



Guaranteed systematic simulation of discrete-time systems defined by polynomial expressions via convex relaxations

Paolo Massioni, Gérard Scorletti

► To cite this version:

Paolo Massioni, Gérard Scorletti. Guaranteed systematic simulation of discrete-time systems defined by polynomial expressions via convex relaxations. International Journal of Robust and Nonlinear Control, 2018, 28 (3), pp.1062-1073. 10.1002/rnc.3920 . hal-01574320

HAL Id: hal-01574320

<https://hal.science/hal-01574320>

Submitted on 5 Apr 2019

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.

Guaranteed systematic simulation of discrete-time systems defined by polynomial expressions via convex relaxations

Paolo Massioni^{1,†,*}, Gérard Scorletti²

¹Laboratoire Ampère, UMR CNRS 5005, INSA Lyon, Université de Lyon, F-69621 Villeurbanne, France

²Laboratoire Ampère, UMR CNRS 5005, Ecole Centrale Lyon, Université de Lyon, F-69134 Ecully, France

SUMMARY

This paper concerns the simulation of a class of nonlinear discrete-time systems under a set of initial conditions described by a bounding ellipsoid. We derive a procedure allowing the propagation of such ellipsoids through time, which makes it possible to set a guaranteed hard bound on the evolution of the state of the system for all the possible initial conditions. Two versions of this procedure are given, the second of which is slightly less general but less computationally demanding. At the end of the paper, we first show an application of the method in the domain of aerospace engineering; subsequently, three academic examples of applications are presented, two of which come from the theory of fractals.

KEY WORDS: Discrete-time systems, sum of squares, robust simulation, attitude dynamics.

1. INTRODUCTION

When dealing with nonlinear systems, it is often of capital importance to be able to predict the evolution of the state for uncertain initial conditions, especially in the case of safety-critical systems like for example in aerospace applications, which are ultimately the main motivation of this work. A practical approach to this problem is the use of systematic simulation, which consists ideally in checking the behaviour of the system with respect to all the possible initial conditions; but this is strictly impossible if initial conditions are assumed to belong to a dense set, leading to approaches based on random tests or Monte Carlo methods [1]. Other approaches have also been investigated, for example based on random exploration [2] or sensitivity analysis [3]. However, all of these methods suffer of the shortcoming on being a sort of “statistical” validation, in the sense that they do not offer a hard bound on the evolution of the system, if not for a computational complexity going to infinity. Another approach is based on evaluating the effect of the initial condition with respect to an output index (i.e. the time integral of the maximum distance from the nominal trajectory), which gives an idea of such effects but does not establish precise bounds on each state variable [4, 5].

In this article we present a radically different approach to the problem, based on the so called “robust simulation” or simulation of sets [6, 7, 8, 9, 10], which offers instead mathematically guaranteed bounds for the evolution of dynamical systems (also called “reachability analysis” in the context of computer science [11, 12]). The specific method developed in this paper focuses on a class of discrete-time systems, and it is based on the sum of squares (SOS) relaxation of polynomial problems [13], which leads to efficiently solvable convex optimisation problems in the form of linear matrix inequalities (LMIs) [14]. This approach can be considered “safe”, as the evolution of all the possible trajectories of the state are hard bounded, but on the other hand it is

*Correspondence to: Laboratoire Ampère, UMR CNRS 5005, INSA de Lyon, Université de Lyon, F-69621 Villeurbanne, France

†e-mail: paolo.massioni@insa-lyon.fr

conservative, i.e. the bounds are not necessarily tight. One of the main contribution of this work is that the first theorem which we provide can be applied to a very general set of nonlinear equations, namely nonlinear implicit dynamical equations, which can stem from continuous-time dynamical equations discretised through implicit discretisation methods. We also propose a computationally cheaper specialised version which applies to a restricted set of dynamical equations.

We would like to stress that robust simulation is a problem that shares several similarities with many works on computations on invariant sets [15], backward reachability [16] and search for regions of attractions [17]. The goal of this paper is to find the envelope containing the state *at each time instant*, and not to understand whether it is stable or whether it will eventually converge to a set. We have also to mention that the direct sum of square approach that we have chosen is not the only possibility to deal with polynomials, i.e. moment relaxations [18] are also a quite powerful approach.

The paper is organised as follows. Section 2 contains the preliminaries and the problem formulation. The two main theoretical results can then be found in Section 3 and Section 4, in the form of theorems with instructions for practical use. After a short discussion on the computational complexity in Section 5, these results are applied to four examples in Section 6. Finally, conclusions are drawn in Section 7.

2. PRELIMINARIES

2.1. Notation

We denote by \mathbb{N} the set of non-negative integers, by \mathbb{R} the set of real numbers and by $\mathbb{R}^{n \times m}$ the set of real $n \times m$ matrices. Let \mathbb{C} be the set of complex numbers, and j the imaginary unit. $\mathbf{R}_n[x]$ is the set of polynomials of degree n in the entries of x , A^\top indicates the transpose of a matrix A , I_n is the identity matrix of size n , and $0_{n \times m}$ is a matrix of zeros of size $n \times m$. The notation $A \succeq 0$ (resp. $A \preceq 0$) indicates that all the eigenvalues of the symmetric matrix A are positive (resp. negative) or equal to zero, whereas $A \succ 0$ (resp. $A \prec 0$) indicates that all such eigenvalues are strictly positive (resp. negative). The symbol $\binom{n}{k}$ indicates the binomial coefficient, for which we have

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

We also define $\mathcal{E}(P, c)$ as the compact ellipsoid of dimension n with matrix $P \in \mathbb{R}^{n \times n}$, $P = P^\top \succ 0$ and centered in $c \in \mathbb{R}^n$, i.e. $\mathcal{E}(P, c) = \{x \in \mathbb{R}^n \mid (x - c)^\top P^{-1}(x - c) \leq 1\}$. At last, we employ the symbol $*$ to complete symmetric matrix expressions avoiding repetitions.

2.2. Problem formulation

We consider a discrete-time dynamical system of order n whose evolution is described by the equation

$$h_m(x(k), x(k+1)) = 0 \tag{1}$$

where $k \in \mathbb{N} \cup \{0\}$ is the discrete time variable, $x \in \mathbb{R}^n$ is the state vector, $h_m \in \mathbb{R}^n$ is a vector of polynomial functions in $x(k)$ and $x(k+1)$ of degree not greater than $m \in \mathbb{N}$, i.e. $\mathbf{R}_m[[x(k)^\top \ x(k+1)^\top]^\top]$. We suppose that the initial condition $x(0)$ is not exactly known, but it is bound to belong to an ellipsoid $\mathcal{E}(P(0), c(0))$, i.e.

$$(x(0) - c(0))^\top P(0)^{-1}(x(0) - c(0)) \leq 1. \tag{2}$$

Under the assumption that (1) is well-posed, i.e. it is possible to solve for $x(k+1)$ if $x(k)$ is given (at least, for $x(k)$ in a restricted subset of \mathbb{R}^n), the problem on which this article focuses is to find the smallest ellipsoid $\mathcal{E}(P(k), c(k))$ that, for all the valid initial conditions, bounds the state at each

time instant k with $0 < k \leq N$, i.e. such that

$$(x(k) - c(k))^\top P(k)^{-1} (x(k) - c(k)) \leq 1. \quad (3)$$

This problem can be decomposed into the iteration of an elementary problem on a single time step, i.e. finding the smallest $\mathcal{E}(P(k+1), c(k+1))$ for a given $x \in \mathcal{E}(P(k), c(k))$.

We have chosen to focus on ellipsoids instead of semialgebraic sets of higher degree (like in [19]) as we intend to use the method recursively. Using different sets would increase the degree and the computational complexity with each time step. Moreover, ellipsoids are quite useful in the practice, as they can be quickly characterised geometrically in terms of center and semiaxes.

3. MAIN RESULT

As stated, h_m is a vector of polynomials in the state variable at two different time steps, namely $x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^\top$, and $x(k+1) = [x_1(k+1), x_2(k+1), \dots, x_n(k+1)]^\top$. As we are going to deal with polynomials up to degree m , we define also the vector $\zeta \in \mathbb{R}^\mu$ which contains all the possible monomials obtainable from $x(k)$ and $x(k+1)$ from degree 0 up to m (for example, if $n = 2, m = 2$, then $\zeta = [1, x_1(k), x_2(k), x_1(k+1), x_2(k+1), x_1(k)^2, x_1(k)x_2(k), x_1(k)x_1(k+1), x_1(k)x_2(k+1), x_2(k)^2, x_2(k)x_1(k+1), \dots]^\top$). We have that

$$\mu = \binom{2n+m}{2n}. \quad (4)$$

In this way, any polynomial in the variables of $x(k), x(k+1)$ up to degree m can be formulated as a linear function of ζ ; so namely we have

$$h_m(x(k), x(k+1)) = H\zeta = [H_1\zeta \ H_2\zeta \ \dots \ H_n\zeta]^\top \quad (5)$$

with $H \in \mathbb{R}^{\mu \times n}$. Moreover, it is also possible to express polynomials p up to degree $2m$ as a quadratic form with respect to ζ , i.e. $p(x(k), x(k+1)) = \zeta^\top \mathcal{P} \zeta$, with $\mathcal{P} = \mathcal{P}^\top \in \mathbb{R}^{\mu \times \mu}$. As reported in the literature related to sum of squares problems [20, 13], this quadratic expression of a polynomial is not unique, due to the fact that different products of monomials can yield the same result, for example x_1^2 is either x_1^2 times 1 or x_1 times x_1 . This implies that there exist linearly independent slack matrices $W_k = W_k^\top \in \mathbb{R}^{\mu \times \mu}$, with $k = 1, \dots, \gamma$ such that $\zeta^\top W_k \zeta = 0$. The number of such matrices [13] is

$$\gamma = \frac{1}{2} \left(\binom{m+2n}{m}^2 + \binom{m+2n}{m} \right) - \binom{2n+2m}{2m}. \quad (6)$$

A polynomial p of degree $2m$ or less can then be expressed with the so called Gram matrix formulation as

$$p(x(k), x(k+1)) = \zeta^\top \left(\mathcal{P} + \sum_{k=1}^{\gamma} \psi_k W_k \right) \zeta \quad (7)$$

with a valid choice of \mathcal{P} , for any $\psi \in \mathbb{R}^\gamma$, $\psi = [\psi_1, \psi_2, \dots, \psi_\gamma]^\top$. Notice that the Gram matrix \mathcal{P} is not unique.

The main result of this paper is in the following theorem, which basically yields a practical solution for the problem in Section 2.2 in its step-by-step formulation.

Theorem 1

Consider the dynamical system described by (1), together with the variables defined in (4), (5), (6). If $x(k)$ is such that $(x(k) - c(k))^\top P(k)^{-1} (x(k) - c(k)) \leq 1$ is satisfied, then $(x(k+1) - c(k+1))^\top P(k+1)^{-1} (x(k+1) - c(k+1)) \leq 1$ is true under (1) if the following inequalities hold:

$$\begin{bmatrix} \mathbb{I} + \Lambda - \Theta + \sum_{k=1}^{\gamma} \psi_k W_k & K^\top \\ K & P(k+1) \end{bmatrix} \succeq 0 \quad (8)$$

$$\Omega + \sum_{k=1}^{\gamma} \phi_k W_k \succeq 0 \quad (9)$$

for some values of

- $\psi \in \mathbb{R}^{\gamma}$, $\psi = [\psi_1, \psi_2, \dots, \psi_{\gamma}]^{\top}$;
- $\phi \in \mathbb{R}^{\gamma}$, $\phi = [\phi_1, \phi_2, \dots, \phi_{\gamma}]^{\top}$;
- $\Omega = \Omega^{\top} \in \mathbb{R}^{\mu \times \mu}$, which is such that $\zeta^{\top} \Omega \zeta \in \mathbf{R}_{2m-2} [[x(k)^{\top} x(k+1)^{\top}]^{\top}]$;
- $\Gamma_i = \Gamma_i^{\top} \in \mathbb{R}^{\mu \times \mu}$, which is such that $\zeta^{\top} \Gamma_i \zeta \in \mathbf{R}_m [[x(k)^{\top} x(k+1)^{\top}]^{\top}]$, for $i = 1, \dots, n$,

where

- $W_k = W_k^{\top} \in \mathbb{R}^{\mu \times \mu}$ satisfies $\zeta^{\top} W_k \zeta = 0$ for $k = 1, \dots, \gamma$;
- $\Theta = \Theta^{\top} \in \mathbb{R}^{\mu \times \mu}$ is such that $\zeta^{\top} \Theta \zeta = (\zeta^{\top} \Omega \zeta) (1 - (x(k) - c(k))^{\top} P(k)^{-1} (x(k) - c(k)))$;
- $\Lambda = \Lambda^{\top} \in \mathbb{R}^{\mu \times \mu}$ is such that $\zeta^{\top} \Lambda \zeta = \sum_{i=1}^n (\zeta^{\top} \Gamma_i \zeta) (H_i \zeta)$;
- $K \in \mathbb{R}^{n \times \mu}$ is such that $K \zeta = x(k+1) - c(k+1)$;
- $\mathbb{I} \in \mathbb{R}^{\mu \times \mu}$ is such that $\zeta^{\top} \mathbb{I} \zeta = 1$.

Proof

The goal is to assure that $(x(k+1) - c(k+1))^{\top} P(k+1)^{-1} (x(k+1) - c(k+1)) \leq 1$ is satisfied under (1) and $x(k) \in \mathcal{E}(P(k), c(k))$. This is true if $1 - ((x(k+1) - c(k+1))^{\top} P(k+1)^{-1} (x(k+1) - c(k+1)))$ is sum of squares under the required circumstances, i.e. if

$$1 - \zeta^{\top} K^{\top} P(k+1)^{-1} K \zeta + \sum_{k=1}^{\gamma} \zeta^{\top} \psi_k W_k \zeta \geq 0 \text{ when } \begin{cases} 1 - (x(k) - c(k))^{\top} P(k)^{-1} (x(k) - c(k)) \geq 0 \\ h_m(x(k), x(k+1)) = 0. \end{cases} \quad (10)$$

The next step is the use the S-procedure [14, 21] employing polynomial multipliers $\tau(x(k), x(k+1)) \geq 0$ and $\nu_i(x(k), x(k+1))$, for $i = 1, \dots, n$ (more on this later). Namely, we choose $\tau(x(k), x(k+1))$ as a polynomial of degree $2m-2$, i.e. $\tau(x(k), x(k+1)) = \zeta^{\top} \Omega \zeta$, and in this way $\tau(x(k), x(k+1)) (1 - (x(k) - c(k))^{\top} P(k)^{-1} (x(k) - c(k)))$ is of degree $2m$ and can be expressed as $\zeta^{\top} \Theta \zeta$; $\nu_i(x(k), x(k+1)) = \zeta^{\top} \Gamma_i \zeta$ has to be constrained to degree m , so that $(\zeta^{\top} \Gamma_i \zeta) H_i \zeta$ is at most of degree $2m$, and it can be expressed as a quadratic form in ζ . The condition $\tau(x(k), x(k+1)) = \zeta^{\top} (\Omega + \sum_{k=1}^{\gamma} \phi_k W_k) \zeta \geq 0$ demanded by the S-procedure for imposing an inequality constraint is implied by (9) (this is a pure sum of squares problem). Then (10) is implied by

$$\zeta^{\top} (\mathbb{I} + \Lambda - \Theta - K^{\top} P(k+1)^{-1} K + \sum_{k=1}^{\gamma} \psi_k W_k) \zeta \geq 0. \quad (11)$$

By the Schur complement [14], thanks to the fact that $P(k+1) \succ 0$, this is equivalent to (8) in the theorem statement. Then (8) and (9) imply (10) when $x(k)$ belongs to the initial ellipsoid, completing the proof. \square

In this proof the passage from (11) to (10), which we have explained as an S-procedure, can also be seen as an application of Schmüdgen's Positivstellensatz (see [22] or [23] page 29), with a special choice of multipliers. Extending the class of S-procedure terms, i.e. including the products of more than one constraint (i.e. elements in the ideals and preorderings, see again [23] or [24] for an example) as well as increasing m will reduce the conservatism of the result, at the cost of substantially increasing the computational cost.

Theorem 1 can then be exploited in order to solve the problem that we have formulated in the beginning, by searching for the smallest possible ellipsoid $\mathcal{E}(P(k+1), c(k+1))$ containing $x(k+1)$ as a function of $P(k)$, $c(k)$ under (1). This can be done by solving a convex optimisation problem consisting in minimising the trace of $P(k+1)$. The procedure is summarised here.

Optimisation problem 2

Consider (1);

given: $H, P(k), c(k)$,

minimise: $\text{trace}(P(k+1))$,

over: $P(k+1), c(k+1), \Omega, \Gamma_i, \psi$ and ϕ ,

under: (8), (9), $\zeta^\top \Omega \zeta \in \mathbf{R}_{2m-2}[[x(k)^\top x(k+1)^\top]^\top]$ and $\zeta^\top \Gamma_i \zeta \in \mathbf{R}_m[[x(k)^\top x(k+1)^\top]^\top]$.

Notice that the ellipsoids found in this way are not necessarily tight (i.e. the ellipsoids are bigger than the locus of $x(k+1)$), for two reasons. First, Theorem 1 only states a sufficient (and not necessary) condition. Second, the ellipsoidal shape has been arbitrarily imposed: the image of the initial set mapped to the following time step does not necessarily have an ellipsoidal shape.

Solving the problem above repeatedly allows going from the bounds on $x(0)$ to the bounds on $x(N)$ for any $N > 0$ (yielding the systematic or robust simulation of the system).

4. A SPECIAL CASE

In this section we introduce a variant on the main result, which can be applied to discrete-time dynamical systems of order n whose evolution is described by the equation

$$g_m(x(k))x(k+1) = f_m(x(k)) \quad (12)$$

where $k \in \mathbb{N} \cup \{0\}$ is the discrete time variable, $x \in \mathbb{R}^n$ is the state vector, $g_m \in \mathbf{R}_m[x(k)]$ with $m \in \mathbb{N}$, and $f_m(x(k)) \in \mathbb{R}^n$ is a vector of polynomials $\mathbf{R}_m[x(k)]$. The reason for focusing on such a special case (which is not indeed quite restrictive), is that we can derive a set of LMIs that are much smaller than in the case of Theorem 1, as it will be shown later on.

In this case, both g_m and f_m only depend on $x(k)$, and not on $x(k+1)$. We then define the vector $\chi \in \mathbb{R}^\rho$ which contains all the possible monomials obtainable from $x(k)$ from degree 0 up to m (for example, if $n = 2, m = 2$, then $\chi = [x_1(k), x_2(k), x_1(k)^2, x_1(k)x_2(k), x_2(k)^2, 1]^\top$). We also have that

$$\rho = \binom{n+m}{n}. \quad (13)$$

As seen before, any polynomial in the class $\mathbf{R}_m[x(k)]$ can be formulated as a linear function of χ , so

$$f_m(x(k)) = F^\top \chi, \quad g_m(x(k)) = G^\top \chi \quad (14)$$

with $F \in \mathbb{R}^{\rho \times n}$, $G \in \mathbb{R}^\rho$. Notice that of course the size ρ of χ is much smaller than the size μ of ζ . As a consequence of working with χ instead of ζ , also the number ι of slack matrices Q_i , with $i = 1, \dots, \iota$, such that $\chi^\top Q_i \chi = 0$, is much smaller than the number γ of slack matrices W_i in ζ , with

$$\iota = \frac{1}{2} \left(\binom{m+n}{m}^2 + \binom{m+n}{m} \right) - \binom{n+2m}{2m}. \quad (15)$$

We can now formulate the theorem which deals with this special case.

Theorem 3

Consider the dynamical system in (12). If $x(k)$ is such that $(x(k) - c(k))^\top P(k)^{-1} (x(k) - c(k)) \leq 1$ is satisfied, then $(x(k+1) - c(k+1))^\top P(k+1)^{-1} (x(k+1) - c(k+1)) \leq 1$ is true under (12) if the following two inequalities hold:

$$\begin{bmatrix} GG^\top - \Theta + \sum_{k=1}^{\iota} \psi_k Q_k & F - Gc(k+1)^\top \\ F^\top - c(k+1)G^\top & P(k+1) \end{bmatrix} \succeq 0 \quad (16)$$

$$\Omega + \sum_{k=1}^{\iota} \phi_k Q_k \succeq 0 \quad (17)$$

for some values of

- $\psi \in \mathbb{R}^\iota$, $\psi = [\psi_1, \psi_2, \dots, \psi_\iota]^\top$;
- $\phi \in \mathbb{R}^\iota$, $\phi = [\phi_1, \phi_2, \dots, \phi_\iota]^\top$;
- $\Omega = \Omega^\top \in \mathbb{R}^{\rho \times \rho}$, which is such that $\chi^\top \Omega \chi \in \mathbf{R}_{2m-2}[x(k)]$,

where

- $Q_k = Q_k^\top \in \mathbb{R}^{\rho \times \rho}$ satisfies $\chi^\top Q_k \chi = 0$ for $k = 1, \dots, \iota$;
- $\Theta = \Theta^\top \in \mathbb{R}^{\rho \times \rho}$ is such that $\chi^\top \Theta \chi = (\chi^\top \Omega \chi) (1 - (x - c(k))^\top P(k)^{-1} (x - c(k)))$.

Proof

We start by rewriting the expression $(x(k+1) - c(k+1))^\top P(k+1)^{-1} (x(k+1) - c(k+1)) \leq 1$; multiplying the terms of this equation on the left by $g_m(x(k)) = \chi(k)^\top G$ and on the right by $g_m(x(k)) = G^\top \chi(k)$, replacing $g_m(x(k))x(k+1)$ with $f_m(x(k))$ according to (12) we get $\chi(k)^\top G G^\top \chi(k) - (F^\top \chi(k) - c(k+1)G^\top \chi(k))^\top P(k+1)^{-1} (F^\top \chi(k) - c(k+1)G^\top \chi(k)) \geq 0$.

We would like to have this last expression verified (in the SOS sense) when $1 - (x(k) - c(k))^\top P(k)^{-1} (x(k) - c(k)) \geq 0$. For this purpose we can use again the S-procedure with an $x(k)$ -varying multiplier $\tau(x(k)) \geq 0$; let $\tau(x(k)) = \chi^\top \Omega \chi$, $\tau(x(k)) \in \mathbf{R}_{2m-2}[x(k)]$, so that $\tau(x(k))(1 - (x(k) - c(k))^\top P(k)^{-1} (x(k) - c(k)))$ is of degree $2m$ and can be expressed as $\chi^\top \Theta \chi$. The condition $\tau(x(k)) = \chi^\top (\Omega + \sum_{k=1}^\iota \phi_k Q_k) \chi \geq 0$ is implied by (17). Then

$$\begin{aligned} \chi(k)^\top G G^\top \chi(k) - (F^\top \chi(k) - c(k+1)G^\top \chi(k))^\top P(k+1)^{-1} (F^\top \chi(k) - c(k+1)G^\top \chi(k))^\top &\geq 0 \\ \text{when } 1 - (x(k) - c(k))^\top P(k)^{-1} (x(k) - c(k)) &\geq 0 \end{aligned} \quad (18)$$

is implied by (S-procedure)

$$\begin{aligned} &\chi(k)^\top (G G^\top - \Theta + \sum_{k=1}^\iota \psi_k Q_k) \chi(k) \\ &- (F^\top \chi(k) - c(k+1)G^\top \chi(k))^\top P(k+1)^{-1} (F^\top \chi(k) - c(k+1)G^\top \chi(k))^\top \geq 0. \end{aligned} \quad (19)$$

By using the Schur complement [14], thanks to the fact that $P(k+1) \succ 0$, this is equivalent to (16) in the theorem statement. Then (16) and (17) imply (18) when $x(k)$ belongs to the initial ellipsoid, concluding the proof. \square

The proof has been formulated again as relying on the S-procedure. The same proof, namely the passage from (19) to (18) can also be seen as an application of Putinar's Positivstellensatz [25]. Such a theorem (see [23] pages 29-30) guarantees that the SOS formulation of the inequality $(x(k+1) - c(k+1))^\top P(k+1)^{-1} (x(k+1) - c(k+1)) \leq 1$ is non conservative if the degree of $\tau(x(k))$ is sufficiently high. This means that one can reduce the conservatism of the theorem by choosing a bigger value of m (it is always possible to do so), at a higher computational cost.

In this case as well, an optimisation problem allows the use of the theorem.

Optimisation problem 4

Consider (12);

given: $F, G, P(k), c(k)$

minimise: $\text{trace}(P(k+1))$

over: $P(k+1), c(k+1), \Omega, \psi$ and ϕ ,

under: (16), (17) and $\chi^\top \Omega \chi \in \mathbf{R}_{2m-2}[x(k)]$.

Notice that again the ellipsoids found in this way are not necessarily tight, as the ellipsoidal shape has been arbitrarily imposed and the image of the initial set is not necessarily an ellipsoid. The advantage of using Optimisation 4 rather than Optimisation 2, if the dynamical equations can be cast into (12), is that the solution of (16), (17) is sensibly less computationally intensive than (8), (9), as explained in the next section.

5. COMPUTATIONAL COMPLEXITY

As all SOS based methods, the proposed theorems and optimisation procedures are of limited applicability for systems of high order (n) or for polynomials of high degree (m). Figure 1 shows the computational complexity in terms of number of unknowns to be found when running Optimisation 2 and Optimisation 4.

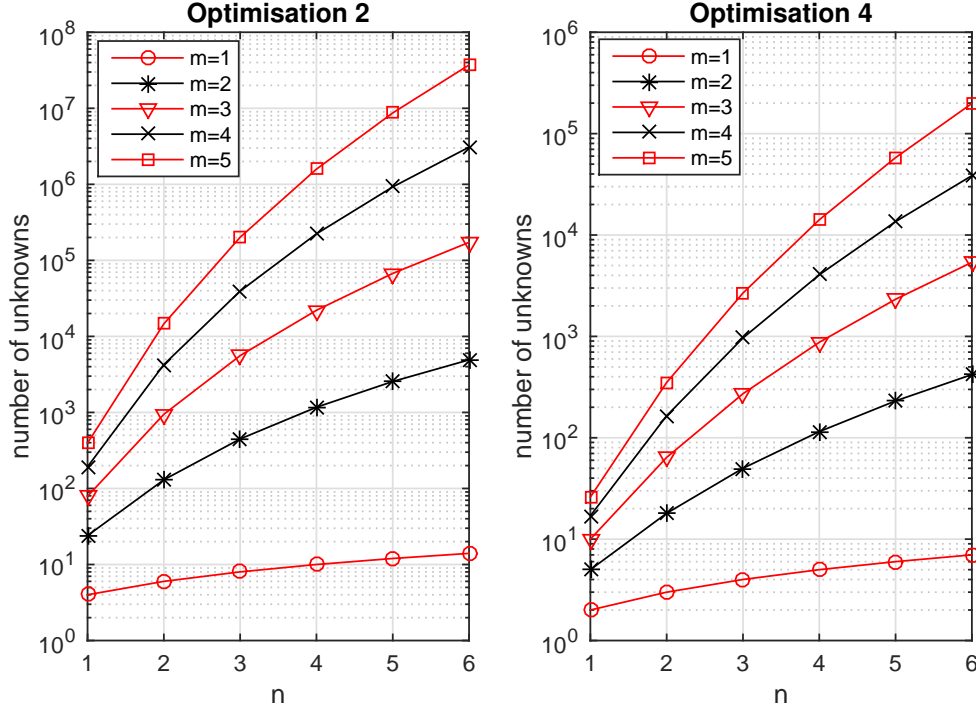


Figure 1. Number of unknowns in the two optimisation problems.

Moreover, the inequalities in (8) (Optimisation 2) are of size μ and $\mu + n$, whereas the ones in (16) (Optimisation 4) are of size ρ and $\rho + n$. As an example, for $n = 2$, $m = 2$, (8) is of size 17×17 and the number of slack matrices is $\gamma = 75$, for a total of 130 unknowns, whereas for (16) the size is 8×8 and the number of slack matrices is $\iota = 6$ for a total of 18 unknowns. In general it is difficult to say which is the maximum number of unknowns that a computer can deal with, but in our experience a number of unknowns in the order of 10^4 should not pose specific problems. In this sense, it is also important that further research efforts (on SOS in general as well) should be focused on devising a method to contain the explosion in complexity.

The next section shows some examples where the computational complexity is manageable and the proposed optimisation problems can successfully be solved.

6. EXAMPLES

6.1. Aerospace applications: attitude dynamics

As a first example, we consider the attitude dynamics of a rigid body in space (like a satellite or a celestial body), under no external torque. The continuous-time dynamic of the three principal body

axis components $\omega_1, \omega_2, \omega_3$ of its rotational velocity is described by Euler's equations [26]

$$\begin{cases} \dot{\omega}_1 = \frac{J_2 - J_3}{J_1} \omega_2 \omega_3 \\ \dot{\omega}_2 = \frac{J_3 - J_1}{J_2} \omega_1 \omega_3 \\ \dot{\omega}_3 = \frac{J_1 - J_2}{J_3} \omega_1 \omega_2 \end{cases} \quad (20)$$

where J_1, J_2, J_3 are the moments of inertia with respect to the principal body axes. Such equations conserve the energy of the system, so a discretization with explicit methods (like forward Euler integration) fails to keep such aspect (in the same way as continuous-time linear systems with poles on the imaginary axis turn into unstable discrete-time systems if forward Euler is used, regardless of how small the sampling time is). For this reason, an implicit discretisation method is preferable, like Tustin's method, which turns (20) into

$$\begin{cases} \omega_1(k+1) = \omega_1(k) - \frac{T}{2} \frac{J_2 - J_3}{J_1} (\omega_2(k)\omega_3(k) - \omega_2(k+1)\omega_3(k+1)) \\ \omega_2(k+1) = \omega_2(k) - \frac{T}{2} \frac{J_3 - J_1}{J_2} (\omega_1(k)\omega_3(k) - \omega_1(k+1)\omega_3(k+1)) \\ \omega_3(k+1) = \omega_3(k) - \frac{T}{2} \frac{J_1 - J_2}{J_3} (\omega_1(k)\omega_2(k) - \omega_1(k+1)\omega_2(k+1)) \end{cases} \quad (21)$$

for any chosen discretisation step T . These discrete-time equations fit into the framework of (1), with $n = 3$ and $m = 2$, which implies $\mu = 28$ and $\gamma = 196$, for a total of 448 unknown variables in Theorem 1. For the sake of this example, we pick $T = 0.5$ s, $J_1 = 1$ kg·m², $J_2 = 1.2$ kg·m², $J_3 = 1.5$ kg·m², and initial conditions in $\mathcal{E}(\alpha^2 I_3, \omega_0)$ (a ball of radius α centered in ω_0), with $\alpha = 0.1$ rad/s and $\omega_0 = [1.8 \ 0 \ 0.5]^\top$ rad/s.

The conditions stated in Theorem 1 have been coded using Matlab and the Yalmip toolbox [27], and SeDuMi [28] has been chosen as solver for the optimisation problem in Optimisation 2. We have propagated the ellipsoid from time 0 s up to 8 s, and compared it with the simulation of the evolution of a set of randomly initial points taken from the initial ellipsoid. Figure 2 shows how the procedure successfully propagates the ellipsoids, as we can see the projection of the results on two different planes, namely the final and initial points/ellipsoid and some intermediate values. On a 2.67 GHz quad-core personal computer, the solver takes 1.17 s to compute each time step.

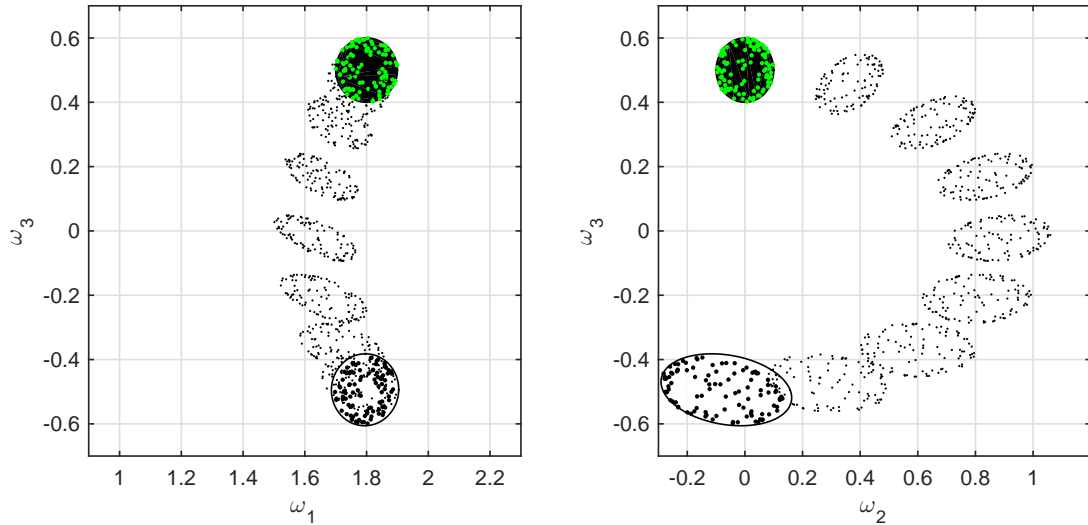


Figure 2. Propagations of the ellipsoids of Example 6.1, projections on the $\omega_1 - \omega_3$ and $\omega_2 - \omega_3$ planes. The black ellipses represent the initial conditions, whereas the empty ellipses are the final result of the propagation after 8 s. The dots are the simulation of randomly generated initial conditions (the bright dots in the initial ellipse), the final values are indicated by the thicker dots.

6.2. Cubic function

We now consider a simple dynamical system described by the scalar expression

$$x(k+1) = \frac{1}{10}x(k)^3 + \frac{8}{10}x(k)^2 + \frac{1}{10}. \quad (22)$$

This equation fits nicely into (12) with $n = 1$, $m = 3$, $\chi(k) = [x(k), x(k)^2, x(k)^3, 1]^\top$,

$$F = \begin{bmatrix} 0 \\ 0.8 \\ 0.1 \\ 0.1 \end{bmatrix}, \quad G = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}. \quad (23)$$

In this case, $\iota = 3$ and $\rho = 4$ for a total of 10 unknowns in Theorem 3, whereas if Theorem 1 were to be used, we would have $\gamma = 27$ and $\mu = 10$ (79 unknowns).

It can be observed that the system in (22) has a locally stable equilibrium point at $x = 0.1$ and a locally unstable one for $x = 1$. Initial conditions between such two points will make the state converge to 0.1, while for $x(0) > 1$ the state will diverge.

Again, by employing Matlab, we have simulated the system for two different ellipsoids of initial conditions:

1. $x(0) = 0.8$, $P(0) = 0.03$;
2. $x(0) = 0.8$, $P(0) = 0.04$.

The first set of initial conditions does not include the unstable equilibrium point. We have iterated the ellipsoid (which in this case of $n = 1$ degenerates to an interval) up to $N = 10$, and we have verified that in fact all its points converge to the stable equilibrium (Figure 3). This is not the case for the second initial condition, for which such convergence is impossible (Figure 4). Notice that in both cases the center of the ellipsoid does not coincide with the propagation of the initial nominal value (i.e. $x(0) = 0.8$). Moreover, in this special case the bounds appear to be tight, by comparing the ellipsoids to the simulation of the upper (“+” symbol) and lower bounds (“o” symbol) in the figure. The solver time is on average 0.15 s per time step.

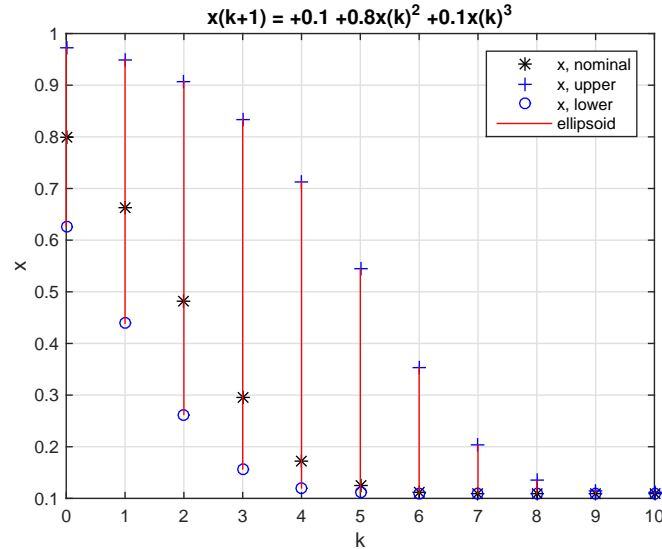


Figure 3. Evolution of the state for Example 6.2, with the initial ellipse not touching the unstable equilibrium point.

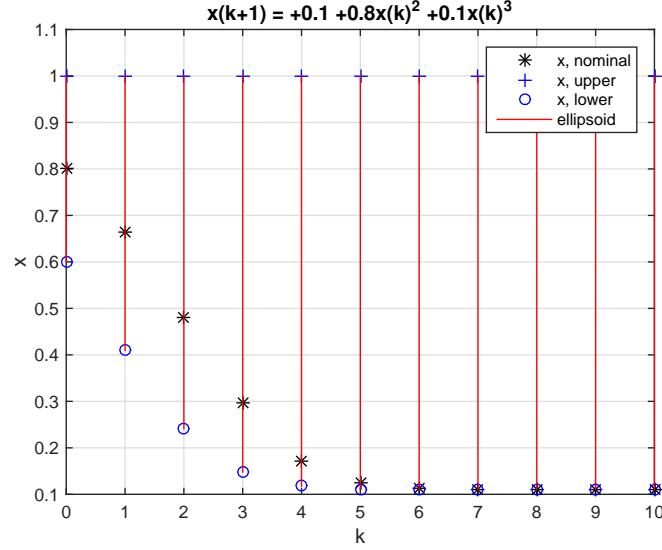


Figure 4. Evolution of the state for Example 6.2, with the initial ellipse ending at the unstable equilibrium point.

6.3. Julia fractal

We now focus on a system with a more complex (and fascinating) dynamics. We consider the following dynamical equation:

$$z(k+1) = z(k)^2 + c \quad (24)$$

where $z, c \in \mathbb{C}$. The invariant sets of such equation are known as Julia sets or Julia fractals [29]. For $z = x_1 + jx_2$, $n = 2$ and $m = 2$ the equation can be recast into the framework of Theorem 3 and Optimisation 4, with $\rho = 6$ and $\iota = 6$ for a total of 18 unknowns. We consider the special case of $c = 0.123 + 0.745j$, for which the invariant set has a fractal shape (nicknamed “the rabbit”), and three attractors defining a three-point limit cycle. We have simulated the evolution of the system for an ellipse located completely inside the invariant set, in its periphery. The simulations, in Figure 5, show that eventually all the points in this initial set converge to the limit cycle, after a few bounces around the plane. The solver takes on average 0.19 s to compute each time step; we have also run, for comparison, the problem in the form of Optimisation 2 ($\mu = 15$, $\gamma = 50$, 130 unknowns), for which we have had an average computational time of 0.38 s per time step, and ellipsoids which can be considered numerically identical.

6.4. Newton fractal

At last, we consider the following dynamical equation:

$$z(k+1) = z(k) + \frac{z(k)^3 - 1}{3z(k)^2} \quad (25)$$

where again $z \in \mathbb{C}$. There are three attractors for this equation; partitioning the plane according to which one the state converges to generates a so-called Newton fractal [29]. The name comes from the fact that (25) is the formula for Newton’s iterations for finding the roots of an equation, namely $z(k)^3 - 1 = 0$ in this case; in fact, the attractors are such roots, i.e. the cubic roots of 1 in the complex plane.

For $z = x_1 + jx_2$, $n = 2$ and $m = 5$ the equation can be recast into the framework of Theorem 3 with $\iota = 165$ and $\rho = 21$ for a total of 351 unknowns (whereas if Theorem 1 were to be used, we would have $\gamma = 7000$, $\mu = 126$ and 14621 unknown variables, which would make the problem too cumbersome to be solved by a common personal computer in a reasonable time). Again we have

simulated the evolution of the system for an ellipse located completely inside one of the invariant, convergent sets, in its periphery. The simulations (Figure 6) show that eventually all the points in this initial set converge to one attractor. The solver time is 0.68 s per time step on average.

7. CONCLUSIONS

In this paper we have shown an efficient method which can be employed in order to validate the properties of dynamical systems through a systematic simulation. The method offers hard mathematical bounds but it requires solving a convex optimisation at each time step, which can be computationally intensive. In any case, we believe that this method can be helpful in relevant application, proposing a different and complementary approach with respect to the popular Monte Carlo approach and other approaches found in the literature. Through examples, we have seen that this systematic simulation can be used to validate regions of attractions for equilibrium points or limit cycles (generalising other results from the literature, like for example [8] which applied to polynomial functions, or [9], which only applied to quadratic functions).

We believe that the main advantage of the proposed method is that it is based on LMIs, so it will be possible to upgrade it exploiting the LMI-based theory on robust control in order to take into account a wide array of possible situations, including for example uncertainties in the model parameters. Concerning other future research directions, we are also interested in applying the method to the simulation of aerospace systems.

REFERENCES

1. Binder K. *Introduction: Theory and technical aspects of Monte Carlo simulations*. Springer, 1986.
2. Donzé A, Maler O. Systematic simulation using sensitivity analysis. *Hybrid Systems: Computation and Control*. Springer, 2007; 174–189.
3. Dang T, Donzé A, Maler O, Shalev N. Sensitive state-space exploration. *47th IEEE Conference on Decision and Control*, IEEE, 2008; 4049–4054.
4. Tierno J, Murray R, Doyle J, Gregory I. Numerically efficient robustness analysis of trajectory tracking for nonlinear systems. *Journal of guidance, control, and dynamics* 1997; **20**(4):640–647.
5. Jönsson U. Robustness of trajectories with finite time extent. *Automatica* 2002; **38**(9):1485–1497.
6. Kantner M, Doyle J. Robust simulation and nonlinear performance. *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, vol. 3, IEEE, 1996; 2622–2623.
7. Kishida M, Braatz R. Ellipsoid bounds on state trajectories for discrete-time systems with time-invariant and time-varying linear fractional uncertainties. *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, IEEE, 2011; 5671–5676.
8. Topcu U, Packard A, Seiler P. Local stability analysis using simulations and sum-of-squares programming. *Automatica* 2008; **44**(10):2669–2675.
9. Calafiore G. Set simulations for quadratic systems. *IEEE Transactions on Automatic Control* 2003; **48**(5):800–805.
10. Ben-Talha H, Massioni P, Scorletti G. Robust simulation of continuous-time systems with rational dynamics. *International Journal of Robust and Nonlinear Control* (in press); .
11. Sassi M, Testylier R, Dang T, Girard A. Reachability analysis of polynomial systems using linear programming relaxations. *International Symposium on Automated Technology for Verification and Analysis*, Springer, 2012; 137–151.
12. Asarin E, Dang T, Girard A. Reachability analysis of nonlinear systems using conservative approximation. *International Workshop on Hybrid Systems: Computation and Control*, Springer, 2003; 20–35.
13. Parrilo P. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming* 2003; **96**(2):293–320.
14. Boyd S, El Ghaoui L, Feron E, Balakrishnan V. *Linear matrix inequalities in system and control theory*, vol. 15. SIAM, 1994.
15. Korda M, Henrion D, Jones C. Convex computation of the maximum controlled invariant set for polynomial control systems. *SIAM Journal on Control and Optimization* 2014; **52**(5):2944–2969.
16. Shia V, Vasudevan R, Bajcsy R, Tedrake R. Convex computation of the reachable set for controlled polynomial hybrid systems. *53rd IEEE Conference on Decision and Control*, IEEE, 2014; 1499–1506.
17. Henrion D, Korda M. Convex computation of the region of attraction of polynomial control systems. *IEEE Transactions on Automatic Control* 2014; **59**(2):297–312.
18. Lasserre J. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization* 2001; **11**(3):796–817.
19. Magron V, Henrion D, Lasserre JB. Semidefinite approximations of projections and polynomial images of semialgebraic sets. *SIAM Journal on Optimization* 2015; **25**(4):2143–2164.
20. Chesi G. LMI techniques for optimization over polynomials in control: a survey. *IEEE Transactions on Automatic Control* 2010; **55**(11):2500–2510.

21. Scherer C. LMI relaxations in robust control. *European Journal of Control* 2006; **12**(1):3–29.
22. Jarvis-Wloszek Z, Feeley R, Tan W, Sun K, Packard A. Control applications of sum of squares programming. *Positive Polynomials in Control* 2005; :580–580.
23. Lasserre J. *Moments, positive polynomials and their applications*, vol. 1. World Scientific, 2009.
24. Pitarch J, Sala A, Lauber J, Guerra TM. Control synthesis for polynomial discrete-time systems under input constraints via delayed-state lyapunov functions. *International Journal of Systems Science* 2016; **47**(5):1176–1184.
25. Putinar M. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal* 1993; **42**(3):969–984.
26. Bong W. *Space vehicle dynamics and control*. AIAA, 1998.
27. Löfberg J. Yalmip: a toolbox for modeling and optimization in MATLAB. *Proc. of the CACSD Conference*, 2004; 284–289.
28. Sturm J. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, vol. 11–12, 1999; 625–653.
29. Barnsley M, Devaney R, Mandelbrot B, Peitgen HO, Saupe D, Voss R, Fisher Y, McGuire M. *The science of fractal images*. Springer Publishing Company, 2011.

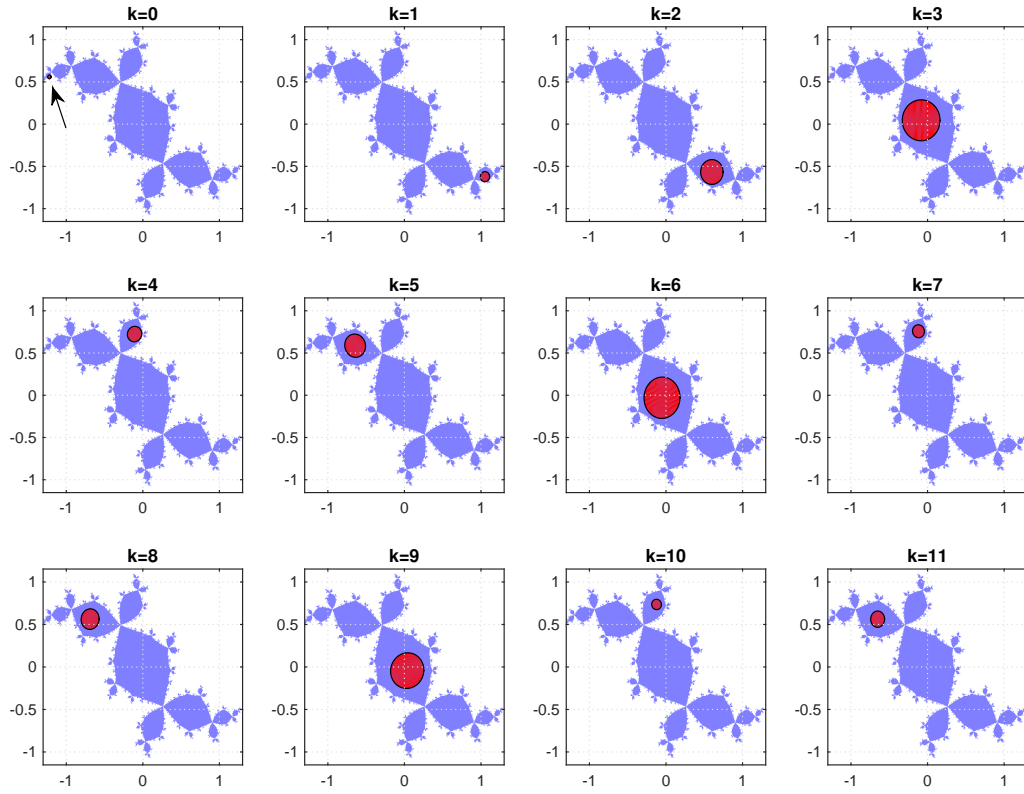


Figure 5. Evolution of the bounding ellipse for Example 6.3. The filled area is the Julia set.

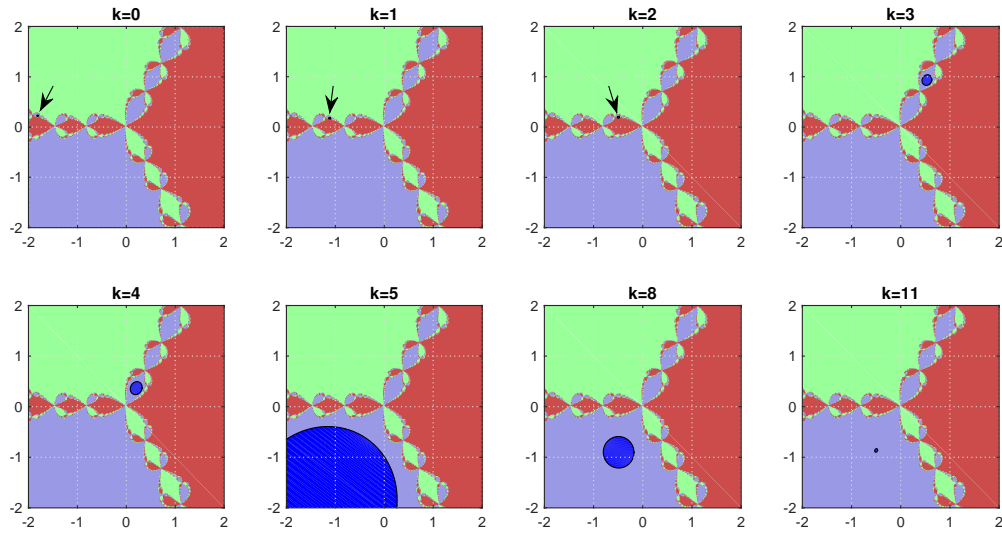


Figure 6. Evolution of the bounding ellipse for Example 6.4. Each area of a different color represents the region of attraction of one of the three solutions.