set. Its* tangent cone *is the set-valued mapping* $\mathbf{T}_K : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ *defined by*

$$\mathbf{T}_K(x) = \{w \in \mathbb{R}^n \mid \langle w, z\rangle \leq 0, \text{ for all } z \in \mathbf{N}_K(x)\}. \tag{S20}$$

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

Fact 5. *Let $K \subseteq \mathbb{R}^n$ be a closed convex set. Then, $\Psi_K(\cdot)$ is proper, convex, lsc, and $\partial\Psi_K(x) = \mathbf{N}_K(x)$ for all $x \in \mathbb{R}^n$.*

Definition 9. *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*

$$f^\star(y) = \sup_{x\in\mathbb{R}^n} \left\{x^\top y - f(x)\right\}. \tag{S21}$$

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: $v \in \partial f(w)$ if, and only if, $w \in \partial f^\star(v)$.*

Theorem 5. *[S13, Theorem 16.23] Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be proper, convex, and lsc. Then:*

$$(w, v) \in \operatorname{gph} \partial f \Leftrightarrow (v, w) \in \operatorname{gph} \partial f^\star \Leftrightarrow w^\top v = f(w) + f^\star(v) \tag{S22}$$

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

$$f^\star(y) = \sup_{x\in\mathbb{R}} \left\{x^\top y - \kappa|x|\right\}. \tag{S23}$$

By concavity, the supremum is actually a maximum whenever the subdifferential of $x^\top y - \kappa|x|$ contains the origin, that is, whenever $y \in \kappa\,\mathbf{sgn}(x)$. 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, $f^\star(\cdot) = \Psi_{[-\kappa,\kappa]}(\cdot)$. Fact 5 is easy to verify in this example:

$$\partial\Psi_{[-\kappa,\kappa]}(y) = \mathbf{N}_{[-\kappa,\kappa]}(y) = \begin{cases} (-\infty, 0] & \text{if } y = -\kappa \\ 0 & \text{if } y \in (-\kappa, \kappa) \\ [0, +\infty) & \text{if } y = \kappa \\ \emptyset & \text{otherwise} \end{cases}. \tag{S24}$$

Verifying Fact 7 is also straightforward:

$$\mathbf{N}_{[-\kappa,\kappa]}(\cdot) = (\kappa\,\mathbf{sgn})^{-1}(\cdot). \tag{S25}$$

More generally, let $\|x\|_p$ be the p norm $(\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$, $1 \leq p < +\infty$, and $\|x\|_\infty = \max_{1\leq i\leq n} |x_i|$. The dual norm is $\|x\|_{p^\star}$ with $1/p+1/p^\star = 1$. We know that, if $f(\cdot) = \|\cdot\|_p$, then $f^\star(\cdot) = \Psi_{\mathcal{B}_{p^\star}}(\cdot)$, where

$$\mathcal{B}_{p^\star} = \{x \in \mathbb{R}^n \mid \|x\|_{p^\star} \leq 1\} \tag{S26}$$

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

$$\partial\|\cdot\|_p = \left(\partial\Psi_{\mathcal{B}_{p^\star}}\right)^{-1}(\cdot) = \left(\mathbf{N}_{\mathcal{B}_{p^\star}}\right)^{-1}(\cdot). \tag{S27}$$

Setting, *e.g.*, $p = 1$ gives $p^\star = \infty$, $\mathcal{B}_\infty = [-1, 1]^n$ and

$$\mathbf{Sgn}(x) = \mathbf{N}_{\mathcal{B}_\infty}^{-1}(x) \tag{S28}$$

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

$$\mathbf{N}_{\mathcal{B}_2}^{-1}(x) = \partial\|x\|_2 = \begin{cases} x/\|x\|_2 & \text{if } x \neq 0 \\ \mathcal{B}_2 & \text{if } x = 0 \end{cases}. \tag{S29}$$

The following result, known as the chain rule of convex analysis, is used at several places of 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 $f(\cdot)$ is a polyhedral function (*i.e.*, its epigraph is a polyhedral set), or that there exists x_0 with $Ax_0 \in \mathrm{dom}(f)$ such that $\mathrm{rge}(A) - \mathbb{R}_+(\mathrm{dom}(f) - Ax_0)$ 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 $x \in \mathbb{R}^m$.*

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

$$\mathrm{Proj}_M[K; y] = \arg\min_{z\in K} \frac{1}{2}\langle z - y, z - y\rangle_M. \tag{S30}$$

For simplicity we will write $\mathrm{Proj}[K;\cdot]$ in place of $\mathrm{Proj}_I[K;\cdot]$.

The projection onto the $\mathcal{B}_2$ is straightforward:

$$\mathrm{Proj}[\mathcal{B}_2, y] = \begin{cases} \frac{y}{\|y\|_2} & \text{if } \|y\|_2 \geq 1 \\ y & \text{otherwise} \end{cases}. \tag{S31}$$

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

$$(\mathrm{Proj}[\mathcal{B}_\infty, y])_i = \min\{|y_i|, 1\}\,\mathbf{sgn}(y_i), \quad i = 1, \ldots, n. \tag{S32}$$

Proposition 5. *Consider a vector $y \in \mathbb{R}^n$, a closed convex set $K \subseteq \mathbb{R}^n$, and a weighted inner product $\langle \cdot, \cdot \rangle_M$. The following statements are equivalent:*

$$x = \mathrm{Proj}_M[K; y], \tag{S33a}$$

$$M(x - y) \in -\,\mathbf{N}_K(x). \tag{S33b}$$

Proof. The proof can be found in [S12, p.79] inside another proof, we reproduce it here for convenience. The optimization problem (S30) can be equivalently written as $\mathrm{Proj}_M[K; y] = \arg\min \frac{1}{2}\langle z - y, z - y\rangle_M + \Psi_K(z)$. By the convexity of the weighted norm and of the indicator function, we have $x = \mathrm{Proj}_M[K; y]$ if, and only if, $0 \in M(x - y) + \partial\Psi_K(x)$, that is, if, and only if, $M(x - y) \in -\,\mathbf{N}_K(x)$. □

If, K is a cone, then the statements (S33) are equivalent to [S10, Corollary 23.5.4]

$$K^\star \ni M(x - y) \perp x \in K, \tag{S34}$$

where

$$K^\star = \{\zeta \in \mathbb{R}^n \mid \langle \zeta, z\rangle \geq 0 \text{ for all } z \in K\} \tag{S35}$$

is the *dual cone* to K.

If K is a polyhedral set, that is, then there exists a matrix $G \in \mathbb{R}^{n\times m}$ and a vector $b \in \mathbb{R}^m$ such that $K = \{x \in \mathbb{R}^n | Gx \leq b\}$. For instance, if $K = \mathcal{B}_\infty$, then $G = [I_n, -I_n]^\top$ and $b = \mathbf{1}_{2n}$. For polyhedral sets, the projection (S33a) can be computed solving a conventional quadratic program. The code below shows an implementation in Python 3 using the open source modelling language for convex optimization CVXPY [119].

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

If the set $K = \mathcal{B}_1$, it is possible to use the code above with $G \in \mathbb{R}^{2^n \times n}$ with the columns 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 x into its positive and negative components, that is $x = x^+ - x^-$, where $x^+, x^- \in \mathbb{R}^n_+$. In such a case the code below solve 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
w_p = cvx.Variable(n)
w_n = cvx.Variable(n)
objective = 0.5*cvx.quad_form(w_p - w_n, M) - y.T@M@(w_p - w_n)
cvx.Problem(cvx.Minimize(objective), H@cvx.bmat([[w_p], [w_n]]) <= c).solve()
x = w_p.value - w_n.value
```

References

[S10] R.T. Rockafellar, Convex Analysis, Princeton Landmarks in Mathematics, 1970.

[S11] J.B. Hiriart Urruty, C. Lemaréchal, Fundamentals of Convex Analysis, Grundlehren Text Editions, Springer-Verlag Berlin Heidelberg, 2001.

[S12] F. Facchinei, J.S. Pang, Finite-Dimensional Variational Inequalities and Complementarity Problems, vol.I, Springer Series in Operations Research, Springer Verlag New York, 2003.

Sidebar:Proximal-Point Algorithm and Proximal Mapping

Proximal-point algorithms and proximal operators are fundamental tools in Optimization. Among the big names and founding fathers of this field we may cite Jean Jacques Moreau [S18]–[S20], Kôsacu Yosida [S21], R. Tyrell Rockafellar [S5], [S23], and Hedy Attouch [S22]. Let us recall that we deal in this article with finite-dimensional systems only. As seen in (10), (12), (70), (76), (92), and (93), 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 "Maximal Monotone Operators"), its set of zeros is denoted by $\operatorname{zero} \mathbf{M} = \mathbf{M}^{-1}(0) = \{x \in \mathbb{R}^n \mid \mathbf{M}(x) = 0\}$.

Definition 10 (Proximal-Point Algorithm [S13, p. 345] [S14])**.** *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* $\mathbf{M}(\cdot)$ *is defined as:*

$$x_{k+1} = \mathcal{J}_{\gamma \mathbf{M}}(x_k), \tag{S36}$$

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

Theorem 1 in [S15] establishes the convergence of (S36) towards $\operatorname{zero} \mathbf{M}$.

Consider a proper, lower semicontinuous, and convex function $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. The operator

$$\operatorname{Prox}_f(\cdot) = (\mathrm{I_d} + \partial f)^{-1}(\cdot) \tag{S37}$$

is called the *proximal mapping* [S13, Definition 12.23]. By definition, when $\mathbf{M}(\cdot) = \partial f(\cdot)$ the algorithm (S36) is equivalent to any of the following iterations:

$$x_{k+1} = \operatorname{Prox}_{\gamma f}(x_k) \tag{S38a}$$

$$x_{k+1} - x_k \in -\gamma \partial f(x_{k+1}) \tag{S38b}$$

$$x_{k+1} = \operatorname*{arg\,min}_{z \in \mathbb{R}^n} \left\{ f(z) + \frac{1}{2\gamma} ||z - x_k||^2 \right\}. \tag{S38c}$$

The form (S38a) gives the algorithm its name. It can be readily seen from (S37) that

$$\operatorname{zero} \operatorname{Prox}_f = \partial f(0). \tag{S39}$$

Also remark that, if $f(\cdot) = \Psi_K(\cdot)$ (see Definition 8) with $K \subseteq \mathbb{R}^n$ a closed convex nonempty set, then:

$$\mathrm{Prox}_f(\cdot) = \mathrm{Proj}(K;\cdot) = (\mathrm{I_d} + \mathbf{N}_K)^{-1}(\cdot). \tag{S40}$$

When implicitly or semi-implicitly discretized, several higher-order differentiators and controllers involve the resolvent of the subdifferential of the proper convex lsc function $f : \mathbb{R} \to \mathbb{R}$, $x \mapsto \frac{1}{2}\gamma x^2 + \Psi_{[-1,1]}(x)$, $\gamma > 0$ *i.e.*, $\partial f(x) = \gamma x + \mathbf{N}_{[-1,1]}(x)$. 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: $\mathcal{J}_{\partial f'}(y) = \frac{1}{1+\gamma}y$ if $y \in [-1-\gamma, 1+\gamma]$, -1 if $y \leq -1-\gamma$, 1 if $y \geq 1+\gamma$.

Remark 8. *It can be readily seen from* (S38b) *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 [S17].*

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

$$x_{k+1} = \mathcal{J}_{\gamma\mathbf{M}}(x_k + w_k). \tag{S41}$$

The next Proposition is the discrete-time counterpart of Proposition 4.

Proposition 6. Robust proximal-point algorithm *Consider the perturbed proximal-point algorithm* (S41) *with $\gamma > 0$ and $\mathbf{M} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ a maximal monotone operator such that*

$$(\gamma\,\mathbf{M})^{-1}\left(\bigcup_{k\in\mathbb{N}} w_k\right) = \{0\}\,. \tag{S42}$$

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

$$(\gamma\,\mathbf{M})^{-1}\left(\bigcup_{k\in\mathbb{N}} w_k + \varepsilon\mathcal{B}\right) = \{0\} \tag{S43}$$

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

Proof. Recalling the definition of Yosida approximation (S14), we have that $x_{k+1} = x_k + w_k - \gamma\mathcal{Y}_{\gamma\mathbf{M}}(x_k + w_k)$ and it follows that

$$\begin{aligned}
\|x_{k+1}\|_2^2 &= x_{k+1}^\top \mathcal{J}_{\gamma\mathbf{M}}(x_k + w_k) \\
&= x_{k+1}^\top \left(x_k + w_k - \gamma\mathcal{Y}_{\gamma\mathbf{M}}(x_k + w_k)\right) \\
&\leq \frac{1}{2}\|x_{k+1}\|_2^2 + \frac{1}{2}\|x_k\|_2^2 - \mathcal{J}_{\gamma\mathbf{M}}(x_k + w_k)^\top \left(\gamma\mathcal{Y}_{\gamma\mathbf{M}}(x_k + w_k) - w_k\right)
\end{aligned} \tag{S44}$$

Setting the Lyapunov function candidate as $V_{k+1} = \frac{1}{2}\|x_{k+1}\|_2^2$, we obtain

$$V_{k+1} - V_k \leq -\mathcal{J}_{\gamma \mathbf{M}}(x_k + w_k)^\top \left(\gamma \mathcal{Y}_{\gamma \mathbf{M}}(x_k + w_k) - w_k\right). \tag{S45}$$

Let $S_k := \mathcal{J}_{\gamma \mathbf{M}}(x_k + w_k)^\top \left(\gamma \mathcal{Y}_{\gamma \mathbf{M}_k}(x_k + w_k) - w_k\right)$. It follows from the monotonicity of $\mathbf{M}$, together with (S16) and (S42), that $S_k \geq 0$. Thus, $V_{k+1} - V_k \leq 0$. Moreover, it follows from (S45) that, for any $k > 0$,

$$\sum_{j=0}^{k} S_j \leq V_0 - V_{k+1} \leq V_0 < +\infty, \tag{S46}$$

so that $S_k \to 0$ as $k \uparrow +\infty$. In what follows we show that $S_k = 0$, if and only if, $x_{k+1} = 0$, implying the global asymptotic stability of the origin. Clearly, S_k can be rewritten as

$$S_k = x_{k+1}^\top (x_k - x_{k+1}) = x_{k+1}^\top x_k - \|x_{k+1}\|_2^2. \tag{S47}$$

Thus, if $x_{k+1} = 0$, then $S_k = 0$. Conversely, if $S_k = 0$, then

$$x_{k+1}^\top x_k = \|x_{k+1}\|_2^2. \tag{S48}$$

Notice that (S48) holds, if and only if, either $x_{k+1} = 0$ (and global asymptotic stability of the origin follows), or $x_{k+1} = x_k$. The latter case implies that $w_k = \gamma \mathcal{Y}_{\gamma \mathbf{M}}(x_k + w_k)$ and it follows from (S16) that $w_k \in \gamma\, \mathbf{M}(\mathcal{J}_{\gamma \mathbf{M}}(x_k + w_k)) = \gamma\, \mathbf{M}(x_{k+1})$ or, equivalently, $x_{k+1} \in (\gamma\, \mathbf{M})^{-1}(w_k)$ and the global asymptotic stability of the origin follows in view of (S42).

Now, the iteration (S41) can also be written as

$$x_k - x_{k+1} + w_k \in \gamma\, \mathbf{M}(x_{k+1}). \tag{S49}$$

By the global asymptotic stability established above, for any $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that $x_k - x_{k+1} \in \varepsilon \mathcal{B}_p$ for all $k \geq N$, $p \in \{1, \ldots, +\infty\}$. Thus, we have $w_k + \varepsilon \mathcal{B}_p \subset \gamma\, \mathbf{M}(x_{k+1})$. The latter inclusion, together with (S43) imply that $x_{k+1} = 0$ for all $k \geq N$. □

References

[S13] 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.

[S14] L. Condat, D. Kitahara, A. Contreras, A. Hirabayashi, Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists, *SIAM Review*, vol.65, no 2, pp.375-435, 2023.

[S15] E.K. Ryu, W. Yin, Large-scale convex optimization via monotone operators, Cambridge University Press, 2023.

[S16] B. Brogliato, B. Maschke, R. Lozano and O. Egeland, Dissipative Systems Analysis and Control: Theory and Applications, Springer London, 2020.

[S17] N. Parikh and S. Boyd, Proximal Algorithms, *Foundations and Trends in Optimization*, vol. 1, no. 3, pp 127-239, 2014.

[S18] J.J. Moreau, Fonctions convexes duales et points proximaux dans un espace hilbertien, *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, vol. 255, pp.2897-2899, 1962.

[S19] J.J. Moreau, Proximité et dualité dans un espace hilbertien, *Bulletin de la Soc. Math. France*, tome 93, pp. 273-299, 1965.

[S20] J.J. Moreau, Propriétés des applications "prox", *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, vol.256, pp.1069-1071, 1963.

[S21] K. Yosida, Functional Analysis, Springer Berlin, Heidelberg, 1974.

[S22] H. Attouch, G. Buttazzo and G. Michaille, Variational Analysis in Sobolev and BV Spaces, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2014.

[S23] R. T. Rockafellar, Monotone operators and the proximal point algorithm, *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877-898, 1976.

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

Implicit 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,

$$\begin{aligned} &\min_{(s_{k+1},u_k)} f(s_{k+1} - h\delta_k) + \frac{h}{2}\left\|u_k\right\|^2, \\ &\text{such that } s_{k+1} = s_k + hu_k + h\delta_k \end{aligned} \tag{S50}$$

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 (S50) is given as,

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

The Karush-Kuhn-Tucker conditions associated with (S50) 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,

$$s_k \in (I + h\partial f)(s_{k+1}^* - h\delta_k) \Leftrightarrow s_{k+1}^* - h\delta_k = \mathrm{Prox}_{hf}(s_k)\,, \tag{S51}$$

and the substitution of (S51) back into KKT-iii) leads to

$$s_{k+1}^* - h\delta_k = s_k + hu_k^* = \mathrm{Prox}_{hf}(s_k) \Leftrightarrow u_k^* = -\frac{1}{h}\left(s_k - \mathrm{Prox}_{hf}(s_k)\right)\,. \tag{S52}$$

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

$$\begin{aligned} s_{k+1}^* &= \tilde{s}_{k+1} + h\delta_k \\ \tilde{s}_{k+1} &= s_k + hu_k^* \quad , \\ -u_k^* &\in \partial f(\tilde{s}_{k+1}) \end{aligned} \tag{S53}$$

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

Sidebar:Projected Dynamical Systems

Projected dynamical systems (PDS) make a class of nonsmooth systems which is, as seen below, closely related to sliding-mode systems. They were introduced in [S24], [S25]. Recently they have witnessed a strong interest in the Automatic Control scientific community, see, *e.g*, [S26], [S27]. PDS may be expressed under different equivalent forms [S28], [S29] (see definitions 6 and 7 in "Convex Analysis Tools"):

$$\dot{x} \in -f(x) - g(t) - \mathbf{N}_\Phi(x)$$

$$\text{with } \dot{x} = -f(x) - g(t) - \mathrm{Proj}[\mathbf{N}_\Phi(x); -f(x) - g(t)] \tag{S54a}$$

$$\dot{x} \in -f(x) - g(t) - \mathbf{N}_{\mathbf{T}_\Phi(x)}(\dot{x}) \tag{S54b}$$

$$\dot{x} = (\mathrm{I_d} + \mathbf{N}_{\mathbf{T}_\Phi(x)})^{-1}(-f(x) - g(t)) \tag{S54c}$$

$$\dot{x} = \mathrm{Proj}[\mathbf{T}_\Phi(x); -f(x) - g(t)] \tag{S54d}$$

The second condition in (S54a) means that solutions are slow (with $-\dot{x}(t)$ of minimal norm inside $f(x) + g(t) + \mathbf{N}_\Phi(x)$). The resolvent of the normal cone to the tangent cone appears in (S54c), and use was made of Fact 5 in "Convex Analysis Tools", (S37) and the comment just after (S39) in "Proximal-Point Algorithm and Proximal Mapping" to obtain (S54d) . The projection is a particular instance of $\mathrm{Prox}(\cdot)$, see (S40), showing that PDS belong to the class of proximal systems.

Discrete-time PDS

Let us now discretize (S54)(a) as:

$$x_{k+1} \in x_k - h(f(x_k) - g_k) - \mathbf{N}_\Phi(x_{k+1}) \tag{S55}$$

This is rewritten equivalently as $x_{k+1} = (\mathrm{I_d} + \mathbf{N}_\Phi)^{-1}(x_k - h(f(x_k) + g_k))$, that is:

$$x_{k+1} = \mathrm{Proj}[\Phi; x_k - h(f(x_k) + g_k)] \tag{S56}$$

where again the same tools have been used as above to pass from (S54c) to (S54d). We see once again that resolvents are ubiquitous in PDS. The system (S56) furnishes a convenient way to obtain solutions whenever the projection onto the set Φ can be computed. This is the case when Φ is a nonempty convex polyhedral: $\Phi = \{x \in \mathbb{R}^n \mid Cx + d \geq 0\}$ for some $C \in \mathbb{R}^{m \times n}$, $d \in \mathbb{R}^m$, $1 \leq m$. Indeed, in this case the normal cone can be expressed as $\mathbf{N}_\Phi = \{\zeta \in \mathbb{R}^n \mid \zeta = -C^\top \lambda,\ 0 \leq \lambda \perp Cx + d \geq 0\}$. In other words, the normal cone is generated by the outwards normals to the active constraints at x. Thus, using (S55) the PDS is equivalently rewritten as a complementarity system:

$$\left\{ \begin{array}{c} x_{k+1} = x_k - h(f(x_k) - g_k) + C^\top \lambda_{k+1} \\ 0 \leq \lambda_{k+1} \perp w_{k+1} = Cx_{k+1} + d \geq 0 \end{array} \right. . \tag{S57}$$

The discrete-time PDS in (S57) is a mixed linear complementarity problem [S30] with unknowns x_{k+1} and λ_{k+1}. It gives the LCP:

$$0 \leq \lambda_{k+1} \perp w_{k+1} = C(x_k - h(f(x_k) - g_k)) + CC^\top \lambda_{k+1} + d \geq 0, \tag{S58}$$

Obviously, $CC^\top \succ 0$ if and only if, C has row-rank m ($\Rightarrow m \leq n$), otherwise $CC^\top \succcurlyeq 0$ and it is also a copositive-plus matrix. Using [S30, Theorem 3.8.6], if $d \in (\mathrm{SOLLCP}(CC^\top, 0))^\star$, then the LCP in (S58) has a solution. Assuming a Slater condition (*i.e.*, there exists λ such that $CC^\top \lambda + C(x_k - h(f(x_k) - g_k)) + d > 0$) then by [S30, Theorem 5.3.9] the LCP in (S58) has a solution. Also if λ_1 and λ_2 are any two solutions, then $CC^\top(\lambda_1 - \lambda_2) = 0$ [S30, Theorem 3.1.7]. Therefore $\lambda_1 - \lambda_2 \in \mathrm{Ker}(CC^\top) = \mathrm{Ker}(C^\top)$ [3, Theorem 3.5.3]. It is easily deduced that, if $x^i_{k+1} = x^i_k + h(f(x^i_k) - g_k) + C^\top \lambda^i_{k+1}$, $i = 1, 2$, then $x^1_k = x^2_k$ implies $x^1_{k+1} = x^2_{k+1}$, $k \geq 0$. As expected, the projection onto Φ is unique. A quite interesting fact is that efficient algorithms can be used to solve the LCP in (S58) [S30], [S31], at each time-step $k \geq 1$.

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 Φ. In this case there exists an element of $\lambda(x, t) \in -\mathbf{N}_\Phi(x)$ (a selection, see Definition 2 in "Set-Valued Mappings and Differential Inclusions") which "compensates" for the part of $-f(x) - g(t)$ which tends to "push" the trajectory outside Φ (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 [S32, Equation (6)] can be transformed into a PDS after the classical state space transformation (see [S29, section 3.4]). It guarantees robust output tracking.

References

[S24] C. Henry, Differential equations with discontinuous right-hand side for planning procedures, *J. Econom. Theory*, vol. 4, pp.545-551, 1972.

[S25] C. Henry, An existence theorem for a class of differential equations with multivalued right- hand side, *J. Math. Anal. Appl.*, vol. 41,pp. 179-186, 1973.

[S26] D. Gadjov, L. Pavel, A passivity-based approach to Nash equilibrium seeking over networks, *IEEE Transactions on Automatic Control*, vol. 64, no. 3, pp. 1077-1092, March 2019.

[S27] L. Pavel, Dissipativity theory in game theory: On the role of dissipativity and passivity in Nash equilibrium seeking, *IEEE Control Systems Magazine*, vol. 42, no. 3, pp.150-164, June 2022.

[S28] B. Brogliato, A. Daniliidis, C. Lemaréchal, V. Acary, On the equivalence between complementarity systems, projected systems and differential inclusions, Systems Control Lett., vol. 55, pp.45-51, 2006

[S29] B. Brogliato, A. Tanwani, Dynamical systems coupled with monotone set-valued operators: Formalisms, applications, well-posedness, and stability, *SIAM Review*, vol. 62, no 1, pp.3-129, 2020.

[S30] R.W. Cottle, J.S. Pang, R.E. Stone, The Linear Complementarity Problem, Academic Press, Computer Science and Science Comuting, 1992.

[S31] V. Acary, B. Brogliato, Numerical Simulation for Nonsmooth Dynamical Systems, Springer, LNACM vol.35, 2008.

[S32] F. A. Miranda, F. Castaños, Robust output regulation of linear passive systems using maximally monotone controls, *54th IEEE Conference on Decision and Control (CDC)*, Osaka, Japan, pp.6897-6902, 2015.

References

[1] W. M Haddad, J. Lee, Finite-time stability of discrete autonomous systems. Automatica, vol. 122, 2020.

[2] Dong, Q.-L., Cho, Y. J., He S., Pardalos, P. M., Rassias, T. M. The Krasnosel'skiĭ-Mann Iterative Method: Recent Progress and Applications, Springer, 2022.

[3] D.S. Bernstein, Scalar, Vector, and Matrix Mathematics. Theory, Facts and Formulas. Revised and expanded edition, Princeton University Press, 2018.

[4] C.I. Byrnes, A. Isidori and J. C. Willems, Passivity, feedback equivalence, and the global stabilization of minimum phase nonlinear systems, in *IEEE Transactions on Automatic Control*, vol. 36, no. 11, pp. 1228-1240, Nov. 1991.

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

[6] G.R. Duan and H.H. Yu, LMIs in Control Systems, Analysis, Design and Applications, CRC Press, 2013.

[7] E. M. Navarro-López, E. Fossas-Colet, Feedback passivity of nonlinear discrete-time systems with direct input–output link, *Automatica*, vol. 40, no. 8, pp. 1423-1428, 2004.

[8] V. Utkin, J. Guldner and J. Shi, Sliding Mode Control in Electro-Mechanical Systems, CRC Press, 2009.

[9] Y. Shtessel, C. Edwards, L. Fridman and A. Levant, Sliding Mode Control and Observation, Birkhäuser New York, NY, 2013.

[10] V. Acary, B. Brogliato, Y. Orlov, Chattering-free digital sliding-mode control with state observer and disturbance rejection, *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1087-1101, May 2012 .

[11] O. Huber, V. Acary, B. Brogliato, Lyapunov stability and performance analysis of the implicit discrete sliding mode control, *IEEE Transactions on Automatic Control*, vol.61, no 10, pp.3016-3030, 2016.

[12] A. Levant, On fixed and finite time stability in sliding mode control, *Proceedings IEEE Conference on Decision and Control*, IEEE, Washington, DC, pp.4260–4265, 2023.

[13] V. Acary, B. Brogliato, Implicit Euler numerical scheme and chattering-free implementation of sliding mode systems, *Systems and Control Letters*, vol.59, no 5, pp.284-293, 2010.

[14] D.J. Hill, P.J. Moylan, Dissipative dynamical systems: baisc input-output and state properties, *Journal of the Franklin Institute*, vol.30, no 5, pp.327-357, 1980.

[15] W. Lin, C.I. Byrnes, Passivity and absolute stabilization of a class of discrete-time nonlinear systems, *Automatica*, vol.31, no 2, pp.263-267, 1995.

[16] X. Yu and G. Chen. Discretization behaviors of equivalent control based sliding-mode control systems, *IEEE Transactions on Automatic Control*, vol.48, no 9, pp.1641–1646,

2003.

[17] 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*, vol.53, no 6, pp.1563–1569, 2008.

[18] Y. Yan, Z. Galias, X. Yu, and C. Sun. Euler's discretization effect on a twisting algorithm based sliding mode control, *Automatica*, vol.6, pp.203–208, 2016.

[19] Y. Yan, S. Yu, and X. Yu. Quantized super-twisting algorithm based sliding mode control, *Automatica*, vol.105, pp.43–48, 2019.

[20] 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)*, Istanbul, Turkey, 2015.

[21] Z. Galias and X. Yu. Complex discretization behaviours of a simple sliding-mode control system. *IEEE Transactions on Circuits and Systems–II: Express Briefs*, vol.53, no 8, pp.652–656, August 2006.

[22] Z. Galias and X. Yu. Euler's discretization of single input sliding-mode control systems, *IEEE Transactions on Automatic Control*, vol. 52, no 9, pp.1726–1730, September 2007.

[23] 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*, vol. 55, no 12, pp.1269–1273, December 2008.

[24] B. Baji, A. Cabot, An inertial proximal algorithm with dry friction: finite convergence results, *Set Valued Analysis*, vol.14, no. 1, pp.1-23, 2006.

[25] B. Brogliato, Comments on "Finite-time stability of discrete autonomous systems [Automatica 122 (2020) 109282]", *Automatica*, vol.156, paper 111206, October 2023.

[26] O. Huber, B. Brogliato, V. Acary, A. Boubakir, F. Plestan, B. Wang, Experimental results on implicit and explicit time-discretization of equivalent-control-based sliding-mode control, in L. Fridman, J.P. Barbot, F. Plestan (eds.), Recent Trends in Sliding Mode Control, vol.102, IET, pp.207- 235, IET Control, Robotics and Sensors Series, 2016.

[27] B. Wang, B. Brogliato, V. Acary, A. Boubakir, 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*, vol.23, no 5, pp.2071-2075, 2015.

[28] D.S. Bernstein, Facing future challenges in feedback control of aerospace systems through scientific experimentation, *Journal of Guidance, Control and Dynamics*, vol.45, no 12, pp.2202-2210, 2022.

[29] B. Brogliato, Dissipative dynamical systems with set-valued feedback loops: Well-posed set-valued Lur'e dynamical systems, *IEEE Control Systems Magazine*, vol. 42, no.3, pp.93-114, June 2022.

[30] B. Brogliato, A. Polyakov, D. Efimov, The implicit discretization of the supertwisting sliding-mode control algorithm, *IEEE Transactions on Automatic Control*, vol. 65, no. 8, pp.3707-3713, Aug. 2020.

[31] J.E. Carvajal-Rubio, J.D. Sánchez-Torres, M. Defoort, M. Djemai, A.G. Loukianov, Implicit and explicit discrete-time realizations of homogeneous differentiators, *Int. J. Robust Nonlinear Control*, vol.31, pp.3606–3630, 2021.

[32] J.E. Carvajal-Rubio, M. Defoort, J.D. Sánchez-Torres, M. Djemai, A.G. Loukianov, Implicit and explicit discrete-time realizations of the robust exact filtering differentiator, *Journal of the Franklin Institute*, vol.359, pp.3951-3978, 2022.

[33] M.R. Mojallizadeh, B. Brogliato, V. Acary, Time-discretization of differentiators: design of implicit algorithms, and comparative analysis, *Int. Journal of Robust and Nonlinear Control*, vol.31, no 16, pp.7679-7723, 2021.

[34] B. Andritsch, L. Watermann, S. Koch, M. Reichhartinger, J. Reger, M. Horn, Modified implicit discretization of the super-twisting controller, https://arxiv.org/pdf/2303.15273.pdf, 27 October 2023.

[35] X. Xiong, G. Chen, Y. Lou, R. Huang, S. Kamal, Discrete-time implementation of super-twisting control with semi-implicit Euler method, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no.1, pp.99–103, 2022.

[36] J. Davila, L. Fridman, A. Levant, Second-order sliding mode observer for mechanical systems, *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp.1785–1789, Nov. 2005.

[37] Y. Z. Tsypkin, Theory of Relay Control Systems, Moscow: Gostechizdat, 1955.

[38] A. Aizerman, F. R. Gantmakher, On certain switching specifics in non-linear automatic control systems with a piecewise-smooth characteristics of nonlinear element, *Automatika i Telemekhanika*, vol. 18, pp.1017–1028, 1957.

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

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

[41] J.J. Slotine, J.K. Hedrick, E.A. Misawa, On sliding observers for nonlinear systems, *J. Dyn. Syst. Meas. Control*, vol.109, no 3, pp.245-252, 1987.

[42] R. Kikuuwe, R. Pasaribu, G. Byun, A first-order differentiator with first-order sliding mode filtering, *IFAC-PapersOnLine*, vol.52, no 16, pp.771-776, 2019.

[43] G. Byun, R. Kikuuwe, An improved sliding mode differentiator combined with sliding mode filter for estimating first and second-order derivatives of noisy signals, *Int. J. Control Autom. Syst.*, vol.18, pp.3001–3014, 2020.

[44] A. Levant, Higher-order sliding modes, differentiation and output-feedback control, *Int. J. Control*, vol.76, no 9-10, pp.924-941, 2003.

[45] A. Levant, M. Livne, Robust exact filtering differentiators, *Eur. J Control*, vol.55, pp.33-44, 2020.

[46] E. Cruz-Zavala, J.A. Moreno, L.M. Fridman, Uniform robust exact differentiator, *IEEE Trans Automat Contr.*, vol. 56, no 11, pp.2727-2733, 2011.

[47] A.F. Filippov, Differential equations with discontinuous right-hand side, *Matematicheskii Sbornik, Novaya Seriya*, Vol. 51(93), No 1, pp. 99–128, 1960.

[48] A.F. Filippov, Differential Equations with Discontinuous Righthand Sides, Springer Science + Business Media, Dordrecht, Mathematics and Its Applications, 1988.

[49] A. Levant, Robust exact differentiation via sliding mode technique, *Automatica*, vol. 34, pp.379–384, 1998.

[50] K.J. Åström, P. Hagander, J. Sternby, Zeros of sampled systems, *Automatica*, vol. 20, no 1, pp.31-38, 1984.

[51] S. Greenhalgh, V. Acary, B. Brogliato, On preserving dissipativity properties of linear complementarity dynamical systems with the θ-method, *Numer. Math.*, vol. 125, pp.601–637, 2013.

[52] P. L. Lions, B. Mercier. Splitting algorithms for the sum of two nonlinear operators, *SIAM Journal of Numerical Analysis*, vol. 16, no. 6, pp. 964-979, 1979.

[53] J. Eckstein, D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, *Mathematical Programming*, vol. 55, pp. 293-318, 1992.

[54] S. Koch, M. Reichhartinger, M. Horn, L. Fridman, Discrete-time implementation of homogeneous differentiators, *IEEE Trans Automat Contr.*, vol.65, no 2, pp.757-762, 2020.

[55] F. Miranda-Villatoro, B. Brogliato, F. Castanos, Multivalued robust tracking control of Lagrange systems: continuous and discrete-time algorithms, *IEEE Transactions on Automatic Control*, vol.62, no 9, pp.4436-4450, 2017

[56] F. Miranda-Villatoro, B. Brogliato, F. Castanos, Set-valued sliding-mode control of uncertain linear systems: continuous and discrete-time analysis, *SIAM Journal on Control and Optimization*, vol.56, no 3, pp.1756-1793, 2018.

[57] F. Miranda-Villatoro, F. Castanos, B. Brogliato, Continuous and discrete-time stability of a robust set-valued nested controller, *Automatica*, vol.107, pp.406-417, September 2019.

[58] R. Kikuuwe, Y. Yamamato, B. Brogliato, Implicit implementation of nonsmooth controllers to nonsmooth actuators, *IEEE Transactions on Automatic Control*, vol. 67, no 9, pp.4645-4657, 2022.

[59] X. Xiong, R. Kikuuwe, S. Kamal, S. Jin, Implicit-Euler implementation of super-twisting observer and twisting controller for second-order systems, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol.67, no.11, pp.2607-2611, November 2020.

[60] R. Kikuuwe, S. Yasukouchi, H. Fujimoto, M. Yamamoto, Proxy-based sliding mode control:

A safer extension of PID position control, *IEEE Transactions on Robotics*, vol.26, No.4, pp.670-683, August 2010.

[61] L. Michel, S. Selvarajan, M. Ghanes, F. Plestan, Y. Aoustin, J.P. Barbot, An experimental investigation of discretized homogeneous differentiators: Pneumatic actuator case, *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, vol.2, no 3, pp.227-236, 2021.

[62] L. Michel, M. Ghanes, F. Plestan, Y. Aoustin, J.P. Barbot, Semi-implicit Euler discretization for homogeneous observer-based control: one dimensional case, *IFAC-PapersOnLine*, vol. 53, no 2, pp.5135-5140, 2020.

[63] L. Michel, M. Ghanes, Y. Aoustin, J.P. Barbot, A third order semi-implicit homogeneous differentiator: Experimental results, *16th International Workshop on Variable Structure Systems (VSS)*, Rio de Janeiro, Brazil, pp.77-82, 2022.

[64] O. Huber, V. Acary, B. Brogliato, Lyapunov stability analysis of the implicit discrete-time twisting control algorithm, *IEEE Transactions on Automatic Control*, vol.65, no 6, pp.2619-2626, June 2020.

[65] G.F. Franklin, J.D. Powell, M.L. Workman. *Digital control of dynamic systems*. Vol. 3. Menlo Park, CA: Addison-wesley, 1998.

[66] A. Winkler, G. Grabmair, J. Reger, Om implementing the implicit discrete-time super-twisting observer on mechanical systems, *Int. Journal Robust and Nonlinear Control*, 2023, DOI: 10.01002/rnc.6764

[67] X. Xiong, Y. Chu, A.D. Udai, S. Kamal, S. Jin, Y. Lou, Implicit discrete-time terminal sliding mode control for second-order systems, *IEEE Transactions on Circuits and Systems–II: Express Briefs*, vol.68, no 7, pp.2508-2512, 2021.

[68] C. Wang, H. Xia, S. Ren, An implicit discretization-based adaptive reaching law for discrete-time sliding mode control systems, *Journal of Vibration and Control*, vol.29, no 5-6, pp.1117-1127, 2023.

[69] O. Huber, Analysis and implementation of discrete-time sliding mode control, Ph.D. Thesis, Université Grenoble Alpes, 2015. https://inria.hal.science/tel-01194430v1/document

[70] O. Huber, V. Acary, B. Brogliato, F. Plestan, Implicit discrete-time twisting controller without numerical chattering: analysis and experimental results, *Control Engineering Practice*, vol.46, pp.129-141, January 2016.

[71] B. Brogliato, A. Polyakov, Globally stable implicit Euler time-discretization of a nonlinear single-input sliding-mode control system, *54th IEEE Conference on Decision and Control (CDC)*, pp.5426-5431, 2015.

[72] A. Polyakov, D. Efimov, B. Brogliato, Consistent discretization of finite-time and fixed-time stable systems, *SIAM Journal on Control and Optimization*, vol.57, no 1, pp.78-103, 2019.

[73] X. Xiong, H. Chen, Y. Lou, Z. Liu, S. Kamal, M. Yamamoto, Implicit discrete-time adaptive

first-order sliding mode control with predefined convergence time, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 12, pp.3562-3566, Dec. 2021.

[74] C. Wang, H. Xia, Y. Wang, S. Ren, Implicit discrete-time fast terminal sliding mode control with disturbance compensation, *Asian J. Control*, vol. 25, no 1, pp.637–643, 2023.

[75] C. Hettiger, L. Watermann, K. Kumari, L. Eisenzopf, F. Weissenberger, M. Horn, S. Koch, J. Reger, M. Reichhartinger, On discretization methods for indirect adaptive sliding mode control, *IEEE 61st Conference on Decision and Control (CDC)*, Cancun, Mexico, pp.4930-4936, 2022.

[76] C. Wang, H. Xia, Y. Wang, S. Ren, Discrete-time sliding mode control with adaptive reaching law via implicit Euler method, *Int. J. Control Autom. Syst.*, vol. 21, pp.109–116, 2023.

[77] R. Seeber, S. Koch, Structural conditions for chattering avoidance in implicitly discretized sliding mode differentiators, *IEEE Control Systems Letters*, vol. 7, pp.2065-2070, 2023.

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

[79] S. Jin, X. Xiong, D. Zhao, C. Jin, Unified framework for implicit-Euler implementation of second-order sliding mode controllers, *Communications in Nonlinear Science and Numerical Simulation*, vol. 111, paper 106430,2022.

[80] R.K. Sharma, X. Xiong, S. Kamal, S. Ghosh, Discrete-time super-twisting fractional-order differentiator with implicit Euler method, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 4, pp.1238-1242, April 2021.

[81] X. Xiong, Y. Bai, R. Shi, S. Kamal, Y. Wang, Y. Lou, Discrete-time twisting algorithm implementation with implicit-Euler ZOH discretization method, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 8, pp.3435-3439, Aug. 2022.

[82] S. Jin, Z. Lv, X. Xiong, J. Yu, A chattering-free sliding mode filter enhanced by first order derivative feedforward, *IEEE Access*, vol. 8, pp.41175-41185, 2020.

[83] X. Xiong, S. Kamal, S. Jin, Adaptive gains to super-twisting technique for sliding mode design, *Asian J Control*, vol. 23, pp.362–373, 2021.

[84] X. Xiong, R. K. Sharma, S. Kamal, S. Ghosh, Y. Bai, Y. Lou, Discrete-time super-twisting fractional-order observer with implicit Euler method, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 6, pp.2787-2791, June 2022.

[85] X. Yang, X. Xiong, Z. Zou, Y. Lou, S. Kamal, J. Li, Discrete-time multivariable super-twisting algorithm with semi-implicit Euler method, *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 11, pp.4443-4447, Nov. 2022.

[86] M.T.S. Aung, Z. Shi, 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, vol.140, no 4, paper 041005, 2018.

[87] D. Luo, X. Xiong, S. Jin, S. Kamal, Adaptive gains of dual level to super-twisting algorithm for sliding mode design, *IET Control Theory Appl.*, vol.12 pp.2347-2356, 2018.

[88] S. Jin, R. Kikuuwe, M. Yamamoto, Improving velocity feedback for position control by using a discrete-time sliding mode filtering with adaptive windowing, *Advanced Robotics*, vol.28, no 14, pp.943-953, 2014.

[89] R. Kikuuwe, Torque-bounded admittance control realized by a set-valued algebraic feedback, *IEEE Transactions on Robotics*, vol. 35, no. 5, pp.1136-1149, Oct. 2019.

[90] R. Kikuuwe, Sliding motion accuracy of proxy-based sliding mode control subjected to measurement noise and disturbance, *European Journal of Control*, vol. 58, pp.114-122, 2021.

[91] G. Chen, X. Xiong, Y. Lou, Multi-state modelling and observation of magneto-rheological clutch with rate-dependent hysteresis characteristic, *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp.2445-2452, April 2021.

[92] G. Leitmann, Guaranteed asymptotic stability for some linear systems with bounded uncertainties, *J. Dyn. Sys., Meas., Control*, 101(3): 212-216, 1979.

[93] S. Gutman, Uncertain dynamical systems–A Lyapunov min-max approach, in *IEEE Transactions on Automatic Control*, vol. 24, no. 3, pp. 437-443, June 1979.

[94] E. Cruz-Zavala, J.A. Moreno, Homogeneous high order sliding mode design: A Lyapunov approach, *Automatica*, Volume 80, pp. 232-238, 2017.

[95] G. Perozzi, A. Polyakov, F.Miranda-Villatoro, B.Brogliato, Upgrading a linear controller to a sliding mode one: Theory and experiments, *Control Engineering Practice*, vol. 123, 105107, June 2022.

[96] M.R. Mojallizadeh, B. Brogliato, A. Polyakov, S. Selvarajan, L. Michel, F. Plestan, M. Ghanes, J.P. Barbot, Y. Aoustin, Discrete-time differentiators in closed-loop control systems: experiments on electro-pneumatic system and rotary inverted pendulum, Research Report, February 2023 https://inria.hal.science/hal-03125960v2/document.

[97] M.R. Mojallizadeh, B. Brogliato, A. Polyakov, S. Selvarajan, L. Michel, F. Plestan, M. Ghanes, J.B. Barbot, Y. Aoustin, A survey on the discrete-time differentiators in closed-loop control systems: Experiments on an electro-pneumatic system, *Control Engineering Practice*, vol. 136, paper 105546, 2023.

[98] M.R. Mojallizadeh, B. Brogliato, Effect of Euler explicit and implicit time discretizations on variable-structure differentiators. In: Oliveira, T.R., Fridman, L., Hsu, L. (eds) *Sliding-Mode Control and Variable-Structure Systems.* Studies in Systems, Decision and Control, vol 490, pp.165-180, Springer, Cham, 2023.

[99] M.A. Alarcón-Carbajal, J.E. Carvajal-Rubio, J.D. Sánchez-Torres, D.E. Castro-Palazuelos, G.J.Rubio-Astorga, An output feedback discrete-time controller for the DC-DC Buck

converter, *Energies*, vol.15, no 14, 5288, 2022.

[100] R. Mae, R. Kikuuwe, An admittance controller with a jerk limiter for position-controlled robots, Research Report, 2023, https://jxiv.jst.go.jp/index.php/jxiv/preprint/view/444/1402, DOI: https://doi.org/10.51094/jxiv.444.

[101] I. Sala-Mira, J.L. Díez, B. Ricarte, J. Bondia, Sliding-mode disturbance observers for an artificial pancreas without meal announcement, *Journal of Process Control*, vol. 78, pp.68-77, 2019.

[102] I. Sala-Mira, M. Siket, L. Kovács, G. Eigner and J. Bondia, Effect of model, observer and their interaction on state and disturbance estimation in artificial pancreas: An in-silico study, *IEEE Access*, vol. 9, pp.143549-143563, 2021.

[103] S. Faccioli, I. Sala-Mira, J.L. Díez, A. Facchinetti, G. Sparacino, S. Del Favero, 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*, vol. 219, 106736, 2022.

[104] R. Nishimoto, R. Kikuuwe, Position-commanding anti-sway controller for 2-D overhead cranes under velocity and acceleration constraints, *IEEE Access*, vol. 11, pp.35069-35079, 2023.

[105] Y. Yamamoto, J. Qiu, T. Doi, T. Nanjo, K. Yamashita, R. Kikuuwe, A position controller for hydraulic excavators with deadtime and regenerative pipelines, Research Report, 2023, https://jxiv.jst.go.jp/index.php/jxiv/preprint/view/440/1564, DOI: https://doi.org/10.51094/jxiv.440.

[106] R. Kikuuwe, T. Okada, H. Yoshihara, T. Doi, T. Nanjo, K. Yamashita, A nonsmooth quasi-static modeling approach for hydraulic actuators, *Transactions of ASME: Journal of Dynamic Systems, Measurement, and Control*, vol.143, no.12, p.121002, December 2021.

[107] Y. Yamamoto, J. Qiu, Y. Munemasa, T. Doi, T. Nanjo, K. Yamashita, R. Kikuuwe, A sliding-mode set-point position controller for hydraulic excavators, *IEEE Access*, vol. 9, pp.153735-153749, 2021.

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

[109] C. B. Vũ, A splitting algorithm for dual monotone inclusions involving cocoercive operators, *Advances in Computational Mathematics*, Volume 38, pp. 667-681, 2013.

[110] P. L. Combettes, J.-C. Pesquet, Proximal splitting methods in signal processing, In: Bauschke, H., Burachik, R., Combettes, P., Elser, V., Luke, D., Wolkowicz, H. (eds), *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer, vol. 49, pp. 185-212, 2011.

[111] X. Xiong, Z. Zou, Y. Lou, X. Yang, X. Zhu, F. Zheng, Semi-implicit Euler realization

of time-delayed super-twisting algorithm with modified Smith predictor, *IEEE 19th International Conference on Automation Science and Engineering (CASE)*, Auckland, New Zealand, pp.1-6, 2023.

[112] G. Chen, X. Xiong, Y. Lou, Z. Li, Modeling and observation of rate-dependent hysteresis and creep phenomena in magnetorheological clutch, *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp.2053-2061, Aug. 2022.

[113] O. Güler, Foundations of Optimization, Springer Science+Business Media, Graduate Texts in Mathematics 258, 2010.

[114] P. Goulart, Y. Chen, Clarabel solver, https://oxfordcontrol.github.io/ClarabelDocs/stable/. Last accessed 23/11/2023.

[115] J.A. Moreno, On discontinuous observers for second order systems: Properties, analysis and design, in B. Bandyopadhyay et al. (Eds.): *Advances in Sliding Mode Control*, Springer-Verlag Berlin Heidelberg, LNCIS 440, pp.243–265, 2013.

[116] I. Nagesh, C. Edwards, A multivariable super-twisting sliding mode approach, *Automatica*, vol. 50, no. 3, pp.984-988, 2014.

[117] T. Liard, I. Balogoun, S. Marx, F. Plestan, Boundary sliding mode control of a system of linear hyperbolic equations: A Lyapunov approach, *Automatica*, vol. 135, paper 109964, 2022.

[118] M.A. Estrada, J.A. Moreno, L. Fridman, Sliding mode controllers design based on control Lyapunov functions for uncertain LTI systems, *IFAC-PapersOnLine*, vol. 56, no 2, pp.615-1620, 2023.

[119] S. Diamond, S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization", *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1-5, 2016.

[120] S. Ding, A. Levant, S. Li, Simple homogeneous sliding-mode controller, *Automatica*, vol. 67, pp.22-32, 2016.

[121] A. Polyakov, Generalized Homogeneity in Systems and Control, Springer Nature Switzerland AG, Communications and Control Engineering, 2020.

[122] O. Huber, http://xhub.github.io/pages/videos.html, 2016.

Authors Biographies

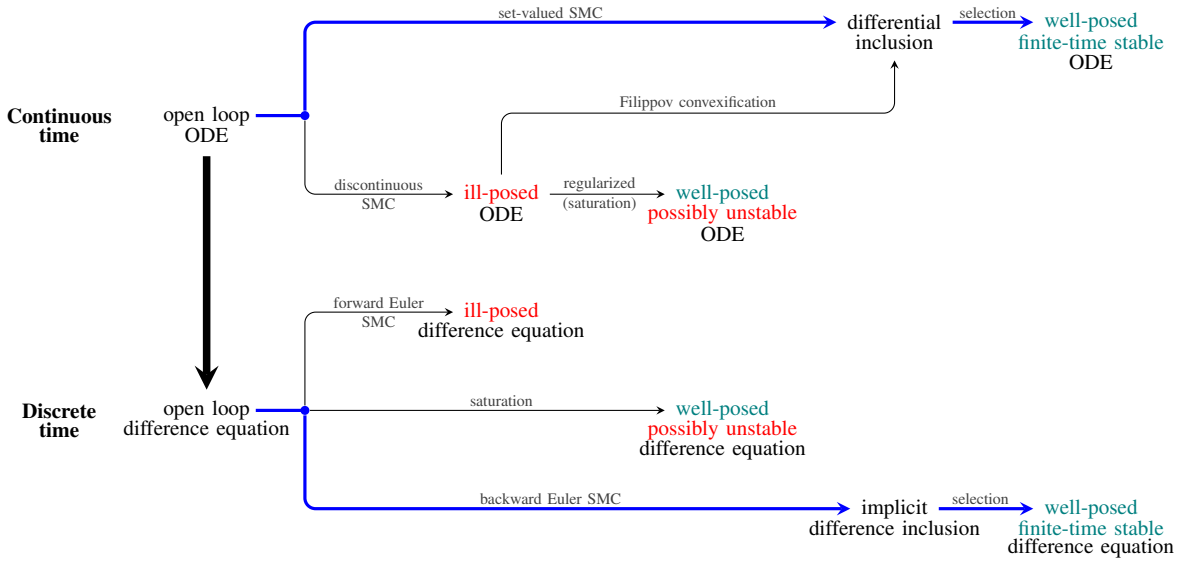

Figure S1: Flowchart of the discretization process of systems controlled with sliding modes.

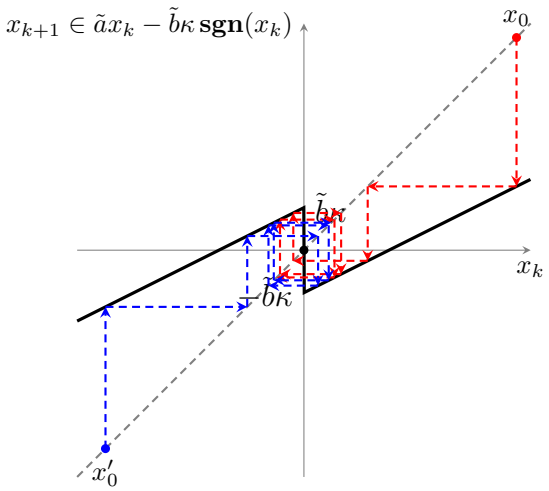

Figure S2: Cobweb diagram of (6) with parameters: $\tilde{a} = 0.5$, $\tilde{b}\kappa = 0.75$ for two different initial conditions. The explicit discretization of the set-valued law (2) yields an iteration with a non-monotone map, such that for almost all initial conditions, the closed-loop shows an oscillatory behavior. The amplitude of the oscillation is proportional to the control gain κ.

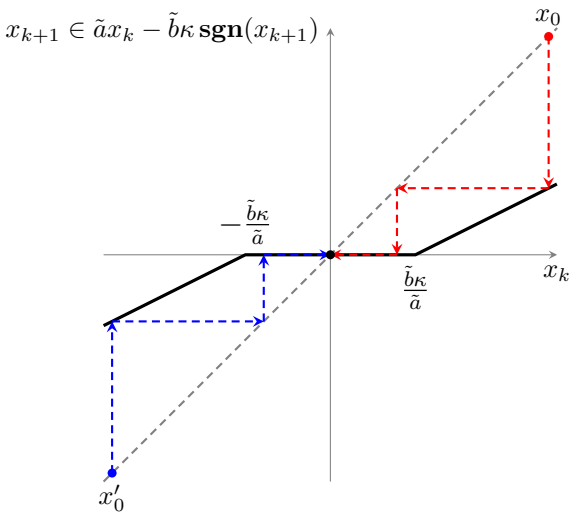

Figure S3: Cobweb diagram of (8) with parameters: $\tilde{a} = 0.5$, $\tilde{b}\kappa = 0.75$. The implicit discretization of the set-valued law (2) yields an iteration with a maximal monotone map, such that for any initial condition the state converges towards the origin after a finite number of steps.

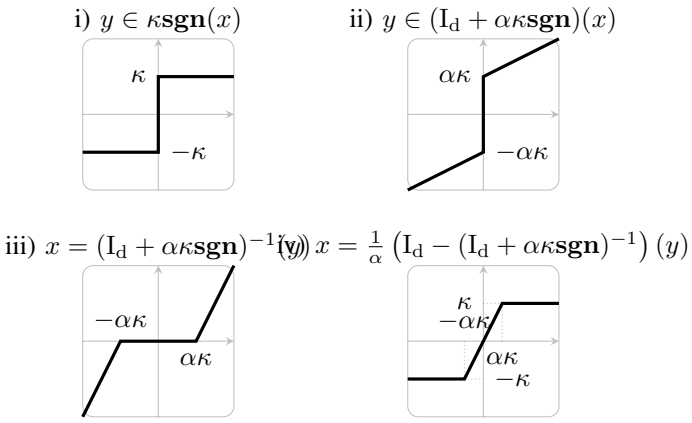

Figure S4: Graphical computation of resolvent and Yosida approximation (lower left and right figures, respectively) for the $\mathbf{sgn}$ multifunction in the scalar case. Note that, at each step, a maximal monotone operator is obtained.

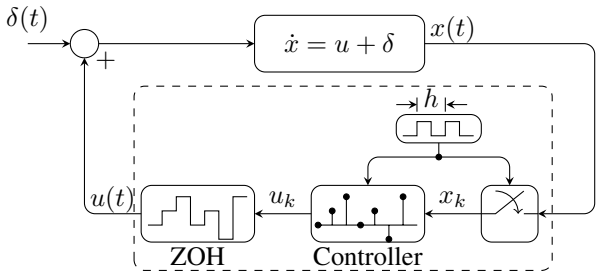

Figure S5: Closed-loop system. The state of the continuous-time dynamics is sampled every h seconds. Such information is used for computing the control input u_k which passes through a classical zero-order-hold mechanism to obtain the feeedback input signal $u(t) = u_k$ for $kh < t \leq (k+1)h$.

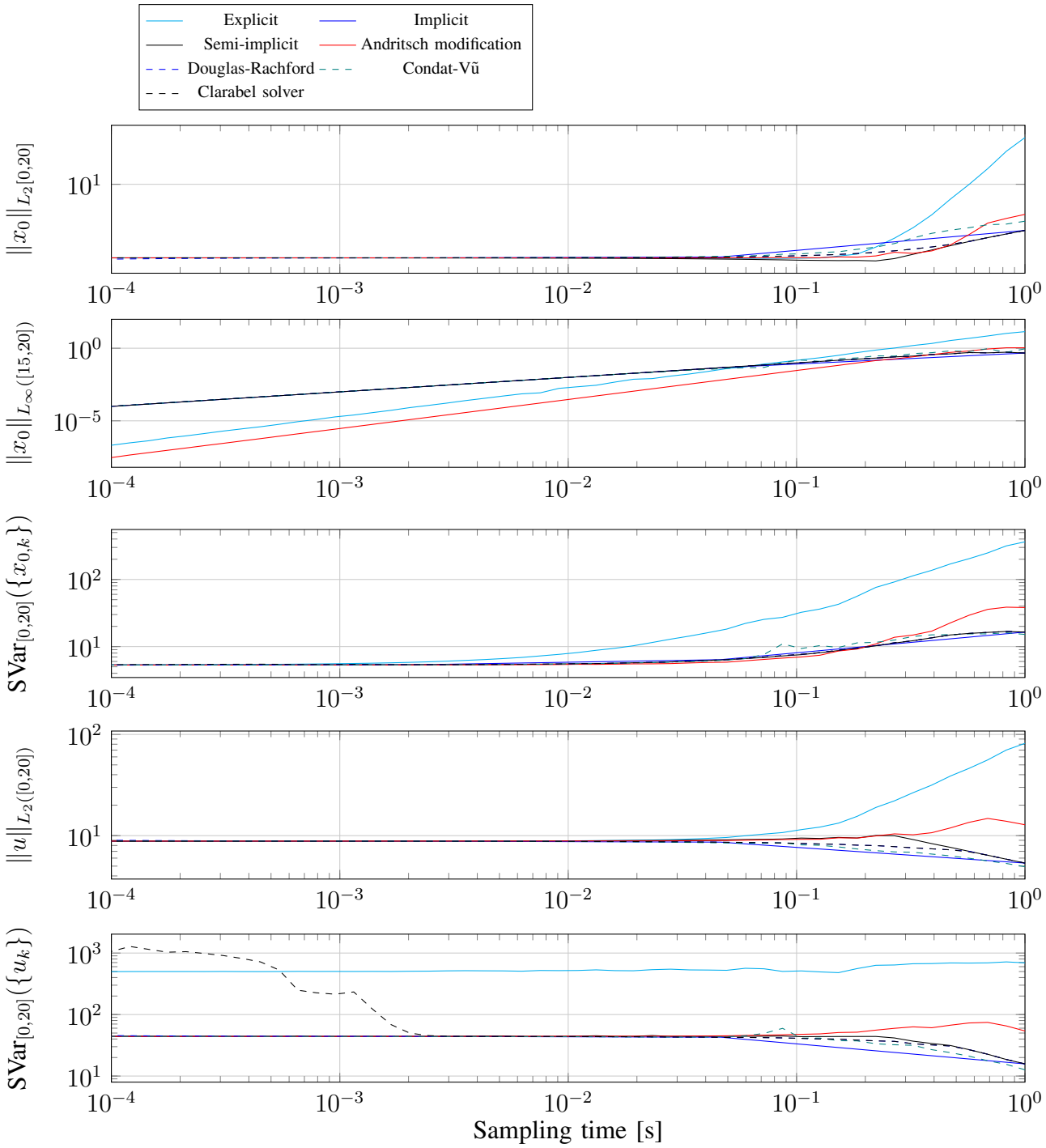

Figure S6: Norms and total variations of state and control signals for several discrete-time super-twisting controllers. In terms of the L_2-norm of the state x_0 and the control input u, the implicit controller (194) (and its associated splittings); the semi-implicit controller (195); and the numerical solver for (197), have a similar behavior for all sampling times. In terms of the step-to-step variation (198) 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 (196) shows the best performance for $h \leq 200ms$.

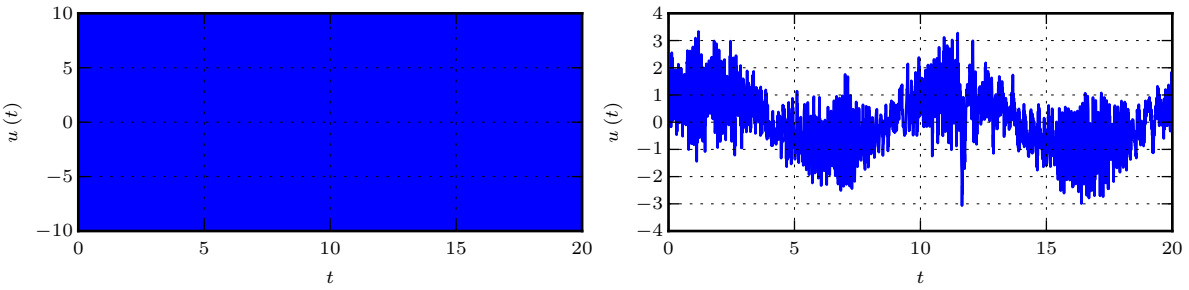

Figure S7: Typical control inputs for explicit (above) and implicit (below) sliding mode controllers, see [70].