

Algorithms for the estimation of the region of attraction with positively invariant sets

Andrea Iannelli, Andrés Marcos and Mark Lowenberg¹

Abstract—This article focuses on the numerical estimation of the Region of Attraction of systems with polynomial vector field. The presented approach, based on a recent theoretical work on positively invariant sets, computes the inner Estimates of the Region of Attraction by means of Sum of Squares techniques. This allows the set containment conditions defining the region to be enforced at the expense of requiring iterative schemes since the ensuing optimization features bilinearities in the decision variables. The main contribution consists of two novel algorithms aimed at addressing some of the shortcomings typically associated with the adoption of iterative schemes. The results confirm the advantages of the proposed approaches, particularly as the size of the system increases.

I. INTRODUCTION

The Region of Attraction (ROA) associated with an equilibrium point x^* of a nonlinear system is the set of all the initial conditions from which the trajectories of the system converge to x^* as time goes to infinity [1]. Finding the exact region of attraction analytically might be difficult and several algorithms have been proposed to numerically calculate inner Estimates of the Region of Attraction (ERA), which can be broadly classified into two categories: Lyapunov and non-Lyapunov methods.

The former build on the invariance and contractiveness properties held by Lyapunov functions (LF) sublevel sets. When the LF space is restricted to quadratic functions, an ERA for polynomial (or even rational) systems is the largest ellipsoid obtained by computing the Lyapunov matrix [2], whose calculation can be posed as a line search involving the solution of a series of Linear Matrix Inequalities. In the general case of higher order LF, Sum of Squares (SOS) techniques can be used to recast the problem as a set of SemiDefinite Programs (SDPs) [3].

Non-Lyapunov methods have also been studied to reduce the conservatism associated with the aforementioned approaches. In [4] it is shown that the problem can be formulated as a convex infinite-dimensional linear program, which is solved (with a hierarchy of convex finite-dimensional Linear Matrix Inequalities with asymptotically vanishing conservatism) by making use of the concept of occupation measures. In recently published works [5], [6], the recipes for calculating ERA are expressed in terms of positively invariant sets. These approaches, prompted by an old result

known as LaSalle’s theorem [1], still use Lyapunov stability concepts but prescribe weaker conditions for the function used to define the ERA.

The work here considers the theoretical result from [5] as starting point, and employs SOS relaxations to enforce the set containment conditions defining the ERA. The usage of these techniques often leads to non-convex Bilinear Matrix Inequalities [7], and generally iteration schemes are adopted. Their initializations and the definition of the iteration’s steps are critical aspects for the achievement of accurate ERA in an efficient manner. In view of this, the article proposes two novel algorithms having the goal of improving run time and ERA accuracy. Based on two case studies from the literature, and featuring increasing complexity, the prowess of the proposed approaches in addressing this task is quantified via a comparative study. Distinct features are commented and the role of different initializations is studied. One of the algorithms developed in this work served as basis in [8] for the formulation of the robust ROA via invariant sets for the case of systems with uncertain parameters in the vector field.

The layout of the article is as follows. Section II provides a cursory introduction to the basics of the work. Section III presents the problem of determining estimates of the ROA as positively invariant sets, and describes the numerical algorithms. These are subsequently applied in Section IV, where results are also discussed. Section V presents the Conclusions.

II. BACKGROUND

The set of functions $g(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ which are m -times continuously differentiable is denoted by \mathcal{C}^m . For $x \in \mathbb{R}^n$, the set of all polynomials in n variables is denoted by $\mathbb{R}[x]$. For $g \in \mathbb{R}[x]$, $\partial(g)$ denotes the degree of g . Given a scalar $c > 0$ and a function g , the level set $\varepsilon(g, c) := \{x \in \mathbb{R}^n : g(x) \leq c\}$ and its boundary $\partial\varepsilon(g, c) := \{x \in \mathbb{R}^n : g(x) = c\}$ can be defined.

A polynomial $g(x)$ is said to be a Sum of Squares if there exists a finite set of polynomials $g_1(x), \dots, g_k(x)$ such that $g(x) = \sum_{i=1}^k g_i^2(x)$. The set of SOS polynomials in x is denoted by $\Sigma[x_1, \dots, x_n]$, abbreviated here with $\Sigma[x]$. The importance of SOS polynomials is due to their connection with convex optimization [9]. Namely, $g \in \Sigma[x]$ if and only if $g = z^T Q z$ and $Q = Q^T \succeq 0$ (i.e. Q is positive semidefinite), where z is a vector gathering the monomials of g . This problem can be recast as a semidefinite program (SDP) and there are freely available software toolboxes to solve this in an efficient manner [10].

*This work has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 636307, project FLEXOP.

¹ Andrea Iannelli, Andrés Marcos and Mark Lowenberg are with the Department of Aerospace Engineering, University of Bristol, BS8 1TR, United Kingdom andrea.iannelli@bristol.ac.uk/m.lowenberg@bristol.ac.uk

The algorithms employed in this work involve finding functions that satisfy set containment conditions. Known results from real algebraic geometry can be employed to tackle this problem when dealing with polynomial functions. In particular, an application of the Positivstellensatz (P-satz) Theorem allows the following property to be stated.

Lemma 1: [9] Given $h, f_0, \dots, f_r \in \mathbb{R}[x]$, the following set containment holds

$$\{x : h(x) = 0, f_1(x) \geq 0, \dots, f_r(x) \geq 0\} \subseteq \{x : f_0(x) \geq 0\} \quad (1)$$

if there exist *multipliers* $p \in \mathbb{R}[x], s_1, \dots, s_r \in \Sigma[x]$ such that

$$p(x)h(x) - \sum_{i=1}^r s_i(x)f_i(x) + f_0(x) \in \Sigma[x] \quad (2)$$

This result will be used to express set containments (1) as SOS constraints (2).

Consider an autonomous nonlinear system of the form

$$\dot{x} = f(x), \quad x(0) = x_0 \quad (3)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is the vector field. The vector $x^* \in \mathbb{R}^n$ is called a *fixed* or *equilibrium* point of (3) if $f(x^*) = 0$. Let $\phi(t, x_0)$ denote the solution of (3) at time t with initial condition x_0 . The ROA associated with x^* is defined as

$$\mathcal{R} := \{x_0 \in \mathbb{R}^n : \lim_{t \rightarrow \infty} \phi(t, x_0) = x^*\} \quad (4)$$

Thus \mathcal{R} is the set of all initial states that eventually converge to x^* . While for linear systems convergence to the equilibrium, if it holds, is a global property, for nonlinear ones it might hold only locally (i.e. $\mathcal{R} \subseteq \mathbb{R}^n$).

The origin will be assumed as fixed point ($x^* = 0$) henceforward without loss of generality.

III. ALGORITHMS FOR THE COMPUTATION OF ERA WITH INVARIANT SETS

This section presents the main theoretical result from [5] and discusses numerical algorithms to find the ERA of (3) with the invariant sets approach. A novel algorithm is proposed (Subsection III-B) that ameliorates the computational effort and improves the level sets estimates. This is also used to devise a hybrid algorithm (Subsection III-C) aimed to tackle some of the pitfalls associated with SOS optimization.

A. Positively invariant sets-based estimation of ROA

A standard approach to calculate inner Estimates of ROA consists of applying Lyapunov's direct method.

Lemma 2: [1] Let $\mathcal{D} \subset \mathbb{R}^n$ and let 0 be contained in \mathcal{D} . If there exists $V : \mathbb{R}^n \rightarrow \mathbb{R}$, with $V \in \mathcal{C}^1$ such that:

$$\begin{aligned} V(0) = 0 \quad \text{and} \quad V(x) > 0 \quad \forall x \in \mathcal{D} \setminus 0 \\ \dot{V}(x) = \nabla V(x)f(x) < 0 \quad \forall x \in \mathcal{D} \setminus 0 \\ \varepsilon(V, c) \quad \text{is bounded and} \\ \varepsilon(V, c) \subseteq \mathcal{D} \end{aligned} \quad (5)$$

then $\varepsilon(V, c)$ is an invariant set of \mathcal{R} .

As pointed out by LeSalle's theorem [1], this characterization is usually conservative due to the fact that contractiveness of the level set defining the ERA is unnecessary. In

fact, it suffices to consider compact positively invariant sets of \mathcal{D} , that is, a compact set $\Omega \subseteq \mathcal{D}$ such that every trajectory starting in Ω stays in Ω for all future time.

Prompted by these observations, the following result has been proposed in the literature:

Theorem 1: ([5], Th. 1) If there exist $R, V_N : \mathbb{R}^n \rightarrow \mathbb{R}$, with $R, V_N \in \mathcal{C}^1$, and a positive scalar γ satisfying:

$$\nabla R(x)f(x) < 0 \quad \forall x \in \partial\varepsilon(R, \gamma) \quad (6a)$$

$$V_N(0) = 0 \quad \text{and} \quad V_N(x) > 0 \quad \forall x \in \varepsilon(R, \gamma) \setminus 0 \quad (6b)$$

$$\nabla V_N(x)f(x) < 0 \quad \forall x \in \varepsilon(R, \gamma) \setminus 0 \quad (6c)$$

$$\varepsilon(R, \gamma) \quad \text{is compact and} \quad 0 \in \varepsilon(R, \gamma) \quad (6d)$$

then $x_0 \in \varepsilon(R, \gamma)$ implies $x_0 \in \mathcal{R}$.

The proof of this result can be found in the reference. The fundamental idea is that $\varepsilon(R, \gamma)$ is a positively invariant set, due to (6a)-(6d), and that all trajectories initiated from it converge to a level set of some LF- which is contractive and invariant because of (6b)-(6c), therefore guaranteeing such set to be an ERA. Note that the function R defining the level set only requires negativity of its gradient on the set boundary.

Theorem (1) involves finding functions that satisfy set containment conditions. In order to make the problem computationally tractable, interest is restricted to polynomial vector fields f , and thus, applying Lemma 1, the problem of finding an ERA can be recast as an SOS optimization.

Program 1:

$$\begin{aligned} \max_{s_1, s_2 \in \Sigma[x]; s_0, V_N, R \in \mathbb{R}[x]} \gamma \\ - \nabla R f - s_0(\gamma - R) \in \Sigma[x] \end{aligned} \quad (7a)$$

$$V_N - s_1(\gamma - R) \in \Sigma[x] \quad (7b)$$

$$- \nabla V_N f - s_2(\gamma - R) \in \Sigma[x] \quad (7c)$$

Note that V_N enters affinely in (7), whereas there are bilinear terms featuring the multipliers s_i, γ and R . When the objective function is one of the two terms in the bilinearity (e.g. $s_0\gamma$), it was demonstrated [11] that the problem is quasiconvex and thus the global optimum can be computed via cost bisection. However, the terms in s_i and R (e.g. s_0R) make the above program non-convex. This can be handled with local Bilinear Matrix Inequalities solvers [12] or by means of iterative schemes. In [5] this is handled with the latter approach and the following algorithm is proposed:

Algorithm 1:

Output: the level set $\varepsilon(R, \gamma)$.

Input: a polynomial R^0 satisfying condition (7a).

Step 1 : solve for $s_0, s_1, s_2, V_N, \gamma$

$$\begin{aligned} \gamma_1 = \max_{s_1, s_2 \in \Sigma[x]; s_0, V_N \in \mathbb{R}[x]} \gamma \\ - \nabla R^0 f - s_0(\gamma - R^0) \in \Sigma[x] \end{aligned}$$

$$V_N - s_1(\gamma - R^0) \in \Sigma[x]$$

$$- \nabla V_N f - s_2(\gamma - R^0) \in \Sigma[x]$$

Step 2 : solve for s_3, R, γ

$$\begin{aligned} & \max_{s_3 \in \Sigma[x]; R \in \mathbb{R}[x]} \gamma \\ & -\nabla R f - \bar{s}_0(\gamma - R) \in \Sigma[x] \\ & \bar{V}_N - \bar{s}_1(\gamma - R) \in \Sigma[x] \\ & -\nabla \bar{V}_N f - \bar{s}_2(\gamma - R) \in \Sigma[x] \\ & (\gamma - R) - s_3(\gamma - R^0) \in \Sigma[x] \\ & \gamma - \gamma_1 \geq 0 \end{aligned}$$

The superscript 0 indicates that the functions hold the value calculated at the end of the previous iteration (or their initializations, if at the first iteration), whereas the symbol *bar* is used for quantities optimised within the same iteration (at previous steps).

This iterative scheme consists of two steps. In Step 1 the multipliers s_i ($i = 0, 1, 2$) and V_N are optimised, whereas Step 2 computes the level set function R , which is updated at the beginning of the new iteration ($R^0 \leftarrow R$). Note that a candidate R is required to initialise the algorithm. A possible choice is any quadratic LF proving asymptotic stability of the linearised system, named here V_{lin} . Another important aspect is that the last two SOS constraints in Step 2 ensure that $\varepsilon(R^0, \gamma) \subseteq \varepsilon(R, \gamma)$, i.e. the solution is a set that strictly contains the previous one.

The iterations terminate when one of the steps fails, i.e. the associated optimization is found unfeasible, and the last optimised values for R and γ are taken to provide the output $\varepsilon(R, \gamma)$. Alternatively, a stopping criterion could be employed to prevent slow progress in the simulations. This is not done here in order to present an objective comparison among the algorithms, free from arbitrariness as for example the choice for the tolerance on the progress.

B. A 3 step iteration scheme

An alternative algorithm to solve Program 1 is here proposed. The main goal is to distribute the computational effort in order to ease the SDP calculations underpinning each step and to improve in this way the estimation of the level set $\varepsilon(R, \gamma)$.

First, a modification of Algorithm 1 is discussed. The property $\varepsilon(R^0, \gamma) \subseteq \varepsilon(R, \gamma)$ discussed before implies that Step 2, in addition to Step 1, is also quasi-convex and thus it entails a bisection search on γ . To overcome this, the last two SOS constraints in Step 2 are replaced with:

$$(\gamma - R) - s_3(\gamma_1 - R^0) \in \Sigma[x] \quad (8)$$

This constraint keeps enforcing the level set increase at each iteration, but does not involve any bilinearity. In the tested cases this modification led to a reduction in simulation time while achieving the same or better accuracy in the results. Therefore, this is implemented in all the analyses performed in this article when Algorithm 1 is used.

In addition, a new iteration strategy is devised which consists of three steps:

Algorithm 2:

Output: the level set $\varepsilon(R, \gamma)$.

Input: polynomials R^0, V_N^0 satisfying (7).

Step 1 : solve for s_0, s_1, s_2, γ

$$\begin{aligned} & \max_{s_1, s_2 \in \Sigma[x]; s_0 \in \mathbb{R}[x]} \gamma \\ & -\nabla R^0 f - s_0(\gamma - R^0) \in \Sigma[x] \\ & V_N^0 - s_1(\gamma - R^0) \in \Sigma[x] \\ & -\nabla V_N^0 f - s_2(\gamma - R^0) \in \Sigma[x] \end{aligned}$$

Step 2 : solve for V_N, γ

$$\begin{aligned} & \gamma_2 = \max_{V_N \in \mathbb{R}[x]} \gamma \\ & -\nabla R^0 f - \bar{s}_0(\gamma - R^0) \in \Sigma[x] \\ & V_N - \bar{s}_1(\gamma - R^0) \in \Sigma[x] \\ & -\nabla V_N f - \bar{s}_2(\gamma - R^0) \in \Sigma[x] \end{aligned}$$

Step 3 : solve for s_3, R, γ

$$\begin{aligned} & \max_{s_3 \in \Sigma[x]; R \in \mathbb{R}[x]} \gamma \\ & -\nabla R f - \bar{s}_0(\gamma - R) \in \Sigma[x] \\ & \bar{V}_N - \bar{s}_1(\gamma - R) \in \Sigma[x] \\ & -\nabla \bar{V}_N f - \bar{s}_2(\gamma - R) \in \Sigma[x] \\ & (\gamma - R) - s_3(\gamma_2 - R^0) \in \Sigma[x] \end{aligned}$$

The scheme consists of one quasi-convex step (Step 1) and two convex steps (Steps 2-3). Each step has a specific task: Step 1 provides the multipliers for the next two steps; Step 2 calculates the function V_N ; and Step 3 evaluates the sought level set $\varepsilon(R, \gamma)$ based on the iterates from the previous steps. The size of the ERA γ is maximised throughout the iteration, although Steps 2-3 can also be solved as simple feasibility problems. In this regard, note that the optimality of the solution is already prevented by the non-convexity of (7), and that the algorithm ensures in any case that the ERA is non-decreasing. Therefore, resorting to just feasibility when maximization fails is a viable solution.

Algorithm 2 requires initializations for R and V_N . A first option is to choose for both V_{lin} , which automatically satisfies (7). Alternatively, the calculation can start with Algorithm 1 which in turn can provide the initializations R^0 and V_N^0 to Algorithm 2. Despite the impossibility to give a conclusive statement in terms of which one is a better option, this paper will investigate the effect of different initializations. This is deemed an important aspect of the search for ERAs since the obtained local optimum is very sensitive to the initial guess. When not specified, the algorithms are initialised with V_{lin} .

C. Hybrid scheme

The issues with the algorithms illustrated in the previous sections are twofold: on the one hand, the non-convexity due to the bilinear terms forces coordinate-wise search algorithms to be adopted, leading to local optima; on the other, the

SDP associated with each iteration can be computationally challenging. While the algorithm proposed in Sec. III-B aims at tackling the aforementioned aspects by providing an alternative less computationally intensive strategy to solve the optimization problem, it is nonetheless affected by the same local optima pitfall.

Note that the issue of dealing with non-convex searches is well-known in the optimization field and one of the proposed solutions is represented by so-called hybrid strategies [13]. The essence of this approach is to cross global optimizers with problem-specific local search algorithms. In the currently investigated ERA programs, the non-convexity is inherent to the adoption of SOS relaxations for the enforcement of set containments. Thus, the adoption of a hybrid scheme meant in the conventional sense does not look viable. However, in this work the availability of two distinct schemes, namely Algorithms 1 and 2, is exploited to implement a unified one which makes the search of ERA more robust to numerical issues.

Taking the cue from this discussion, the following *hybrid* algorithm is proposed.

Algorithm 3:

Output: the level set $\varepsilon(R, \gamma)$.

Input: a switching criterion sw_{cr} ; polynomials R^0, V_N^0 satisfying (7).

Stage 1 :

Execute Algorithm 1

if Stage 1 converged **then** set $R^0 \leftarrow R$ and $V_N^0 \leftarrow V_N$

if sw_{cr} is true **then** go to Stage 2

else restart Stage 1

else set $R^0 \leftarrow R^0$ and $V_N^0 \leftarrow V_N^0$, and go to Stage 2

Stage 2 :

Execute Algorithm 2

if Stage 2 converged **then** set $R^0 \leftarrow R$

if sw_{cr} is true **then** go to Stage 1

else restart Stage 2

else set $R^0 \leftarrow R^0$ and go to Stage 1

The iterative scheme builds on the advantageous capability of switching from one algorithm to the other in case of failed solution or slow progress. The latter can be employed to devise the selection of the switching criterion sw_{cr} , and an example is commented next. γ is by definition the size of the level set and a possible metric to quantify the expansion rate of the ERA for a certain algorithm is obtained comparing values of γ referred to the *same* shape function R . This is the case, for example, of Step 1 (of both the algorithms) which is performed with the R obtained at the previous iteration (with an associated size γ_0). Thus, sw_{cr} can be defined as a tolerance on the ratio $\frac{\gamma_1}{\gamma_0}$, where γ_1 is the size of the ERA after Step 1. Note that when Algorithm 2 is employed, $\frac{\gamma_2}{\gamma_1}$ (with γ_2 the size of the ERA after Step 2) can also be considered, because the shape R is held fixed over this step, too. The idea behind the choice of these ratios is that they

provide a measure of the increase in the size of the estimated ROA achieved with a particular algorithm. Other metrics able to quantify the progress in enlarging the provable ERA could be also adopted as switching criteria. Since this is likely to lead to different results, testing different sw_{cr} represents a further strategy to improve the estimation of the ROA.

In the following analyses, sw_{cr} is defined such that the algorithm crosses the Stages sequentially (i.e. the inner **if** condition is always true). In fact, it should not be underestimated the utility of a scheme whose goal is simply to carry on the optimization in case of failed solution of one algorithm, as it is often the case that unfeasibility of one of the steps is not caused by the fact that the ERA is close to the actual ROA, but by numerical issues of the SDPs (exacerbated when the size of the program increases).

The selection of the switching criterion is deemed a problem-specific feature and thus the study of the benefits in adopting the aforementioned criteria can be undertaken in future works focused on the study of ROA for specific case studies.

IV. RESULTS

The different algorithms discussed in Section III are applied here to two case studies (labelled *VdP* and *SP*). All the analyses are performed on a 3.6 GHz desktop PC with 16 GB RAM, with the library SOSOPT [10] used in conjunction with the SDP solver Sedumi [14]. Before presenting the case studies and detailed results, Tab. I summarises the computational statistics. Recall that each algorithm is iterative and each iteration features two or three steps. Therefore, for each algorithm only the number of decision variables N_{var} and size of the Gram matrix N_{SDP} for the most demanding step are reported. In addition, the averaged processing time per iteration T_{iter} and the overall time T_{tot} required to determine the ERA are given. Even though these latter metrics could vary depending on the adopted SOS-SDP solvers, it is believed that the differences in their values among the algorithms would remain similar. The same reasoning applies to the size of the ERA discussed in the remainder of the section.

TABLE I
COMPUTATIONAL STATISTICS

Case study	Algorithm	N_{var}	N_{SDP}	T_{iter} [s]	T_{tot} [s]
VdP	Alg. 1	27	362	4.8	88
VdP	Alg. 2	15	362	4.4	114
VdP	Alg. 3	15	362	4.6	123
SP	Alg. 1	246	12322	78	1872
SP	Alg. 2	126	12322	70	1260
SP	Alg. 3	246	12322	72	2742

A. Van der Pol oscillator

The Van der Pol (*VdP*) oscillator [15] is a nonlinear system with 2 states given by:

$$\begin{aligned} \dot{x}_1 &= -x_2 \\ \dot{x}_2 &= x_1 + (x_1^2 - 1)x_2 \end{aligned} \quad (9)$$

The *VdP* steady-state solutions are characterized by an unstable limit cycle and a stable equilibrium at the origin. The

latter is only locally stable, its ROA being the region enclosed by the limit cycle, which thus can be easily obtained from time-marching simulations. Fig. 1 shows different estimates of the ROA for this system (the actual ROA is also reported for reference): *Alg. 1* (Algorithm 1); *Alg. 2* (Algorithm 2); and *Alg. 3* (Algorithm 3). The degree of the optimized polynomials V_N and R is 4.

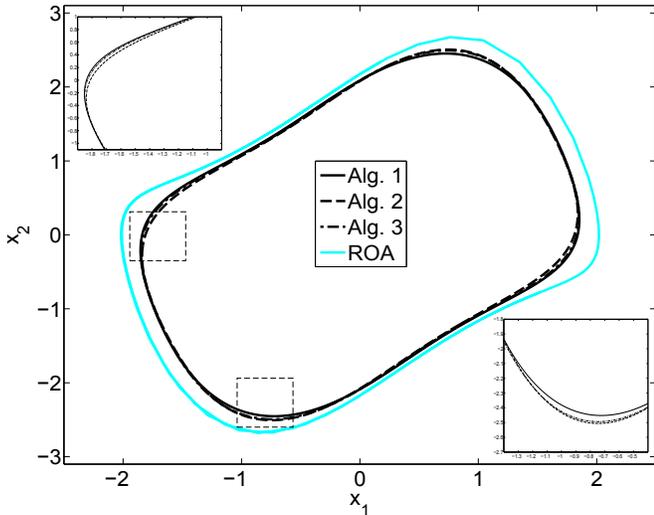


Fig. 1. ERA of VdP ($\partial(V_N, R) = 4$).

As a preamble, it should be noted that all the algorithms provide a good approximation of the ROA of the system. In fact, VdP is often used as a benchmark study for newly developed algorithms and in the same spirit was also adopted in this work. The small differences are better seen in the insets of Fig. 1, which help to highlight the effect of adopting a hybrid strategy. The curve *Alg. 3* is shown to envelope the curves obtained with the two iteration schemes, providing in this way the largest estimation of ROA among the considered approaches.

As for the computational statistics reported in Tab. I, Algorithm 2 features a smaller T_{iter} than Algorithm 1 (bearing in mind the similarity of the results commented before). Even though the former features 3 steps (as opposed to Algorithm 1 which has only 2), this performance is a result of the redistribution of the computational effort which is the main idea behind the proposal of Algorithm 2 (which also benefits Algorithm 3). However, when looking at the overall time T_{tot} the trend is opposite, despite the fact that the two achieved estimations of the ROA are very close. This is due to the fact that the algorithms are carried out until an optimization step is unfeasible, and there is no condition preventing slow progress. Algorithm 2 performs more iterations before reaching unfeasibility, but with no tangible improvement on the estimation for this case study, and this results in a greater T_{tot} . This could be overcome by using a stopping condition based for example on one of the options discussed in Sec. III-C for the switching criterion sw_{cr} . Tests carried out with this rationale lead to an overall smaller time T_{tot} for Algorithm 2, too.

B. Controlled short-period aircraft dynamics

This case study consists of a closed-loop nonlinear short-period (SP) model of the longitudinal dynamics of an aircraft [16]. It features 3 open-loop states (pitch rate z_1 , angle of attack z_2 , pitch angle z_3) and 2 controller states η_1, η_2 .

$$\begin{aligned} \dot{z} &= \begin{bmatrix} -3 & -1.35 & -0.56 \\ -0.91 & -0.64 & -0.02 \\ 1 & 0 & 0 \end{bmatrix} z + \begin{bmatrix} 1.35 & -0.04z_2 \\ 0.4 \\ 1 \end{bmatrix} u \\ &+ \begin{bmatrix} 0.08z_1z_2 + 0.44z_2^2 + 0.01z_2z_3 + 0.22z_2^3 \\ -0.05z_2^2 + 0.11z_2z_3 - 0.05z_3^2 \\ 0 \end{bmatrix} \\ \dot{\eta} &= \begin{bmatrix} -0.6 & 0.09 \\ 0 & 0 \end{bmatrix} \eta + \begin{bmatrix} -0.06 & -0.02 \\ -0.75 & -0.28 \end{bmatrix} y \\ y &= [z_1 \quad z_3]^T; \quad u = \eta_1 + 2.2\eta_2; \end{aligned} \quad (10)$$

Since the system has more than 2 states, projections of the ERA onto particular planes are employed to visualize the results. In general, the analyst will focus on the states which are supposed to experience larger perturbations during the operation of the system. In this work, the z_1 - z_2 phase-plane (Fig. 2) will be displayed since the studied nonlinearities arise primarily from their dynamics. Further, to provide as much information relative to the analyses as possible, Fig. 3 shows the projections onto the z_1 - η_1 plane. The degree of the optimized polynomials V_N and R is 4.

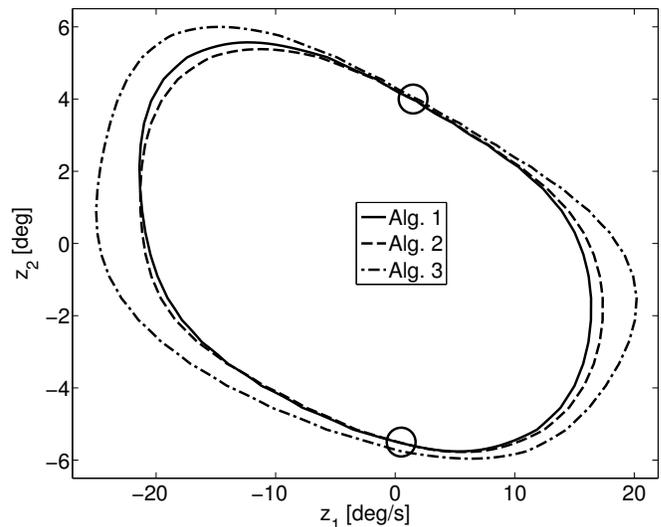


Fig. 2. Projection of the ERA of SP onto z_1 - z_2 plane.

The effect of the adopted iteration scheme is, as expected, more significant here. Note indeed from N_{var} and N_{SDP} in Tab. I that this case study has a greater level of computational complexity when compared to the VdP dynamics. Although no conclusive remarks can be stated based only on these results, it is worth discussing some trends observed also in other analyses which considered different definitions for the multipliers and for the level set functions degrees. The algorithms are sensitive to the initialization, especially Algorithm 2 which requires a guess for both V_N and R . A

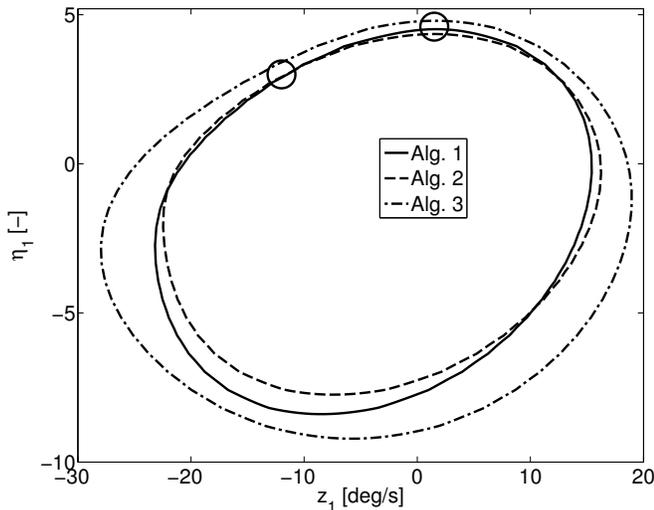


Fig. 3. Projection of the ERA of SP onto z_1 - η_1 plane.

promising strategy in this regard consists of initializing it with the first iterate from Algorithm 1. The analyses also revealed that the estimates computed with Algorithm 3 were typically the largest (this is noticeable for example in Figs. 2-3).

Other valuable information provided by the comparison of different ERA is an enhanced insight into the actual boundaries of the ROA, which is not known for the SP case. For example, from a closer inspection of Figs. 2-3, a dense presence of curves in some regions, marked with circles in the plot, can be identified. It can then be speculated that these correspond to boundaries of the actual region of attraction of the system. These insights can be of great help in the analysis of nonlinear systems since they can inform extensive refined time-marching simulations to check safety regions for the operation of the plant. In addition to this, they can inform the initializations for further analyses which, in view of their importance associated with the local search nature of the algorithms, can drastically improve the estimation.

The conservatism associated with the estimates provided by the algorithms can be investigated by simulating the nonlinear systems using as initial conditions random points on the boundary $\partial\epsilon(R, \gamma)$ and enlarging γ until trajectories steering away from the equilibrium point are detected (providing in this way an upper bound γ_f on the size of the ERA for the shape associated with R). These analyses revealed that the hybrid scheme led to the most accurate estimate, featuring a gap between γ_f and γ within 6%. Moreover, the initial conditions associated with escaping trajectories lie in the regions marked with circles in Figs. 2-3, confirming the inferences discussed before.

Finally, it can be appreciated from Tab. I that the reduction in T_{iter} from Algorithm 1 to Algorithm 2 is even greater for this case study. This was expected since it features a larger size in both number of states n and polynomial degree $\partial(f)$, thus the effect of lowering the SDPs dimension (which drove the design of Algorithm 2) is magnified. Algorithm

2 also features the smallest T_{tot} . Note that the high value observed for Algorithm 3 can be motivated observing that the associated ERA is markedly larger than the others and thus more iterations will be involved in its computation. This is an important aspect to keep in mind when using the metric T_{tot} to compare different algorithms.

V. CONCLUSION

This article considers the problem of estimating the region of attraction of systems described by polynomial vector fields. A recent formulation based on invariant level sets is adopted as the theoretical foundation to propose computationally more efficient algorithms. Well-known numerical issues associated with Sum Of Squares relaxations are also commented on, and possible strategies to ameliorate them are discussed. An iteration scheme to tackle the bilinearities arising in the corresponding programs is proposed, and the advantages of jointly using different iterative algorithms in order to overcome unfeasibility or slow progress is demonstrated via two example systems. These confirm the advantages, both in run time and size of the estimated ROA, in adopting the iterative schemes proposed in this article, especially as the size of the analysed system increases.

REFERENCES

- [1] H. K. Khalil, *Nonlinear systems*. Prentice Hall, 1996.
- [2] G. Valmorbida, S. Tarbouriech, and G. Garcia, "Region of attraction estimates for polynomial systems," in *48th IEEE Conference on Decision and Control (CDC)*, 2009.
- [3] G. Chesi, *Domain of Attraction: Analysis and Control via SOS Programming*. Springer, 2011.
- [4] D. Henrion and M. Korda, "Convex Computation of the Region of Attraction of Polynomial Control Systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 297–312, 2014.
- [5] G. Valmorbida and J. Anderson, "Region of attraction estimation using invariant sets and rational Lyapunov functions," *Automatica*, vol. 75, pp. 37 – 45, 2017.
- [6] D. Han, A. El-Guindy, and M. Althoff, "Estimating the Domain of Attraction Based on the Invariance Principle," in *55th IEEE Conference on Decision and Control (CDC)*, 2016.
- [7] J. Anderson and A. Papachristodoulou, "Advances in computational Lyapunov analysis using sum-of-squares programming," *Discrete and Continuous Dynamical Systems - Series B*, vol. 20, no. 8, pp. 2361–2381, 2015.
- [8] A. Iannelli, A. Marcos, and M. Lowenberg, "Estimating the region of attraction of uncertain systems with invariant sets," 9th IFAC Symposium on Robust Control Design (ROCOND), 2018.
- [9] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, no. 2, pp. 293–320, 2003.
- [10] G. J. Balas, A. K. Packard, P. Seiler, and U. Topcu, "Robustness analysis of nonlinear systems," <http://www.aem.umn.edu/Aerospace-Control/>, 2009.
- [11] P. Seiler and G. Balas, "Quasiconvex Sum-of-Squares Programming," in *49th IEEE Conference on Decision and Control (CDC)*, 2010.
- [12] M. Kocvara and M. Stingl, *PENBMI Users Guide (Version 2.1)*, 2006.
- [13] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley, 1989.
- [14] J. F. Sturm, "Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1-4, pp. 625–653, 1999.
- [15] W. Tan and A. Packard, "Stability Region Analysis Using Polynomial and Composite Polynomial Lyapunov Functions and Sum-of-Squares Programming," *IEEE Transactions on Automatic Control*, vol. 53, no. 2, pp. 565–571, 2008.
- [16] U. Topcu and A. Packard, "Local Stability Analysis for Uncertain Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1042–1047, 2009.