HAL
open science

Using Interval Analysis to Compute the Invariant Set of a Nonlinear Closed-Loop Control System

Swantje Romig, Luc Jaulin, Andreas Rauh

To cite this version:

Swantje Romig, Luc Jaulin, Andreas Rauh. Using Interval Analysis to Compute the Invariant Set of a Nonlinear Closed-Loop Control System. Algorithms, 2019, 12 (12), pp.262. 10.3390/a12120262 . hal-02434964

HAL Id: hal-02434964

https://ensta-bretagne.hal.science/hal-02434964v1

Submitted on 13 Dec 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.

algorithms

MDPI

Article

Using Interval Analysis to Compute the Invariant Set of a Nonlinear Closed-Loop Control System

Swantje Romig [1]*, Luc Jaulin [2] and Andreas Rauh [1,*]

[1] Chair of Mechatronics, University of Rostock, D-18059 Rostock, Germany
[2] Lab-STICC of ENSTA Bretange, 29806 Brest, France; lucjaulin@gmail.com
* Correspondence: swantje.romig@uni-rostock.de (S.R.); andreas.rauh@uni-rostock.de (A.R.)

Received: 30 October 2019; Accepted: 3 December 2019; Published: 6 December 2019

Abstract: In recent years, many applications, as well as theoretical properties of interval analysis have been investigated. Without any claim for completeness, such applications and methodologies range from enclosing the effect of round-off errors in highly accurate numerical computations over simulating guaranteed enclosures of all reachable states of a dynamic system model with bounded uncertainty in parameters and initial conditions, to the solution of global optimization tasks. By exploiting the fundamental enclosure properties of interval analysis, this paper aims at computing invariant sets of nonlinear closed-loop control systems. For that purpose, Lyapunov-like functions and interval analysis are combined in a novel manner. To demonstrate the proposed techniques for enclosing invariant sets, the systems examined in this paper are controlled via sliding mode techniques with subsequently enclosing the invariant sets by an interval based set inversion technique. The applied methods for the control synthesis make use of a suitably chosen Gröbner basis, which is employed to solve Bézout's identity. Illustrating simulation results conclude this paper to visualize the novel combination of sliding mode control with an interval based computation of invariant sets.

Keywords: interval analysis; Lyapunov stability; sliding mode control; invariant sets

1. Introduction

During the last few decades, interval analysis has been shown to provide efficient numerical approaches for solving various tasks in scientific computing, as well as in application-oriented disciplines in which numerical techniques with a result verification are helpful or required. In such a way, they help to solve problems in control engineering, robotics, including rigorous methods for localization and/or path planning, as well as guaranteed obstacle avoidance, computational (bio-)mechanics, civil engineering, or bio-engineering, just to mention a few of the large number of different areas [1–7].

In these applications, not only the possibility of interval analysis to treat the effects of round-off and truncation errors of numerical solution techniques (such as truncated temporal Taylor series expansions for the solution of initial value problems to ordinary differential equations) is exploited [8–10]. In addition, these applications also benefit from a further fundamental property of interval analysis, namely that it provides guaranteed outer enclosures of all possible solutions to a mathematical problem at hand in the presence of ambiguities in the solutions or in the presence of bounded uncertainty in selected parameters.

Here, a selection of useful techniques ranges from the extension of fundamental arithmetic operations to interval valued expressions [1–3], which were recently standardized in the IEEE standard 1788 [11], over the development of set valued counterparts for zero finding techniques for sets of algebraic equations such as the Krawczyk operator [12–14], interval based techniques for reachability analysis [15], the verified global optimization [16–18], to the solution of identification tasks by

means of set inversion techniques via interval analysis (e.g., the algorithms set inversion via interval analysis (SIVIA) and problem specific generalizations) [2,19]. If control applications are concerned, interval analysis can be applied both to an offline design and verification of control procedures under consideration of feasibility and safety requirements such as input and state constraints and to an online interval evaluation in terms of real-time capable robust control strategies generalizing the ideas of variable-structure control techniques and backstepping [20–25]. Further, approaches for the control of systems with input saturations that do not rely on interval analysis were handled, for example, in [26] and the references therein. In addition to the implementation of robust controllers, also the dual task of interval based state estimation and observer design can be solved; see, for example, [27–32].

This paper addresses the following two issues: First, a method for finding a nonlinear state-feedback controller is derived for the exact tracking of predefined paths in the phase-space. These paths are assumed to be described in terms of a sliding surface for a nonlinear dynamic system. Second, this control design procedure is interfaced with techniques from the field of interval analysis that are employed to determine enclosures of the invariant sets of the resulting controlled system in a reliable and computationally efficient way. For a continuous-time system model with the state vector $\boldsymbol{x} \in \mathbb{R}^n$, a subset $S \subset \mathbb{R}^n$ is called positively invariant if the relation $\boldsymbol{x}(t_1) \in S$ is satisfied for all possible $\boldsymbol{x}(t_0) \in S$ if $t_1 \geq t_0$ holds [33]. Due to the fact that we are only interested in a forward-in-time propagation of state trajectories, these sets are briefly called invariant throughout this paper. For the proposed control design, an appropriate Gröbner basis is used to check if the occurring Bézout identity has a solution, such that a controller can be designed that does not contain any structural singularities.

To find the invariant set of the respective closed-loop control system, Lyapunov-like functions are employed, which transform the stability properties of dynamical systems described by a set of ordinary differential equations into inequalities. Those inequalities can be evaluated using interval computation and finally lead to the desired enclosures of the invariant sets to be determined. Note that the proposed technique does not focus on determining invariant sets by using the La Salle invariance theorem. Instead, only temporal derivatives of the aforementioned Lyapunov-like functions are required during the solution procedure.

To depict the examined problem and to distinguish it from a state-of-the-art sliding mode control design [34–39], consider a single-input system:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), u(\boldsymbol{x}(t))), \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector and $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^n$ is the evolution function. For simplicity, assume that the input in (1) is given by the state dependent expression $u(\boldsymbol{x}(t)) \in \mathbb{R}$. The time dependencies of the state vector $\boldsymbol{x} := \boldsymbol{x}(t) \in \mathbb{R}^n$ and of the control vector $u := u(t)$ are furthermore not explicitly denoted. Additionally, consider a sliding surface:

$$s(\boldsymbol{x}) = 0\,, \quad \text{where} \quad s : \mathbb{R}^n \mapsto \mathbb{R}. \tag{2}$$

For the system model defined in Equation (1), the control input u should be found in such a way that the function $\frac{1}{2}s^2(\boldsymbol{x})$, which can be interpreted as a Lyapunov function candidate, decreases monotonically up to the point where perfect tracking of the predefined path (respectively, trajectory) occurs due to $s(\boldsymbol{x}) = 0$. As a consequence, the state trajectories of the controlled system will converge to an invariant set in which the sliding surface should be contained.

The paper is structured in the following way. Firstly, the general concept of sliding mode approaches for the control design of nonlinear systems is described in Section 2 with the aim of distinguishing the investigated problem from classical techniques in this field. Then, the proposed novel method for a sliding mode-type controller design is explained in Section 3 in combination with the use of a Gröbner basis to solve Bézout's identity. Section 4 deals with the conditions to be satisfied for an invariant set, which are evaluated numerically by using interval analysis as described in Section 5. In Section 6, several applications of the developed method are presented. These applications

include cases with sliding surfaces represented by closed curves in the phase space. Moreover, straight line segments are considered as well. In addition, a concluding example is investigated where Bézout's identity is not solvable. However, the interval based method for enclosing the resulting invariant set of the controlled system can still be employed successfully if a bounded control input for the considered nonlinear system of dimension three is analyzed. Finally, conclusions and an outlook toward future work together with an explanation of how the presented interval technique can be applied to other nonlinear control techniques are given in Section 7.

2. Classical Sliding Mode Control

The main concept in sliding mode control is to replace an nth order tracking problem by a stabilization problem of the order $(n-1)$ [40]. Looking at the dynamic system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, u), \tag{3}$$

the guaranteed stabilizing, variable structure input signal u is chosen according to:

$$u = \begin{cases} u^{+}, & \text{if } s(\boldsymbol{x}, t) > 0 \\ u_{\text{eq}}, & \text{if } s(\boldsymbol{x}, t) = 0 \\ u^{-}, & \text{if } s(\boldsymbol{x}, t) < 0. \end{cases} \tag{4}$$

For that purpose, a switching function $s(\boldsymbol{x}, t)$ needs to be defined. In classical sliding mode procedures, which are applied to solve trajectory tracking tasks, this switching function is usually composed of the output tracking error and a weighted linear combination of the temporal derivatives of this error signal up to the order $\delta - 1$, where δ denotes the system's relative degree [24]. In the remainder of this section, it is assumed that the relative degree δ equals the system order according to $n = \delta$. Then, a stability proof of the closed-loop dynamics can be performed in terms of a square Lyapunov function candidate as was mentioned in the Introduction of this paper.

In general, the switching function $s(\boldsymbol{x}, t)$ determines under which conditions the input signal of the system changes structurally, so that u becomes discontinuous across $s(\boldsymbol{x}, t) = 0$ [40,41]. In such a way, the sliding surface, which is defined by $s(\boldsymbol{x}, t) = 0$, divides the state space into two mutually disjoint domains where either $s(\boldsymbol{x}, t) > 0$ or $s(\boldsymbol{x}, t) < 0$ holds. In those domains, different dynamics of the system occur due to different definitions of the system's input u. The sliding surface is included in neither of these subdomains as a trajectory, whereas each point of the sliding surface $s(\boldsymbol{x}, t) = 0$ has to be accessible by a trajectory from either side [41]. To make the system converge to the sliding surface, the input has to be chosen in such a way that it always makes the state trajectories approach the sliding surface, i.e., they have to be "pushed" towards $s(\boldsymbol{x}, t) = 0$. This phase is called the transient or reaching phase. The possibilities for the design of controllers for this reaching phase can be found, for example, in [42,43] and the references therein. Note that the design of the corresponding control for this phase needs to account for input constraints in terms of the limited range of the admissible system inputs u. If scenarios occur, in which the system input reaches its lower or upper saturation value in such a phase, a careful stability proof is inevitable, cf. [26].

The next phase, following the completion of the reaching phase, is called sliding mode. The system's dynamics are then represented by the switching function itself. As mentioned before, this function typically has a lower order than the open-loop system [38,39].

Thus, in this state, the system's dynamic behavior only depends on the parameters included in the switching function, which means that this form of control is not only robust against uncertainties in plant parameters, but also against disturbances satisfying suitable matching conditions (for example, disturbances entering the state Equations (3) via the same channels as the system inputs u). From an intuitive point of view, this results from the fact that even if the system leaves the sliding surface

slightly, it will immediately return to it, because of the switching-type behavior of the input u and the proven stability (i.e., attraction property) of the sliding surface.

However, the switching between structurally different inputs as defined in (4) may lead to a phenomenon called chattering, which implies high control activity and thus leads to a rapid wear of the actuator. In addition, the associated high frequencies may also excite dynamics that were not considered in the model [40]. Strategies aiming at a reduction of the chattering phenomenon by a stability preserving interval based gain adaptation strategy were investigated in [20–25].

To ensure that the system will reach the sliding surface in finite time, the condition:

$$\frac{1}{2}\frac{\mathrm{d}s^2(\boldsymbol{x})}{\mathrm{d}t} = s(\boldsymbol{x}) \cdot \dot{s}(\boldsymbol{x}) \leq -\eta|s(\boldsymbol{x})|, \text{ where } \eta > 0, \tag{5}$$

is typically introduced in a classical sliding mode control synthesis. The inequality (5) means that the squared distance to the sliding surface has to decrease along all possible trajectories with a constant velocity η. This condition makes the set $s = 0$ an invariant set due to the invariance theorem of La Salle. If this condition is satisfied for the complete state space, the system will always converge to the sliding surface.

To reach the sliding mode, the state dependent controller has to be designed. There are various ways to do that, where we restrict ourselves to one of the available options. To deal with possible singularities in the control law and to avoid them during perfect trajectory tracking in the sliding mode, the input u is chosen according to the concept of equivalent control. To find the equivalent control u_{eq} that ensures perfect tracking, $\dot{s} = 0$ should be valid on the sliding surface in order for s not to change and thus the system to remain in sliding mode. For a system like (1) with a (nonlinear) feedback controller $u(\boldsymbol{x})$, the time derivative of $s(\boldsymbol{x})$ is calculated by the following Lie derivative of $s(\boldsymbol{x})$ in the direction of the vector field $\boldsymbol{f}(\boldsymbol{x}, u)$, assuming that $s(\boldsymbol{x})$ does not explicitly depend on the time variable t:

$$\dot{s}(\boldsymbol{x}) = \left(\frac{\partial s(\boldsymbol{x})}{\partial \boldsymbol{x}}\right)^T \cdot \boldsymbol{f}(\boldsymbol{x}, u). \tag{6}$$

To find the equivalent input $u := u_{\mathrm{eq}}$, this expression is solved with respect to u. However, if the system states are not on the sliding surface during the reaching phase, u should make it converge to $s(\boldsymbol{x}) = 0$. To ensure this behavior, a further term depending on the sign of s resulting from the right hand side of (5) is typically added to the equivalent control u_{eq} to obtain the overall control signal stated in Equation (4). This can be achieved by using a sign function (often regularized by means of smooth sigmoid-type expressions) depending on the current value of $s(\boldsymbol{x})$ [40,44].

3. Controller Design

To find the controller for the system model (1) with a time invariant sliding surface as considered in Equation (6), the following approach is used, which is similar to the one that can be found in [45] by setting:

$$\dot{s}(\boldsymbol{x}) = -s(\boldsymbol{x}) \cdot p(\boldsymbol{x}), \tag{7}$$

where $p(\boldsymbol{x}) > 0$. Note that this assumption is not only useful for the classical tracking control problem mentioned in the previous section, but it can also be employed to design control procedures allowing for following paths in the phase space that satisfy the design procedures listed in the following. From a dynamics point of view, the factor $p(\boldsymbol{x})$ in Equation (7) can be interpreted as a state dependent decay rate for the convergence of tracking errors towards the desired sliding mode.

With the choice given in Equation (7), $s(\boldsymbol{x})$ will be forced towards zero, because the resulting derivative $\dot{s}(\boldsymbol{x})$ always has the opposite sign of $s(\boldsymbol{x})$, leading to the strict inequality:

$$\frac{\mathrm{d}s^2(\boldsymbol{x})}{\mathrm{d}t} = s(\boldsymbol{x}) \cdot \dot{s}(\boldsymbol{x}) = -s^2(\boldsymbol{x}) \cdot p(\boldsymbol{x}) < 0 \quad \text{for all} \quad s(\boldsymbol{x}) \neq 0. \tag{8}$$

With the help of the relation (7), the control signal can be found in symbolic form. As an example, consider a two-dimensional input affine system of the form:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= c(\boldsymbol{x}) + u, \end{aligned} \tag{9}$$

for which Equation (7) becomes equal to:

$$\dot{s}(\boldsymbol{x}) = \frac{\partial s(\boldsymbol{x})}{\partial x_1}\dot{x}_1 + \frac{\partial s(\boldsymbol{x})}{\partial x_2}\dot{x}_2 = -s(\boldsymbol{x}) \cdot p(\boldsymbol{x}) \tag{10}$$

as well as equal to:

$$\dot{s}(\boldsymbol{x}) = \frac{\partial s(\boldsymbol{x})}{\partial x_1}x_2 + \frac{\partial s(\boldsymbol{x})}{\partial x_2}(c(\boldsymbol{x}) + u) = -s(\boldsymbol{x}) \cdot p(\boldsymbol{x}). \tag{11}$$

The latter expression can now be solved for the control input according to:

$$u = u(\boldsymbol{x}) = \left(-s(\boldsymbol{x}) \cdot p(\boldsymbol{x}) - \frac{\partial s(\boldsymbol{x})}{\partial x_1}x_2\right) \cdot \frac{1}{\frac{\partial s(\boldsymbol{x})}{\partial x_2}} - c(\boldsymbol{x}). \tag{12}$$

To avoid singularities in the controller and in this way in the closed-loop system dynamics, the following condition has to be satisfied:

$$-s(\boldsymbol{x}) \cdot p(\boldsymbol{x}) - \frac{\partial s(\boldsymbol{x})}{\partial x_1}x_2 = k(\boldsymbol{x})\frac{\partial s(\boldsymbol{x})}{\partial x_2}, \tag{13}$$

so that all fractions in the expression for the input u cancel out analytically. After simple rearrangement, this expression becomes:

$$\frac{\partial s(\boldsymbol{x})}{\partial x_2}k(\boldsymbol{x}) + s(\boldsymbol{x}) \cdot p(\boldsymbol{x}) = -\frac{\partial s(\boldsymbol{x})}{\partial x_1}x_2, \tag{14}$$

which is Bézout's identity, where $p(\boldsymbol{x})$ and $k(\boldsymbol{x})$ have to be found in terms of non-singular analytic (polynomial) expressions.

Remark 1. *The design procedure summarized above is more general than the classical sliding mode design, where the goal of tracking sufficiently smooth reference trajectories $y_\mathrm{d} := y_\mathrm{d}(t)$ for the desired temporal evolution of the state variable x_1 would lead to the system input:*

$$u = \alpha \cdot (\dot{y}_\mathrm{d} - x_2) + \ddot{y}_\mathrm{d} + \eta \cdot \mathrm{sign}\,(\tilde{s}(\boldsymbol{x}, t)) - c(\boldsymbol{x}). \tag{15}$$

Here, the parameter α is chosen as a positive constant (generally satisfying the Hurwitz criterion [20,22]) to ensure stability on the sliding surface and $\eta > 0$ ensures a stable motion during the reaching phase towards the classical (time varying) sliding surface:

$$\tilde{s}(\boldsymbol{x}, t) = \alpha \cdot (y_\mathrm{d} - x_1) + (\dot{y}_\mathrm{d} - x_2) = 0. \tag{16}$$

To solve the Bézout identity (14) in the design procedure suggested in this paper, it has to be checked firstly whether a solution for this polynomial equation exists. This can be done using a reduced Gröbner basis. If a polynomial belongs to an ideal, it can be expressed as a linear combination of the polynomials that belong to the generating set of the ideal in a polynomial ring. This means if the term:

$$-\frac{\partial s(\boldsymbol{x})}{\partial x_1}x_2 \tag{17}$$

included in the system input defined in Equation (12) is part of the ideal:

$$I_1 = \left\langle s(\boldsymbol{x}), \frac{\partial s(\boldsymbol{x})}{\partial x_2} \right\rangle, \tag{18}$$

Equation (14) has a solution.

The Gröbner basis can provide this generating set of polynomials, and the reduced Gröbner basis makes sure that none of the leading coefficients of the polynomials are divided by one another. All polynomials belonging to this ideal can be expressed by linear combinations of the reduced Gröbner basis elements.

Therefore, if the reduced Gröbner bases of I_1 and:

$$I_2 = \left\langle s(\boldsymbol{x}), \frac{\partial s(\boldsymbol{x})}{\partial x_2}, -\frac{\partial s(\boldsymbol{x})}{\partial x_1} x_2 \right\rangle \tag{19}$$

are the same, Equation (14) has a solution. As soon as a first solution for $p(\boldsymbol{x})$ and $k(\boldsymbol{x})$ has been found, multiple further solutions can be obtained by exploiting the fact that if:

$$p(\boldsymbol{x}) \cdot G_0 + k(\boldsymbol{x}) \cdot G_1 = B(\boldsymbol{x}) \tag{20}$$

holds true, it directly implies that also the equality:

$$\begin{gathered} (p(\boldsymbol{x}) + Y(\boldsymbol{x}) \cdot G_1) \cdot G_0 + (k(\boldsymbol{x}) - Y(\boldsymbol{x}) \cdot G_0) \cdot G_1 = \\ p(\boldsymbol{x}) \cdot G_0 + Y(\boldsymbol{x}) \cdot G_0 \cdot G_1 + k(\boldsymbol{x}) \cdot G_1 - Y(\boldsymbol{x}) \cdot G_0 \cdot G_1 = B(\boldsymbol{x}) \end{gathered} \tag{21}$$

is satisfied, where G_0 and G_1 are the elements of the Gröbner basis [46,47].

4. Computation of the Invariant Set of the Closed-Loop Control System

Once the controller $u(\boldsymbol{x})$ in Equation (12) is found, it should be shown that the system that converges to the sliding surface $s(\boldsymbol{x}) = 0$ that is characterized by (7) stays on the surface for all times. This is the definition of an invariant set due to La Salle. If a trajectory enters an invariant set, it will stay inside this set forever [33,40]. To find this set, the switching function $s(\boldsymbol{x})$ is used as a Lyapunov-like function $\tilde{V}(\boldsymbol{x}) = s(\boldsymbol{x})$, which does not necessarily comply with all requirements of a Lyapunov function, as its derivative may also be zero for states $\boldsymbol{x}$, where $V = \frac{1}{2}s^2(\boldsymbol{x}) \neq 0$. Moreover, the function $\tilde{V}(\boldsymbol{x})$ may exhibit a change of sign.

Under this preliminary consideration of the characteristics of invariant sets, states are not part of such sets if they are related to an inflection point in the form of a saddle point of the function $V(\boldsymbol{x})$ (see Figure 1).

In fact, it suffices if any finite time derivative of $V(\boldsymbol{x})$ is not equal to zero for a point not belonging to the invariant set. This can also be confirmed when thinking of the Barbalat lemma, which states that, if a function has a finite limit for $t \to \infty$, its uniformly continuous derivative will tend to zero [40]. This can also be applied for the first derivative being the function and the second derivative being its first derivative, and so on:

$$\begin{gathered} \lim_{t\to\infty} V(\boldsymbol{x}) = c \implies \lim_{t\to\infty} \dot{V}(\boldsymbol{x}) = 0 \implies \\ \lim_{t\to\infty} \ddot{V}(\boldsymbol{x}) = 0 \implies \lim_{t\to\infty} \dddot{V}(\boldsymbol{x}) = 0 \ldots, \\ \text{where } c \text{ is a constant.} \end{gathered} \tag{22}$$

Using the fact that the desired sliding surface $s(\boldsymbol{x})$ itself and the first three derivatives of $s(\boldsymbol{x})$ have to be zero as a necessary condition if $V(\boldsymbol{x}) = 0$ holds constantly, the invariant limit set of the system can be found.

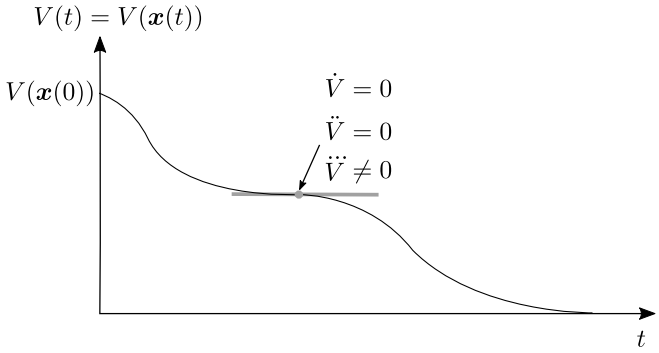

Figure 1. Possible behavior of $V(t) = V(\boldsymbol{x}(t))$.

This statement can be verified by means of the following relations:

$$V(\boldsymbol{x}) = \frac{1}{2}s^2(\boldsymbol{x}) = 0 \implies s(\boldsymbol{x}) = 0 \tag{23}$$

$$\dot{V}(\boldsymbol{x}) = s(\boldsymbol{x}) \cdot \dot{s}(\boldsymbol{x}) = 0 \implies s(\boldsymbol{x}) = 0 \tag{24}$$

satisfied due to (23)

$$\ddot{V}(\boldsymbol{x}) = s(\boldsymbol{x}) \cdot \ddot{s}(\boldsymbol{x}) + \dot{s}^2(\boldsymbol{x}) = 0 \implies s(\boldsymbol{x}) = 0 \tag{25}$$

satisfied due to (23) $\wedge$ $\dot{s}(\boldsymbol{x}) = 0$

$$\dddot{V}(\boldsymbol{x}) = s(\boldsymbol{x}) \cdot \dddot{s}(\boldsymbol{x}) + 3\dot{s}(\boldsymbol{x}) \cdot \ddot{s}(\boldsymbol{x}) = 0 \implies s(\boldsymbol{x}) = 0 \ \wedge \ \dot{s}(\boldsymbol{x}) = 0. \tag{26}$$

satisfied due to (23) and (25).

Requiring furthermore that $\ddot{s}(\boldsymbol{x}) = 0$ and $\dddot{s}(\boldsymbol{x}) = 0$ hold, all further higher order derivatives of $V(\boldsymbol{x})$ up to order seven are directly proven to be equal to zero.

Moreover, accounting for the fact that $\dot{s}(\boldsymbol{x}) = -s(\boldsymbol{x}) \cdot p(\boldsymbol{x})$ holds by construction (see Equation (7)), the relations (23)–(25) above turn into:

$$V(\boldsymbol{x}) = \frac{1}{2}s^2(\boldsymbol{x}) = 0 \implies s(\boldsymbol{x}) = 0 \tag{27}$$

$$\dot{V}(\boldsymbol{x}) = -s^2(\boldsymbol{x}) \cdot p(\boldsymbol{x}) = 0 \implies s(\boldsymbol{x}) = 0 \tag{28}$$

satisfied due to (27)

$$\ddot{V}(\boldsymbol{x}) = 2s^2(\boldsymbol{x}) \cdot p^2(\boldsymbol{x}) - s^2(\boldsymbol{x}) \cdot \dot{p}(\boldsymbol{x}) = 0 \implies s(\boldsymbol{x}) = 0, \tag{29}$$

satisfied due to (27), where $\ddot{V}(\boldsymbol{x})$ in (29) can be rewritten into:

$$\ddot{V}(\boldsymbol{x}) = s^2(\boldsymbol{x}) \cdot \left(2p^2(\boldsymbol{x}) - \dot{p}(\boldsymbol{x})\right). \tag{30}$$

Hence, the equivalent control designed in this paper guarantees that all temporal derivatives of $V(\boldsymbol{x})$ are perfectly equal to zero as soon as $s(\boldsymbol{x}) = 0$ has been reached. Enforcing furthermore that $\dot{s}(\boldsymbol{x}) = \ddot{s}(\boldsymbol{x}) = \dddot{s}(\boldsymbol{x}) = 0$ (involved in (23)–(26)) holds after expressing these terms as functions of the controlled system dynamics on the invariant set during the interval based solution procedure in Section 5 helps to improve the computational efficiency by means of defining so-called contractors that quickly shrink the search space for the invariant set towards the actual solution domain.

Remark 2. *If an invariant set is computed according to the definition:*

$$S_1 = \{\boldsymbol{x} | \dot{V}(\boldsymbol{x}) = \ddot{V}(\boldsymbol{x}) = \dddot{V}(\boldsymbol{x}) = 0\}, \tag{31}$$

it is not guaranteed that the system converges under all circumstances to the set:

$$S_0 = \{\boldsymbol{x} | V(\boldsymbol{x}) = \dot{V}(\boldsymbol{x}) = \ddot{V}(\boldsymbol{x}) = \dddot{V}(\boldsymbol{x}) = 0\}, \tag{32}$$

where the system is on the sliding surface. If the condition $s(\boldsymbol{x}) = 0$ is not enforced explicitly in the interval based procedure described in the following section, the set S_1 may not only consist of the invariant set that is characterized by the sliding mode, but may also contain further fixed points.

For example, for systems of dimension two, in which the examined sliding surface describes a closed trajectory in the state space in terms of a limit cycle, the Poincaré index theorem can be considered to visualize this fact. This theorem states that every limit cycle encloses at least one equilibrium point of the system, which will therefore also belong to the invariant set defined by S_1 [40].

5. Interval Computation

The outer interval enclosure of the invariant set S_1 of the controlled system in $\boldsymbol{x}$, denoted exemplarily by (x_1, x_2), is calculated using interval computation. This is possible due to the fact that the Lyapunov-like functions defined before provide inequalities. Therefore, it is checked for subintervals (i.e, boxes) covering the operating domain of interest if the conditions:

$$\begin{aligned} &s(\boldsymbol{x}) = 0 \text{ (which is not included in the definition of } S_1\text{, but only in } S_0\text{)}, \\ &\dot{s}(\boldsymbol{x}) = 0, \\ &\ddot{s}(\boldsymbol{x}) = 0, \text{ and} \\ &\dddot{s}(\boldsymbol{x}) = 0 \end{aligned} \tag{33}$$

are satisfied in a domain that is a priori restricted to the set $s^2(\boldsymbol{x}) \leq \mu, \mu > 0$. This last condition restricts the search space to a maximum admissible squared distance from the sliding surface.

To solve a problem according to the specification (33), a general approach for interval techniques aiming at set inversion should be briefly described. The following description is based on Figure 2. In the examined case, a set is known in the domain of (y_1, y_2), and the set that should be found by the inversion technique is located in the (x_1, x_2)-plane. The function $\boldsymbol{f}$ performing the mapping, $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^2, \boldsymbol{y} \in \mathbb{R}^2$, from the (x_1, x_2)-plane to the (y_1, y_2)-plane is assumed to be known together with a suitable inclusion function $[\boldsymbol{f}]$ that finds a rectangular box that encloses the set (gray). This function will not necessarily lead to the tightest enclosing box, but it is convergent with respect to interval bisections. The inverse mapping of $\boldsymbol{f}$ does not necessarily have to be known explicitly, so (x_1, x_2) either does not need to be or even cannot be determined analytically.

Starting with a sufficiently large box $[\boldsymbol{x}_0]$ in the (x_1, x_2)-plane that encloses the whole solution domain of interest, the corresponding box in (y_1, y_2) is calculated. It is then checked if it is located either outside the set (blue), inside the set (red), or if it is overlapping with both, the outer and interior domains, i.e., if the location is undecided (yellow). Whenever it is not sure if the box is inside or outside, the box in the (x_1, x_2)-plane will be split at the midpoint of each edge, and the procedure will be repeated for each of the new boxes. This is continued until a certain accuracy is reached. The red boxes will represent the corresponding set in the (x_1, x_2)-plane. This algorithm is called set inversion via interval analysis (SIVIA) [2].

Note that interval computations are guaranteed in the following sense: If a box in the (x_1, x_2)-plane has been found that is mapped into the interior of $\mathcal{Y}$, it is guaranteed to be part of the true solution set $\mathcal{Y}$, and vice versa, that if a box lies outside the solution set $\mathcal{Y}$ in the (y_1, y_2)-plane, its counterpart in (x_1, x_2) certainly contains no admissible solutions. If both cases are included in one box, it will, as mentioned before, either be split or, if the volume of the respective hypercuboid falls below some threshold value, be labeled as undecided.

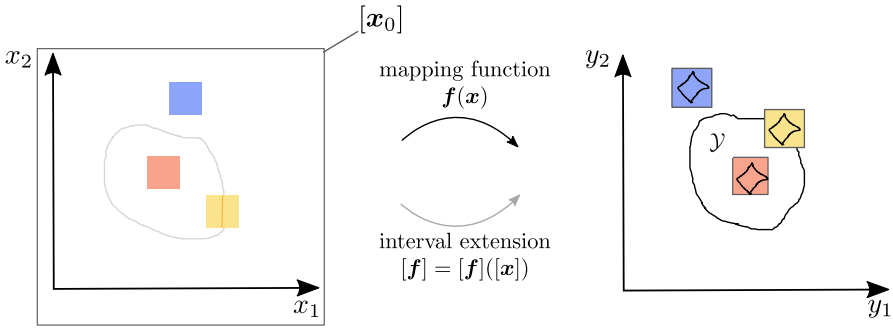

Figure 2. Visualization of the set inversion via interval analysis (SIVIA) algorithm.

As this successive bisectioning scheme leads to lots of boxes that have to be checked (2^n boxes for a simultaneous subdivision of an n-dimensional box in each of its edges) the concept of contractors can effectively be applied to shrink a box as far as possible to the solution set. Here, the interval contraction is performed by means of each component of the vector valued function $\boldsymbol{f}$.

A contractor can be built by looking at the syntactic tree of the function $\boldsymbol{f}$, which means a decomposition into primitive equations by defining auxiliary variables [2]. In this way, the intervals can be contracted in a forward-backward manner. The forward part is where the definitions of the auxiliary variables are executed, and the backward part is where the original intervals are intersected with the newly calculated ones through the auxiliary variables' definition.

This type of algorithm is described in more detail in [2] and has been employed for an interval based parameter identification scheme in [19]. Using this contractor, the first box will be blue, and inside this box, there will be a yellow one that is the smallest one to contain the whole solution set included in the first blue box, if the contractor is minimal. Then, the yellow box containing the solution set is split, and the procedure is repeated for each of the boxes. In this way, the boxes become smaller more quickly than in the case where every box is repeatedly divided. Figure 3 illustrates this fact [2].

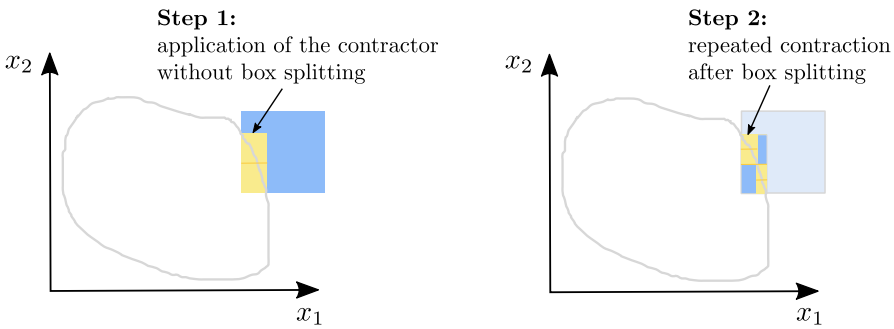

Figure 3. Visualization of the effect of contractors in SIVIA-like algorithms.

To transfer this contractor approach to the problem defined in this paper according to Equation (33), the following analogues are exploited: (i) the first condition $\tilde{V} = 0$ corresponds to $\boldsymbol{f}$; (ii) (x_1, x_2) is generalized in terms of the n-dimensional state vector; (iii) the value of zero is considered as the known set in the image space $\boldsymbol{y}$. Analogously, contractors are also built for the further derivatives in (33). All of these contractors are intersected with each other because all conditions should be satisfied simultaneously to provide an (outer) interval enclosure of the system's invariant set.

6. Application Scenarios

6.1. Pendulum Motion on a Circle in the Phase Plane

A pendulum with velocity proportional linear friction, as well as a nonlinear restoring sine-type force shall be described by the following state equations:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\sin(x_1) - x_2 + u, \end{aligned} \tag{34}$$

where x_1 is the angle, x_2 the angular velocity, and u is the control torque. A controller $u(\boldsymbol{x})$ should be found to make the pendulum move on a circle in the phase plane. This means, according to the sliding mode control design presented in Section 3, the switching function $s(\boldsymbol{x})$ is defined as:

$$s(\boldsymbol{x}) = x_1^2 + x_2^2 - 1 \tag{35}$$

describing a circle with a radius of one around the origin.

6.1.1. Controller Design

From Equations (12) and (35), the control input can be found as:

$$u = \left(-\left(x_1^2 + x_2^2 - 1\right) \cdot p(\boldsymbol{x}) - 2x_1x_2\right) \cdot \frac{1}{2x_2} + \sin(x_1) + x_2. \tag{36}$$

The Bézout identity to solve is hence:

$$\left(x_1^2 + x_2^2 - 1\right) \cdot p(\boldsymbol{x}) + 2x_2k(\boldsymbol{x}) = -2x_1x_2. \tag{37}$$

The reduced Gröbner bases for both ideals:

$$I_1 = \left\langle x_1^2 + x_2^2 - 1, 2x_2 \right\rangle \quad \text{and} \quad I_2 = \left\langle x_1^2 + x_2^2 - 1, 2x_2, -2x_1x_2 \right\rangle \tag{38}$$

are the same as:

$$G_1 = G_2 = \left\{x_1^2 - 1, x_2\right\}. \tag{39}$$

This means there is a solution for Equation (37). Choose, e.g., $p(\boldsymbol{x}) = 4x_2^2 = \left(\frac{\partial s}{\partial x_2}\right)^2$, which makes:

$$k(\boldsymbol{x}) = -2x_1^2x_2 - x_1 - 2x_2^3 + 2x_2 \tag{40}$$

and, finally, leads to the nonlinear equivalent control:

$$u = -2x_2 \cdot \left(x_1^2 + x_2^2 - 1\right) - x_1 + \sin(x_1) + x_2. \tag{41}$$

6.1.2. Verification of Stability

The stability of the closed-loop control structure can be verified by computing the total time derivative of the sliding surface $s(\boldsymbol{x})$ defined in (35) according to:

$$\begin{aligned}\dot{s}(\boldsymbol{x}) &= \frac{\partial s(\boldsymbol{x})}{\partial x_1} \cdot \dot{x}_1 + \frac{\partial s(\boldsymbol{x})}{\partial x_2} \cdot \dot{x}_2 \\ &= 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \\ &= 2x_1x_2 + 2x_2 \cdot \left(-\sin(x_1) - x_2 + \left(-\left(x_1^2 + x_2^2 - 1 \right) \cdot p(\boldsymbol{x}) - 2x_1x_2 \right) \cdot \frac{1}{2x_2} + \sin(x_1) + x_2 \right) \\ &= -s(\boldsymbol{x}) \cdot p(\boldsymbol{x}) \quad \text{with} \quad p(\boldsymbol{x}) = 4x_2^2 > 0 \quad \text{for} \quad s(\boldsymbol{x}) \neq 0 \ \wedge \ x_2 \neq 0 \end{aligned} \tag{42}$$

along the trajectories of the controlled system. Hence, $\dot{V}(\boldsymbol{x}) = s(\boldsymbol{x}) \cdot \dot{s}(\boldsymbol{x}) < 0$ is guaranteed for all states that do not belong to the desired invariant set, except for the fixed point at the origin of the phase space.

Remark 3. *The control law defined in (41) ensures perfect trajectory tracking as soon as the system has reached the sliding surface $s(\boldsymbol{x}) = 0$. However, it does not possess the variable structure characteristics defined in (5). Therefore, typically, it only leads to an asymptotic convergence towards the desired invariant set. However, here, and analogously in all other examples shown in the remainder of the paper, finite time convergence towards $s(\boldsymbol{x}) = 0$ can be achieved if u is replaced by:*

$$\tilde{u} = u - \frac{\eta}{2x_2} \cdot \operatorname{sign}(s(\boldsymbol{x})) \quad \textit{with} \quad \eta > 0 \tag{43}$$

for states that are not contained in the invariant set. Then, the control $\tilde{u}$ turns into the equivalent control u as soon as $s(\boldsymbol{x}) = 0$ starts to hold, and hence, the computation procedure for invariant sets derived in Section 4 becomes again applicable.

6.1.3. Computation of the Invariant Set

The invariant set S_1 of the closed-loop system, where (33) is true, can be calculated via interval computation (see Figure 4 in which the same color code as in Figure 3 is employed).

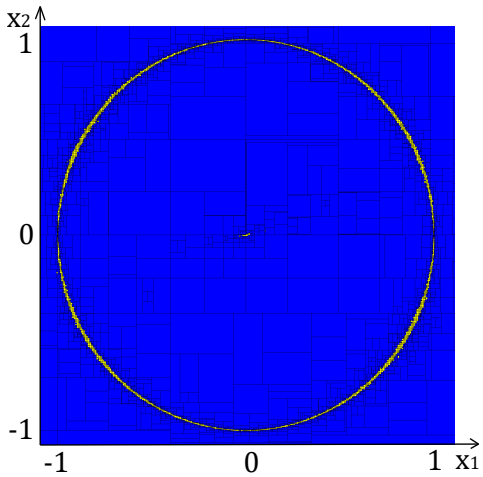

Figure 4. Invariant set of the pendulum, where the sliding surface is a circle.

The vector field in Figure 5 supports this result. As is shown in these figures, the pendulum converges to the circle or the equilibrium point at the origin, that is inside the circle due to the Poincaré fixed point theorem.

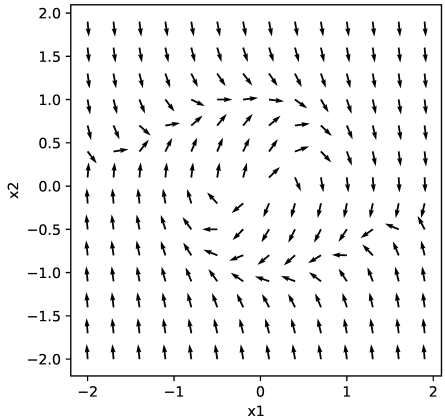

Figure 5. Vector field of the controlled pendulum, where the sliding surface is a circle.

6.2. Pendulum Motion on an Ellipse in the Phase Plane

The pendulum described by Equation (34) will now move on an ellipse in the phase plane. Thus, the switching function (35) is replaced by:

$$s(\boldsymbol{x}) = x_1^2 + x_2^2 + x_1 x_2 - 1. \tag{44}$$

Controller Design

The Bézout identity to be solved for this case is:

$$(2x_2 + x_1) \cdot k(\boldsymbol{x}) + \left(x_1^2 + x_2^2 + x_1 x_2 - 1\right) \cdot p(\boldsymbol{x}) = -\left(2x_1 + x_2\right) \cdot x_2. \tag{45}$$

As the reduced Gröbner bases:

$$G_1 = \left\{x_1 + 2x_2, 3x_2^2 - 1\right\} \quad \text{and} \quad G_2 = \{1\} \tag{46}$$

are not the same, there is no solution for an input without singularity, and thus, the pendulum cannot be made to move on this specific ellipse.

6.3. Pendulum with an Iso-Energy Constraint

Now, the pendulum should keep a constant energy. The switching function is then defined as a combination of kinetic and potential energy:

$$E(\boldsymbol{x}) = x_2^2 - \cos(x_1) + 1 = 2 \tag{47}$$

leading to:

$$s(\boldsymbol{x}) = x_2^2 - \cos(x_1) - 1. \tag{48}$$

6.3.1. Controller Design

Due to Equation (12), the controller is now parameterized by:

$$u = \frac{1}{2x_2}\left(-\left(x_2^2 - \cos(x_1) - 1\right) \cdot p(\boldsymbol{x}) - \sin(x_1) \cdot x_2\right) + \sin(x_1) + x_2. \tag{49}$$

As this is not a polynomial equation, the Bézout identity is not of any help. However, luckily, due to the surface definition, no singularity occurs, if $p(\boldsymbol{x})$ is chosen as $p(\boldsymbol{x}) = 2x_2^2$.

6.3.2. Verification of Stability

The stability of the closed-loop control structure can be verified by computing the total time derivative of the sliding surface $s(\boldsymbol{x})$ defined in (48) according to:

$$\begin{aligned}\dot{s}(\boldsymbol{x}) &= \frac{\partial s(\boldsymbol{x})}{\partial x_1} \cdot \dot{x}_1 + \frac{\partial s(\boldsymbol{x})}{\partial x_2} \cdot \dot{x}_2 \\ &= \sin(x_1) \cdot \dot{x}_1 + 2x_2\dot{x}_2 \\ &= \sin(x_1) \cdot x_2 + 2x_2 \cdot \Big(-\sin(x_1) - x_2 \\ &\qquad + \frac{1}{2x_2}\Big(-\Big(x_2^2 - \cos(x_1) - 1\Big) \cdot p(\boldsymbol{x}) - \sin(x_1) \cdot x_2 \Big) + \sin(x_1) + x_2 \Big) \\ &= -s(\boldsymbol{x}) \cdot p(\boldsymbol{x}) \quad \text{with} \quad p(\boldsymbol{x}) = 2x_2^2 > 0 \quad \text{for} \quad s(\boldsymbol{x}) \neq 0 \ \wedge \ x_2 \neq 0\end{aligned} \tag{50}$$

along the trajectories of the controlled system. Hence, $\dot{V}(\boldsymbol{x}) = s(\boldsymbol{x}) \cdot \dot{s}(\boldsymbol{x}) < 0$ is guaranteed for all states that do not belong to the desired invariant set, except for the fixed point at the origin of the phase space.

6.3.3. Computation of the Invariant Set

The invariant set S_1 for the iso-energy constraint of the controlled pendulum is depicted in Figure 6.

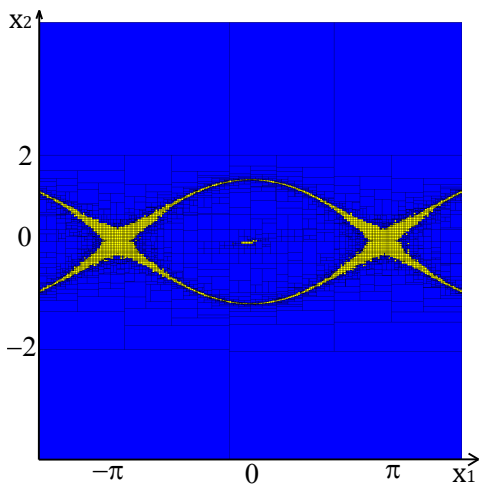

Figure 6. Invariant set of the pendulum, where the sliding surface corresponds to an iso-energy.

This corresponds to the pendulum turning all around, and from the top $x_1 = \pm\pi$, it is possible to turn to either of the sides. Similarly, as in Figure 4, a fixed point at the origin of the phase space was verified by means of the interval evaluation in this scenario.

6.4. Stabilization of the Pendulum Motion towards Zero

In this case, the pendulum should be brought to its naturally stable equilibrium, where:

$$x_1 = 0 \ \wedge \ x_2 = 0. \tag{51}$$

Therefore, the switching function is given as:

$$s(\boldsymbol{x}) = x_1 + x_2, \tag{52}$$

which corresponds to a straight line passing through the origin. This is a typical choice of s to bring the system to the origin.

6.4.1. Controller Design

Due to Equation (12), the control input has to be chosen as:

$$u = (-(x_1 + x_2) \cdot p(\boldsymbol{x}) - x_2) + \sin(x_1) + x_2. \tag{53}$$

Then, the Bézout identity trivially yields:

$$(x_1 + x_2) \cdot p(\boldsymbol{x}) + k(\boldsymbol{x}) = -x_2. \tag{54}$$

Choose $p(\boldsymbol{x}) = 1$, such that the input:

$$u = \sin(x_1) - x_2 - x_1 \tag{55}$$

is obtained.

6.4.2. Verification of Stability

The stability of the closed-loop control structure can be verified by computing the total time derivative of the sliding surface $s(\boldsymbol{x})$ defined in (52) according to:

$$\begin{aligned} \dot{s}(\boldsymbol{x}) &= \dot{x}_1 + \dot{x}_2 \\ &= x_2 - \sin(x_1) - x_2 + \sin(x_1) - x_2 - x_1 = -s(\boldsymbol{x}) \end{aligned} \tag{56}$$

along the trajectories of the controlled system. Hence, $\dot{V}(\boldsymbol{x}) = -s^2(\boldsymbol{x}) < 0$ is guaranteed for all states that do not belong to the desired invariant set.

6.4.3. Computation of the Invariant Set

The invariant set for this case is depicted in Figure 7. As can be seen, the whole line is part of the invariant set, implying that once on the surface, the system will stay there for ever. To see that the pendulum will actually converge to its equilibrium point, the vector field in Figure 8 is helpful. It can be seen that once on the surface, the trajectories tend to the origin.

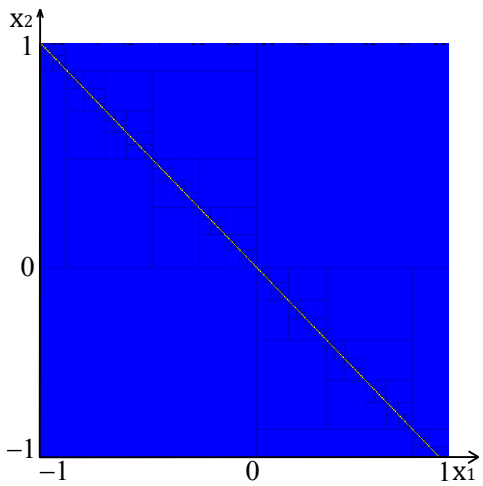

Figure 7. Invariant set of the pendulum, where the sliding surface is a straight line through the origin.

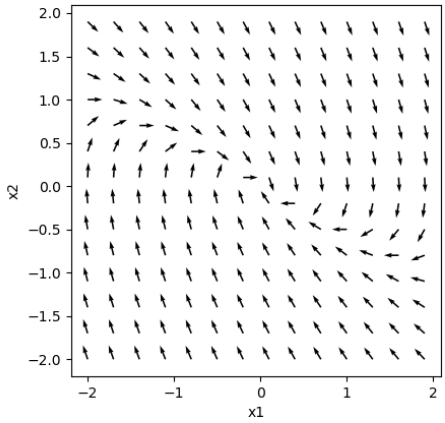

Figure 8. Vector field of the pendulum, where the sliding surface is a straight line.

6.5. Line Following for a Dubins Car

As a final example, consider the Dubins car [45], which is described by the state equations:

$$\begin{aligned} \dot{x}_1 &= \cos(x_3) \\ \dot{x}_2 &= \sin(x_3) \\ \dot{x}_3 &= u, \end{aligned} \tag{57}$$

where x_3 is the heading of the car and (x_1, x_2) are the position coordinates of its center. It should follow a line $x_2 = 0$, which is defined by the switching function:

$$s(\boldsymbol{x}) = x_3 + \arctan(x_2), \tag{58}$$

with the heading being zero, if the car is on $x_2 = 0$ and $\arctan(x_2) = 0$ for $x_2 = 0$. The saturation function arctan is used to exclude the (unbounded) case $x_2 = -x_3 \neq 0$ from the surface.

6.5.1. Controller Design

Due to Equation (7), the system's input in terms of a nonlinear feedback controller becomes:

$$u = -\left(x_3 + \arctan(x_2)\right) \cdot p(\boldsymbol{x}) - \frac{\sin(x_3)}{1 + x_2^2}. \tag{59}$$

6.5.2. Verification of Stability

Stability of the closed-loop control structure can be verified by computing the total time derivative of the sliding surface $s(\boldsymbol{x})$ defined in (58) according to:

$$\begin{aligned} \dot{s}(\boldsymbol{x}) &= \frac{\dot{x}_2}{1 + x_2^2} + \dot{x}_3 \\ &= \frac{\sin(x_3)}{1 + x_2^2} - \left(x_3 + \arctan(x_2)\right) \cdot p(\boldsymbol{x}) - \frac{\sin(x_3)}{1 + x_2^2} \\ &= -s(\boldsymbol{x}) \cdot p(\boldsymbol{x}) \quad \text{with} \quad p(\boldsymbol{x}) > 0 \quad \text{for} \quad s(\boldsymbol{x}) \neq 0 \end{aligned} \tag{60}$$

along the trajectories of the controlled system. Hence, $\dot{V}(\boldsymbol{x}) = s(\boldsymbol{x}) \cdot \dot{s}(\boldsymbol{x}) < 0$ is guaranteed for all states that do not belong to the desired invariant set.

6.5.3. Consideration of Input Range Constraints

As a new aspect, range constraints for this input should be examined, as it is not a polynomial equation and thus not Bézout's identity. Let $u \in [-1\ ;\ 1]$. This makes:

$$p(\boldsymbol{x}) \in \frac{\sin(x_3)}{-(x_3 + \arctan(x_2)) \cdot (1 + x_2^2)} + \frac{1}{x_3 + \arctan(x_2)} \cdot [-1\ ;\ 1]. \tag{61}$$

The extremal choice for $p(\boldsymbol{x})$ is:

$$p(\boldsymbol{x}) = \frac{\sin(x_3)}{-(x_3 + \arctan(x_2)) \cdot (1 + x_2^2)} + \frac{1}{|x_3 + \arctan(x_2)|} > 0. \tag{62}$$

The extremal choice of $p(\boldsymbol{x})$ leads to the largest absolute value of u. If this is inserted in u, the result is:

$$\begin{aligned} u &= -(x_3 + \arctan(x_2)) \cdot \left(\frac{\sin(x_3)}{-(x_3 + \arctan(x_2)) \cdot (1 + x_2^2)} + \frac{1}{|x_3 + \arctan(x_2)|} \right) - \frac{\sin(x_3)}{1 + x_2^2} \\ &= \frac{-(x_3 + \arctan(x_2))}{|x_3 + \arctan(x_2)|} \\ &= -\mathrm{sign}(x_3 + \arctan(x_2)) \ . \end{aligned} \tag{63}$$

To make the input differentiable, the hard switching function is replaced by:

$$u = -\tanh(x_3 + \arctan(x_2)). \tag{64}$$

6.5.4. Computation of the Invariant Set

As this example is three dimensional, the result is projected into the (x_1, x_2)-plane. As can be seen in Figure 9, the invariant set is the line, where $x_2 = 0$ as desired.

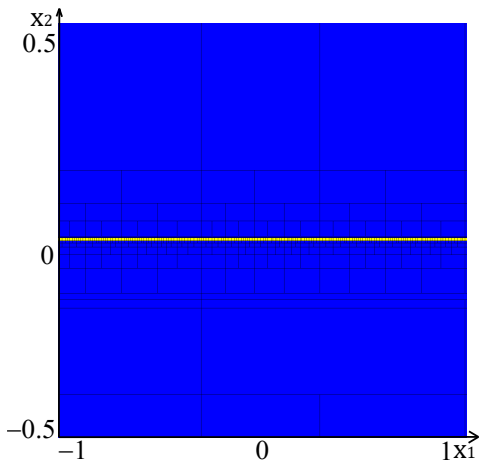

Figure 9. Projection of the invariant set for the Dubins car into the (x_1, x_2)-plane.

The control input (64) stays within its required bounds for a simulation with randomly chosen initial conditions $x(0)$, as can be seen in Figure 10.

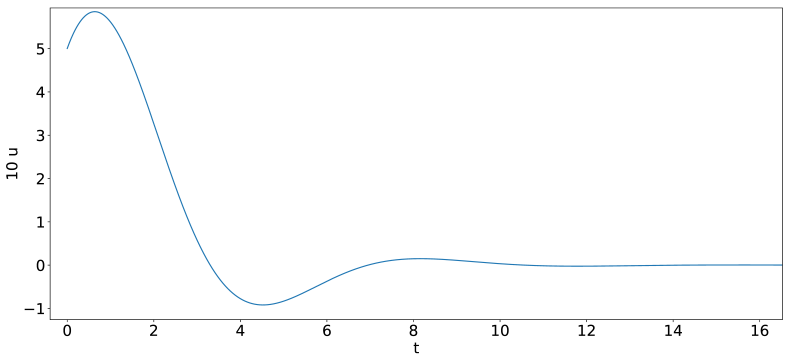

Figure 10. Input for random initial conditions.

7. Conclusions and Outlook on Future Work

In this paper, a method to compute the invariant set of a nonlinear system controlled by sliding mode approaches, which were designed in such a way that structural singularities were guaranteed to be prevented, was suggested. Numerically, the invariant set could be obtained using interval computation, if the controller was designed according to the equation:

$$\dot{s}(\boldsymbol{x}) = -s(\boldsymbol{x}) \cdot p(\boldsymbol{x}) \,, \;\; p(\boldsymbol{x}) > 0. \tag{65}$$

Avoiding singularities in u was ensured using a suitable Gröbner basis representation. The various investigated examples showed a surface represented by a closed curve, a case where Bézout's identity had no solution and thus no controller can be found, and a surface that was a line. Additionally, a three-dimensional example was given and bounds, on the control input were considered.

The presented interval technique for the identification of invariant sets can easily be generalized towards the computation of the region of attraction of asymptotically stable equilibria of nonlinear feedback control systems by verifying those domains in the state space in which the time derivative $\dot{V}(\boldsymbol{x})$ of a Lyapunov function candidate was strictly non-positive within a level set characterized by $V(\boldsymbol{x}) = c > 0$, where this curve needed to be proven by the suggested approach not to coincide with an invariant set. Straightforwardly, also generalizations to the identification of domains in the state space, for which the stability of stochastic differential equations cannot be verified, became possible. Such domains were derived in [48].

In future work, the suggested procedure for the computation of invariant sets can be extended by the consideration of set valued uncertainty in the state variables included in the control signal $u(\boldsymbol{x})$, which result either from an imperfect state measurement or from uncertainty if the state information included in the feedback control law is reconstructed by means of an interval observer [49]. Moreover, procedures for enhanced control parameterizations can be developed on the basis of the presented identification of invariant sets so that the volume of the intervals containing these sets is minimized. Hence, the aim will be to make the guaranteed provable stability domains as large as possible, as well as to maximize the regions of attraction of asymptotically stable equilibria by control re-parameterizations, to minimize the non-provable stability domains in the aforementioned stochastic settings, and to tune Lyapunov functions so that the provable stability domains show a minimum amount of pessimism.

Author Contributions: The implementation and algorithm design were performed by S.R. under the supervision of L.J. The paper was jointly written by S.R., L.J., and A.R.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Moore, R.E. *Interval Arithmetic*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1966.
2. Jaulin, L.; Kieffer, M.; Didrit, O.; Walter, E. *Applied Interval Analysis*; Springer: London, UK, 2001.
3. Moore, R.E.; Kearfott, R.B.; Cloud, M.J. *Introduction to Interval Analysis*; SIAM: Philadelphia, PA, USA, 2009.
4. Merlet, J.P. Interval Analysis for Certified Numerical Solution of Problems in Robotics. *Appl. Med. Eng.* **2009**, *19*, 399–412. [CrossRef]
5. Freihold, M.; Hofer, E.P. Derivation of Physically Motivated Constraints For Efficient Interval Simulations Applied to the Analysis of Uncertain Dynamical Systems. *Appl. Med. Eng.* **2009**, *19*, 485–499. [CrossRef]
6. Pepy, R.; Kieffer, M.; Walter, E. Reliable Robust Path Planning with Application to Mobile Robots. *Appl. Med. Eng.* **2009**, *19*, 413–424. [CrossRef]
7. Rauh, A.; Kersten, J.; Auer, E.; Aschemann, H. Sensitivity-Based Feedforward and Feedback Control for Uncertain Systems. *Computing* **2012**, *94*, 357–367. [CrossRef]
8. Nedialkov, N.S. Interval Tools for ODEs and DAEs. In Proceedings of the 12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2006, Duisburg, Germany, 26–29 September 2006; IEEE Computer Society: Duisburg, Germany, 2007.
9. Nedialkov, N.S. Computing Rigorous Bounds on the Solution of an Initial Value Problem for an Ordinary Differential Equation. Ph.D. Thesis, Graduate Department of Computer Science, University of Toronto, Toronto, ON, Canada, 1999.
10. Lin, Y.; Stadtherr, M.A. Validated Solution of Initial Value Problems for ODEs with Interval Parameters. *Appl. Numer. Math.* **2007**, *57*, 1145–1162. [CrossRef]
11. IEEE Computer Society. IEEE 1788–2015—IEEE Standard for Interval Arithmetic. Technical Report IEEE Std. 1788–2015, American National Standards Institute. 2015. Available online: http://standards.ieee.org (accessed on 30 October 2019).
12. Neumaier, A. *Interval Methods for Systems of Equations*; Cambridge University Press, Encyclopedia of Mathematics: Cambridge, UK, 1990.
13. Krawczyk, R. Newton-Algorithmen zur Bestimmung von Nullstellen mit Fehlerschranken. *Computing* **1969**, *4*, 189–201. (In German) [CrossRef]
14. John, K.; Rauh, A.; Bruschewski, M.; Grundmann, S. Towards Analyzing the Influence of Measurement Errors in Magnetic Resonance Imaging of Fluid Flows. *Acta Cybern.* **2020**, under review.
15. Kersten, J.; Rauh, A.; Aschemann, H. Interval Methods for Robust Gain Scheduling Controllers. *Granul. Comput.* **2018**, doi:10.1007/s41066-018-00147-1. [CrossRef]
16. Csendes, T.; Ratz, D. Subdivision Direction Selection in Interval Methods for Global Optimization. *SIAM J. Numer. Anal.* **1997**, *34*, 922–938. [CrossRef]
17. Hansen, E.R. *Global Optimization Using Interval Analysis*; Marcel Dekker: New York, NY, USA, 1992.
18. Kearfott, R.B. Globsol: History, Composition, and Advice on Use. In *Global Optimization and Constraint Satisfaction, Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 17–31.
19. Rauh, A.; Kersten, J.; Aschemann, H. Interval Methods and Contractor-Based Branch-and-Bound Procedures for Verified Parameter Identification of Quasi-Linear Cooperative System Models. *J. Comput. Appl. Math.* **2020**, *367*, 112484. [CrossRef]
20. Rauh, A.; Senkel, L.; Aschemann, H. Interval-Based Sliding Mode Control Design for Solid Oxide Fuel Cells with State and Actuator Constraints. *IEEE Trans. Ind. Electron.* **2015**, *62*, 5208–5217. [CrossRef]
21. Rauh, A.; Senkel, L.; Auer, E.; Aschemann, H. Interval Methods for the Implementation of Real-Time Capable Robust Controllers for Solid Oxide Fuel Cell Systems. *Math. Comput. Sci.* **2014**, *8*, 525–542. [CrossRef]
22. Senkel, L. *Sliding Mode Techniques for Robust Control, State Estimation and Parameter Identification of Uncertain Dynamic Systems*; Shaker Verlag: Aachen, Germany, 2018.
23. Rauh, A.; Senkel, L.; Kersten, J.; Aschemann, H. Reliable Control of High-Temperature Fuel Cell Systems using Interval-Based Sliding Mode Techniques. *IMA J. Math. Control Inf.* **2016**, *33*, 457–484. [CrossRef]

24. Rauh, A.; Senkel, L.; Aschemann, H. Reliable Sliding Mode Approaches for the Temperature Control of Solid Oxide Fuel Cells with Input and Input Rate Constraints. In Proceedings of the 1st IFAC Conference on Modelling, Identification and Control of Nonlinear Systems, MICNON 2015, Saint Petersburg, Russia, 24–26 June 2015.
25. Rauh, A.; Senkel, L.; Aschemann, H. Interval Methods for Variable-Structure Control of Dynamic Systems with State Constraints. In Proceedings of the 3rd International Conference on Control and Fault-Tolerant Systems, SysTol'16, Barcelona, Spain, 7–9 September 2016.
26. Zhao, X.; Yang, H.; Xia, W.; Wang, X. Adaptive Fuzzy Hierarchical Sliding-Mode Control for a Class of MIMO Nonlinear Time-Delay Systems with Input Saturation. *IEEE Trans. Fuzzy Syst.* **2017**, *25*, 1062–1077. [CrossRef]
27. Raïssi, T.; Efimov, D.; Zolghadri, A. Interval State Estimation for a Class of Nonlinear Systems. *IEEE Trans. Automat. Control* **2012**, *57*, 260–265. [CrossRef]
28. Efimov, D.; Raïssi, T.; Chebotarev, S.; Zolghadri, A. Interval State Observer for Nonlinear Time Varying Systems. *Automatica* **2013**, *49*, 200–205. [CrossRef]
29. Mazenc, F.; Bernard, O. Asymptotically Stable Interval Observers for Planar Systems With Complex Poles. *IEEE Trans. Autom. Control* **2010**, *55*, 523–527. [CrossRef]
30. Mazenc, F.; Andrieu, V.; Malisoff, M. Design of Continuous-Discrete Observers for Time-Varying Nonlinear Systems. *Automatica* **2015**, *57*, 135–144. [CrossRef]
31. Rauh, A.; Kersten, J.; Aschemann, H. Linear Matrix Inequality Techniques for the Optimization of Interval Observers for Spatially Distributed Heating Systems. In Proceedings of the 23rd IEEE International Conference on Methods and Models in Automation and Robotics MMAR 2018, Międzyzdroje, Poland, 27–30 August 2018.
32. Kharkovskaya, T.; Efimov, D.; Polyakov, A.; Richard, J.P. Design of Interval Observers and Controls for PDEs Using Finite-Eelement Approximations. *Automatica* **2018**, *93*, 302–310. [CrossRef]
33. Blanchini, F. Set Invariance in Control. *Automatica* **1999**, *35*, 1747–1767. [CrossRef]
34. Eker, İ. Second-Order Sliding Mode Control with Experimental Application. *ISA Trans.* **2010**, *49*, 394–405. [CrossRef]
35. Fridman, L.; Levant, A. Higher Order Sliding Modes. In *Sliding Mode Control in Engineering*; Barbot, J.P., Perruguetti, W., Eds.; Marcel Dekker: New York, NY, USA, 2002; pp. 53–101.
36. Bartolini, G.; Pisano, A.; Punta, E.; Usai, E. A Survey of Applications of Second-Order Sliding Mode Control to Mechanical Systems. *Int. J. Control* **2003**, *76*, 875–892. [CrossRef]
37. Shtessel, Y.; Edwards, C.; Fridman, L.; Levant, A. *Sliding Mode Control and Observation*; Springer: New York, NY, USA, 2014.
38. Utkin, V.I. *Sliding Modes in Control and Optimization*; Springer: Berlin/Heidelberg, Germany, 1992.
39. Utkin, V.I. Sliding Mode Control Design Principles and Applications to Electric Drives. *IEEE Trans. Ind. Electron.* **1993**, *40*, 23–36. [CrossRef]
40. Soltine, J.E.; Li, W. *Applied Nonlinear Control*; Prentice Hall Inc.: Englewood Cliffs, NJ, USA, 1991.
41. Hebisch, H. Grundlagen der Sliding Mode Regelung. 1995. Available online: www.uni-due.de/imperia/md/content/srs/forschung/msrt_paper/1995/fb15_95.pdf (accessed on 30 October 2019). (In German)
42. Bartoszewicz, A.; Nowacka-Leverton, A. *Time-Varying Sliding Modes for Second and Third Order Systems*; Springer: Berlin/Heidelberg, Germany, 2009.
43. Bartoszewicz, A.; Latosiński, P. Discrete Time Sliding Mode Control with Reduced Switching—A New Reaching Law Approach. *Int. J. Robust Nonlinear Control* **2016**, *26*, 47–68. [CrossRef]
44. Adamy, J. *Nichtlineare Regelungen*; Springer: Berlin/Heidelberg, Germany, 2009. (In German)
45. Le Gallic, M.; Tillet, J.; Jaulin, L.; Le Bars, F. Tight Slalom Control for Sailboat Robots. In Proceedings of the International Robotic Sailing Conference IRSC 2018, Southampton, U, 31 August–1 September 2018.
46. Cox, D.; Little, J.; O'Shea, D. *Ideals, Varieties and Algorithms*; Springer: New York, NY, USA, 2007.
47. Tuncer, Y.B. Solving Multivariate Diophantine Equations and Their Role in Multivariate Polynomial Factorization. 2017. Available online: www.cecm.sfu.ca/CAG/theses/baris.pdf (accessed on 30 October 2019).

48. Rauh, A.; Romig, S.; Aschemann, H. When is Naive Low-Pass Filtering of Noisy Measurements Counter-Productive for the Dynamics of Controlled Systems? In Proceedings of the 23rd IEEE International Conference on Methods and Models in Automation and Robotics MMAR 2018, Międzyzdroje, Poland, 27–30 August 2018.
49. Rauh, A.; Kersten, J.; Aschemann, H. An Interval Observer Approach for the Online Temperature Estimation in Solid Oxide Fuel Cell Stacks. In Proceedings of the European Control Conference ECC, Limassol, Cyprus, 12–15 June 2018.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).