Experimental results on implicit and explicit time-discretization of equivalent-control-based sliding-mode control

Olivier Huber, Bernard Brogliato, Vincent Acary, Ahcene Boubakir, Franck Plestan, Bin Wang

Experimental Results on Implicit and Explicit Time-Discretization of Equivalent-Control-Based Sliding-Mode Control

Olivier Huber, Bernard Brogliato, Vincent Acary, Ahcene Boubakir, Franck Plestan, Bin Wang

This chapter presents a set of experimental results concerning the sliding mode control of an electro-pneumatic system. The controller is implemented *via* a micro-processor as a discrete-time input. Three discrete-time control strategies are considered for the implementation of the discontinuous part of the sliding mode controller: explicit discretizations with and without saturation, and an implicit discretization (that is very easy to implement as a projection on the interval $[-1,1]$). While the explicit implementation is known to generate numerical chattering, the implicit one is expected to significantly reduce chattering while keeping the accuracy. The experimental results reported in this work remarkably confirm that the implicit discrete-time sliding mode supersedes the explicit ones, with several important features: chattering in the control input is almost eliminated (while the explicit and saturated controllers behave like high-frequency bang-bang inputs), the input magnitude depends only on the perturbation size and is "independent" of the controller gain and sampling time. On the contrary the explicit controller shows obvious chattering for all sampling times, its magnitude increases as the controller gain increases, and it does not reduce when the sampling period augments. The tracking errors are comparable for both methods, though the implicit method keeps the precision when the control gain increases, which is not the case for the explicit one. Introducing a saturation in the explicit controller does not allow to significantly improve the explicit controller behaviour if one does not take care of the saturation width.

1.1 Introduction

Sliding-mode control has very attractive features like robustness and simplicity of implementation, with few gains to tune. Its main drawback is the existence of the so-called chattering phenomenon, which may be due to actuators limitations, unmodelled dynamics, or time-discretization. Several works recently focussed on the time-discretization effects, showing that an *explicit* implementation

of either Euler of ZOH discretizations yields limit cycles [9, 10, 30, 32], while the *implicit* form suppresses, in theory, the numerical chattering [1, 2, 31] due to the time–discretization. The goal of this chapter is to show that the superiority of the implicit discretization is verified in practice. Note that even if in the following the sliding variable is always scalar (e.g. the sliding surface co-dimension is one), the implicit method works with an arbitrary number of sliding surfaces. This is in sharp contrast with the previous approaches. Before going further let us briefly recall what is meant by explicit and implicit discrete-time sliding mode controllers.

Explicit vs *implicit discrete sliding mode control:*

To illustrate the difference between the explicit and the implicit discretizations, we consider the scalar system $\dot{x}(t) = u(t) + d(t)$, with $u(t) \in -\mathrm{sgn}(x(t))$, where $\mathrm{sgn}(\cdot)$ is the set-valued signum function: $\mathrm{sgn}(0) = [-1,1]$, $\mathrm{sgn}(x) = \{1\}$ if $x > 0$, and $\mathrm{sgn}(x) = \{-1\}$ if $x < 0$. Let the disturbance $d(t)$ satisfy $|d(t)| \leq \delta < 1$ for some δ. Recall that using Filippov's mathematical framework of differential inclusions [8], one deduces that for any $x(0)$, the state $x(t)$ reaches the "sliding surface" $x = 0$ in a finite time t^*, and then $x(t) = 0$ for all $t \geq t^*$. Using terminology from the differential inclusions literature, $u(t)$ is defined from $\xi(t)$, a *selection* of sgn(0) (the interval $[-1,1]$) for $t \geq t^*$, and it satisfies $\xi(t) = u(t) = -d(t)$ after t^*. In a sense, the set-valued controller acts as a disturbance observer once the sliding mode is attained. It is clear that if one multiplies the signum by a gain $a > 0$, *i.e.* $u(t) \in -a\,\mathrm{sgn}(x(t))$, then one still has $u(t) = -d(t)$ in the sliding phase after t^*. However this time the value of the selection $\xi(t)$ inside the set-valued part of $\mathrm{sgn}(x(t))$ is divided by a, *i.e.* $\xi(t) = -\frac{d(t)}{a}$.

Let us now consider the Euler discretization of this system. It reads: $x_{k+1} = x_k + hu_k + hd_k$, where $f_k = f(t_k)$ for a function $f(\cdot)$, and $t_k = t_0 + kh$, $k \in \mathbb{N}$, are the sampling times, $h > 0$ is the sampling period. In such a simple case, the Euler and ZOH discretizations are the same, except for the disturbance $d_k = \int_{t_k}^{t_{k+1}} d(t)\,\mathrm{d}t$ for the ZOH method. Our focus is on how to choose u_k. The explicit method yields $u_k \in -\mathrm{sgn}(x_k)$, yielding the closed-loop $x_{k+1} - x_k - hd_k \in -h\,\mathrm{sgn}(x_k)$. As alluded to above, limit cycles exist and create oscillations around the sliding surface (here the origin), known as the *numerical chattering in the output*. One of the consequences is that the explicit controller keeps switching between the two values 1 and -1, and never attains any point in $(-1,1)$. In particular the explicit controller cannot approximate the continuous-time selection $\xi(\cdot) = u(\cdot)$ when the system evolves close to the sliding surface. If a gain $a > 0$ premultiplies $u(\cdot)$ then the explicit controller switches between two discrete values a and $-a$, the switching frequency being inversely proportional to the sampling period: this is the *numerical chattering in the input*. It is noteworthy that the mere notion of a sliding surface does not exist in this case, since the discrete trajectories cannot attain the origin, and the controller cannot take values in the set-valued part equal to $(-1,1)$. One then has to resort to so-called quasi-sliding surfaces [27].

The implicit method is implemented as follows. Since $d(t)$ is unknown, one first constructs a *nominal unperturbed system* with state $\tilde{x}_k$, from which the input is computed: $\tilde{x}_{k+1} = x_k + hu_k$, $u_k \in -\mathrm{sgn}(\tilde{x}_{k+1})$. This is a so-called *generalized equation* with unknown $\tilde{x}_{k+1}$. Its solution yields after few manipulations $u_k = \mathrm{proj}\left([-1,1]; -\frac{x_k}{h}\right)$, that is the projection on the interval $[-1,1]$ of $-\frac{x_k}{h}$,

and is a causal input (not depending on any future values of the state). In such a simple case, the closed-loop expression of u_k is the same as the saturation function with a width h^{-1}. However in case of sliding surfaces of higher co-dimension, such an analogy with the well-known saturated sign function, is not trivial. Notice that in the unperturbed case, $\tilde{x}_k$ and x_k are the same. As proved in [1, 2], the implicit controller guarantees convergence of $\tilde{x}_k$ to the origin in a finite number of steps (the chattering is suppressed), and a disturbance attenuation by a factor h during the sliding mode (defined here from the fact that $u_k \in (-1,1)$). Most importantly, the control input takes values in $(-1,1)$ once $\tilde{x}_k$ has reached the origin, as may be seen from the generalized equation from which it is calculated, and one has during that phase $u_k = -d_k$: u_k is a selection τ_{k+1} of the discrete-time differential inclusion $\tilde{x}_{k+1} = x_k + hu_k$, $u_k \in -\mathrm{sgn}(\tilde{x}_{k+1})$, and the discrete-time input observes the disturbance when the sliding mode is attained. Similarly to the continuous-time case, if the controller is multiplied by a gain $a > 0$, then the selection $\tau_k = \frac{-d_k}{a}$. For the sake of completeness of this chapter, let us reproduce here one of the results in [2]. Let us start by considering the following scalar sliding mode system:

$$\begin{cases} \dot{x}(t) = -a\tau(t) + d(t) \\ \tau(t) \in \mathrm{sgn}(x(t)), \end{cases} \tag{1.1}$$

where $d(\cdot)$ is a Lebesgue measurable perturbation such that $\|d\|_\infty < \rho < a$. The control input is here $u(t) = \tau(t)$. The discrete-time sliding mode system is implemented as follows:

$$\begin{cases} \tilde{x}_{k+1} = x_k - ah\tau_{k+1} \\ \tau_{k+1} \in \mathrm{sgn}(\tilde{x}_{k+1}) \\ x_{k+1} = x_k - ah\tau_{k+1} + hd_k \end{cases} \tag{1.2}$$

The first two lines of (1.9) may be considered as the *nominal unperturbed* plant, from which one computes the input at time t_k. The input is said implicit since it involves $\tilde{x}_{k+1}$ in the sign multifunction. It is however a causal input as shown next, and $\tilde{x}_{k+1}$ is just an intermediate variable which does not explicitly enter into the controller. The third line is the Euler approximation of the plant, on which the disturbance is acting. One has $u(t) = \tau_{k+1}$ on the time-interval $[t_k, t_{k+1})$.

Proposition 1.1.1. *Let x_0 be the given initial state. Then after a finite number of steps k_0 one obtains that $\tilde{x}_k = 0$ and $x_k = hd_{k-1}$ for all $k > k_0 > 0$. In other words, the disturbance is attenuated by a factor h. Moreover the approximated derivative of the state satisfies $\frac{x_{k+1}-x_k}{h} = d_k - d_{k-1}$ for all $k > k_0 + 1$ whereas $\frac{\tilde{x}_{k+1}-\tilde{x}_k}{h} = 0$ for all $k > k_0$. The control input takes values inside the sign multifunction multivalued part on the sliding surface for all $k > k_0$.*

Proof: Let us start with the case $|x_0| > ah > 0$. The generalized equation $\tilde{x}_{k+1} = x_k - ah\tau_{k+1}$ and $\tau_{k+1} \in \mathrm{sgn}(\tilde{x}_{k+1})$ is equivalent, using the material in the Appendix .1, to the inclusion $\tau_{k+1} - \frac{x_k}{ah} \in -N_{[-1,1]}(\tau_{k+1})$, and to $\tau_{k+1} = \mathrm{proj}([-1,1]; \frac{x_k}{ah})$. Thus one obtains:

- If $x_k > ah$ then $\tilde{x}_{k+1} = x_k - ah$ and $\mathrm{sgn}(\tilde{x}_{k+1}) = 1$,

- If $x_k < -ah$ then $\tilde{x}_{k+1} = x_k + ah$ and $\mathrm{sgn}(\tilde{x}_{k+1}) = -1$,
- If $x_k \in [-ah, ah]$ then $\tilde{x}_{k+1} = 0$ and $\mathrm{sgn}(\tilde{x}_{k+1}) \in [-1, 1]$[1].

From the above we infer that

- If $x_k > ah$ then $x_{k+1} = x_k + hd_k - ah = x_k + h(d_k - a) < x_k + h(\rho - a)$. Since $\rho - a < 0$ the state is strictly decreased from step k to step $k+1$.
- If $x_k < -ah$ then $x_{k+1} = x_k + hd_k + ah = x_k + h(d_k + a) > x_k + h(a - \rho)$. Since $a - \rho > 0$ the state is strictly increased from step k to step $k+1$.

One deduces that if the initial data satisfies $|x_0| > ah$ then after $k_0 = \lceil \frac{x_0}{h|a-\rho|} \rceil$ steps one gets $\tilde{x}_{k_0} = 0$, where $\lceil \cdot \rceil$ is the ceiling function. Indeed at k_0 the state x_k reaches the interval $[-ah, ah]$ and then the unique solution for $\tilde{x}_k$ is zero. From $\tilde{x}_{k_0} = 0$ one deduces that $|x_{k_0}| < ah$. In the case that $|x_0| \leq ah$, it is easy to check that $k_0 = 1$. Indeed one has to solve the generalized equation

$$\begin{cases} \tilde{x}_{k_0+1} = x_{k_0} - ah\tau_{k_0+1} \\ \tau_{k_0+1} \in \mathrm{sgn}(\tilde{x}_{k_0+1}), \end{cases} \tag{1.3}$$

whose unique solution is found by inspection to be $\tilde{x}_{k_0+1} = 0$ [2]. The reasoning can be repeated to conclude that $\tilde{x}_k = 0$ for all $k \geq k_0$. Therefore $\frac{\tilde{x}_{k+1} - \tilde{x}_k}{h} = 0$ for all $k > k_0$. Now let us assume that for $k \geq k_0$ we have

$$\tilde{x}_{k+1} = x_k - ah\tau_{k+1} = 0, \quad k \geq k_0, \tag{1.4}$$

that is $\tau_{k+1} = \frac{x_k}{ha}$. In this case, the state x_{k+1} is given by $x_{k+1} = hd_k$, and therefore $x_k = hd_k, \tau_{k+1} = \frac{d_k}{a}$ for all $k \geq k_0 + 1$, so that $\frac{x_{k+1} - x_k}{h} = d_k - d_{k-1}$ for all $k > k_0 + 1$.

Remark 1.1.1. *The implicit discretization of set-valued sign controllers has been independently introduced in [18] with the so-called proxy-based sliding mode control. The authors noted that the implicit Euler discretization allows one to obtain a perfect (at the machine precision) vanishing of the sliding variable. Related work is also in [12], which however only applies to a simple scalar case. These methods have their roots in the numerical analysis and simulation of mechanical systems with unilateral constraints, impacts and Coulomb friction [21, 22] and of linear complementarity systems for switched circuits with ideal diodes [5, 20, 4].*

Therefore the implicit controller has the same features as its continuous-time counterpart. We may summarize them as follows:

1. When there is no perturbation, the sliding surface is reached after a finite number of steps and there is no chattering.

[1] This replaces the third and fourth items in the proof of Proposition 1 in [2], which contains an unfortunate error.

[2] The underlying crucial property that makes this hold is the maximal monotonicity of the sign multifunction.

2. When a perturbation acts on the system, the state of the nominal system reaches the sliding surface after a finite number of steps, while the perturbation effect is attenuated by a factor h on the state of the system.

3. Despite the system's state x_k never attains its sliding surface due to the disturbance, the notion of discrete-time sliding mode *does* exist, and corresponds to the nominal system's state $\tilde{x}_k$ vanishing, or equivalently to the set-valued controller evolving strictly inside the interval $[-1,1]$. In this mode the controller compensates for the disturbance, with a delay due to the discretization. Therefore, its magnitude is independent, in the sliding phase, of the controller gain, and there is no need to adapt the gain (denoted as a above, and as G in the sequel) online.

4. Theoretically there is no numerical chattering during the sliding mode, neither in the sliding variable, nor in the input.

5. The discrete-time controller keeps the simplicity of its continuous-time counterpart, with no added gain to tune.

6. Computing the input at each step boils down to solving a simple generalized equation, equivalently a projection on $[-1,1]$ in the codimension one sliding surface case. With more sliding surfaces, we can either compute the control input by enumeration (for a co-dimension up to 3), or solve an optimisation problem like a quadratic program. This is quite easy to implement in a code.

7. The implicit discretization allows to prove Lyapunov stability of the sliding variable dynamics, mimicking the continuous-time Lyapunov functions [15, 16].

The main objective of this work is to confirm these features experimentally, while they have been analysed and numerically illustrated in [1, 2, 13, 15, 14, 16, 17]. Preliminary experimental results were presented in [31].

The chapter is organized as follows: in Section 1.2 the dynamics and the various controllers of the electropneumatic actuators are detailed. Section 1.3 is dedicated to the experimental results: the explicit and the implicit discrete-time algorithms are applied to the system and compared in terms of their overall performance, comprising the tracking accuracy, the input chattering, the input magnitude, the disturbance rejection, when the controller gain and the sampling period are varied. In addition the saturated explicit controller is also tested. Conclusions end the chapter in Section 1.5.

1.2 Dynamics of the plant and controllers

1.2.1 Implicit controller design

To start let us explain in details how the so-called implicit controller (which might be also named the *projected sliding-mode controller*) is calculated in case of tracking of a reference output $y_d(t)$

(only the regulation case has been described above). We consider the same scalar system as in the introduction, *i.e.* $\dot{x}(t) = u$, $y = x$, and first we disregard the disturbance for the sake of simplicity. In this case, the Euler and ZOH discretizations are the same. Let the sliding variable be $\sigma = x - x_d$. The controller is set to $u(x,t) \in -\text{sgn}(\sigma) + \dot{x}_d(t)$, so that the closed-loop system is $\dot{\sigma}(t) \in -\text{sgn}(\sigma(t))$. The plant discretization is

$$\frac{x_{k+1} - x_k}{h} = u_k \tag{1.5}$$

and the implicit input is set equal to

$$u_k \in -\text{sgn}(x_{k+1} - x_{d,k+1}) + \frac{x_{d,k+1} - x_{d,k}}{h} \tag{1.6}$$

where the last term accounts for the Euler approximation of $\dot{x}_d(t)$. The discrete-time sliding variable is given by $\sigma_k = x_k - x_{d,k}$. We therefore obtain

$$x_{k+1} - x_k \in -h\,\text{sgn}(\sigma_{k+1}) + x_{d,k+1} - x_{d,k} \Leftrightarrow \sigma_{k+1} - \sigma_k \in -h\,\text{sgn}(\sigma_{k+1}). \tag{1.7}$$

Let $w_{k+1} \stackrel{\Delta}{=} \sigma_{k+1} - \sigma_k$. Using convex analysis we may write $w_{k+1} \in -h\,\text{sgn}(\sigma_{k+1}) \Leftrightarrow \sigma_{k+1} \in -N_{[-h,h]}(w_{k+1})$, where $N_{[-h,h]}(w_{k+1})$ is the normal cone to $[-h,h]$ calculated at $w_{k+1} \in [-h,h]$, given in this case by

$$N_{[-h,h]}(w_{k+1}) = \begin{cases} 0 & \text{if} \quad |w_{k+1}| < h \\ [0,+\infty) & \text{if} \quad w_{k+1} = h \\ (-\infty,0] & \text{if} \quad w_{k+1} = -h. \end{cases} \tag{1.8}$$

Inserting this in (1.7) yields

$$w_{k+1} + \sigma_k \in -N_{[-h,h]}(w_{k+1}) \Leftrightarrow -N_{[-h,h]}(w_{k+1}) - \sigma_k \ni w_{k+1}. \tag{1.9}$$

By using basic convex analysis one finds equivalently

$$w_{k+1} = \text{proj}([-h,h]; -\sigma_k) = h\,\text{proj}([-1,1]; -\frac{\sigma_k}{h}) \tag{1.10}$$

where proj denotes the orthogonal projection (details on the above manipulations can be found in [3, Appendix A], and in Appendix .1). From (1.5), we have

$$u_k = \frac{1}{h} w_{k+1} + \frac{x_{d,k+1} - x_{d,k}}{h}.$$

Thus we obtain

$$\begin{aligned} u_k &= \tfrac{1}{h}\text{proj}([-h,h]; -\sigma_k) + \tfrac{x_{d,k+1} - x_{d,k}}{h} = \text{proj}([-1,1]; -\tfrac{\sigma_k}{h}) + \tfrac{x_{d,k+1} - x_{d,k}}{h} \\ &= \begin{cases} -\frac{\sigma_k}{h} + \frac{x_{d,k+1} - x_{d,k}}{h} & \text{if} \quad |\sigma_k| \le h \\ -1 + \frac{x_{d,k+1} - x_{d,k}}{h} & \text{if} \quad \sigma_k > h \\ 1 + \frac{x_{d,k+1} - x_{d,k}}{h} & \text{if} \quad \sigma_k < -h. \end{cases} \end{aligned} \tag{1.11}$$

The implicit controller is thus bounded whatever the value of the sampling period $h > 0$. It is obviously quite easy to implement in a code. When a disturbance acts on the system $\dot{x}(t) = u + d(t)$, $d(t) \leq \delta < 1$ for some known δ, then a similar procedure applied on a nominal, unperturbed system yields the generalized equation:

$$\begin{cases} \tilde{\sigma}_{k+1} = \sigma_k + hu_k - x_{d,k+1} + x_{d,k} \\ \\ u_k \in -\mathrm{sgn}(\tilde{\sigma}_{k+1}) + \frac{x_{d,k+1} - x_{d,k}}{h} \end{cases} \tag{1.12}$$

from which the controller is calculated again to be $u_k = \mathrm{proj}([-1,1]; -\frac{\sigma_k}{h}) + \frac{x_{d,k+1} - x_{d,k}}{h}$. Following the developments in [2] breifly recalled in the introduction, such a controller guarantees interesting properties of the closed-loop system like disturbance attenuation by a factor h, and *convergence to the nominal state sliding surface* $\tilde{\sigma}_k = 0$ *in a finite number of steps*. Moreover it also has the features listed in 1 to 7 above, and this will be confirmed by the experimental results given in Section 1.3. It is noteworthy that an explicit implementation of the input yields

$$u_k = -\mathrm{sgn}(\sigma_k) + \frac{x_{d,k+1} - x_{d,k}}{h}. \tag{1.13}$$

It is not necessary to write an inclusion $u_k \in -\mathrm{sgn}(\sigma_k) + \frac{x_{d,k+1} - x_{d,k}}{h}$ in (1.13), because the *multivalued part of the sign function cannot be realized with an explicit controller*. Indeed the fact is not only that the zero value does not exist numerically, but even if it did, one would not be able to choose in a unique way the controller value inside $[-1,1]$ (in numerical analysis of differential inclusions, this is known as the *selection procedure* [3, §9.2, 9.4]). On the contrary the implicit implementation *does* realize the set-valuedness of the input. Moreover the computed controller value is the unique selection of the discrete-time inclusion, as a result of solving the above generalized equation.

Remark 1.2.1 (Chattering). *The oscillations around the sliding surface which are solely due to the time discretization, are known as the* output numerical chattering. *They have been analysed with explicit discretizations in [9, 10, 32, 30]. The input numerical chattering is focussed on in this work.*

Remark 1.2.2 (Controller commutations). *It is widely accepted in the Control literature that sliding mode controllers have to be implemented through infinite-frequency commutations of some actuator, and that the infinitely fast switching strategy is necessary to approximate the continuous-time solution obtained from Filippov's mathematical framework. This is false when one considers the implicit implementation which is briefly summarized above. The implicit implementation takes the form of a projection onto a finite interval (a hypercube if there is more than one switching surface) as shown in (1.11), hence it is a Lipschitz continuous function of the state. As it will be shown in Section 1.3, the* input numerical chattering *is drastically reduced when the implicit controller is used. It is quite possible that the input shape may be in practice as important as the output shape, because input chattering is known to demand a lot of the actuators.*

Remark 1.2.3 (Sliding surface with codimension $m \geq 2$). *The above implicit discretization method extends to the case when more than one attractive sliding surface is designed: then one computes*

the controller by solving a generalized equation that extends (1.12) [2, Equation (15)]. Its well-posedness relies on CB (or a discretized version of it, if the zero-order-hold method is used [2, 15, 17]) being a P-matrix. However CB needs not be diagonal: couplings between the sliding surfaces are allowed. *It also extends to the twisting controller [14, 17, 15], and a preliminary analysis for the super-twisting algorithm can be found in [2, Lemmas 5 and 6]. As shown above, in case $m = 1$, the implicit discretization gives a saturated discrete-time input, with a specific saturation width. In higher codimensions, such an interpretation is not trivial, however one may say that the implicit discretization provides a suitable regularization of the controller which allows to almost suppress chattering, to guarantee some Lyapunov stability as well as finite-time convergence properties [16]: this proves that it is a correct approximation of the set-valued continuous-time controller.*

1.2.2 Plant dynamics and controllers

The electropneumatic system used for the controllers evaluation consists in two actuators which are controlled by two servodistributors (see Figure 1.1). Each actuator is composed by two chambers denoted by P (positive) and N (negative). The controllers proposed in the sequel are designed in order to control the position of one of these two actuators, named "Main actuator", whereas the second actuator, named "Perturbation actuator" and mechanically connected to the Main one, is used in order to produce an external perturbation force. With a nominal 7-bar source pressure, the maximum produced force is 2720 N; furthermore, both actuators have the same physical features: piston diameter is 80 *mm* and rod diameter 25 *mm*. The external perturbation force controller is not under interest in this chapter and has been designed and tuned by Sitia Co. (`http://www.sitia.fr/`) which has built the experimental set-up. The air mass flow rates q_m entering in the chambers are modulated by two three-way servodistributors. The pneumatic jack horizontally moves a load carriage of mass M.

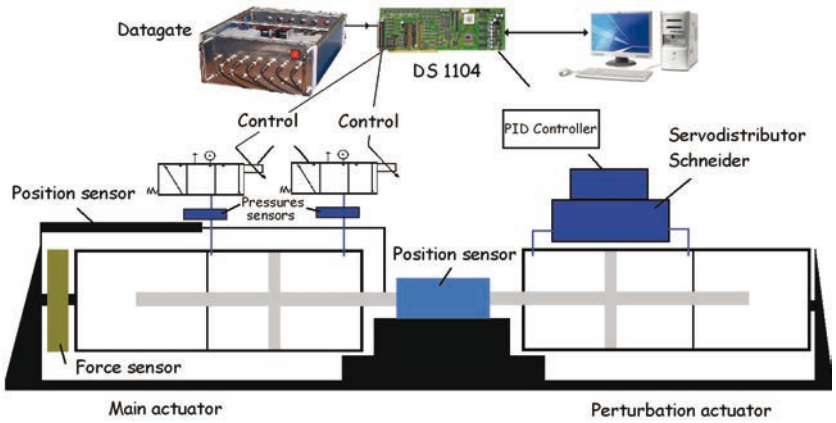


Figure 1.1: [26] Photography and scheme of the electropneumatic system.

Under some assumptions detailed in [26], the dynamic model of the pneumatic actuator can be written as a nonlinear system which is affine in the control input $[u_P\ u_N]^T$, u_P (resp. u_N) being the control input of the servodistributor connected to the P (resp. N) chamber. The model is divided in two parts: two first equations concern the pressure dynamics in each chamber whereas the motion of the actuator is described by the two last equations. Then the model of the electropneumatic experimental set-up reads as

$$\begin{cases} \dot{p}_P &= \dfrac{krT}{V_P(y)}[\varphi_P + \psi_P \cdot u_P - \dfrac{S}{rT} p_P v] \\ \dot{p}_N &= \dfrac{krT}{V_N(y)}[\varphi_N + \psi_N \cdot u_N + \dfrac{S}{rT} p_N v] \\ \dot{v} &= \dfrac{1}{M}[S(p_P - p_N) - b_v v - F] \\ \dot{y} &= v, \end{cases} \tag{1.14}$$

with p_P (reps. p_N) the pressure in the P (resp. N) chamber, y and v being the position and velocity of the actuator. The force F is a disturbance that takes into account dry friction and unknown external forces. Note that the previous system appears to have two control inputs given that there is one servo distributor connected to each chamber. In the sequel, only the main actuator position is controlled: given that there is a single control objective, one states[3]:

$$u = u_P = -u_N.$$

The constant k is the polytropic constant, r is the ideal gas constant, T is the temperature which is supposed the same inside or outside the chambers and b_v is the viscous friction constant. V_P and V_N are the volumes in both chambers. These volumes depend on the actuator position y. S is the piston section and is constant. Finally, φ_X and ψ_X (X being P or N) are both 5^{th} order polynomial functions versus p_X [25] and allow to model the mass flow rate q_X in the chamber X such that

$$q_X = \varphi_X(p_X) + \psi_X(p_X)u_X. \tag{1.15}$$

This kind of system is uncertain: in fact, the uncertainties on the polytropic constant, on the mass flow, on the temperature, on the mass, on the viscous friction coefficient and on the disturbance force can be modeled by additive bounded functions added to the nominal part of each parameter [11]. As an example, the mass M can be viewed as the sum of a nominal part and an uncertain one

$$M = M_n + \Delta M,$$

[3]Multivariable control can be designed [11], in case of position and pressure (in a chamber) control; an advantage of control pressure is that the rigidity of the actuator is improved.

with ΔM a bounded uncertainty and M_n the nominal value. By considering that the system (1.14) with a single input reads as $\dot{x} = f(x) + g(x)u$ with $f(\cdot)$ and $g(\cdot)$ uncertain vector fields defined as

$$f(x) = \begin{bmatrix} \frac{krT}{V_P(y)}[\varphi_P - \frac{S}{rT}p_P v] \\ \frac{krT}{V_N(y)}[\varphi_N + \frac{S}{rT}p_N v] \\ \frac{1}{M}[S(p_P - p_N) - b_v v - F] \\ v \end{bmatrix}, \quad g(x) = \begin{bmatrix} \frac{krT}{V_P(y)}\psi_P \\ -\frac{krT}{V_N(y)}\psi_N \\ 0 \\ 0 \end{bmatrix}, \tag{1.16}$$

it may be rewritten as

$$\dot{x} = (f_n + \Delta f)(x) + (g_n + \Delta g)(x)\, u \tag{1.17}$$

with $f_n(\cdot)$, $g_n(\cdot)$ the *nominal* dynamics parts of $f(\cdot)$ and $g(\cdot)$, and Δf, Δg the *uncertainties* and *perturbations*. Due to the presence of uncertainties and perturbations, a robust controller is required in order to get high performances (in terms of accuracy, for example). The choice of sliding mode controller [28, 29] has been made because of its intrinsic features of robustness. Let us define the so-called *sliding variable* as

$$\sigma(x,t) = \ddot{e} + \lambda_1 \dot{e} + \lambda_0 e \tag{1.18}$$

with $e \stackrel{\Delta}{=} y - y_d(t)$, $y_d(t)$ being the desired trajectory, supposed to be sufficiently differentiable. The coefficients λ_1, λ_0 are defined such that, given z a complex variable, the polynomial $Q(z) = z^2 + \lambda_1 z + \lambda_0$ is Hurwitz. The first and second derivatives of e are computed by direct numerical differentiation with appropriate first-order filters (see remark 1.3.2 on the influence of those filters on the closed-loop behaviour). The idea of the continuous-time sliding mode controller is the following: the control ensures, in spite of uncertainties and perturbations and thanks to a discontinuous term, the finite time convergence to the so-called *sliding surface* $\sigma = 0$ (if the controller is well-tuned). Once the system trajectories have reached this domain, they are evolving on it and the closed-loop system dynamics is governed by the definition of σ, *i.e.* when $\sigma = 0$, one has $\ddot{e} = -\lambda_1 \dot{e} - \lambda_0 e$ which ensures exponential convergence to $(e, \dot{e}) = (0,0)$. Note that once $\sigma = 0$, this convergence is not influenced by the uncertainties or perturbations. One gets

$$\begin{aligned} \dot{\sigma} &= e^{(3)} + \lambda_1 \ddot{e} + \lambda_0 \dot{e} \\ &= \frac{1}{M}\left[S(\dot{p}_P - \dot{p}_N) - b_v \dot{v} - \dot{F}\right] - y_d^{(3)}(t) + \frac{\lambda_1}{M}\left[S(p_P - p_N) - b_v v - F\right] - \lambda_1 \ddot{y}_d(t) \\ &\quad + \lambda_0 \left(\dot{y} - \dot{y}_d(t)\right), \end{aligned} \tag{1.19}$$

where we assumed that the disturbance $F(\cdot)$ is differentiable, for simplicity (rigorously, dry friction may introduce some non differentiability at zero relative tangential velocity, depending on the used

model). As shown in [19, 11] and given the system (1.17), the first time derivative of σ in (1.19) can be written as

$$\begin{aligned} \dot{\sigma} &= \Psi(x,t) + \Phi(x)u \\ &= \Psi_n(x,t) + \Delta\Psi(t) + [\Phi_n(x) + \Delta\Phi(t)]\ u \end{aligned} \tag{1.20}$$

such that Ψ_n, Φ_n are the nominal functions and $\Delta\Psi, \Delta\Phi$ are the uncertain terms. From [19, 11], the functions Ψ and Φ are bounded in the physical working domain (which gives that the uncertain terms are also bounded). Furthermore, one supposes that $\Delta\Phi$ is sufficiently small with respect to Φ_n to ensure that $1 + \frac{\Delta\Phi}{\Phi_n} > 0$. From a practical point of view, this assumption is not too strong: it simply means that the uncertainties are small compared to the nominal values. Let us consider the control law[4]:

$$u = \frac{1}{\Phi_n}\left[-\Psi_n + v\right]. \tag{1.21}$$

By applying (1.21) in (1.20), one gets

$$\dot{\sigma} = \frac{\Delta\Phi}{\Phi_n}\Psi_n + \Delta\Psi + \left[1 + \frac{\Delta\Phi}{\Phi_n}\right] v. \tag{1.22}$$

The controller v is a set-valued input defined as

$$v \in -G\,\mathrm{sgn}(\sigma) \tag{1.23}$$

with G tuned sufficiently large[5] to ensure the sliding condition [28, 29] $\sigma\dot{\sigma} \leq -\eta|\sigma|$ ($\eta > 0$). Each controller has been implemented under its discrete forms as follows (with $k \geq 0$, $\sigma_k = \sigma(kh)$, h being the sampling period)

- Explicit sliding mode control (with $\mathrm{sgn}(\cdot)$ function)

$$v_k = -G\,\mathrm{sgn}(\sigma_k), \tag{1.24}$$

- Explicit saturated sliding mode control (with $\mathrm{sat}(\cdot)$ function)

$$v_k = -G\,\mathrm{sat}(\sigma_k, \varepsilon), \tag{1.25}$$

[4] As shown in [6], such a control law allows to reduce the magnitude of the sliding mode controller by using the nominal informations in the controller.

[5] Following the sliding condition, the gain has to be tuned as $G > \dfrac{\mathrm{Max}\left|\dfrac{\Delta\Phi}{\Phi_n}\Psi_n + \Delta\Psi\right| + \eta}{\min\left[1 + \dfrac{\Delta\Phi}{\Phi_n}\right]}$. By a similar way than [23], it can be shown that, over the trajectories and in the working domain, the term $\dfrac{\Delta\Phi}{\Phi_n}\Psi_n + \Delta\Psi$ is bounded whenever $1 + \dfrac{\Delta\Phi}{\Phi_n} > 0$.

with

$$\text{sat}(\sigma_k, \varepsilon) \quad = \quad \begin{cases} \text{sgn}(\sigma_k) \text{ if } |\sigma_k| \geq \varepsilon \\ \sigma_k \text{ if } |\sigma_k| < \varepsilon. \end{cases} \tag{1.26}$$

- Implicit sliding mode control (with $\text{sgn}(\cdot)$ multifunction)

$$v_k \in -G\text{sgn}(\sigma_{k+1}) \tag{1.27}$$

(implemented with a projection as indicated in Section 1.2.1).

1.3 Experimental results

This section is devoted to analyze the experimental data. The controllers have been implemented with three feedback gains $G = 10^4$, $G = 10^5$, $G = 10^6$ and five sampling times 1 ms, 2 ms, 5 ms, 10 ms and 15 ms. The length of the interval of study is 20 seconds. The saturation input has been tested for six different values of the saturation width, with the sampling time $h = 1$ ms. In the data reported below, the unitless width of the saturation is $\varepsilon = 0.1$ (the other widths which have been tested yielded similar results and the results obtained with them are therefore omitted). The comparisons are mainly made with respect to: the magnitude and chattering of the inputs u and v, and the tracking error e.

1.3.1 Comparison of the tracking errors e

Data in Tables 1.1–1.3 characterise the position tracking error e obtained by the three different implementation methods, from the aspects of average, range, standard deviation and variation with four different sampling periods. The symbol Avg denotes the average of the tracking error over the duration of the test, abs is the absolute value of tracking error. The total variation of a real-valued function $f(\cdot)$ defined on an interval $[a,b] \subset R$ is approximated by the quantity

$$Var_{[a,b]}(f) = \sum_{i=0}^{N-1} |f(t_{i+1}) - f(t_i)|, \tag{1.28}$$

where the set of time instants $\{t_0, t_1, \cdots, t_N\}$ is a partition of $[a,b]$. In the following, the variations of the position error e for the three different implementation methods with the different gains G, have been calculated by choosing the partition times t_i in (1.28), as the sampling times.

Remark 1.3.1. *The variation in (1.28) as a quantity to characterise the analyzed signals, is not common in Control Engineering. It is thought here in the context of sliding mode control, that such a quantity is useful to measure the chattering level of a signal, since it does represent how much the signal varies. However due to the partition that has been chosen (the sampling times) the results are*

not comparable from one sampling period to the next, but only between the three controllers for a fixed h. In other words, in Table 1.3 data have to be compared inside a single column, but not from one column to another one.

We first compare the controllers performance in terms of the tracking error, for two gains $G = 10^4$ and $G = 10^5$. All the data concerning e are reported in Tables 1.1, 1.2, 1.3 and on Figures 1.2 and 1.3. Table 1.1 data and Figures 1.2(a), (b), (c) show that when $G = 10^4$ and $h = 2$ or 5 ms, the implicit method does not bring any improvement over the explicit ones, but has lower precision capabilities for small time steps. It is only for the larger time step $h = 15$ ms that the results for the implicit controller (Table 1.1 last column) become the same as those of the other two controllers. However it is confirmed in Table 1.3 (a), that the variation of the implicit input starts to be significantly smaller than that of the other two, for $h \geq 5$ ms, the improvement being huge for $h = 15$ ms. These first data tend to indicate that, in the case of the implicit input, its variation is drastically smaller for larger sampling periods (for $h = 15$ ms: 1.4462×10^3 for the explicit method, 196 for the implicit one with $G = 10^4$), confirming that chattering on e is reduced when the implicit controller (1.27) is used. The fact that the output signal is smooth for the implicit method, while it chatters for the other two controllers for large sampling time, is obvious in Figures 1.2(d), (e) and (f).

Table 1.2 concerns $G = 10^5$, that is the gain is now multiplied by 10. All three methods show similar results in terms of average, range and standard deviation of e, the implicit one providing slightly better results. One infers that augmenting the gain G from 10^4 to 10^5 allows to significantly improve the tracking performance of the implicit control (1.21) (1.27) compared to that of the explicit inputs, especially in terms of the variation which is a good quantification of the chattering. In fact, comparing Table 1.1 (c) and Table 1.2 (c), one sees that the performance of the implicit input is almost unchanged when the gain is multiplied by 10, which is not the case of the other two methods: for these both latter, e is clearly increased. We shall observe this insensitivity property of the implicit method, again in Section 1.3.2. In addition the output produced by the implicit method is smoothed, even for small $h = 1$ ms, see Figure 1.3(a) (b) and (c), or (d) (e) and (f).

The variation values are given in Table 1.3 (b) with $G = 10^5$, and is quite visible in Figures 1.3(d) (e) and (f): the variation of e with the implicit input is much smaller than with the other two controllers, except for $h = 1$ ms where the obtained values are of same order. This indicates that the chattering on e is drastically reduced with the implicit input (1.21) (1.27).

$\rightsquigarrow$ *A first conclusion, that will be strengthened in the next paragraph, is that the implicit control method allows to take larger gains without decreasing the performance. This means that it is possible to reject/counteract larger perturbations/uncertainties without more chattering, and makes the process of gain G tuning much easier. The performance of implicit control is better when G is larger, while it is less good with the explicit and saturation controllers.*

h	2ms	5ms	10ms	15ms
Avg(abs(e))	0.32601	0.4791	0.97802	3.8759
Range of e	(-2.0701, 2.0189)	(-3.0067, 2.1572)	(-3.2599, 4.4023)	(-10.3767, 12.1426)
Standard Deviation of e	0.4274	0.6327	1.0366	4.2553

(a) Explicit control

h	2ms	5ms	10ms	15ms
Avg(abs(e))	0.28053	0.51399	0.99017	3.6119
Range of e	(-1.7255, 1.1288)	(-1.7006, 2.4793)	(-4.5846, 2.6004)	(-14.4069, 12.1394)
Standard Deviation of e	0.3319	0.6132	1.1394	4.4131

(b) Saturation control

h	2ms	5ms	10ms	15ms
Avg(abs(e))	0.71254	1.7138	3.3861	5.1387
Range of e	(-1.6760, 1.2200)	(-3.4182, 3.2213)	(-7.9230, 6.4083)	(-9.4997, 6.5194)
Standard Deviation of e	0.7731	1.8780	3.7182	5.4749

(c) Implicit Control

Table 1.1: Comparisons of position error e when $G = 10^4$.

1.3.2 Comparison of control inputs u (1.21) and v (1.24) (1.25) (1.27)

The features of the control inputs is a key-point in this work, given that one of the objectives is to show the influence of implicit control to the chattering effect. Let us now pass to the control inputs comparisons, with data reported in Tables 1.4–1.7 and on Figures 1.4 and 1.5. Data given in Tables 1.4 and 1.5 characterise the "switching functions" for these three methods by considering the range and variation for each of them. Remark 1.3.1 applies also for the variation of the control, so that in Tables 1.4 (b), 1.5 (b), 1.6 (b) and 1.7 (b), data have to be compared inside a single column, but not from one column to another one.

$\rightsquigarrow$ *What we call the* switching functions *are* $sgn(\sigma_k)$ *in (1.24),* $sat(\sigma_k, \varepsilon)$ *in (1.25), and the selection (remember* ξ *in the introduction) of* $sgn(\sigma_{k+1})$ *in (1.27). For the implicit controller, this*

h	2ms	5ms	10ms	15ms
Avg(abs(e))	0.98336	1.1363	2.4372	5.5254
Range of e	(-4.3911, 3.9936)	(-4.7722, 3.9665)	(-11.3641, 6.6129)	(-17.7670, 19.0185)
Standard Deviation of e	1.2430	1.3412	2.8063	6.4330

(a) Explicit control

h	2ms	5ms	10ms	15ms
Avg(abs(e))	1.2502	1.7987	4.4362	5.4374
Range of e	(-4.2085, 4.9032)	(-2.3505, 7.6094)	(-4.7248, 14.8659)	(-11.9105, 19.3981)
Standard Deviation of e	1.5220	1.4996	2.8328	6.6223

(b) Saturation control

h	2ms	5ms	10ms	15ms
Avg(abs(e))	0.72598	1.7017	3.2844	5.0835
Range of e	(-1.8663, 2.3094)	(-5.8677, 4.6001)	(-8.1843, 6.3261)	(-9.2313, 8.1833)
Standard Deviation of e	0.7941	1.9237	3.5816	5.4152

(c) Implicit control

Table 1.2: Comparisons of position error e when $G = 10^5$.

is what we called the selection ξ_k in Introduction. This is not to be confused with the discontinuous control v in (1.23).

Comparisons of the inputs u in three methods are given in Tables 1.6 and 1.7 from three aspects, that is: range, variation, and standard deviation. In addition, the three controllers are depicted on Figures 1.4 and 1.5, for various time steps and gains.

Globally, the experimental results show that the implicit method drastically reduces the input chattering and magnitude compared with the other two methods. The explicit and saturation switching inputs keep oscillating between the maximum and minimum values like a bang-bang controller (see data in Tables 1.4(a) and 1.5(a), and Figures 1.4(a) (b)). This results in a large amplitude of inputs u as well (see Tables 1.6(a) and 1.7(a), as well as Figures 1.5(a) 1.5(b) 1.5(c) 1.5(f) 1.5(g) 1.5(h). Notice that the explicit and saturation inputs behave slightly better when the time step increases. This is visible by comparing Figures 1.5(a) and (f), (b) and (g), (c) and (h) which all concern $h = 2$ ms and $h = 15$ ms, respectively, for both gains G. However the magnitude of the implicit input is far much better in all cases (see Figures 1.5(d) 1.5(e) 1.5(i) 1.5(j).

The magnitude of the ranges of the switching function and control u in the implicit method is much less than the other two methods, see Tables 1.6(a) and 1.7(a). These facts are well supported by Figures 1.4 and 1.5. Consider Figure 1.5: when $h = 15ms$, while the ranges of the control law u in explicit method and explicit saturation method are both between -10 and 10 (see Figures 1.5(h) and 1.5(g)), the range of u for the implicit case is strictly between -2 and 2 (see Figures 1.5(i) and

h	2ms	5ms	10ms	15ms
Explicit	1.7838e+03	904.1336	844.2871	1.4462e+03
Saturation	1.6527e+03	914.4627	838.3387	1.6821e+03
Implicit	1.6452e+03	657.6504	428.0244	196.0669

(a) $G = 10^4$

h	2ms	5ms	10ms	15ms
Explicit	2.5724e+03	1.7742e+03	1.6081e+03	2.5070e+03
Saturation	2.5691e+03	2.0749e+03	2.1638e+03	2.5756e+03
Implicit	1.6360e+03	650.2710	480.1660	228.8022

(b) $G = 10^5$

Table 1.3: Variation of position error e.

h	2ms	5ms	10ms	15ms
Explicit (1.24)	(-1.000, 1.000)	(-1.000, 1.000)	(-1.000, 1.000)	(-1.000, 1.000)
Saturation (1.25)	(-1.000, 1.000)	(-1.000, 1.000)	(-1.000, 1.000)	(-1.000, 1.000)
Implicit (1.27)	(-0.4635, 0.5385)	(-0.3247, 0.3338)	(-0.2969, 0.3117)	(-0.1935, 0.2194)

(a) Range of the switching function.

h	2ms	5ms	10ms	15ms
Explicit (1.24)	6926	2822	2258	1936
Saturation (1.25)	6.6197e+03	2.7224e+03	2.2199e+03	2008
Implicit (1.27)	1.8416e+03	357.9547	211.4038	79.1096

(b) Variation of the switching function.

Table 1.4: Switching function, gain $G = 10^4$.

1.5(j)). The comparison between Figures 1.4(c) through 1.4(j), 1.5(d) and 1.5(e), 1.5(i) 1.5(j), which concern the implicit controller switching function for various gains and sampling times, show that for $h \geq 2$ ms, the implicit input v in (1.27) is largely independent of the gain and sampling time. From Tables 1.4 (a) and 1.5 (a), the data in the rows corresponding to the implicit controller allow to obtain a confirmation of this fact. Furthermore the switching function range for the implicit controller, is divided by ten when the gain G passes from 10^4 to 10^5, which implies that the sliding mode input v_k in (1.27) has a magnitude that does not vary with the gain (recall that what we call the switching function, has to be multiplied by the gain G to obtain the input v). This is in very good agreement with theoretical predictions (item 3) in the introduction). One can also have a look at Tables 1.6 (a) (b) (c), and 1.7 (a) (b) (c), to obtain the same conclusions, that the range (magnitude), the variation and the standard deviation (chattering) of u for (1.27) are drastically smaller than for (1.24) and (1.25). The magnitudes of the switching function for the implicit controller, for 6 different gains G and two different sampling periods h, are reported in Table 1.8. It confirms that the magnitude of the

h	2ms	5ms	10ms	15ms
Explicit (1.24)	(-1.000, 1.000)	(-1.000, 1.000)	(-1.000, 1.000)	(-1.000, 1.000)
Saturation (1.25)	(-1.000, 1.000)	(-1.000, 1.000)	(-1.000, 1.000)	(-1.000, 1.000)
Implicit (1.27)	(-0.0606, 0.0545)	(-0.0360, 0.0417)	(-0.0289, 0.0349)	(-0.0173, 0.0247)

(a) Range of the switching function.

h	2ms	5ms	10ms	15ms
Explicit (1.24)	2980	2050	1932	1836
Saturation (1.25)	2.3486e+03	1.9858e+03	1902	1860
Implicit (1.27)	183.1965	34.7510	25.2005	8.1039

(b) Variation of the switching function.

Table 1.5: Switching function, gain $G = 10^5$.

input v in (1.23), which is the switching function times the gain G, does not depend neither on G nor on h in this range of sampling times (see a comment in remark 1.3.2).

$\rightsquigarrow$ *This insensitivity property is believed to be a fundamental property of the implicit method introduced in [1, 2], compared to explicit implementations which drastically differ when h and/or G are varied.*

The results depicted on Figures 1.4 and 1.5 clearly demonstrate that while the explicit and saturation controllers tend to approximate a signal that switches infinitely fast between two extreme values like bang-bang inputs, this is not at all the case for the implicit controller that behaves in a totally different way. This is a nice confirmation of both theoretical and numerical predictions [1, 2], that the implicit controller does represent the discrete-time approximation of the selection of the differential inclusion according to Filippov's mathematical framework.

Input chattering is also visible in Tables 1.4 (b), 1.5 (b), 1.6 (b) and (c), 1.7 (b) and (c). Variation of the implicit switching function is much smaller than the other two, and standard deviation of u as well. These results demonstrate that the switching function chattering and magnitude, strongly influences the input u in (1.21).

Remark 1.3.2. *All the results tend to show that when the sampling period is too small (typically in our experiments $h = 2$ or $h = 2$ ms), then the implicit controller performance (output precision and chattering, input magnitude and chattering) is decreasing. This is visible on Figure 1.4 with the evolution of the implicit signum function from subfigure (c) to subfigure (g) for gain $G = 10^4$, and from subfigure (h) to subfigure (l) when the gain is $G = 10^5$. In theory the implicit switching function should not vary by changing the sampling period. This phenomenon is due to bandwidth limitations in the first-order filters used to estimate velocities and accelerations from position measurement, in order to calculate the sliding variable in (1.18). It results in a deterioration of the closed-loop performance and controller chattering. Further work on proper tuning of these filters to accommodate for smaller sampling periods h, proves that the performance of the implicit discretization can be further improved for small sampling times.*

h	2ms	5ms	10ms	15ms
Explicit	(-7.8876 8.4594)	(-8.1550 8.6118)	(-8.7349 8.3970)	(-10 10)
Saturation	(-8.0737 8.1963)	(-7.9095 8.0899)	(-8.5541 8.7543)	(-10 10)
Implicit	(-3.2500 3.5871)	(-1.9990 2.6204)	(-1.9399 2.1267)	(-1.8990 1.9484)

(a) range of u.

h	2ms	5ms	10ms	15ms
Explicit	4.1102e+04	1.7731e+04	1.3816e+04	1.2759e+04
Saturation	4.0209e+04	1.6864e+04	1.3838e+04	1.3671e+04
Implicit	9.5190e+03	1.5731e+03	963.2736	609.5058

(b) variation of u.

h	2ms	5ms	10ms	15ms
Explicit	5.7570	5.8144	5.8437	6.3526
Saturation	5.7270	5.6808	5.8811	6.5001
Implicit	1.1183	0.8915	0.8519	0.8650

(c) standard deviation of u.

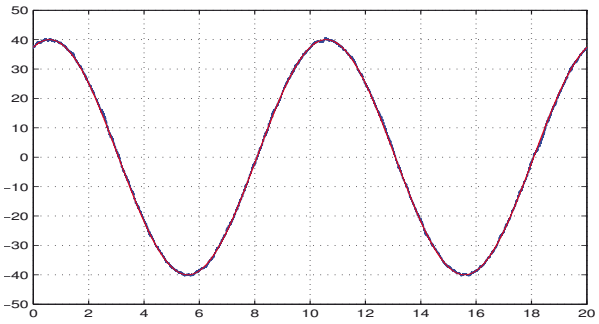
Table 1.6: Comparisons of u when $G = 10^4$.

1.3.3 Summary

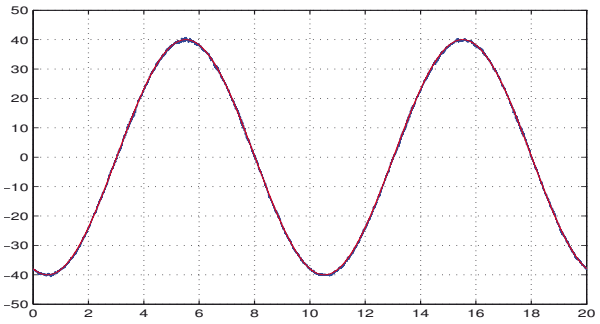
These extensive experimental tests prove that items 3) 4) 5) 6) 7) in the Introduction, are not only theoretical and numerical predictions obtained in [1, 2], but significantly influence the discrete-time implemented sliding-mode controller. The implicit method (1.27) allows to drastically reduce the input chattering and magnitude, while enhancing the tracking capabilities (output chattering is almost entirely eliminated), and simplifying the tuning of the gain G. It also allows the designer to choose larger sampling periods, which may be of strong interest in practice, where one wants to make less calculations to save time and energy. Perhaps counter-intuitively for control engineers, the performance and robustness increase when the gain G increases, which is thought to considerably simplify the controller gain tuning process. A video of the experimental tests is available at `http://nullptr.fr/pages/videos.html`.

1.4 Numerical analysis of the saturation controller

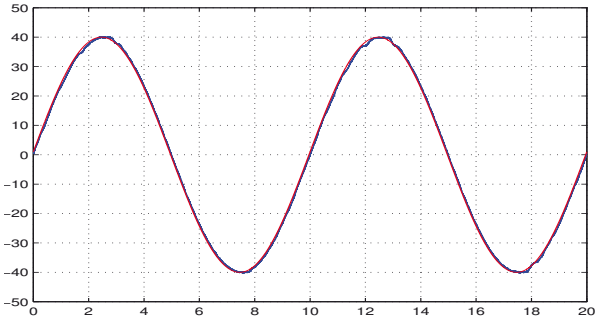
Saturating the set-valued signum function is often presented in the literature as the absolute remedy to the chattering effects. However no analysis seems to be available to confirm this assertion. In order to corroborate our above conclusions on the saturation input (which is shown in general not to decrease the input and output numerical chattering compared to the explicit discretization without



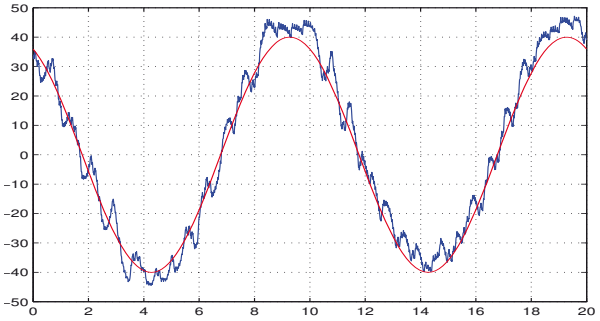
(a) $h = 2$ms. Explicit method.



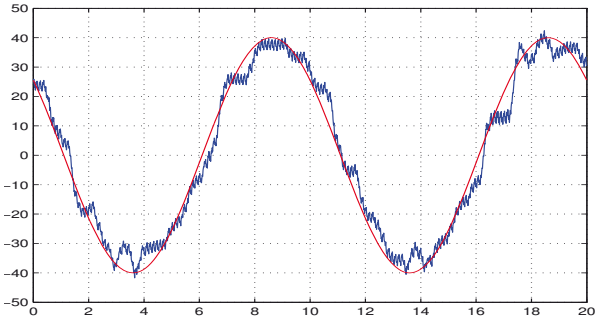
(b) $h = 2$ms. Saturation method.



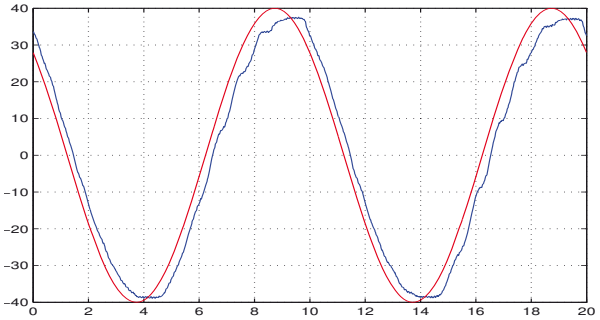
(c) $h = 2$ms. Implicit method.



(d) $h = 15$ms. Explicit method.

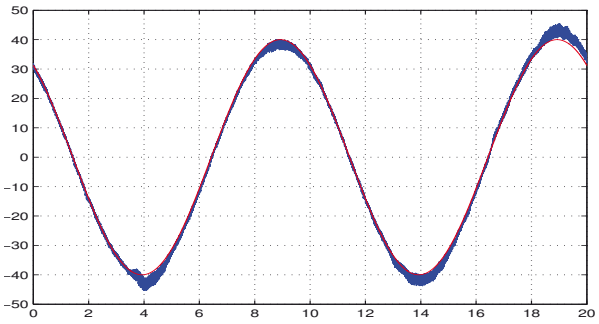


(e) $h = 15$ms. Saturation method.

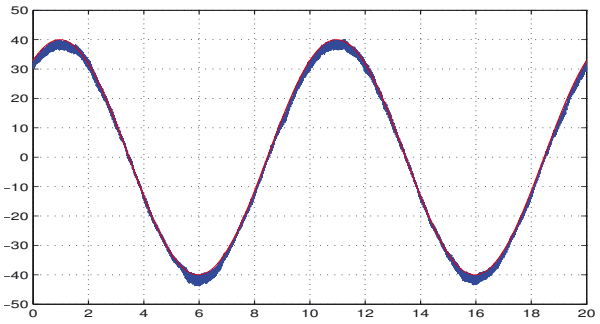


(f) $h = 15$ms. Implicit method.

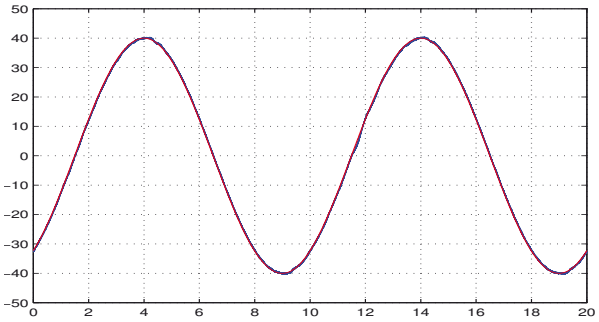
Figure 1.2: Real position y (mm) in blue and y_d (mm) in red, under $h = 2$ms and $h = 15$ms for $G = 10^4$. Real position y in blue and y_d in red.



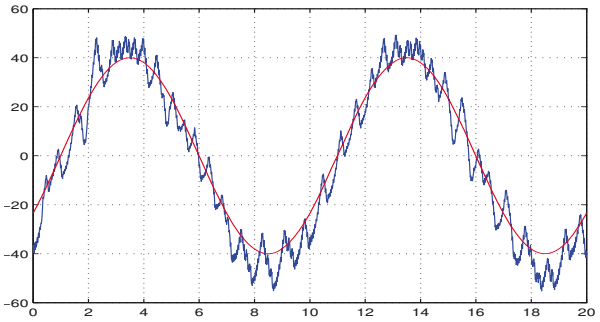
(a) $h = 2$ms. Explicit method.



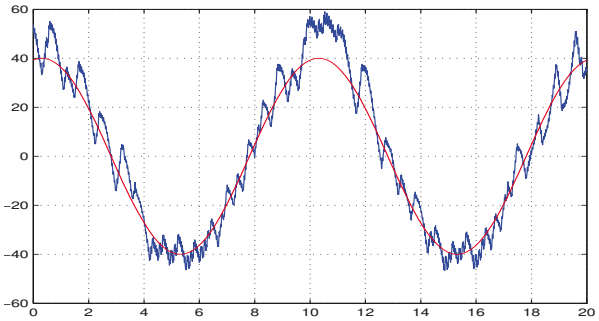
(b) $h = 2$ms. Saturation method.



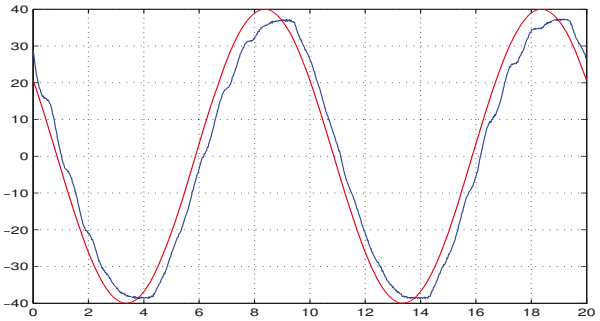
(c) $h = 2$ms. Implicit method.



(d) $h = 15$ms. Explicit method.

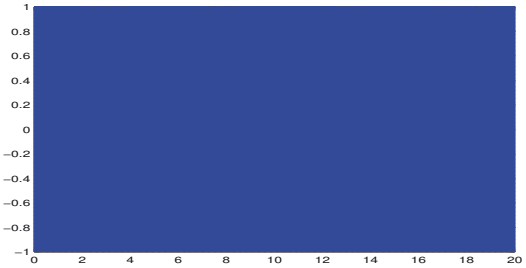


(e) $h = 15$ms. Saturation method.

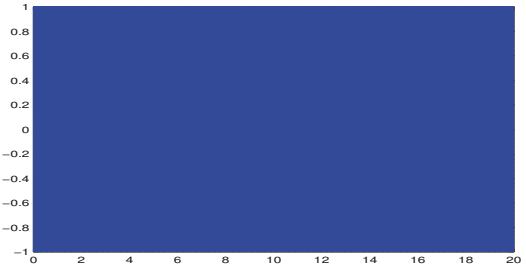


(f) $h = 15$ms. Implicit method.

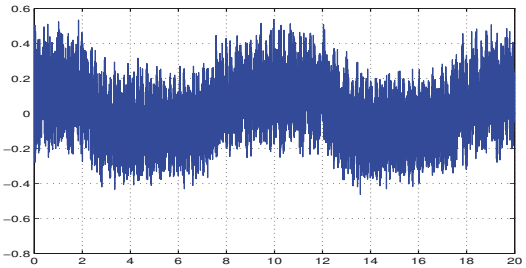
Figure 1.3: Real position y (mm) in blue and y_d (mm) in red, under $h = 2$ms and $h = 15$ms for $G = 10^5$.



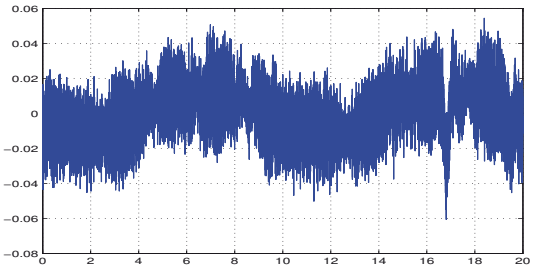
(a) Explicit. $sgn(\sigma_k)$. $G = 10^4$, $h = 2$ms.



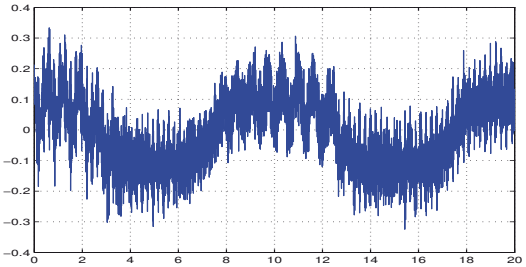
(b) Saturation. $sat(\sigma_k)$. $G = 10^4$, $h = 2$ms



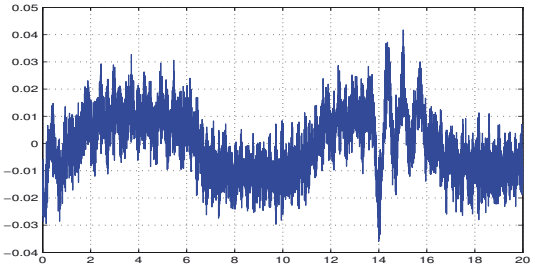
(c) Implicit. $sgn(\sigma_{k+1})$. $G = 10^4$, $h = 2$ms.



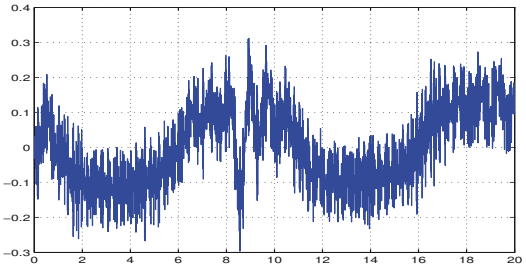
(d) Implicit. $sgn(\sigma_{k+1})$. $G = 10^5$, $h = 2$ms.



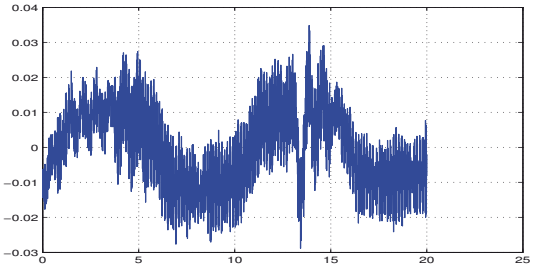
(e) Implicit. $sgn(\sigma_{k+1})$. $G = 10^4$, $h = 5$ms.



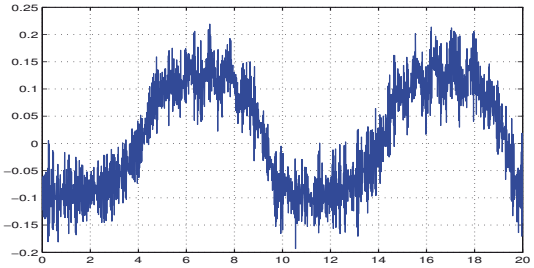
(f) Implicit. $sgn(\sigma_{k+1})$. $G = 10^5$, $h = 5$ms.



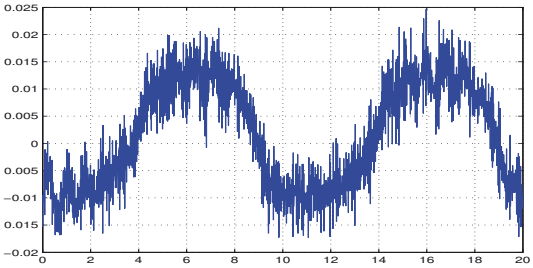
(g) Implicit. $sgn(s_{k+1})$. $G = 10^4$, $h = 10$ms.



(h) Implicit. $sgn(\sigma_{k+1})$. $G = 10^5$, $h = 10$ms.

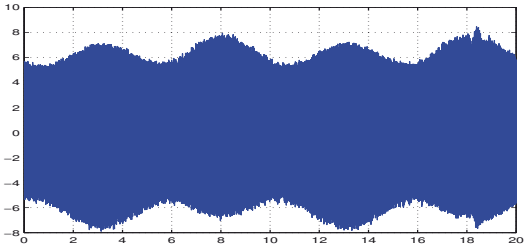


(i) Implicit. $sgn(\sigma_{k+1})$. $G = 10^4$, $h = 15$ms.

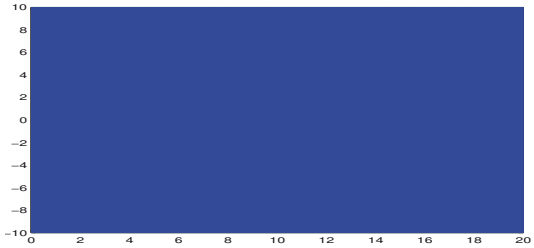


(j) Implicit. $sgn(\sigma_{k+1})$. $G = 10^5$, $h = 15$ms.

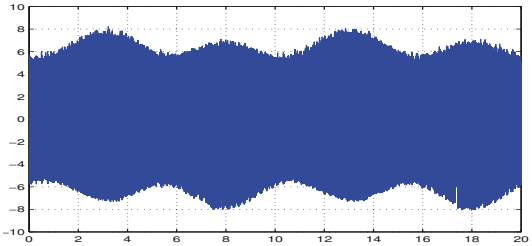
Figure 1.4: Switching function: Comparison between explicit ($sgn(s_k)$), saturation ($sat(s_k)$) and implicit ($sgn(s_{k+1})$) algorithms.



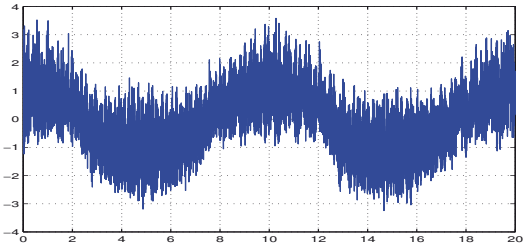
(a) Explicit. $G = 10^4$, $h = 2$ms.



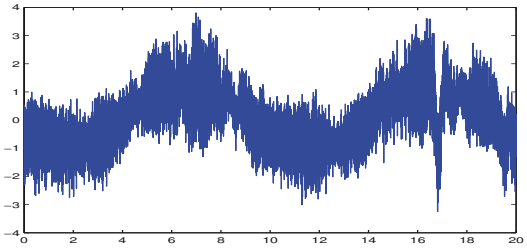
(b) Explicit and Saturation. $G = 10^5$, $h = 2$ms.



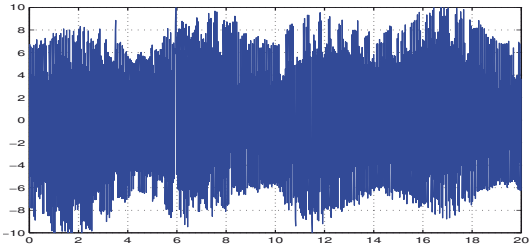
(c) Saturation. $G = 10^4$, $h = 2$ms



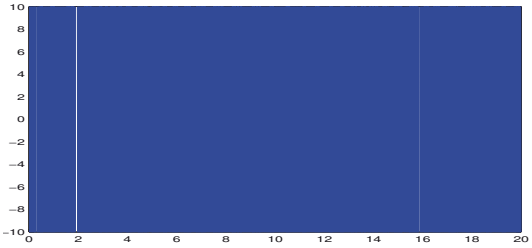
(d) Implicit. $G = 10^4$, $h = 2$ms.



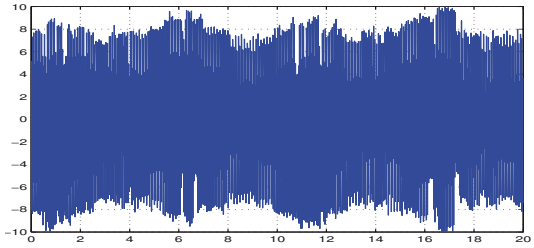
(e) Implicit. $G = 10^5$, $h = 2$ms.



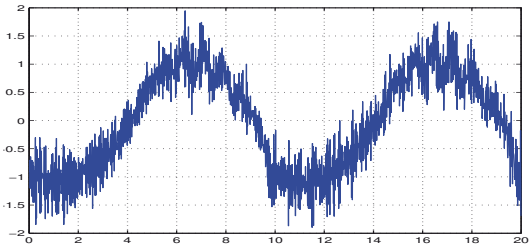
(f) Explicit. $G = 10^4$, $h = 15$ms.



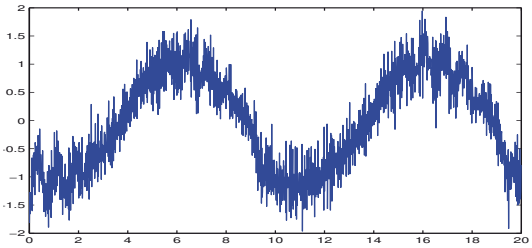
(g) Explicit and Saturation. $G = 10^5$, $h = 15$ms.



(h) Saturation. $G = 10^4$, $h = 15$ms



(i) Implicit. $G = 10^4$, $h = 15$ms.



(j) Implicit. $G = 10^5$, $h = 15$ms.

Figure 1.5: Comparison of control u between explicit, saturation and implicit methods.

h	2ms	5ms	10ms	15ms
Explicit	(-10 10)	(-10 10)	(-10 10)	(-10 10)
Saturation	(-10 10)	(-10 10)	(-10 10)	(-10 10)
Implicit	(-3.2541 3.8092)	(-2.0772 2.6066)	(-2.0325 2.3656)	(-1.9642 1.9461)

(a) range of u.

h	2ms	5ms	10ms	15ms
Explicit	29800	20500	19320	18360
Saturation	2.3516e+04	1.9846e+04	19020	18600
Implicit	9.3245e+03	1.5389e+03	1.1560e+03	629.0904

(b) variation of u.

h	2ms	5ms	10ms	15ms
Explicit	10.0004	9.9729	9.9903	9.9547
Saturation	9.9746	9.9080	9.9262	9.9561
Implicit	1.0974	0.9113	0.9037	0.8630

(c) standard deviation of u.

Table 1.7: Comparisons of u when $G = 10^5$.

G	10^4	10^5	10^6	5.10^6
$h = 5$ ms	$(-0.3, 0.35)$	$(-0.03, 0.035)$	$(-0.003, 0.003)$	$(-0.0006, 0.00065)$
$h = 10$ ms	$(-0.25, 0.3)$	$(-0.025, 0.03)$	$(-0.0025, 0.0025)$	$(-0.0005, 0.0005)$

Table 1.8: Magnitude of implicit switching function $\text{sgn}(x_{k+1})$ for varying gains G and sampling period h.

saturation), we have led a preliminary simulation study on a perturbed system to analyse the effect of adding a saturation, on both the tracking error and the input chattering. The dynamics is given by:

$$\begin{cases} \dot{x}(t) = A(t) + Bu(t) + \sin(4\pi t) & A = \begin{pmatrix} 0 & 1 \\ 19 & -2 \end{pmatrix}, \\ \sigma = Cx & \\ u(t) = u^{eq}(t) + u^{s}(t) & B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, C^T = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \end{cases} \tag{1.29}$$

where $u^{eq}(x) = -(CB)^{-1}CAx$, $u^s(x) \in -\text{sgn}(Cx)$. On Figures 1.6 and 1.7 the performance index is the sum of the $|\sigma_k|$ for the last 20 seconds. On Figures 1.8 and 1.9 the performance index is the sum of the $|u^s_{k+1} - u^s_k|$ for the last 20 seconds. The conclusions should not be considered as generic because the study has been made for only one disturbance: results might change if another disturbance is considered. Nevertheless Figures 1.6, 1.7, 1.8 and 1.9 allow us to draw some conclusions. With both indexes, we can divide the space into 3 cones, numbered 1, 2 and 3 on Figures 1.6, 1.7, 1.8 and 1.9. This separation helps us to compare both controllers. In Figure 1.6 the performance in terms of chattering is presented. For large values of ε, the chattering does not change when the sampling

period varies: the control action does not attenuate the effect of the perturbation. With a small ε, the behavior is more complex, as depicted in Figure 1.7: the overall best performance is obtained with small values for both ε and h. But for small values of ε, the performance can rapidly degrade if the sampling period h is not small enough, as seen in Region 1. The dark points indicate for each value of h the pair (ε, h) of parameters yielding the best performance. It seems that there is a linear relationship between those values. However, it is unclear if this observation on one particular system remains valid with a different perturbation. The level sets in Figure 1.7 are used to compare the performance of the implicit and the saturated explicit controllers. On Figures 1.8 and 1.9, the performance in terms of control cost is presented. The best performance is achieved for large ε since the slope of the saturated function is gentle. On the other hand in Figure 1.9, with a small ε, the cost increases and explodes with ε close to 0, as in Region 1. The level sets indicate the difference between the costs of the two different controllers. It is worth noting that in Region 2 where the saturated controller is better in Figure 1.7, it has a higher cost in term of control (Figure 1.9). In Region 3, where the saturated controller performs less in terms of chattering (Figure 1.6), it has a smaller cost in terms of control (Figure 1.8). Indeed with a large ε, the control input is small when the closed-loop system is close to the sliding manifold. The cost is then very small, but the disturbance is not attenuated at all. The implicit controller appeals to us as the best compromise between the input and output chattering. It is also very easy to use, since it requires no particular tuning with respect to the sampling period or the perturbation.

1.5 Conclusion

Experiments have been conducted on an electropneumatic system, with three different implementations of the sliding mode controller: explicit, saturated explicit, and implicit discretizations. The results demonstrate that the theoretical and numerical predictions of [1, 2] are true: the implicit implementation, which is very easy to implement in a code, drastically supersedes the other two. The output and input chattering are reduced in a significant way, without changing the controller basic structure (*i.e.*, no additional filter, observer, or dynamic controller is added compared to the original, basic sliding mode controller) and keeping its simplicity (in particular the gain tuning is easy, which is a strong feature of the ECB-SMC method). The main feature of the implicit discretization, is that it keeps, in discrete-time, the multivalued feature of the theoretical continuous-time sliding-mode controller, as it is mathematically imposed in Filippov's framework. The proposed implicit discretization method is generic in the sense that it could apply to any kind of sliding mode, set valued control. These conclusions have been confirmed elsewhere on another experimental setup for both the ECB-SMC and the twisting controllers [14, 17, 15].

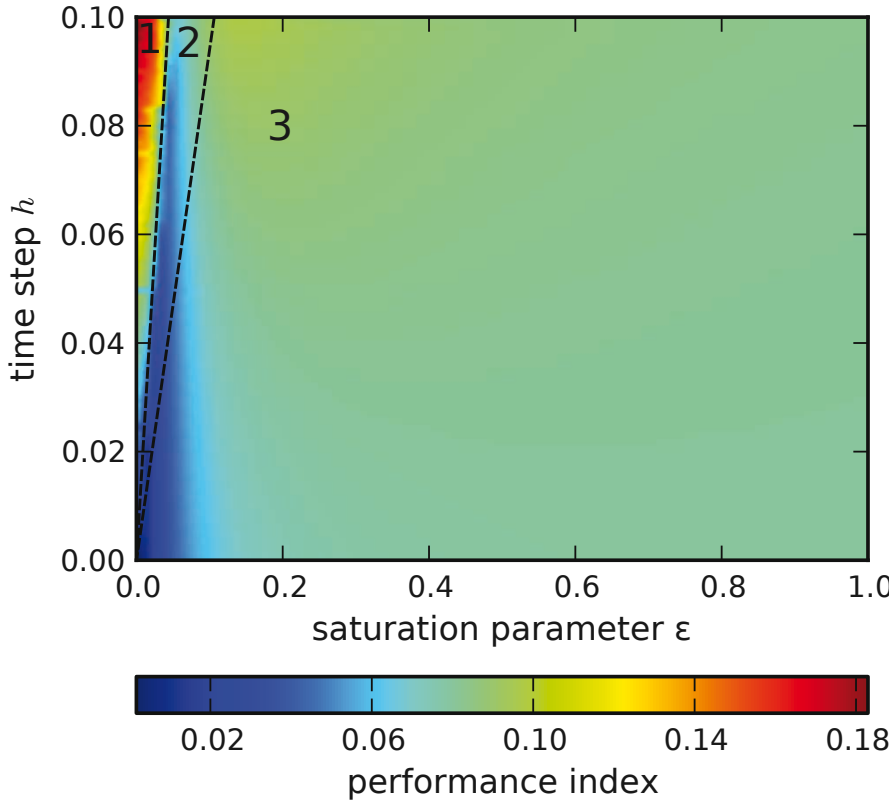


Figure 1.6: Simulation results with 100 regularly spaced values for the sampling period h and 100 logarithmically spaced values for the saturation parameter ε.

.1 Some basic convex analysis tools

In this section we provide few results which are useful to calculate the controller in Section 1.2.1. From [24, p.115] it follows that the conjugate function of the indicator function of the set $[-1,1]$, denoted as $\psi_{[-1,1]}(\cdot)$, is the support function $\psi^{\star}_{[-1,1]}(\cdot)$ of the set $[-1,1]$, given by the absolute value function $x \mapsto |x|$. From [24, Theorem 23.5], one has that the subdifferentials (in the sense of convex analysis) of these two conjugate functions satisfy: $x \in \partial\psi_{[-1,1]}(z) \Leftrightarrow z \in \partial\psi^{\star}_{[-1,1]}(x)$. From the definition of the subgradient, one has $\partial\psi^{\star}_{[-1,1]}(x) = \mathrm{sgn}(x)$ where sgn is the multivalued signum function as defined in the introduction. By definition of the subdifferential of a convex set, $\partial\psi_{[-1,1]}(z)$ is the normal cone to the set $[-1,1]$ at z. These results allow one to derive (1.9) from (1.7).

Consider now the inclusion $x - y \in -N_C(x)$ for some convex, non empty closed set C of $\mathbf{R}^n$, and two vectors x and y of $\mathbf{R}^n$. Using [7, Theorem 1.5.5] one finds that x is the Euclidean projection of y onto C. This allows us to deduce (1.10) from (1.9).

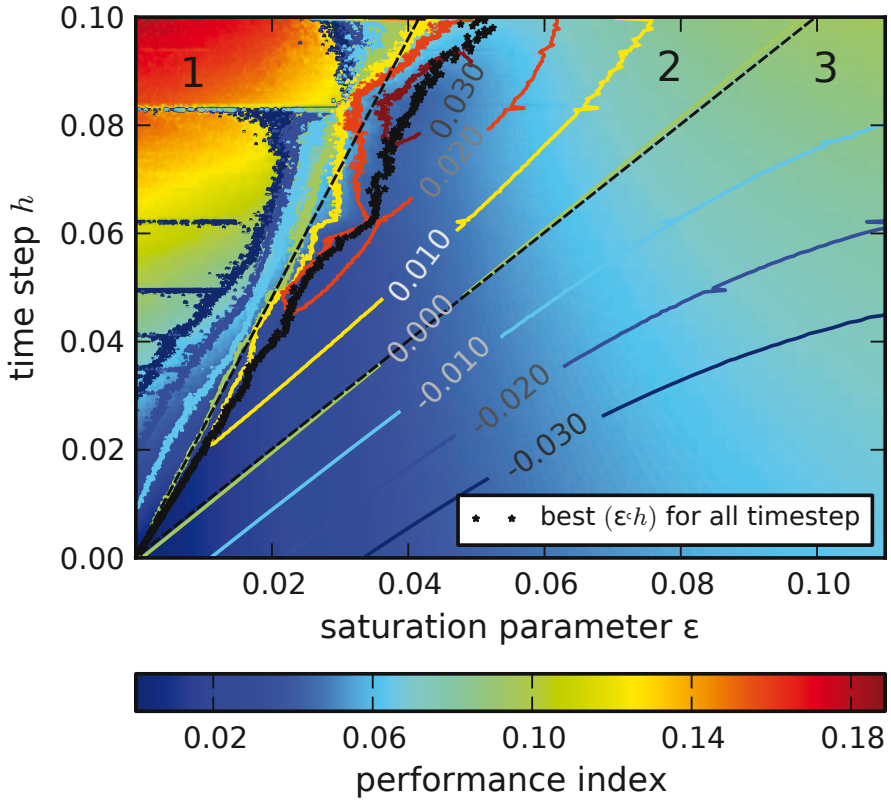


Figure 1.7: Detail of Figure 1.6, 300 values for h and 1000 values for ε, forming a regular grid. Level sets were also added to show the difference in performance between the implicit discretization and the explicit one with saturation. If the difference is positive, the explicit saturated control is performing better than the implicit one.

Acknowledgements: this work has been performed with the support of the French National Research Agency (ANR) project ChaSliM (ANR 2011 BS03 007 01).

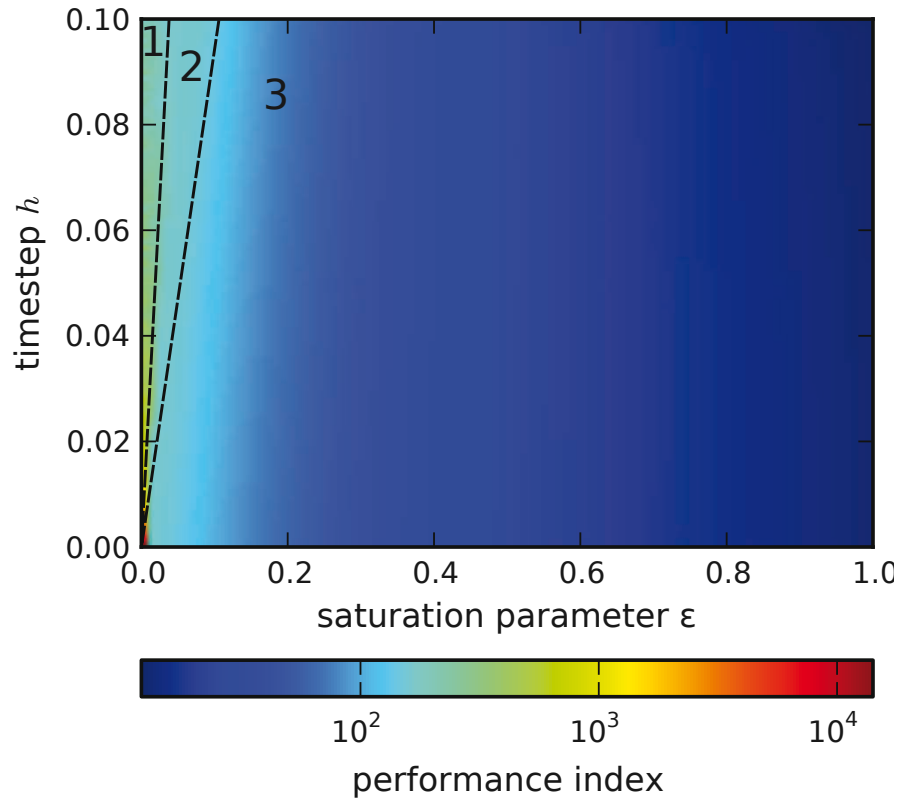


Figure 1.8: Simulation results with 100 regularly spaced values for the sampling period h and 100 logarithmically spaced values for the saturation parameter ε.

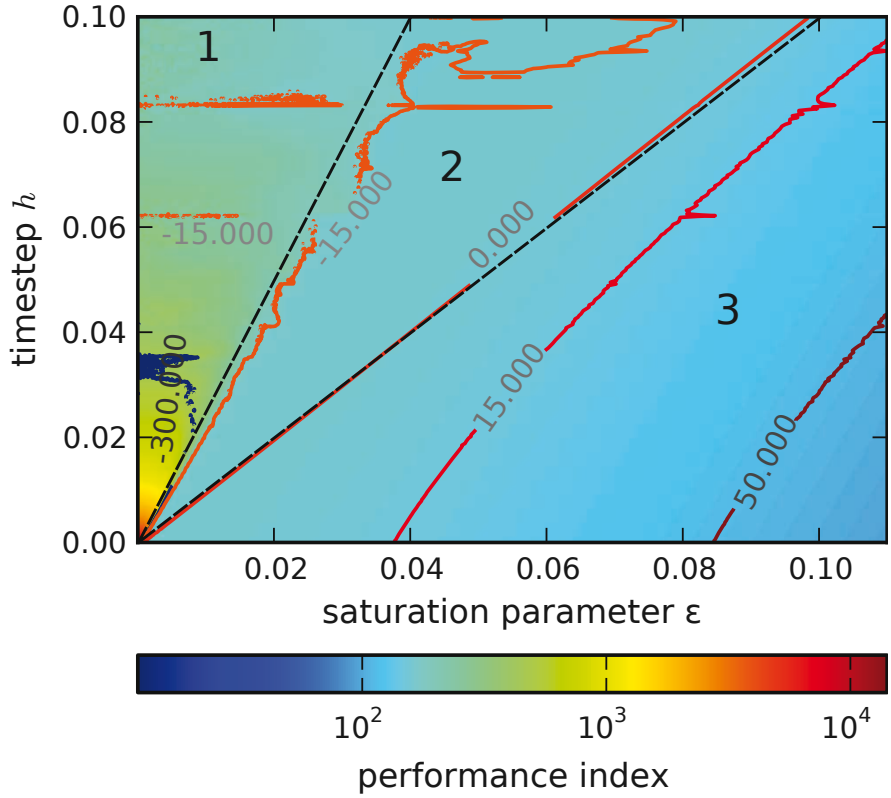


Figure 1.9: Detail of Figure 1.8, 300 values for h and 1000 values for ε, forming a regular grid. Level sets were also added to show the difference in performance between the implicit discretization and the explicit one with saturation. If the difference is positive, the explicit saturated control is performing better than the implicit one.

Bibliography

[1] V. ACARY, B. BROGLIATO, *Implicit Euler numerical scheme and chattering-free implementation of sliding mode systems*, Systems and Control Letters, vol.59, pp.284-293, 2010.

[2] V. ACARY, B. BROGLIATO, Y. ORLOV, *Chattering-free digital sliding-mode control with state observer and disturbance rejection*, IEEE Trans. Automatic Control, vol.57, no 5, pp.1087-1101, May 2012.

[3] V. ACARY AND B. BROGLIATO. *Numerical Methods for Nonsmooth Dynamical Systems. Applications in Mechanics and Electronics.* Lecture Notes in Applied and Computational Mechanics 35. Berlin: Springer. xxi, 525 p. , 2008.

[4] V. ACARY, O. BONNEFON, B. BROGLIATO, *Nonsmooth Modeling and Simulations for Switched Circuits*, Springer Verlag, Lecture Notes in Electrical Engineering, vol.69, 2011.

[5] M.K. CAMLIBEL, W.P.M.H. HEEMELS, J.M. SCHUMACHER *Consistency of a time-stepping method for a class of piecewise linear networks*, IEEE Transactions on Circuits and Systems I, vol.49, no 3, pp.349-357, 2002.

[6] R. CASTRO-LINARÈS, S. LAGHROUCHE, A. GLUMINEAU, AND F. PLESTAN, *Higher order sliding mode observer-based control*, IFAC Symposium on System, Structure and Control, Oaxaca, Mexico, 2004.

[7] F. FACCHINEI, J.S. PANG, *Finite-Dimensional Variational Inequalities and Complementarity Problems. Volume I*, Springer-Verlag New-York, Springer Series in Operations Research, 2003.

[8] A.F. FILIPPOV, *Differential Equations with Discontinuous Righthand Sides*, Kluwer Academic Publishers, Mathematics and its Applications, 1988.

[9] Z. GALIAS, 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.

[10] Z. GALIAS, 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.

[11] A. GIRIN, F. PLESTAN, X. BRUN, AND A. GLUMINEAU, *Robust control of an electropneumatic actuator: application to an aeronautical benchmark*, IEEE Transactions on Control Systems Technology, vol.17, no.3, pp.633-645, 2009.

[12] G. GOLO, A. J. VAN DER SCHAFT, Č. MILOSAVLJEVIĆ, *Discretization of control law for a class of variable structure control systems*, in Advances in Variable Structure Systems: Analysis, Integration and Applications, pp. 45-54, X. Yu and J.-X. Xu (editors), World Scientific, 2000.

[13] O. HUBER, V. ACARY, B. BROGLIATO, *Comparison between explicit and implicit discrete-time implementations of sliding-mode controllers*, Proceedings of the 52nd IEEE Conference on Decision and Control, pp.2870-2875, 10-13 December 2013, Florence, Italy.

[14] O. HUBER, V. ACARY, B. BROGLIATO, F. PLESTAN, *Discrete-time twisting controller without numerical chattering: analysis and experimental results with an implicit method*, Proceedings of the 53rd IEEE Conference on Decision and Control, pp.4373-4378, 15-17 December 2014, Los Angeles, USA.

[15] O. HUBER, *Analysis and implementation of discrete-time sliding mode control*, Ph.D. Thesis, University Grenoble-Alpes, May 2015 (https://hal.inria.fr/tel-01194430/document).

[16] O. HUBER, V. ACARY, B. BROGLIATO, *Analysis of explicit and implicit discrete-time equivalent-control based sliding mode controllers*, INRIA Research Report 8383, October 2013 (http://arxiv.org/pdf/1310.6004v1.pdf).

[17] O. HUBER, V. ACARY, B. BROGLIATO, F. PLESTAN, *Implicit discrete-time twisting controller without numerical chattering: analysis and experimental results*, Control Engineering Practice, *in press*, DOI: 10.1016/j.conengprac.2015.10.013.

[18] R. KIKUUWE, S. YASUKOUCHI, H. FUJIMOTO, M. YAMAMOTO, *Proxy-Based Sliding Mode Control: A Safer Extension of PID Position Control*, IEEE Transactions on Robotics, vol.26, no 4, pp.670-683, 2010.

[19] S. LAGHROUCHE, M. SMAOUI, F. PLESTAN, AND X. BRUN, *Higher order sliding mode control based on optimal approach of an electropneumatic actuator*, International Journal of Control, vol.79, no.2, pp.119-131, 2006.

[20] D. LEENAERTS, *On linear dynamic complementary systems*, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications, vol.46, no 8, pp.1022-1026, 1999.

[21] J.J. MOREAU, *Standard inelastic shocks and the dynamics of unilateral constraints*, in CISM Courses and Lectures, no 288, Springer-Verlag, pp.173-221, 1985.

[22] J.J. MOREAU, *Unilateral contact and dry friction in finite freedom dynamics*, in J.J. Moreau, P.D. Panagiotopoulos, (Eds.), 1988 *Nonsmooth mechanics and applications*, CISM Courses and Lectures no 302, International Centre for Mechanical Sciences, Springer-Verlag, pp.1-82.

[23] F. PLESTAN, Y. SHTESSEL, V. BRÉGEAULT, AND A. POZNYAK, *Sliding mode control with gain adaptation–Application to an electropneumatic actuator*, Control Engineering Practice, vol.21, no.5, pp.679-688, 2013.

[24] R.T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton Landmarks in Mathematics, 1970.

[25] S. SESMAT, AND S. SCAVARDA, *Static characteristics of a three way servovalve*, Proc. Conf. Fluid Power Tech., Aachen, Germany, 1996.

[26] Y. SHTESSEL, M. TALEB, AND F. PLESTAN, *A novel adaptive-gain supertwisting sliding mode controller: methodology and application*, Automatica, vol.48, no.5, pp. 759-769, 2012.

[27] SIRA–RAMIREZ, H., *Non–linear discrete variable structure systems in quasi–sliding mode*, International Journal of Control, vol.54, no.5, pp. 1171–1187, 1991.

[28] V. UTKIN, *Variable structure systems with sliding mode*, IEEE Transactions on Automatic Control, vol.22, no 2, pp.212-222, April 1977.

[29] V. UTKIN, *Sliding Modes in Control and Optimization*, Springer-Verlag Berlin, Communications and Control Engineering, 1992.

[30] B. WANG, X. YU, G. CHEN, *ZOH discretization effect on single-input sliding mode control systems with matched uncertainties*, Automatica, vol.45, pp.118-125, 2009.

[31] 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, DOI: 10.1109/TCST.2015.2396473.

[32] X. YU, B. WANG, Z. GALIAS, G. CHEN, *Discretization effect on equivalent control-based multi-input sliding-mode control systems*, IEEE Transactions on Automatic Control, vol.53, no 6, pp.1563-1569, July 2008.