



HAL
open science

Second order sliding mode twisting controller tuning based on two-level optimization process

Dominique Monnet, Alexandre Goldsztejn, Franck Plestan, Charles Audet

► **To cite this version:**

Dominique Monnet, Alexandre Goldsztejn, Franck Plestan, Charles Audet. Second order sliding mode twisting controller tuning based on two-level optimization process. Conference on Decision and Control, Dec 2022, Cancun, Mexico. hal-03963726

HAL Id: hal-03963726

<https://hal.science/hal-03963726>

Submitted on 30 Jan 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Second order sliding mode twisting controller tuning based on two-level optimization process

Dominique Monnet¹, Alexandre Goldsztejn², Franck Plestan³ and Charles Audet¹

Abstract—State-of-the-art finite time convergence conditions for the sliding mode controllers rely on bounds on perturbation terms. These bounds are often over-approximated, leading to conservative designs, *i.e.*, high gains that amplify undesired behaviors such as chattering. This paper proposes to evaluate precisely the bounds on the perturbation terms to avoid conservative designs by using branch-and-bound algorithms dedicated to nonlinear programming. This leads to non-linear, a priori non-convex, non-differentiable constraints on the controller's gains, which is shown to be solvable using a modern black-box optimization algorithm. We propose a new methodology employing branch-and-bound and blackbox solvers to generate gains as small as possible ensuring finite time convergence for the twisting algorithm. It is investigated using both a classical and a recently proposed sufficient conditions for finite time convergence. The applicability of the approach is illustrated over a numerical example.

I. INTRODUCTION

One motivation to the development of second order sliding mode (2-SM) techniques has been the reduction of the chattering effect [2], [8], [5] inherent to the classical first order techniques [16]. Several 2-SM techniques have been proposed [7]. The convergence proofs to the sliding surface rely on bounds on the perturbation terms appearing in the second time derivative of the sliding variable, whether these proofs rely on the majorant curve [15] or Lyapunov theory [10], [11]. These bounds are generally difficult to obtain, and over-approximating them leads to conservative designs, *i.e.*, large magnitude of the control input that amplifies the chattering effect. As a consequence, the control law is often tuned by simulation rather than derived from the convergence conditions. Although adaptive gain strategies allow to reduce the chattering, investigating sharper theoretical bounds and their usage through numerical optimization is of central importance.

In this paper, the classical 2-SM twisting controller is considered. The aim is to focus on the existing convergence conditions, and to use optimization tools to precisely evaluate the bounds they involve. Using the conditions to tune the twisting controller's gains, the convergence is guaranteed (contrary to a simulation-based tuning), and the convergence conditions are as less conservative as they can be.

¹ GERAD and Département de mathématiques et génie industriel, Polytechnique Montréal, 2500 chemin de Polytechnique, Montréal, QC, Canada. dominique.monnet@polymtl.ca

² CNRS-Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004, 1 rue de la Noë, Nantes, France. alexandre.goldsztejn@ls2n.fr

³ École Centrale de Nantes-Laboratoire des Sciences du Numérique de Nantes, UMR CNRS 6004, 1 rue de la Noë, Nantes, France. franck.plestan@ec-nantes.fr

This approach follows previous works that studied optimization-based tuning for first order sliding mode [12], where SIP solvers were used to solve simple instances of first order sliding mode. Such solvers turn out to be unable to solve larger instances of second order SM. We propose a mixed local/global approach in this paper, where a branch-and-bound algorithm dedicated to nonlinear optimization is used to solve globally the finite time convergence conditions, and a local blackbox algorithm [1] is used to optimize gains subject to the blackbox constraints.

The paper is organized as follow. Section II states the state-of-the-art convergence conditions of the twisting controller, as well as new less conservative conditions. Section III provides notions on global optimization tools, and how to employ them to compute the bounds involved in the convergence conditions. Section IV illustrates the proposed optimization-based approach on an example, and Section V concludes the paper and proposes directions for future work.

II. TWISTING CONVERGENCE CONDITIONS

Let us consider a dynamical system given by

$$\dot{x} = f(p(t), x, u) \quad (1)$$

where $t \geq 0$ is the time variable, p is a perturbation input, $x \in \mathbb{R}^n$ the state, $u \in \mathbb{R}$ the control input. We assume that the sliding variable $\sigma(p(t), x)$ depends on perturbations and state variables, as well as its first and second time derivative. We further assume that $\ddot{\sigma}(p(t), x)$ is affine in the control input and is assumed to be linear in the control input,

$$\ddot{\sigma} = h(p(t), x) + g(p(t), x)u. \quad (2)$$

p is therefore a vector that includes external input signals such as the reference signal, perturbations, etc. and their time derivatives. It is assumed that σ , $\dot{\sigma}$, h and g have known explicit expressions depending on p and x . This assumption is needed to employ optimization tools. Several algorithms have been proposed to ensure the convergence of the system on the sliding surface [7]. This paper focuses on the twisting algorithm, whose control structure is given by ([15])

$$u = -r_1 \text{sign}(\sigma) - r_2 \text{sign}(\dot{\sigma}). \quad (3)$$

We assume in the following that finite bounds P are known on $p(t)$,

$$p(t) \in P, \forall t \geq 0. \quad (4)$$

This realistic assumption is required to allow to employ numerical solvers to compute bounds on the right-hand side of Equation (2). These bounds are also needed to compute

the gains (r_1, r_2) using the convergence conditions stated in Subsection II-A.

Since the control signal is discontinuous and not defined for $\sigma = 0$ and $\dot{\sigma} = 0$, the solution of (1) is understood in the Filippov sense [4].

The following two subsections present conditions that guarantee the convergence of both σ and $\dot{\sigma}$ to zero in finite time, hence ensuring the expected sliding motion. The first conditions are classical, the second ones have been recently proposed by the authors in [9], where the conservatism is thoroughly discussed and illustrated. All convergence conditions assume the state is maintained in a given compact $X \subseteq \mathbb{R}^n$. This is another condition to employ optimization tools. Note that in the literature, this condition is implicit whenever bounds on g and h are assumed and they are not bounded over the state space.

A. State-of-the-art convergence conditions

The classical convergence conditions for the twisting controller rely on bounds $C, k, K \geq 0$ on $h(p(t), x)$ and $g(p(t), x)$ that hold for all $t \geq 0$ and $x \in X$. These bounds are defined such that

$$\begin{aligned} h(p(t), x) &\in [-C, C], \quad \forall t \geq 0, \quad \forall x \in X, \\ g(p(t), x) &\in [k, K], \quad \forall t \geq 0, \quad \forall x \in X. \end{aligned} \quad (5)$$

The convergence conditions based on the bounds (5) are given by Theorem 1 [15, pp.148]

Theorem 1: Consider System (2) with u defined by (3). If $r_1 > r_2 > 0$,

$$C < k(r_1 - r_2), \quad (6)$$

and

$$(r_1 - r_2)K + C < (r_1 + r_2)k - C, \quad (7)$$

then σ and $\dot{\sigma}$ converge to 0 in finite time.

The convergence conditions of Theorem 1 present the advantage to be easy to use in order to tune the gains r_1 and r_2 , since these conditions are expressed as linear inequalities on r_1 and r_2 .

B. Less conservative convergence conditions

Recent work [9] has proposed new convergence conditions for the twisting algorithm that are less conservative than the conditions of Theorem 1. That is, these new conditions define larger sets for the gains (r_1, r_2) ensuring the finite time convergence. Therefore, one is allowed to choose smaller gains limiting chattering effect while ensuring finite time convergence. The new conditions depend on the bounds on $\ddot{\sigma}$ in the four quadrants Σ_i in the space $(\sigma, \dot{\sigma})$ defined as,

$$\begin{aligned} \Sigma_1 &= \{(\sigma, \dot{\sigma}) \mid \sigma > 0 \text{ and } \dot{\sigma} > 0\}, \\ \Sigma_2 &= \{(\sigma, \dot{\sigma}) \mid \sigma > 0 \text{ and } \dot{\sigma} < 0\}, \\ \Sigma_3 &= \{(\sigma, \dot{\sigma}) \mid \sigma < 0 \text{ and } \dot{\sigma} < 0\}, \\ \Sigma_4 &= \{(\sigma, \dot{\sigma}) \mid \sigma < 0 \text{ and } \dot{\sigma} > 0\}. \end{aligned} \quad (8)$$

The sub-spaces in X corresponding to these quadrants are denoted X_i , for $i = 1, 2, 3, 4$,

$$X_i(p(t)) = \{x \in X \mid (\sigma(p(t), x), \dot{\sigma}(p(t), x)) \in \Sigma_i\}. \quad (9)$$

Notice that, since σ and $\dot{\sigma}$ depends on $p(t)$, the sub-spaces X_i also depend on $p(t)$. The bounds on $\ddot{\sigma}$ in the four quadrants thus satisfy,

$$\ddot{\sigma}(p(t), x) \in [m_i, M_i], \quad \forall t \geq 0, \quad \forall x \in X_i(p(t)). \quad (10)$$

Theorem 2 [9] states the finite time convergence based on the bounds (10).

Theorem 2: Consider System (2) with u defined by (3). If the bounds (10) hold,

$$M_1, M_2 < 0, \quad (11)$$

$$m_3, m_4 > 0, \quad (12)$$

and

$$M_1 m_3 < m_2 M_4, \quad (13)$$

then σ and $\dot{\sigma}$ converge to zero in finite time.

Remark 1: The proof of Theorem 1 relies on the study of the majorant curve that is obtained from the bounds on $\ddot{\sigma}$ [15, pp. 148]. With the bounds C, k, K on h and g , and the twisting controller (3), one derives the differential inclusion,

$$\begin{aligned} \ddot{\sigma} &\in [-C + K(-r_1 - r_2), C + k(-r_1 - r_2)] && \text{in } \Sigma_1, \\ \ddot{\sigma} &\in [-C + K(-r_1 + r_2), C + k(-r_1 + r_2)] && \text{in } \Sigma_2, \\ \ddot{\sigma} &\in [-C + k(r_1 + r_2), C + K(r_1 + r_2)] && \text{in } \Sigma_3, \\ \ddot{\sigma} &\in [-C + k(r_1 - r_2), C + K(r_1 - r_2)] && \text{in } \Sigma_4. \end{aligned} \quad (14)$$

The conditions of Theorem 2 can be retrieved by following the same reasoning than the proof of Theorem 1 by replacing the bounds (14) by the generic bounds (10). One can remark that under the condition $r_1 > r_2 > 0$ necessary in Theorem 1, injecting the bounds (14) in Conditions (11) and (12) of Theorem 2 leads to Condition (6) of Theorem 1, with the additional constraint $C < K(r_1 + r_2)$ which is ignored since it is necessarily satisfied if (6) holds. Condition (13) becomes Condition (7) in Theorem 1.

Remark 2: It is proved in [9] that the conditions of Theorem 2 are less conservative than the conditions of Theorem 1 provided that the same bounding process is used in both cases. This conservatism is illustrated in Section IV.

In the general case, it is not possible to obtain explicitly the dependency with respect to r_1 and r_2 for the bounds (10). However, since $\ddot{\sigma}$ has an explicit expression, it is possible to employ numerical solvers to compute such bounds in a guaranteed way.

III. NUMERICAL OPTIMIZATION FOR GAINS COMPUTATION

This section proposes a two-level optimization process to compute the optimal gains r_1, r_2 with respect to a given performance criterion, for example minimizing $r_1 + r_2$. The lower level of the process consists in computing the bounds (10) in a guaranteed way. Tightest bounds are aimed to be computed, and since h, g and $\ddot{\sigma}$ are continuous nonlinear

non-convex functions of the states and the perturbations, global solvers must be employed to obtain numerical guaranteed bounds. The upper level of the optimization process consists in exploring the space (r_1, r_2) to find the best gains, that is, the gains that minimize a given performance criterion such that the bounds computed at the lower level respect the convergence conditions. In [13], a similar two-level approach for 1-SM tuning is proposed, the upper level being solve with a global solver. However, the underlying optimization problem of the upper level for tuning the twisting controller is more difficult since it involves a greater number of constraints to ensure the convergence, more complex expressions, and more variables. This makes a global approach intractable because of the high complexity of the global optimization (GO) algorithms. That is why we chose here a novel approach by employing an algorithm that converges locally and not globally, but has lower complexity and can be employed for tuning the twisting controller's gains, *i.e.*, solving the upper problem.

With

A. Overall optimal gain design

The gains r_1 and r_2 are chosen by optimizing a performance index $f(r_1, r_2)$ subject to the requirement that $\sigma(p(t))$ and $\dot{\sigma}(p(t))$ converge to 0 in finite time. Having small gains is a typical requirement, *e.g.*, $f(r) = r_1 + r_2$. The finite time convergence requirement is enforced using the sufficient conditions presented in the previous section. Although Theorem 2 is less conservative than Theorem 1, its bounds do not depend linearly on the gains and require additional work¹: on the one hand, once the bounds C, k, K are computed, Theorem 1 provides linear inequalities on the values of r_1 and r_2 that enforce finite time convergence. The performance index can be optimized inside these linear inequalities. On the other hand, the bounds m_i and M_i used in Theorem 2 depend non-linearly in r_1 and r_2 , and cannot be expressed explicitly as functions of r_1 and r_2 . It follows that the conditions of Theorem 2 cannot be expressed explicitly, and checking that a pair (r_1, r_2) enforces finite-time convergence requires to employ a global solver to compute m_i and M_i . Then, it is possible to check if the convergence conditions of Theorem 2 are ensured. This corresponds to solving the lower problem mentioned above.

If one can solve the lower problem, it is possible to discretize the search for r_1 and r_2 , and for each sample check if the conditions of Theorem 2 are satisfied. This is done in Section IV but only to characterize the set of gains satisfying the conditions of Theorem 2 and to illustrate the conservatism of the conditions of Theorem 1 (see Figure 2).

In order to generate gains that are optimal with respect to $f(r_1, r_2)$, we propose to employ a Derivative Free Optimization (DFO) algorithm. The most common optimization algorithms relies on the derivative of f and the constraints given by the convergence conditions to compute (r_1, r_2) that minimizes f . However, in our case, we do not know

explicit expressions for m_i and M_i , as they are the result of the lower level optimization process, and therefore are not differentiable. That is why we need to employ a DFO algorithm that can solve this class of problems. The DFO algorithm chooses iteratively potential candidates in the space (r_1, r_2) , provides it to the lower level which computes the bounds m_i, M_i and returns them to the upper level as illustrated in Figure 1. From the values of f and m_i, M_i at the provided candidate, the DFO chooses a new potential best candidate at the next iteration and eventually converges to a locally optimal pair of gains.

The two-level approach we propose generates a pair (r_1, r_2) which is guaranteed to ensure finite time convergence of σ and $\dot{\sigma}$ to zero. The pair is locally optimal with respect to f in the set of gains ensuring finite time convergence.

B. Lower level: computing optimal bounds with global optimization

The bounds (5) and (10) have the form $b(t, x) \in [l, u]$, where b is alternatively $g, h,$ or $\ddot{\sigma}$. The bounds l and u are to be computed so that $b(p(t), x) \in [l, u]$ holds for all $t \geq 0$ and all $x \in X(p(t))$. When b is either g or h in (5) then the set $X(p(t))$ does not depend on t , but we discuss here this more general case only. One can remark that, the tighter are the computed bounds the larger is set of gains proved to enforce finite time convergence by theorem. Therefore, we aim at computing the optimal bounds

$$l^* = \min_{\substack{t \geq 0 \\ x \in X(p(t))}} b(p(t), x), \quad u^* = \max_{\substack{t \geq 0 \\ x \in X(p(t))}} b(p(t), x). \quad (15)$$

Local minima or maxima cannot be used because inequalities would not be guaranteed to hold. Therefore, deterministic global optimization such as branch-and-bound algorithms are required to maintain a global lower bound (respectively an upper bound) that converges to the minimum (respectively to the maximum).

The dependency with respect to time in the optimization problems (15) causes difficulties. The first one is that explicit expressions with respect to time of perturbations are unavailable, but only bounds on perturbations are known. The second one is that $t \in [0, \infty)$ is unbounded while branch-and-bound algorithms require bounded feasible search spaces. As explained in Section I, we assume that all time dependencies arise through some unknown function $p(t)$ bounded in P holds for all $t \geq 0$. Theorem 3 allows solving problems involving a variable $p \in P$ instead of $t \geq 0$.

Theorem 3: Define

$$l^+ = \min_{\substack{p \in P \\ x \in X(p)}} b(p, x), \quad u^+ = \max_{\substack{p \in P \\ x \in X(p)}} b(p, x). \quad (16)$$

Then $[l^+, u^+] \supseteq [l^*, u^*]$.

Proof: Let $t^* \geq 0$ $x^* \in X(t^*)$ be minimizers of (15), *i.e.*, $b(p(t^*), x^*) \leq 0$. Then $p^* \in P$, and $x^* \in X(p(t^*)) = X(p^*)$ therefore p^* and x^* are feasible for the problem (16), which implies $l^+ \leq l^*$. A symmetric reasoning shows that $u^+ \geq u^*$. ■

¹This additional work is worthwhile in an optimal design process.

Note that the bounds $[l^+, u^+]$ are conservative with respect to $[l^*, u^*]$ because the impact of time correlation on $b(p(t))$ is lost when only considering the information $p(t) \in P$. One cannot expect better if no accurate knowledge on $p(t)$ is known.

Off-the-shelf global solvers are available [14], [3], and discussing in detail how they are implemented is beyond the scope of this paper.

In the end, global solvers enable to compute guaranteed lower (resp. upper) bounds on l^+ (resp. u^+) as defined in Theorem 3 within a given relative precision, and by extension guaranteed bounds on the optimal bounds l^* and u^* as stated by Theorem 3. One can obtain the bounds C , k , K , and from the conditions of Theorem 1 define a set of gains (r_1, r_2) by linear inequalities. Furthermore, provided numerical value for (r_1, r_2) , one can compute guaranteed bounds m_i , M_i on $\ddot{\sigma}$, and thus concluding if the conditions of Theorem 2 are satisfied.

C. Upper level: optimizing the gains

The upper problem consist in finding r_1, r_2 that minimize a performance criterion $f(r_1, r_2)$, such that (r_1, r_2) satisfies the convergence constraints of Theorem 2. We recall that the convergence constraints are nonlinear, non convex, and non-differentiable and do not have explicit expressions. Such problems can be solved by DFO algorithms, that do not need to have access to the derivative of f or the constraints to converge to a local solution. In this paper, we propose to use the NOMAD solver [6] to solve the upper level problem. NOMAD has been developed to solve blackbox (BB) problems, meaning that the evaluation of the performance criterion f and the constraints is the result of some BB time-consuming process, for example simulations [?]. In our case, the BB process is the lower problem, which takes as input a pair (r_1, r_2) and returns the value of the bounds by employing a GO solver as explained in the previous subsection. From an initial pair (r_1^0, r_2^0) , NOMAD explore the space of gains by repeatedly choosing points in that space and calling the GO solver to ensure that this points respect the convergence constraint, and eventually converges to a local optimum (r_1^*, r_2^*) . Figure 1 schematizes the two-levels optimization process.

NOMAD implements a Mesh Adaptive Direct Search (MADS) algorithm. Devising on the principle of this kind of algorithm is beyond the scope of this paper, but the interested reader can refer to [?] for the latest development in MADS.

IV. NUMERICAL EXAMPLE

Consider the pendulum example adapted from [7]. The system is a rigid pendulum loaded with a mass m tracking a reference signal $\theta_c(t)$, with a varying arm length $R(t)$ to model a perturbation. The states of the pendulum are $x = (\theta, \dot{\theta})$, the angle of the arm and its angular velocity. The state $\theta = 0$ corresponds to the vertical position (down). We suppose that an actuator provides a torque u considered as the control input. The system dynamic is given by the

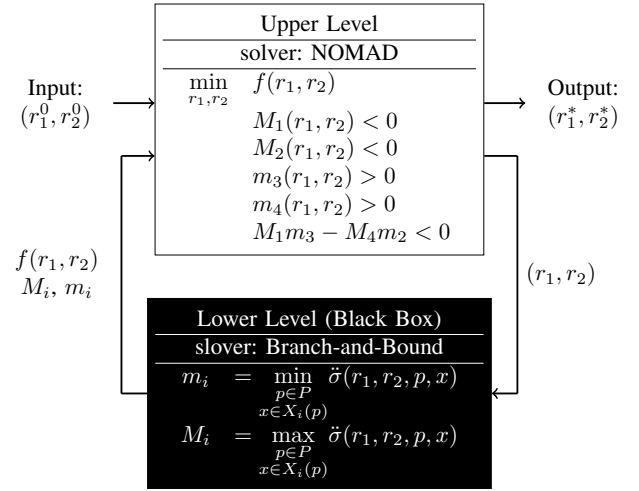


Fig. 1. Two-levels optimization process for optimizing the twisting controller gains.

differential equation

$$\ddot{\theta} = -2\frac{\dot{R}}{R}\dot{\theta} - g\frac{1}{R}\sin(\theta) + \frac{1}{mR^2}u. \quad (17)$$

We take the mass $m = 1$ and $g = 9.81$.

The reference signal is given by

$$\theta_c(t) = 0.5 \sin(0.5t) + 0.5 \cos(0.5t), \quad (18)$$

and the perturbation is modeled by

$$R(t) = 0.8 + 0.1 \sin(2t) + 0.2 \cos(t). \quad (19)$$

The sliding variable is

$$\sigma = \theta - \theta_c. \quad (20)$$

It follows that

$$\dot{\sigma} = \dot{\theta} - \dot{\theta}_c \quad (21)$$

and

$$\ddot{\sigma} = h(t, x) + g(t, x)u \quad (22)$$

with

$$g(p, x) = \frac{1}{mR^2}, \quad h(p, x) = -2\frac{\dot{R}}{R}\dot{\theta} - 9.81\frac{1}{R}\sin(\theta) - \ddot{\theta}_c. \quad (23)$$

The vector of perturbations is

$$p(t) = (R(t), \dot{R}(t), \theta_c(t), \dot{\theta}_c(t), \ddot{\theta}_c(t)). \quad (24)$$

From the expressions of $R(t)$ and $\theta_c(t)$, we derive the bounds

$$p(t) \in P = [0.5, 1.1] \times [-0.4, 0.4] \times [-1, 1] \times [-0.5, 0.5] \times [-0.25, 0.25]. \quad (25)$$

We propose now to generate the set of controllers satisfying the convergence conditions of Theorem 1 and 2. We limit the study to $x \in [-\pi, \pi] \times [-4, 4]$.

In the following, all the problems are solved with the global solver implemented in the C++ library IBEX [3] with a relative precision $\epsilon = 0.01$. We obtain the bounds $C = 26.65$, $k = 0.82$ and $K = 4.04$. The resulting

set of feasible values of (r_1, r_2) with respect to the linear constraints of Theorem 1 is displayed by the gray set in Figure 2.

We propose to generate a set of feasible gains from the conditions of Theorem 2 for illustrative comparison with the conditions of Theorem 1. To do so, we discretize the search space (r_1, r_2) , and for every point we compute the optimal bounds m_i, M_i , as explained in Subsection III-B and verify if the conditions of Theorem 2 are satisfied. The discretization points are $r_1 = \{1, 3, 5, \dots, 199\}$, $r_2 = \{1, 3, 5, \dots, 199\}$ and generate a total number of 10,000 points. The points that satisfy the convergence conditions of Theorem 2 are represented by the dots in Figure 2. Processing these 10,000 points takes approximately 70 minutes on a standard laptop.

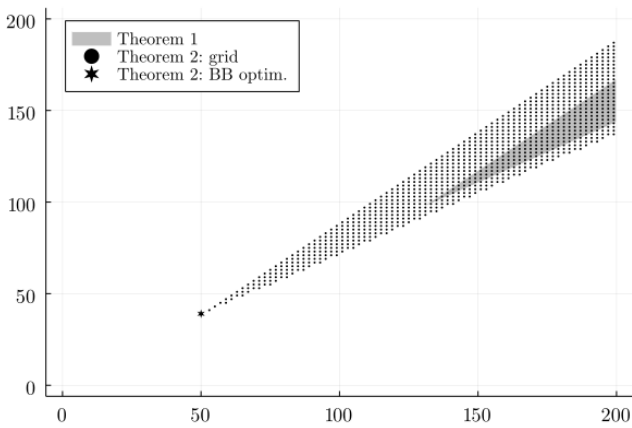


Fig. 2. Sets of gains (r_1, r_2) satisfying the convergence conditions of Theorem 1 (classical conditions) and the ones of Theorem 2 (less conservative) tested over a grid of 10,000 points. The star represents the solution obtained with the proposed Black-Box optimization process.

In Figure 2, the set of feasible gains described by the conditions of Theorem 2 is larger than the one described by the conditions of the classical conditions provided by Theorem 1. However, a grid search procedure is very inefficient to locate the point that minimizes $r_1 + r_2$.

We now focus on reducing the chattering effect by employing the two-level optimization process described in Section III to compute a pair of gains (r_1, r_2) that minimizes the function $f(r_1, r_2) = r_1 + r_2$, corresponding to the maximum amplitude of the control input. We select as initial point (r_1^0, r_2^0) for the NOMAD solver the point that minimizes f and enforces the convergence condition of Theorem 1. This point $(r_1^0, r_2^0) = (129, 96.5)$ corresponds to the leftmost point of the gray triangle in Figure 2. We run NOMAD with the default parameters and set the maximum number of blackbox evaluations, *i.e.*, solving the lower level problem, to 500. In 8 minutes, NOMAD produces the point $(47.1, 36.6)$ represented by the star in Figure 2.

The simulations of (17) with (3) tuned with these pairs of gains are performed using an explicit Euler method with 10^{-5} s time step. The control input is updated every 10^{-2} s to simulate the sampling time of a real system. The simulation results are displayed in Figure 3. The simulations show

that tuning the controller's gains using the new conditions decrease the chattering effect. Between 2 and 10 seconds (sliding motion set), the L2 norm of the error $(x_c - x)$ is decreases 59.99 % using the gains obtained with Theorem 2.

To conclude, using the new conditions, although computationally expensive, provides a larger set of gains than the classical conditions. Having a larger set provides more freedom in choosing the controller's gains, that can help reducing the chattering effect while guaranteeing the convergence to the sliding surface.

V. CONCLUSION

This paper showed how numerical optimization can be employed for tuning the twisting controller's gains. Using a global solver enabled to compute tight bounds on which are based the convergence conditions, and therefore helped in reducing the conservatism due to over-approximations of the bounds. Employing global optimization was necessary since the bounds are found by solving programs involving non-linear functions. Blackbox optimization has been used to solve the higher optimization problem consisting in finding optimal gains subject to finite time convergence. The study of a numerical example showed that using such optimization-based methods with the new convergence conditions of Theorem 2, despite being computationally expensive, enables more freedom in choosing controller's gain while preserving the finite-time convergence.

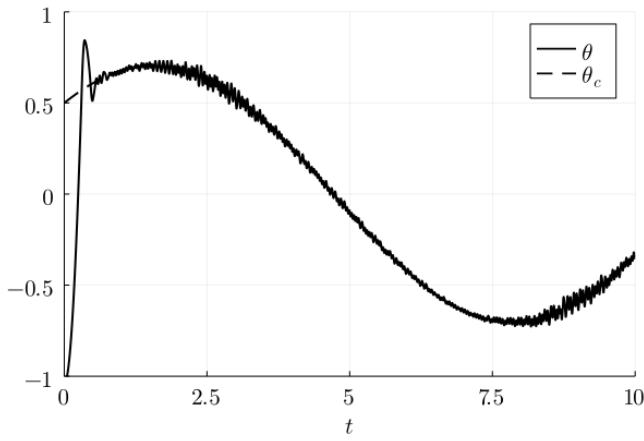
Future works include finding even less conservative conditions than the ones of Theorem 2 for the twisting algorithm, and extending this approach to other 2-SM controllers.

ACKNOWLEDGMENT

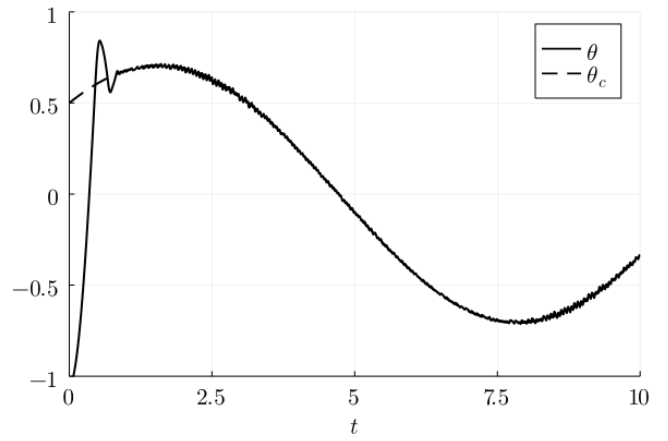
This work is supported by Région Pays de la Loire, Centrale Nantes, D-ICE engineering, West Atlantic Marine Energy Community (WEAMEC), Atlanstic 2020, LS2N and CNRS as part of the O2GRACE project. Part of this work is supported by the NSERC Alliance grant 544900-19 in collaboration with Huawei-Canada. The authors thank Sébastien Le Digabel for his help with the NOMAD solver.

REFERENCES

- [1] C. Audet and W. Hare. *Derivative-Free and Blackbox Optimization*. Springer Series in Operations Research and Financial Engineering. Springer International Publishing, Cham, Switzerland, 2017.
- [2] I. Boiko. Analysis of sliding modes in the frequency domain. *International Journal of Control*, 78(13):969–981, 2005.
- [3] G. Chabert and L. Jaulin. Contractor programming. *Artificial Intelligence*, 173(11):1079–1100, 2009.
- [4] A. Filippov. *Differential Equations with Discontinuous Righthand Sides: Control Systems*. Kluwer, Dordrecht, The Netherlands, 2010.
- [5] L. Fridman. Chattering analysis in sliding mode systems with inertial sensors. *International Journal of Control*, 76(9-10):906–912, 2003.
- [6] S. Le Digabel. Algorithm 909: Nomad: Nonlinear optimization with the mads algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 37(4):1–15, 2011.
- [7] A. Levant. Principles of 2-sliding mode design. *Automatica*, 43(4):576–586, 2007.
- [8] A. Levant. Chattering analysis. *IEEE Transactions on Automatic Control*, 55(6):1380–1389, 2010.
- [9] Dominique Monnet, Alexandre Goldsztejn, and Franck Plestan. Generalization of the twisting convergence conditions to non-affine systems. *Automatica*, 137:110019, 2022.



(a) Tuning with conditions of Theorem 1, $(r_1, r_2) = (129, 96.5)$.



(b) Tuning with conditions of Theorem 2, $(r_1, r_2) = (47.1, 36.6)$.

Fig. 3. Simulations performed with the smallest gains (r_1, r_2) obtained from the conditions of Theorem 1 and Theorem 2.

- [10] J. Moreno and M. Osorio. A Lyapunov approach to second-order sliding mode controllers and observers. In *IEEE Conference on Decision and Control*, pages 2856–2861, 2008.
- [11] A. Polyakov and A. Poznyak. Lyapunov function design for finite-time convergence analysis: “twisting” controller for second-order sliding mode realization. *Automatica*, 45(2):444–448, 2009.
- [12] J. L. Rosendo, D. Monnet, H. De Battista, J. Ninin, B. Clement, and F. Garelli. A global optimization approach for sliding mode tuning and existence maps generation. *International Journal of Dynamics and Control*, pages 1–13, 2020.
- [13] J. L. Rosendo, D. Monnet, H. De Battista, J. Ninin, B. Clement, and F. Garelli. A global optimization approach for sliding mode tuning and existence maps generation. *International Journal of Dynamics and Control*, 9(2):658–670, 2021.
- [14] N. V. Sahinidis. Baron: A general purpose global optimization software package. *Journal of global optimization*, 8(2):201–205, 1996.
- [15] Y. B. Shtessel, C. Edwards, L. Fridman, and A. Levant. *Sliding mode control and observation*. Springer, New York, USA, 2014.
- [16] V. Utkin, J. Guldner, and S. Jingxin. *Sliding mode control in electro-mechanical systems, Second Edition*. CRC press, 2009.