SOStab – a Sum-of-squares toolbox for stability analysis, with application to transients of a droop controlled power converter

Stéphane Drobot

Ecole polytechnique fédérale de Lausanne (EPFL)

Laboratoire d'automatique: Semester project

Pr. Colin Jones Dr. Matteo Tacchi

SOStab – a Sum-of-squares toolbox for stability analysis, with application to transients of a droop controlled power converter

Drobot Stéphane

Thursday 19th January, 2023

Contents

1 Introduction

Power converters are a key technology in the new generation of electrical power grids, characterized by fluctuating energy sources (e.g. wind or solar power), for which a power conversion is needed to transform the supply into a normalized, high voltage 50 Hz signal. Especially, the large scale replacement of traditional energy sources with renewables induces a loss in the mechanical inertia provided by fuelled powered turbines, that used to maintain stability at the grid level. A challenge for future power grids is to compensate for this loss in inertia by coming up with innovative power converter control laws, such as droop control.

The study of non-linear control laws such as the droop control is very challenging, as the problem of finding a solution to a non-linear differential system in a reasonable time is complex. This project proposes to use the sum-of-squares programming approach to explore the transient stability of a droop control law. Using polynomials is a convenient way to assess the stability of a system. They can be exactly represented on a computer and they are also dense in most functional spaces, so they can be make good approximations of many generic functions. However, the use of polynomials in spaces with more than 3 dimension comes at a high computational cost.

As the framework for solving a differential dynamic through SOS programming is always the same, the matlab code used to solve the droop control system was generalized into a toolbox in order to simplify the use of polynomial programming for future researches. It allowed to re-obtain results from the recent scientific literature with a minimum of coding.

1.1 Droop-controlled power converter

The characteristic of power converters is that they are able to produce sinusoidal electric signals with phase θ, frequency ω and amplitude ρ, through fast, high frequency switching. As a consequence, θ, ω and ρ can be fixed quite arbitrarily, provided that the converter receives an appropriate amount of power to generate the desired signal. However, to provide inertia similarly to standard power generation, these state variables can be assigned *ad hoc*, slow dynamics that would enforce a desired behaviour of the converter.

Droop control is an example of such dynamics, with inputs ω^*, p^* and q^*, the desired set points for frequency and active and reactive power. It mimics the speed control used in synchronous generators.

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = \omega \tag{1}$$

$$\frac{\mathrm{d}\omega}{\mathrm{d}t} = \frac{1}{T_\omega}\left[-\omega + \omega_b(\omega^* - \omega_{dq} + m_p(p^* - \mathbf{v}_c \cdot \mathbf{i}_g))\right] \tag{2}$$

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = k_i\left(v^* - \rho + n_q(q^* - \mathbf{v}_c \cdot \mathbf{J}\mathbf{i}_g)\right) \tag{3}$$

where the states variables are:

- θ the angle of the voltage of the power converter
- ω the rotational speed of the voltage
- ρ is the amplitude of the voltage

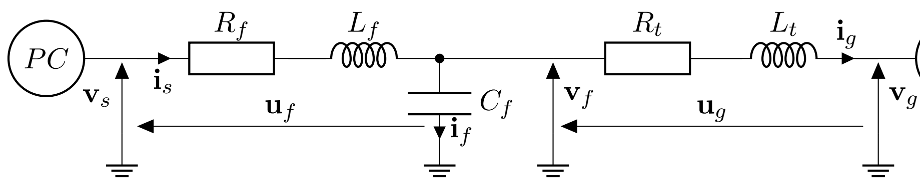


Figure 1: Power converter PC connected to an infinite bus PG.

- $\mathbf{i}_g$ is the current in the line

and $\mathbf{v}_c = \rho \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}$ is the voltage of the converter.

It must be noted that these three equations do not describe any physical reality, they are a way to control the state of the power converter in order to simulate the inertia needed for the electric grid stability.

The Kirchhoff and Ohms law applied on the power converter give a fourth equation – which is electrical this time –, that complete the differential system:

$$\frac{\mathrm{d}\mathbf{i}_g}{\mathrm{d}t} = \frac{1}{L_t}(\mathbf{v}_c - \mathbf{v}_g - (R_t\mathbf{I} + L_t\omega_{dq}\mathbf{J})\mathbf{i}_g) \tag{4}$$

where $\mathbf{v}_g = v_\infty \begin{pmatrix} \cos((\omega_\infty - \omega_{dq})\omega_b t) \\ \sin((\omega_\infty - \omega_{dq})\omega_b t) \end{pmatrix}$ is the voltage applied by the grid

1.2 Region of attraction and its approximations with SOS optimization

In order to assess the security of this control, the stability of the system is measured thanks to the calculation of the region of attraction of the equilibrium point. As we consider fast systems, the study focuses on "constrained finite horizon Region of attraction" (ROA): we look for all the initial states that will go back to a set K – around the equilibrium point – in a finite time T.

Mathematically, it can be defined as follows: for a dynamical system $\dot{\mathbf{x}} = f(\mathbf{x})$, the ROA of K at time $T > 0$ is

$$A_T^M(K) := \left\{\mathbf{x}_0 \in M : \forall t \in [0,T], \mathbf{x}(t|\mathbf{x}_0) \in M \wedge \mathbf{x}(T|\mathbf{x}_0) \in K\right\}$$

where M is the admissible state and $\mathbf{x}(t|\mathbf{x}_0)$ describes the state of the system at time t, given that the initial state was $\mathbf{x}_0$, *ie* $t \mapsto \mathbf{x}(t|\mathbf{x}_0)$ is the trajectory of the system starting at $\mathbf{x}_0$.

In the formula as well as in the report, $\mathbf{x} \in \mathbb{R}^n$ describes the state variables of the system, n being the dimension of the dynamical system, $\mathbf{X}$ designates an indeterminate variable of the same size as the state variable and $\mathbf{T}$ designates an indeterminate variable (of size 1) related to the time t. This notation will be used whenever mentioning a generic optimization problem.

In order to describe this region of attraction, the optimization problem is strengthened using the Lasserre hierarchy, which enables the use of Sum of squares (SOS) programming.

An SOS optimization program is a type of optimization problem, with a linear objective and constraints on polynomial variables that are forced to be in the form of sum of squares of polynomials. When bounding the degree of the polynomials, the problem becomes equivalent to a semi-definite program, which implies that existing solvers for SOS programming are quite efficient. The Lasserre hierarchy transforms the initial optimization problem with variables in an infinite dimension space (trajectories of the system are functions) into a tractable problem, equivalent to SDP.

As the Lasserre hierarchy gives a strengthening of the original problem, the obtained region of attraction will be bigger than the real one: it is an outer approximation. It is possible to calculate an inner approximation by solving the reciprocal problem: an outer approximation of the ROA of the complementary to the target set will give an inner approximation of the initial ROA. The implementation of the calculation of both outer and inner approximation was done for the droop controlled power converter.

1.3 A toolbox for SOS programming of region of attraction

As the framework used to solve computations of regions of attractions through SOS programming is always the same and a bit tedious, the second goal of the project – after solving the problem for the power converter – was to generalize the used code into a toolbox. Its goal is to provide a relatively simple and user-friendly way to calculate region of attraction from the minimal information required:

- the dynamic of the system
- the equilibrium point
- the distance to the equilibrium point that defines the feasible set
- the size of the ball around the equilibrium that defines the target set
- the horizon time at which the target set has to be attained.

The toolbox can calculate both inner and outer approximations of the region of attraction, but the calculation of the inner approximation is very sensitive to the parameters given as input.

1.4 Report organization

The report includes:

- A presentation of the Lasserre hierarchy in section 2.
- The explanation of the application of the Lasserre hierarchy to the calculation of a volume – in the first place – and then the region of attraction of a dynamical system in section 3. It is then applied to the calculation of the ROA of the droop controlled power converter.
- Finally, section 4 presents the toolbox derived from the codes used in the previous section.

2 Lasserre hierarchy on a simple optimization problem

In order to explain the implementation of SOS programming for transient stability analysis, it is needed to explain the Lasserre hierarchy. The Lasserre hierarchy is a sequence of relaxation of an functional optimization problem, *ie* on infinite dimension sets, which allows to get a tractable polynomial optimization problem to approach the optimum solution. The following introduction to the hierarchy is mostly based on [Henrion and Korda, 2013].

2.1 Preliminary: sums of squares polynomials

2.1.1 Sum of squares

Definition: A multivariate polynomial $p \in \mathbb{R}[\mathbf{X}]$ is SOS if it can be written as a sum of squares of polynomials:

$$p = \sum_{i=1}^{N} q_j^2$$

where $\forall i, q_i \in \mathbb{R}[\mathbf{X}]$. We then define the cone of polynomial sums of squares:

$$\Sigma[\mathbf{X}] = \{p_1^2 + \cdots + \mathcal{P}_d^2 : d \in \mathbb{N}, p_1, \ldots, p_d \in \mathbb{R}[\mathbf{X}]\}$$

as well as the bounded degree cone:

$$\Sigma_\delta[\mathbf{X}] = \{p_1^2 + \cdots + p_d^2 : d \in \mathbb{N}, p_1, \ldots, p_d \in \mathbb{R}[\mathbf{x}]_\delta\}$$

Lemma: The interest of using SOS polynomials as constraints of optimization problems comes from the fact that a SOS polynomial can be interpreted as positive semi-definite matrix, hence SOS programming in equivalent to SDP Programming:

If p is SOS, then it can be written

$$\begin{aligned}
p = \sum_{i=1}^{N} q_i^2 &= \sum_{i=1}^{N} \left(\sum_{k=1}^{d} q_i^k \mathbf{X}^{\alpha_k} \right)^2 \\
&= \sum_{i=1}^{N} \left(\sum_{k=1}^{d} \sum_{l=1}^{d} q_i^k q_i^l \mathbf{X}^{\alpha_k} \mathbf{X}^{\alpha_l} \right) \\
&= \sum_{i=1}^{N} \left(\sum_{k=1}^{d} \mathbf{X}^{\alpha_k} \sum_{l=1}^{d} q_i^k q_i^l \mathbf{X}^{\alpha_l} \right) \\
&= \sum_{i=1}^{N} \left(\mathbf{X}^{\alpha} q_i^\intercal q_i \mathbf{X}^{\alpha} \right)
\end{aligned}$$

where $\mathbf{X}^\alpha = (\mathbf{X}^{\alpha_k})_{k=1\ldots d}$ is a vector of monomials of $\mathbf{X}$ and $q_i = (q_i^k)_{k=1\ldots d}$ is the vector of coefficient of the polynomial q_i. Defining $\forall i, Q_i = q_i^\intercal q_i \in \mathcal{M}(\mathbb{R})^d$, by definition, the Q_i are semi definite positive matrices.

If now Q is an SDP matrix, it can be interpreted as the square of a polynomial. We diagonalize the matrix $Q = P^\intercal D P$, with P being orthogonal and D is the diagonal $(d_1, \ldots, d_d)$ – all the d_i

being positive, as Q is SDP. Then

$$\begin{aligned}(\mathbf{X}^\alpha)^\intercal Q\mathbf{X}^\alpha =& (\mathbf{X}^\alpha)^\intercal P^\intercal DP\mathbf{X}^\alpha \\ =& (P\mathbf{X}^\alpha)^\intercal D(P\mathbf{X}^\alpha) \\ =& \sum_{i=1}^{d} d_i\left((P\mathbf{X}^\alpha)_i\right)^2 \\ =& \sum_{i=1}^{d} d_i\left(\sum_{k=1}^{d} P_{ik}\mathbf{X}^{\alpha_k}\right)^2 \\ =& \sum_{i=1}^{d}\left(\sqrt{d_i}\sum_{k=1}^{d} P_{ik}\mathbf{X}^{\alpha_k}\right)^2\end{aligned}$$

which can be interpreted as the sum of the squares of the polynomials $p_i = \sum_{k=1}^{d}\sqrt{d_i}P_{ik}\mathbf{X}^{\alpha_k}$

2.1.2 Putinar's Positivestellensatz

Definition: Semi-algebraic set A vector $\mathbf{g} \in \mathbb{R}[\mathbf{X}]^m$ of polynomials defines a semi-algebraic set

$$K(\mathbf{g}) = \{\mathbf{x} \in \mathbb{R}^n : \quad \forall i \in [\![1,m]\!], g_i(\mathbf{x}) \geqslant 0\}$$

where $[\![1,m]\!]$ defines the set of all the integers between 1 and m: $\{1,\ldots,m\}$

Definition: Quadratic module The quadratic module of a set of polynomials $\mathbf{g}$ is the set defined by

$$\Sigma[\mathbf{g}] = \{q_0 + \mathbf{q}\cdot\mathbf{g} : \quad q_0 \in \Sigma[\mathbf{X}],\ \mathbf{q} \in \Sigma[\mathbf{X}]^m\}$$

where the q_i are all sum of squares.

Theorem (Putinar): The following theorem, due to [Putinar, 1993, Lemma 3.2] is essential to the Lasserre hierarchy.

Let $K(\mathbf{g})$ a semi-algebraic set as defined above be included in a ball B. If a polynomial p is strictly positive on $K(\mathbf{g})$, then it is in the quadratic module $\Sigma[\mathbf{g}]$. In other terms, we have the implication

$$\forall p \in \mathbb{R}[\mathbf{x}], p > 0 \text{ on } K(\mathbf{g}) \Longrightarrow \exists q_0,\ldots,q_m \in \Sigma[\mathbf{X}], p = q_0 + \sum_{i=1}^{m} q_i g_i$$

2.2 Definition of the initial problem on measures

Let us define the type of problem that will be solved. This is a very common optimization problem: maximizing a function $f : M \subset \mathbb{R}^n \to \mathbb{R}$ on a set K, that is imposed to be semi-algebraic. This problem can be formulated as a problem on measures maximizing the function is the same as finding a measure that maximize the integration of f. For a set S, $\mathcal{M}(S)$ designates the set of measures on S, while $\mathcal{M}(S)_+$ is the set of positive measures.

$$\begin{aligned} p_f^* = \quad & \sup \quad \int f \, \mathrm{d}\mu \\ & \text{s. t.} \quad \mu \in \mathcal{M}(K)_+ \\ & \qquad\quad \mu(K) = 1 \end{aligned} \tag{5}$$

The Lagrangian of the problem is

$$\mathcal{L}(\mu, c) = \int f \, \mathrm{d}\mu + (1 - \mu(K))c = \int (f - c) \, \mathrm{d}\mu + c$$

with $c \in \mathbb{R}$ being the only multiplier, which can be negative or positive because of the constraint being an equality.

If there is a,n $x \in K$ such that $f(x) - c > 0$, then the measure $\mu = A\delta_x$ with δ the Dirac measure at point x and A a positive value, will give an arbitrarily large value to the Lagrangian, hence the maximization over the measure $\mu \in \mathcal{M}(K)_+$ will go to infinity (note that μ is a positive measure). Then, the maximization of $\mathcal{L}$ over μ gives two cases:

$$\max_{\mu \in \mathcal{M}(K)_+} \mathcal{L}(\mu, c) = \begin{cases} +\infty & \text{if } \exists x \in K, f(x) - c > 0 \\ c & \text{if } f - c \leqslant 0 \text{ on all } K \end{cases}$$

The dual of this problem is then the minimization of the previous maximization, which corresponds to the minimization of an upper bound of f:

$$\begin{aligned} d_f^* = \quad & \inf \quad c \\ & \text{s. t.} \quad c \in \mathbb{R} \\ & \qquad\quad f - c \leqslant 0 \text{ on } K \end{aligned} \tag{6}$$

2.3 Lasserre hierarchy for the dual problem

The study here will focus on the Lasserre hierarchy for the dual problem – there is one for the primal aswell. It consits in following the sequence of strengthenings of the problem of minimizing f on $K(\mathbf{g})$):

1. Choose f to be a polynomial: $f \in \mathbb{R}[\mathbf{x}]$

$$\begin{aligned} d_f^* = \quad & \inf \quad c \\ & \text{s. t.} \quad c \in \mathbb{R} \\ & \qquad\quad c - f \geqslant 0 \text{ on } K(\mathbf{g}) \end{aligned} \tag{6}$$

2. Use Putinar's Positevestellensatz 2.1.2 to get an SOS condition from the inequality constraint

$$\begin{aligned} d_{\mathbb{R}[X]}^* = \quad & \inf \quad c \\ & \text{s. t.} \quad c \in \mathbb{R}, q_0 \in \Sigma, \mathbf{q} \in \Sigma^m \\ & \qquad\quad c - f = q_0 + \mathbf{q} \cdot \mathbf{g} \end{aligned} \tag{6 - $\mathbb{R}[\mathbf{x}]$}$$

3. Restrict the degree of the polynomials to save calculation time

$$
\begin{aligned}
d^*_\delta = \quad & \inf \quad c \\
& \text{s. t.} \quad c \in \mathbb{R}, q_0 \in \Sigma_\delta, \mathbf{q} \in \Sigma^m_\delta \\
& \qquad\quad c - f = q_0 + \mathbf{q} \cdot \mathbf{g}
\end{aligned}
\tag{6 - δ}
$$

If f is a polynomial, the reformulation 6 - $\mathbb{R}[\mathbf{x}]$ does not change the value of the optimal solution. Indeed, thanks to the Putinar's positivestellensatz we have that the constraint $c - f = q_0 + \mathbf{q} \cdot \mathbf{g}$ is equivalent to $c - f > 0$ – the implication in the opposite direction to the theorem is trivial. Then, changing the constraint from $c - f > 0$ to $c - f \geqslant 0$ does not change the value of the lower bound – but can change it from an attainable minimum to a unattained lower bound – so, we have:

$$d^*_{\mathbb{R}[\mathbf{x}]} = d^*_f$$

Then, the restriction of the polynomial degree of the variables, decreases the size of the admissible set, so the lower bound increases. We have $d^*_\delta \geqslant d^*_f$ for all degree δ and when the degree goes to infinity, the solution converges towards d^*_f:

$$\lim_{\delta \to \infty} d^*_\delta = d^*_f$$

The goal is now to apply the principle of this hierarchy to the calculation of a region of attraction of a dynamic system.

3 Calculation of a region of attraction

3.1 The volume of a set as an optimization problem

The volume of the Region of attraction of a dynamical system (1.2) can be expressed as an optimization problem which can then be solved to compute the ROA.

In terms of measures, the volume of a set $B \subset M$ is by definition the integral of the Lebesgue measure λ_B of the set. Then, this Lebesgue measure can be expressed as the measure that maximize the integral over the set, while being inferior to the Lebesgue measure on the set B, which gives the following expression of the volume:

$$\begin{aligned} \mathrm{Vol}(B) = \lambda(B) = \quad & \sup \quad \int \mathrm{d}\mu_0 \\ & \text{s. t.} \quad \mu_0 \in \mathcal{M}(B)_+, \overline{\mu}_0 \in \mathcal{M}(M)_+ \\ & \qquad \mu_0 + \overline{\mu}_0 = \lambda_M \end{aligned}$$

The sum of two measures equal to λ_M – the Lebesgue measure on M – is simply another way to express the fact that μ_0 is less or equal to the Lebesgue measure on the set B: $\mu_0 \leqslant \lambda_B$. However it allows not to use the Lebesgue measure on the set B.

This reformulation is then useful when the Lebesgue measure of the set M is known, but not the one of the set B, which is the case for the region of attraction. In terms of physical intuition, this definition of the volume is analogous to defining the volume of a physical object as the maximum possible mass of a liquid which occupy only the volume of the object and whose density is less or equal to 1 kg.L^{-1}. The optimum liquid would be water filling the object which then give the volume of the object the density of water is known.

For the ROA, it gives the expression of its volume as:

$$\begin{aligned} \mathrm{Vol}(A_T^M(K)) = \quad & \sup \quad \int \mathrm{d}\mu_0 \\ & \text{s. t.} \quad \mu_0 \in \mathcal{M}\left(A_T^M(K)\right)_+, \overline{\mu}_0 \in \mathcal{M}(M)_+ \\ & \qquad \mu_0 + \overline{\mu}_0 = \lambda_M \end{aligned} \tag{7}$$

And the expression of the set will be obtained simply by taking the argument of the maximization problem – as it is obtained simultaneously to the solution when using a solver for the problem:

$$\begin{aligned} A_T^M(K) = \quad & \text{spt argmax} \quad \int \mathrm{d}\mu_0 \\ & \text{s. t.} \quad \mu_0 \in \mathcal{M}\left(A_T^M(K)\right)_+, \overline{\mu}_0 \in \mathcal{M}(M)_+ \\ & \qquad \mu_0 + \overline{\mu}_0 = \lambda_M \end{aligned}$$

The Lagrangian of 7 is harder to resolve than the previous one. The multiplier for the constraint on measures $\mu_0 + \overline{\mu}_0 = \lambda_M$ will be a continuous real function w (an equality gives a single real multiplier, a continuous equality gives a continuous multiplier).

The Lagrangian is then

$$\begin{aligned} \mathcal{L} : \mathcal{M}\left(A_T^M(K)\right)_+ \times \mathcal{M}(M)_+ \times \mathcal{C}(M) \quad & \longmapsto \quad \mathbb{R} \\ (\mu_0, \overline{\mu}_0, w) \quad & \longrightarrow \quad \int \mathrm{d}\mu_0 + \int w \left(\mathrm{d}\lambda_M - \mathrm{d}\mu_0 - \mathrm{d}\overline{\mu}_0\right) \end{aligned}$$

The separation between integrals on M and only $A_T^M(K)$ gives:

$$\mathcal{L}(\mu_0, \overline{\mu}_0, w) = \int_{A_T^M(K)} (1-w)\ \mathrm{d}\mu_0 - \int_M w\ \mathrm{d}\overline{\mu}_0 + \int_M w\ \mathrm{d}\lambda_M$$

The maximization of the Lagrangian gives:

on μ_0**:** if there is an $x \in A_T^M(K)$ such that $1 - w(x) > 0$, then the Lagrangian is unbounded: a weighted Dirac measure on x can result in an arbitrarily large value. If $1 - w$ is negative on $A_T^M(K)$ then the maximization of the first integral would be 0, with the null measure.

on $\overline{\mu}_0$**:** if there is an $x \in M$ such that $w(x) < 0$, then the Lagrangian is unbounded. If w is non-negative on M, then the maximum of this integral is 0.

Finally, the dual of 7 is an optimization problem on functions, with two constraints: w non-negative on M and $w \leqslant 1$ on $A_T^M(K)$.

$$\begin{aligned} \mathrm{Vol}(A_T^M(K)) = \quad \inf \quad & \int_M w \mathrm{d}\lambda_M \\ \text{s. t.} \quad & w \in \mathcal{C}(M) \\ & w \geqslant 0 \text{ on } M \\ & w \geqslant 1 \text{ on } A_T^M(K) \end{aligned} \tag{8}$$

3.2 Reformulation of the problem with time

In order to make the dynamics of the system appear in the optimization problem, it is needed to reformulate it to introduce the notion of trajectories of the system.

As the set $A_T^M(K)$ is unknown, the constraint $\mu_0 \in \mathcal{M}(A_T^M(K))$ is expressed thanks to the notion of occupation measure. An occupation measure $\mu \in \mathcal{M}([0,T], M)$ is a measure on both space of the system (M) and time ($[0,T]$). It is subject to Liouville's transport Partial Differential Equation, *ie*

$$\exists \mu_0, \mu_T, \frac{\partial \mu}{\partial t} + \nabla \cdot (f\mu) - \delta_0 \mu_0 + \delta_T \mu_T = 0$$

where $\delta_{0/T}$ are the (time) Dirac measures for $t = 0$ and $t = T$ respectively.

Mathematically speaking, the first part of the Liouville's PDE ($\frac{\partial \mu}{\partial t} + \nabla \cdot (f\mu)$) is the hermitian adjoint of the derivative along the trajectories: for a dynamical system $\dot{\mathbf{x}} = f(\mathbf{x})$, the derivative along the trajectories of a function $v : [0,T] \times M \to \mathbb{R}$ is

$$\frac{\mathrm{D}v}{\mathrm{D}t} = \frac{\partial v}{\partial t} + f \cdot \nabla v \tag{9}$$

Physically speaking, the occupation measure can be understood as a measure that follows the trajectories of the dynamical system with some probabilities on each trajectory. Instead of considering one initial state $\mathbf{x}_0$ that we follow along its trajectory, we consider a continuous set of initial states, a distribution of those states[1], defined by a measure μ_0. The trajectory of the measure – expressed by the occupation measure – is the weighted trajectory of all those initial states and the final distribution – at time T – of the initial states is the measure μ_T.

Then, if we go backwards: when μ_T is a measure on the target space K, then μ_0 is a measure on the ROA, because by definition the occupation measure follows the trajectories of the system.

Then, the constraint $\mu_0 \in \mathcal{M}(A_T^M(K))$ is reformulated as:

[1] *ie* each initial state has a "probability" of happening

- $\mu_0 \in \mathcal{M}(M)_+$: note that the measure is not defined on $A_T^M(K)$ only, but the following constraints ensure that it is null outside of the region of attraction
- $\mu_T \in \mathcal{M}(K)_+$
- $\mu \in \mathcal{M}([0,T], M)_+$
- $\frac{\partial \mu}{\partial t} + \nabla \cdot (f\mu) - \delta_0 \mu_0 + \delta_T \mu_T = 0$

The optimization problem is now

$$\begin{aligned} p_M^* = \quad & \sup \quad \int \mathrm{d}\mu_0 \\ & \text{s. t.} \quad \mu \in \mathcal{M}([0,T], M)_+, \mu_0 \in \mathcal{M}(M)_+, \overline{\mu}_0 \in \mathcal{M}(M)_+, \mu_T \in \mathcal{M}(K)_+ \\ & \qquad \tfrac{\partial \mu}{\partial t} + \nabla \cdot (f\mu) - \delta_0 \mu_0 + \delta_T \mu_T = 0 \\ & \qquad \mu_0 + \overline{\mu}_0 = \lambda \end{aligned} \tag{10}$$

The Lagrangian multiplier to the Liouville PDE is again a function, on both time and state: $v \in \mathcal{C}^1([0,T] \times M)$. This time, it must be continuously derivable. The Lagrangian has two more variables:

$$\mathcal{L} : \mathcal{M}([0,T], M)_+ \times \mathcal{M}\left(M\right)_+ \times \mathcal{M}(M)_+ \times \mathcal{C}^1([0,T], M) \times \mathcal{C}(M) \longmapsto \mathbb{R}$$

and one supplementary term:

$$\mathcal{L}(\mu, \mu_0, \overline{\mu}_0, v, w) = \int \mathrm{d}\mu_0 + \int w\ (\mathrm{d}\lambda_M - \mathrm{d}\mu_0 - \mathrm{d}\overline{\mu}_0) - \left\langle v, \frac{\partial \mu}{\partial t} + \nabla \cdot (f\mu) - \delta_0 \mu_0 + \delta_T \mu_T \right\rangle$$

where the operator $\langle\ ,\ \rangle$ is in simple terms the integration of v with the measures on the right side. The integration of a time Dirac gives the value of v at $0/T$, so the last terms are simply:

$$\langle v, \delta_0 \mu_0 - \delta_T \mu_T \rangle = \iint v\ \mathrm{d}\left(\delta_0 \mu_0 - \delta_T \mu_T\right) = \int v(0, \cdot)\ \mathrm{d}\mu_0 - \int v(T, \cdot)\ \mathrm{d}\mu_T$$

As the derivative of a measure is ill-defined, the first two terms will be expressed using the hermitian adjoint of the operator:

$$\left\langle v, \frac{\partial \mu}{\partial t} + \nabla \cdot (f\mu) \right\rangle = -\left\langle \frac{\partial v}{\partial t} + f \cdot \nabla v, \mu \right\rangle = -\left\langle \frac{\mathrm{D}v}{\mathrm{D}t}, \mu \right\rangle = -\iint \frac{\mathrm{D}v}{\mathrm{D}t}\ \mathrm{d}\mu$$

Then the Lagrangian is rewritten:

$$\begin{aligned} \mathcal{L}(\mu, \mu_0, \overline{\mu}_0, v, w) &= \int \mathrm{d}\mu_0 + \int w\ (\mathrm{d}\lambda_M - \mathrm{d}\mu_0 - \mathrm{d}\overline{\mu}_0) + \int v(0, \cdot)\ \mathrm{d}\mu_0 - \int v(T, \cdot)\ \mathrm{d}\mu_T + \iint \frac{\mathrm{D}v}{\mathrm{D}t}\ \mathrm{d}\mu \\ &= \int w\ \mathrm{d}\lambda_M + \int (1 - w + v(0, \cdot))\ \mathrm{d}\mu_0 - \int w\ \mathrm{d}\overline{\mu}_0 - \int v(T, \cdot)\ \mathrm{d}\mu_T + \iint \frac{\mathrm{D}v}{\mathrm{D}t}\ \mathrm{d}\mu \end{aligned}$$

The maximization of the Lagrangian gives:

on μ_0**:** if there is an $x \in M$ such that $1 - w(x) + v(0, x) > 0$, then the Lagrangian is unbounded: a weighted Dirac measure on x can result in an arbitrarily large value. If $1 - w + v(0, \cdot)$ is negative on M then the maximization of the first integral would be 0, with the null measure.

on $\overline{\mu}_0$: if there is an $x \in M$ such that $w(x) < 0$, then the Lagrangian is unbounded. If w is non-negative on M, then the maximum of this integral is 0.

on μ_T: if there is an $x \in M$ such that $v(T,x) < 0$, then the Lagrangian is unbounded. If $v(T,\cdot)$ is non-negative on M, then the maximum of this integral is 0.

on μ: if there is a couple $t \in [0,T], x \in M$ such that $\frac{\mathrm{D}v}{\mathrm{D}t}(t,x) > 0$, then the Lagrangian is unbounded. If $\frac{\mathrm{D}v}{\mathrm{D}t}$ is non-positive on M, then the maximum of this integral is 0.

The dual problem has then two additional constraints, as well as an additional variable. Its new formulation is:

$$\begin{aligned} d_M^* = \quad & \inf \quad \int_M w(\mathbf{x})\mathrm{d}\mathbf{x} \\ & \text{s. t.} \quad v \in \mathcal{C}\left([0,T] \times M\right), w \in \mathcal{C}(M) \\ & \qquad -\tfrac{\partial v}{\partial t} - f \cdot \nabla v \geqslant 0 \text{ on } [0,T] \times M \\ & \qquad v(T,\cdot) \geqslant 0 \text{ on } K \\ & \qquad w - v(0,\cdot) - 1 \geqslant 0 \text{ on } M \\ & \qquad w \geqslant 0 \text{ on } M \end{aligned} \tag{11}$$

3.3 Applying the Lasserre hierarchy

Similarly to the example given in subsection 2.3, the Lasserre hierarchy is applied on the dual problem.

First, the sets $M = K(\mathbf{h})$ and $K = K(\mathbf{g})$ need to be semi-algebraic in order to be able to use the Putinar's theorem:

$$\exists \mathbf{g} \in \mathbb{R}[\mathbf{X}]^{N_1}, \mathbf{h} \in \mathbb{R}[\mathbf{X}]^{N_2}, K = \{\mathbf{g}(\mathbf{x}) \geqslant 0 : \mathbf{x} \in \mathbb{R}^n\}, M = \{\mathbf{h}(\mathbf{x}) \geqslant 0 : \mathbf{x} \in \mathbb{R}^n\}$$

Then, the dynamic of the system f needs to be also expressed as a polynomial function of the state $\mathbf{x}$:

$$f \in \mathbb{R}[\mathbf{X}]$$

The variables of the optimization problem – v and w – need also to be polynomials. The application of the hierarchy will impose a bounded degree to those two polynomials.

Finally, all the inequality constraints are transformed into equality constraints through the use of Putinar's theorem 2.1.2.

The problem solved on matlab is then, for a given degree δ:

$$\begin{aligned} d_\delta^* = \quad & \inf \quad \int_M w(\mathbf{x})\mathrm{d}\mathbf{x} \\ & \text{s. t.} \quad v \in \mathbb{R}_\delta[\mathbf{X},\mathbf{T}], w \in \mathbb{R}_\delta \mathbf{X}] \\ & \qquad -\tfrac{\partial v}{\partial t} - f \cdot \nabla v = q_{10} + \mathbf{q}_1 \cdot \mathbf{h}, \quad q_{10} \in \Sigma_\delta[\mathbf{X},\mathbf{T}], \mathbf{q}_1 \in \Sigma_\delta[\mathbf{X},\mathbf{T}]^{N_2} \\ & \qquad v(T,\cdot) = q_{20} + \mathbf{q}_2 \cdot \mathbf{g}, \quad q_{20} \in \Sigma_\delta[\mathbf{X}], \mathbf{q}_2 \in \Sigma_\delta[\mathbf{X}]^{N_1} \\ & \qquad w - v(0,\cdot) - 1 = q_{30} + \mathbf{q}_3 \cdot \mathbf{h}, \quad q_{30} \in \Sigma_\delta[\mathbf{X}], \mathbf{q}_3 \in \Sigma_\delta[\mathbf{X}]^{N_2} \\ & \qquad w = q_{40} + \mathbf{q}_4 \cdot \mathbf{h}, \quad q_{40} \in \Sigma_\delta[\mathbf{X}], \mathbf{q}_4 \in \Sigma_\delta[\mathbf{X}]^{N_2} \end{aligned} \tag{12}$$

3.4 Inner approximation of the ROA

As said in section 1, the strengthening of the problem by restricting the degrees of the polynomials reduces the size of the admissible set, so it increases the solution $d_\delta^* \geqslant d_M^*$, the calculated volume is larger than the volume of the ROA, hence the obtained set is an outer approximation of the ROA.

However, in terms of security for the energy systems, it would be more relevant to have inner approximation of the set, in order to make sure that there are no false positives in the obtained ROA approximation. This can be done by calculating the complementary set of the ROA with the same method, this way the outer approximation of the complementary set would be an inner approximation of the ROA.

A state $\mathbf{x}_0$ is not in the ROA if either

- the trajectory starting at $\mathbf{x}_0$, $(\mathbf{x}(t)|\mathbf{x}_0)$ goes out of the feasible set of the system at some time $t \in [0, T]$
- or the trajectory does not end up in the target set: $(\mathbf{x}(T)|\mathbf{x}_0) \notin K$

Problem 12 is then reformulated. The second alternative changes the condition of the problem into $v(T, \cdot) \geqslant 0$ on K^c. The complementary of K inside M is the set where $\mathbf{g}$ is negative (not in K) but $\mathbf{h}$ is positive (as it is still in M), so $K^c = K(-\mathbf{g}, \mathbf{h})$.

The first alternative has no counterpart in Equation 12, so a condition has to be added: v has to be positive on the boundary of M (at all times t), this way a trajectory that ends up on the boundary has necessarily started at a state where $v(0, \cdot)$ is positive. The boundary of M is the set of points where one of the inequality defining it is an equality, so it is defined by $\{\mathbf{x} : \mathbf{h}(\mathbf{x}) \geqslant 0 \wedge \exists i, h_i(\mathbf{x}) = 0\}$ This corresponds to a condition

$$\forall t \in [0, T], \forall \mathbf{x} \in M, \prod_i h_i(\mathbf{x}) = 0 \Rightarrow v(t, \mathbf{x}) \geqslant 0$$

which can be formulated as an equality:

$$\exists q_{50}, q_{50'} \in \Sigma_\delta[\mathbf{X}, \mathbf{T}], \mathbf{q}_5 \in \Sigma_\delta[\mathbf{X}, \mathbf{T}]^{N_2}, p_5 \in \mathbb{R}_{2\delta}[\mathbf{X}, \mathbf{T}],\ v = q_{50} + \mathbf{T}(T - \mathbf{T})q_{50'} + p_5 \prod_i h_i + \mathbf{q}_5 \cdot \mathbf{h}$$

where the second term expresses that $t \in [0, T]$ and the third term uses a simple polynomial because it is an equality. One can easily check with this definition of v that if the conditions of the implication above are met, then v is positive.

The (dual) inner approximation problem is then:

$$\begin{aligned}
d_\delta^* = \quad \inf \quad & \int_M w(\mathbf{x}) \mathrm{d}\mathbf{x} \\
\text{s. t.} \quad & v \in \mathbb{R}_\delta[\mathbf{X}, \mathbf{T}], w \in \mathbb{R}_\delta[\mathbf{X}] \\
& -\tfrac{\partial v}{\partial t} - f \cdot \nabla v = q_{10} + \mathbf{q}_1 \cdot \mathbf{h}, && q_{10} \in \Sigma_\delta[\mathbf{X}, \mathbf{T}], \mathbf{q}_1 \in \Sigma_\delta[\mathbf{X}, \mathbf{T}]^{N_2} \\
& v(T, \cdot) = q_{20} - \mathbf{q}_2 \cdot \mathbf{g} + \mathbf{q}_{2'} \cdot \mathbf{h}, && q_{20} \in \Sigma_\delta[\mathbf{X}], \mathbf{q}_2 \in \Sigma_\delta[\mathbf{X}]^{N_1}, \mathbf{q}_{2'} \in \Sigma_\delta[\mathbf{X}, \mathbf{T}]^{N_2} \\
& w - v(0, \cdot) - 1 = q_{30} + \mathbf{q}_3 \cdot \mathbf{h}, && q_{30} \in \Sigma_\delta[\mathbf{X}], \mathbf{q}_3 \in \Sigma_\delta[\mathbf{X}]^{N_2} \\
& w = q_{40} + \mathbf{q}_4 \cdot \mathbf{h}, && q_{40} \in \Sigma_\delta[\mathbf{X}], \mathbf{q}_4 \in \Sigma_\delta[\mathbf{X}]^{N_2} \\
& v = q_{50} + \mathbf{T}(T - \mathbf{T})q_{50'} + p_5 \textstyle\prod_i h_i && q_{50}, q_{50'} \in \Sigma_\delta[\mathbf{X}, \mathbf{T}], \mathbf{q}_5 \in \Sigma_\delta[\mathbf{X}, \mathbf{T}]^{N_2}, \\
& \quad + \mathbf{q}_5 \cdot \mathbf{h} && p_5 \in \mathbb{R}_{2\delta}[\mathbf{X}, \mathbf{T}]
\end{aligned} \tag{13}$$

The calculation of the inner approximation of the ROA was implemented in the toolbox. We will see later, that the modification of one constraint and the adding of another one have a lot of impact on the performances of the optimization.

3.5 Lasserre hierarchy for the droop controlled Power Converter

We recall the 5 dimension dynamical system of the power converter, composed of the three laws of the droop control and the electrical law on the current $\mathbf{i}_g$:

$$\begin{aligned}\frac{\mathrm{d}\theta}{\mathrm{d}t} &= \omega \\ \frac{\mathrm{d}\omega}{\mathrm{d}t} &= \frac{1}{T_\omega}\left[-\omega + \omega_b(\omega^* - \omega_{dq} + m_p(p^* - \mathbf{v}_c \cdot \mathbf{i}_g))\right] \\ \frac{\mathrm{d}\rho}{\mathrm{d}t} &= k_i\left(v^* - \rho + n_q(q^* - \mathbf{v}_c \cdot \mathbf{J}\mathbf{i}_g)\right) \\ \frac{\mathrm{d}\mathbf{i}_g}{\mathrm{d}t} &= \frac{1}{L_t}(\mathbf{v}_c - \mathbf{v}_g - (R_t\mathbf{I} + L_t\omega_{dq}\mathbf{J})\mathbf{i}_g)\end{aligned} \tag{PC}$$

Where:

- State variables of the dynamical system are in blue: $(\theta, \rho, \omega, \mathbf{i}_g)^\intercal$ – 5 dimension state, as $\mathbf{i_g}$ has two dimensions
- Starred quantities are the set objectives
- $\mathbf{v}_c = \rho(\cos\theta, \sin\theta)^\intercal$ is a function of the variables
- $\mathbf{v}_g = v_\infty(\cos((\omega_\infty - \omega_{dq})\omega_b t), \sin((\omega_\infty - \omega_{dq})\omega_b t))^\intercal$ depends on the external grid quantities

For the hierarchy to be usable, the function describing the dynamics of the system needs to be a polynomial of the state variables, which is not the case here, due to the presence of the sine and cosine function. A change of variable into

$$(\cos\theta, \sin\theta, \rho, \omega, \mathbf{i_g})$$

makes the dynamic polynomial (as a function of the new variables):

$$\begin{aligned}\frac{\mathrm{d}\cos\theta}{\mathrm{d}t} &= -\omega\sin\theta \\ \frac{\mathrm{d}\sin\theta}{\mathrm{d}t} &= \omega\cos\theta \\ \frac{\mathrm{d}\omega}{\mathrm{d}t} &= \frac{1}{T_\omega}\left[-\omega + \omega_b(\omega^* - \omega_{dq} + m_p(p^* - \rho(\cos\theta, \sin\theta)^\intercal \cdot \mathbf{i}_g))\right] \\ \frac{\mathrm{d}\rho}{\mathrm{d}t} &= k_i\left(v^* - \rho + n_q(q^* - \rho(\cos\theta, \sin\theta)^\intercal \cdot \mathbf{J}\mathbf{i}_g)\right) \\ \frac{\mathrm{d}\mathbf{i}_g}{\mathrm{d}t} &= \frac{1}{L_t}(\rho(\cos\theta, \sin\theta)^\intercal - \mathbf{v}_g - (R_t\mathbf{I} + L_t\omega_{dq}\mathbf{J})\mathbf{i}_g)\end{aligned} \tag{PC polynomial}$$

Then, for the order of magnitude of the solution to remain in a bounded range, all variables, are centered around 0 and scaled to a maximum absolute value of 1. This way, the admissible set

M will be a subset of the hyper cube $[-1,1]^n$, so the values of all polynomials in M can't diverge even when going to high degree polynomials.

Finally, the used "polynomial" and scaled variables are:

$$\mathbf{x} = \begin{pmatrix} \sin(\theta - \theta_{eq}) \\ \frac{1-\cos(\theta-\theta_{eq})}{2} \\ \frac{\omega - \omega_{eq}}{\Delta\omega} \\ \frac{\rho - \rho_{eq}}{\Delta\rho} \\ \frac{i_{g1} - i_{g,eq1}}{\Delta i_g} \\ \frac{i_{g2} - i_{g,eq2}}{\Delta i_g} \end{pmatrix}$$

3.6 Matlab implementation

The calculation of the region of attraction was done on Matlab, using the Yalmip toolbox on Sum-of-Square programming [Löfberg, 2009], with the solver mosek [ApS, 2022].

The equilibrium point $\mathbf{x}_{eq}$ was calculated as the point where the derivative is null $\frac{\mathrm{d}\mathbf{x}_{eq}}{\mathrm{d}t} = 0$, using symbolic variables in Matlab.

The admissible set used was defined by:

$$\Delta\mathbf{x} = (\Delta\sin, \Delta\cos, \Delta\omega, \Delta\rho, \Delta i_{g1}, \Delta i_{g2})^\intercal = (1, 1, 10\pi \text{ rad.s}^{-1}, 1 \text{ pu}, 10 \text{ pu}, 10 \text{ pu})^\intercal$$

Note that:

- the angle θ is unconstrained, so the range for its sine and cosine has no particular meaning
- the range for the current is far above real permitted value, but as this modelling of the power converter does not take into account fast controls on the current, it was not realistic to give the real upper bound of the system (which would be around 1.5 A).

The values of the different parameters are:

- $\omega_b = 100\pi$ rad.s^{-1}
- $\omega^* = 1$ pu
- $\omega_\infty = 1$ pu
- $\omega_{dq} = 1$ s^{-1} pu
- $T_\omega = \frac{1}{20\pi}$ s
- $m_p = 0.001$ pu
- $n_q = 0.022$ pu
- $p^* = 0.95$ pu
- $q^* = 0.31$ pu
- $v^* = 1$ pu
- $v_\infty = 1$ pu
- $k_i = 20\pi$ pu
- $R_t = 0.015$ pu
- $Lt = 0.25$ pu

The feasible set $K(\mathbf{h}) = [-1,1]^n$ is the semi-algebraic set defined by n polynomials

$$\forall i \in [\![1, n]\!], h_i(\mathbf{x}) = 1 - \mathbf{x}_i^2$$

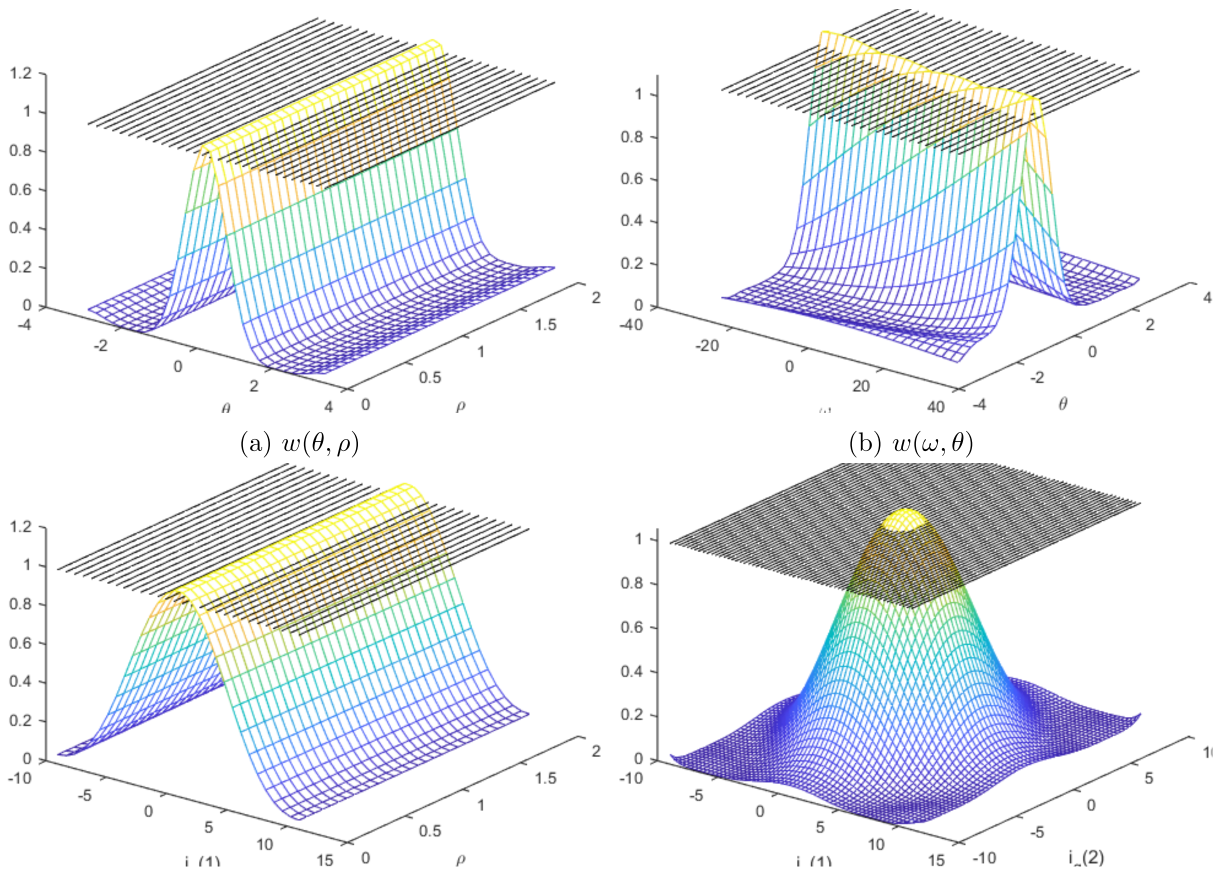
The target set $K(g)$ is the ball of radius $\varepsilon = 0.1$ around 0, a semi-algebraic set defined by

$$g(\mathbf{x}) = \varepsilon^2 - \mathbf{x}^\intercal\mathbf{x}$$

3.7 Results: outer approximation of the region of attraction

The optimization problem (12) is then solved, for different degree of polynomials. It must be noted that in dimension 6, the size of the polynomials increases polynomially with the degree, which makes the calculation intractable above a certain degree. My computer – with 16 GB of RAM – has not enough memory to run the problem in degree 10, where the polynomials can have up to 20'000 coefficients in dimension 7 (6 space dimensions and one temporal).

Here are presented the results for maximum polynomial degrees of 4, 6 and 8($ie\ \delta = 2, 3, 4$)[2]. The plots are projected on two dimensions of the system, all other variables are taken at the equilibrium.

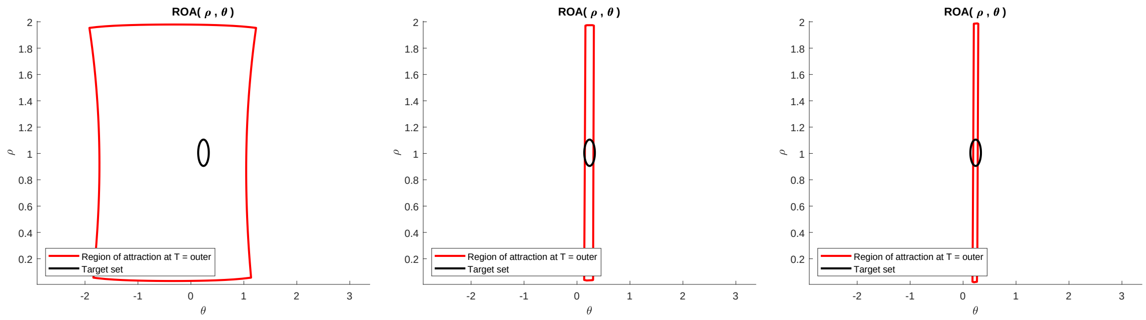


(a) $w(\theta, \rho)$ (b) $w(\omega, \theta)$

(c) $w(i_{g1}, \rho)$ (d) $w(i_{g1}, i_{g2})$

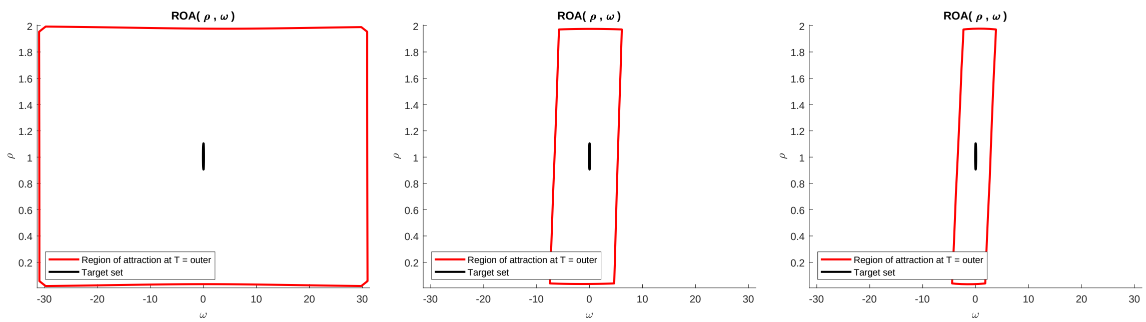
Figure 2: Plots of w as a function of pairs of variables, at degree $\delta = 4$

[2]Computational time is respectively around 60s, 267s and 722s



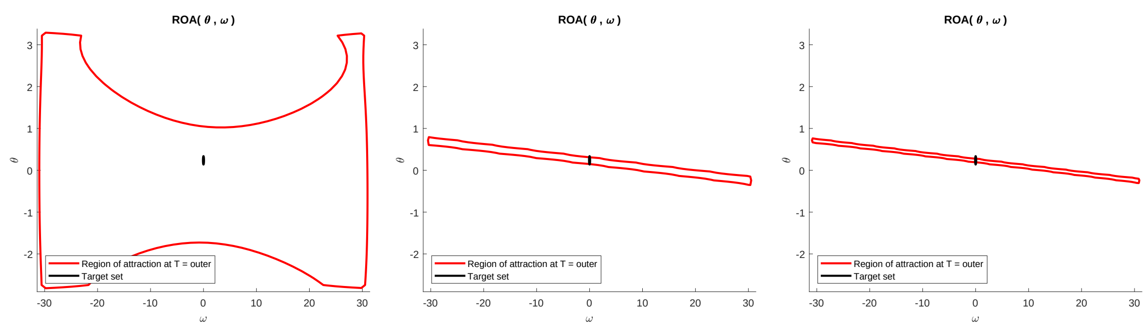
(a) ROA in degree $\delta = 2$ (b) ROA in degree $\delta = 3$ (c) ROA in degree $\delta = 4$

Figure 3: ROA calculation as a function of ρ, θ



(a) ROA in degree $\delta = 2$ (b) ROA in degree $\delta = 3$ (c) ROA in degree $\delta = 4$

Figure 4: ROA calculation as a function of ρ, ω



(a) ROA in degree $\delta = 2$ (b) ROA in degree $\delta = 3$ (c) ROA in degree $\delta = 4$

Figure 5: ROA calculation as a function of ρ, θ

4 Creating a toolbox for SOS programming

The optimization problem in Equation 12 solved for describing the region of attraction of a dynamical system depends only on the definitions of the sets and the dynamic of the system. On the other hand, there are many constraints and the code implementation of the problem is not trivial.

Hence, the goal of this toolbox is to implement the problem in a generic form, that enable an easy use of polynomial optimization for finite time ROA calculations, with the minimum input required. This toolbox is available at https://github.com/droste89/SOStab.

4.1 Preprocessing by the user

While the toolbox integrates a part of the standardization of the problem, it is asked to the user to give as input a preprocessed version of his problem.

Let us say that we want to study the region of attraction of the following generic (*ie* non polynomial) problem:

- the system follows a dynamic $\dot{\mathbf{\Psi}} = h(\mathbf{\Psi}, t)$ where h is a Lipschitz continuous function[3]
- the equilibrium point is defined by $0 = h(\mathbf{\Psi}_{eq})$
- the feasible set is defined by $\mathbf{\Psi} \in [\mathbf{\Psi}_l, \mathbf{\Psi}_u]$
- the target set is K_Ψ, that has to be attained in a time T

In the 5 dimension droop control problem, this corresponds to the dynamic PC.

The user has to transform this problem into a polynomial problem and describe the target set as an ellipsoid around the equilibrium point. The problem becomes:

- a polynomial dynamic $\dot{\mathbf{\Phi}} = g(\mathbf{\Phi}, t)$ where g is a polynomial function $g \in \mathbb{R}[\mathbf{\Phi}, t]^n$
- the equilibrium point is defined by $0 = g(\mathbf{\Phi}_{eq})$
- the feasible set is defined by $\mathbf{\Phi} \in [\mathbf{\Phi}_{eq} - \Delta\mathbf{\Phi}, \mathbf{\Phi}_{eq} + \Delta\mathbf{\Phi}]$
- the target set K_Φ is an ellipsoid $K_\Phi = \{\mathbf{\Phi} : \|A(\mathbf{\Phi} - \mathbf{\Phi}_{\boldsymbol{eq}})\| \leqslant \varepsilon\}$, where A is a symmetric definite positive matrix $A \in \mathcal{S}_n^+, \det(A) = 1$ and ε is a strictly positive value

In the 5 dimension droop control problem, this corresponds to the dynamic PC polynomial, with $A = I_6$ being the identity matrix and $\varepsilon = 0.1$.

Note, that the dimension of the system can increase in this process. For example, usually an angle (*ie* a variable whose sine or cosine is used) is put in polynomial form by replacing it by its sine and cosine. In this toolbox, for a proper plotting of those angle variables, the change of a variable α into its sine and cosine must be precisely into the pair of variable $\sin(\alpha - \alpha_{eq}), 0.5(\cos(\alpha - \alpha_{eq}) - 1)$.

[3] The dynamic can be time-dependant

4.2 Principle of the toolbox

The toolbox takes as inputs the polynomial problem formulated by the user. Its working principle is then the following:

1. The input dynamic is scaled in order for the feasible set to be included in $[-1,1]^n$:
   - The time variable t is scaled $s = t/T$
   - The state variable is scaled

   $$\mathbf{x} = D\left(\mathbf{\Phi} - \mathbf{\Phi}_{eq}\right) = \begin{pmatrix} 1/\Delta\mathbf{\Phi}_1 & 0 & \cdots & 0 \\ 0 & 1/\Delta\mathbf{\Phi}_2 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1/\Delta\mathbf{\Phi}_n \end{pmatrix} \left(\mathbf{\Phi} - \mathbf{\Phi}_{eq}\right)$$

   - The equilibrium point is now 0 and the feasible set is the same as the one defined in subsection 3.6 $K(\mathbf{h}) = [-1,1]^n$
2. The dynamic f is deduced from g

   $$\dot{\mathbf{x}} = D\left(\dot{\mathbf{\Phi}}\right) = Dg\left(\mathbf{\Phi}, t\right) = Dg\left(\mathbf{\Phi}_{eq} + D^{-1}\mathbf{x}, Ts\right) = f(\mathbf{x}, s)$$

3. the target set is defined by $a(\mathbf{x}) = \varepsilon^2 - \left(AD^{-1}\mathbf{x}\right)^\intercal AD^{-1}\mathbf{x}$, as

   $$\|A(\mathbf{\Phi} - \mathbf{\Phi_{eq}})\| \leqslant \varepsilon \Longleftrightarrow \left\|AD^{-1}\mathbf{x}\right\| \leqslant \varepsilon \Longleftrightarrow \varepsilon^2 - \left(AD^{-1}\mathbf{x}\right)^\intercal AD^{-1}\mathbf{x} \geqslant 0$$

4. All the constraints of Equation 12 are implemented inside the toolbox.

4.3 Precise description of the toolbox

More specifically, the toolbox is a `matlab` class `sostab`, composed of a number of properties and methods.

4.3.1 Properties of the class

Properties can be divided into properties defining the dynamic and properties related to a particular solution of the optimization problem. The dynamic is defined by

- **dimension**, the dimension of the problem
- **phi_eq**, the equilibrium state $\mathbf{\Phi}_{eq}$
- **phi_str**, a list of stings giving the name of the variables (for plotting purpose)
- **delta_phi**, the distance to the equilibrium defining the feasible set $\Delta\mathbf{\Phi}$
- **angle_eq**, the equilibrium angles if the problem involves angles – empty if no variables correspond to an angle
- **angle_str**, a list of name of the angles

- **phi**, a Yalmip sdpvar of the size of the problem used to define the dynamic g in matlab
- **t**, a Yalmip sdpvar of size 1, representing time and used to define the dynamic g
- **polynomial_dynamic**, a Yalmip polynomial defining the dynamic of the system g
- **D**, the matrix of variable change from $\mathbf{\Phi}$ to $\mathbf{x}$
- **invD**, the inverse of the matrix D
- **solver**, the solver to use in the optimization, defined as Mosek [ApS, 2022] by default
- **verbose**, the value of the verbose parameters of the Yalmip optimization call, defined at 2 by default

Parameters related to a specific optimization, calculated at each call of the optimization are:

- **d**, the degree of the variable polynomials of the optimization (v and w)
- **A**, the matrix defining the target set
- **epsilon**, the positive value defining the target set
- **vcoef_outer**, coefficients of the solution v for the last calculated outer approximation of the ROA
- **wcoef_outer**, coefficients of the solution w for the last calculated outer approximation of the ROA
- **vcoef_inner**, coefficients of the solution v for the last calculated inner approximation of the ROA
- **wcoef_inner**, coefficients of the solution w for the last calculated inner approximation of the ROA
- **solution**, the volume of the last calculated ROA, *ie* the solution of the optimization

4.3.2 Methods

The class is composed of four methods.

- The class initialization **sos_optimisation** takes as input the equilibrium point $\mathbf{\Phi}_{eq}$, the distance $\Delta\mathbf{\Phi}$ and the names of the variables as three vector, that must have the same length – the dimension of the problem. It initializes an object of the class `sos_optimisation`.
- **integration** is a method called inside the other methods, which calculates the integration of all the monomials on the feasible set $[-1, 1]^n$.
- **solveoptim** takes as input a matrix A and a positive real ε – that define the target set –, a degree d for the polynomials and a time horizon T. It solves the optimization problem and returns the volume of the calculated outer approximation ROA and the coefficients of the polynomials v and w.

- **solveoptim_inner** takes the same inputs and solves the counterpart of the previous problem, which is the inner approximation optimization problem. The explanation of the inner approximation is given in subsection 3.4. It returns the same values as the previous method.
- **plot_roa** takes as inputs the two indices of the variables on which to project the ROA, a string which states which approximation to plot (outer or inner) and an optional vector giving the size of the plotting mesh. It plots the expected projection of the ROA. If both inner and outer approximations are called sequentially for the same variables, the two plots will appear on the same figure. This corresponds to the plots in Figure 3.
- **plot_w** takes the same inputs. It plots the expected projection of w in 3D. This corresponds to the plots in Figure 2.
- **plot_v** takes the same inputs, as well as a time $t \in [0, T]$. It plots the expected projection of $v(t, \cdot)$ in 3D.

4.4 Application of the toolbox on finite time ROA calculated in the literature

To verify the performances of the toolbox and verify its pertinence, its results were compared to published studies of ROA calculations using SOS programming.

4.4.1 Van der Pol oscillator

The first problem is the Van der Pol oscillator, studied in [Korda et al., 2013]. It is a problem in 2 dimensions corresponding to the following dynamic:

$$\begin{pmatrix} \dot{x_1} \\ \dot{x_2} \end{pmatrix} = \begin{pmatrix} -2x_2 \\ 0.8x_1 + 10(x_1^2 - 0.21)x_2 \end{pmatrix}$$

For that problem, the toolbox gives the expected results under certain conditions for both outer and inner approximations.

The computational time of those calculations is approximately 50% higher for the inner approximation:

Degree	10	12	14	16	18	20
Outer ROA	86.773	102.787	123.184	150.327	195.956	227.600
Inner ROA	121.169	144.026	173.790	213.685	279.038	335.784

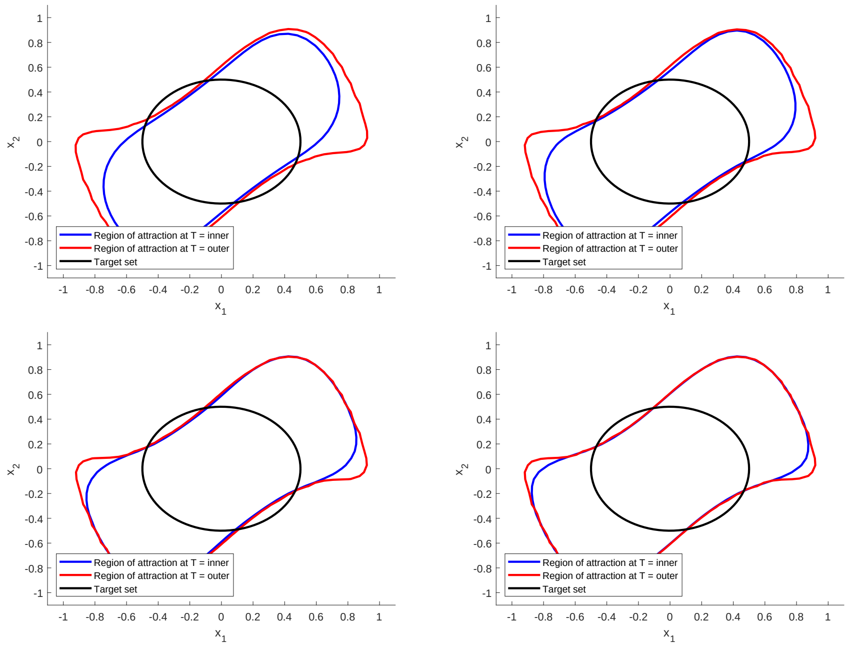


Figure 6: Plots of the inner (blue) and outer (red) ROA for the Van Der Pol oscillator for polynomial maximum degree in $\{12, 14, 16, 18\}$ ($\delta \in \{6, 7, 8, 9\}$)

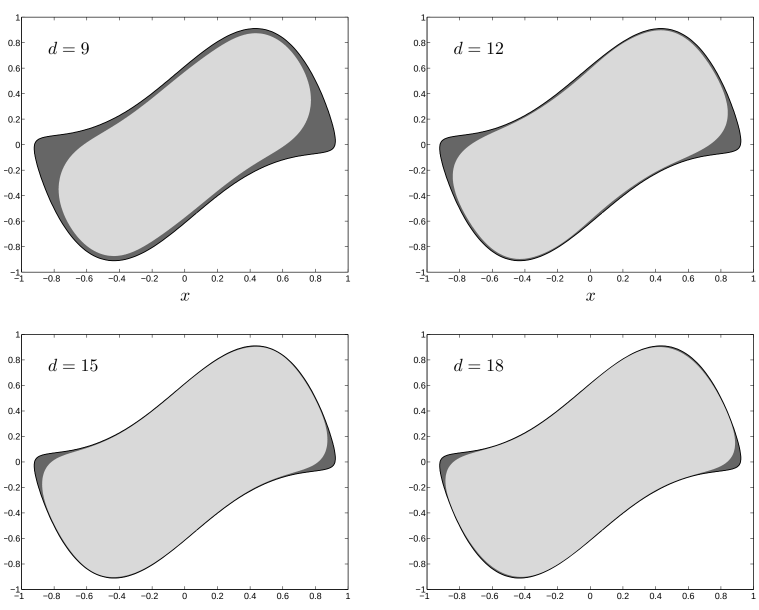


Figure 7: Van der Pol oscillator – polynomial inner approximations (light gray) to the ROA (dark gray) for degrees $d \in \{9, 12, 15, 18\}$ from [Korda et al., 2013]

4.4.2 Three synchronous machines

The second studied problem is an energy grid problem, with three coupled synchronous machines. One is taken as the reference, so the problem has four variables: 2 angles $\theta_{1,2}$ and 2 rotational speed $\omega_{1,2}$. The problem is then lifted to dimension 6 as the angles are replaced by their sine and cosine.

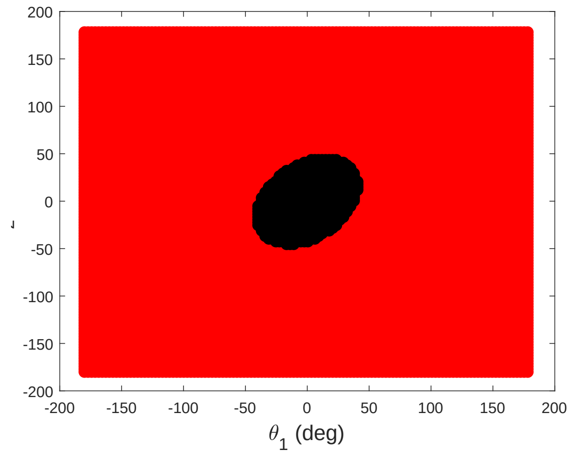


Figure 8: Triple synchronous machines – polynomial outer approximations (in black) for the ROA for degrees 6 ($\delta = 3$) from [Josz et al., 2019]. Projection on θ_1, θ_2

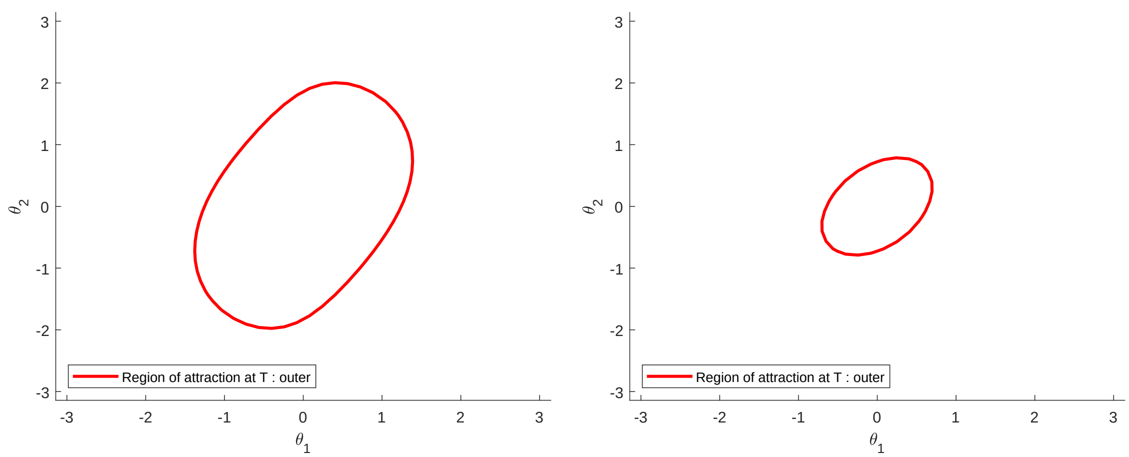


Figure 9: Plots of the outer (red) ROA for the TSM for degrees in $\{6, 8\}$ with the toolbox

Here, the inner approximation of the toolbox does not give any relevant results, which is certainly due to the too low degree of the polynomials on that problem – which is more constrained than the outer one.

However, the approximation of polynomial degree 8 from the toolbox is very close to the ROA of [Josz et al., 2019][4], while the degree 6 approximation is a bit over-sizing the set.

[4]To be precise, this plot is not the one appearing in the article, but it was obtained using the same matlab code

4.5 Limits of the toolbox

4.5.1 Degree limitation

The precision of the calculation depends on the degree of the problem but so does the computational time. The problem being reformulated as Semi-Definite Programming, the computational time will grow polynomially with the degree. However, the issues with solving big SDPs are not only about time but deals also with the computational material. Above some threshold – met quite rapidly – a standard amount of 16 Go of RAM will be saturated and will just make it impossible to solve the optimization.

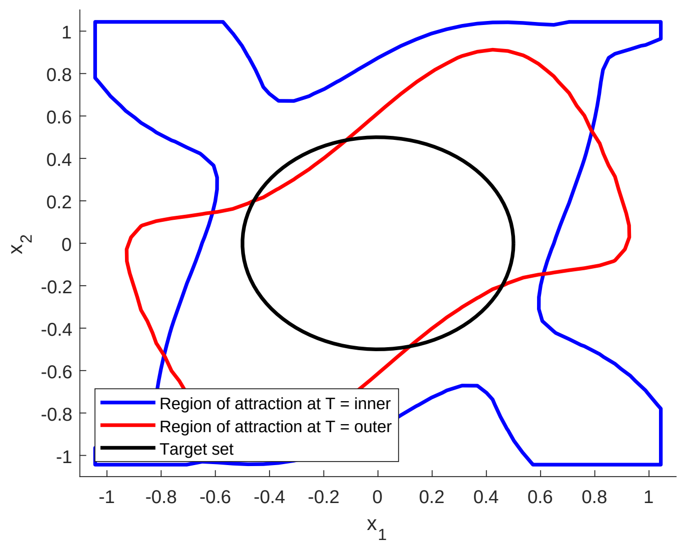


Figure 10: Plots of the inner (blue) and outer (red) ROA for the Van Der Pol oscillator for degrees 10

For example on the Van der Pol oscillator, for degree 10 and below (see Figure 10), while the outer approximation is still valid, the inner one does not work anymore and shows irrelevant artefacts.

4.5.2 Inner approximation quality

The inner approximation of the ROA has more constraints than the outer approximation, which makes it longer to proceed for the same polynomial degree. Moreover, on complex problems, the inner calculation is often non conclusive. That issue could be due to the fact that, when the target set is too small – relatively – the constraints of positivity of v on the complementary of the target forces v to be positive almost everywhere.

That issue could be resolved by making the negativity of v on the target set an additional constraint. In practice, it indeed improves slightly the precision of the result, however it also significantly increases the computational time, which makes the trade-off not worth it.

4.5.3 Influence of the time horizon and the variable range

The relative size of the target and feasible sets have a considerable impact on the results, especially on the inner approximation.

For example, on the Van der Pol oscillator, decreasing the size of the target set considerably impacts the inner calculations.

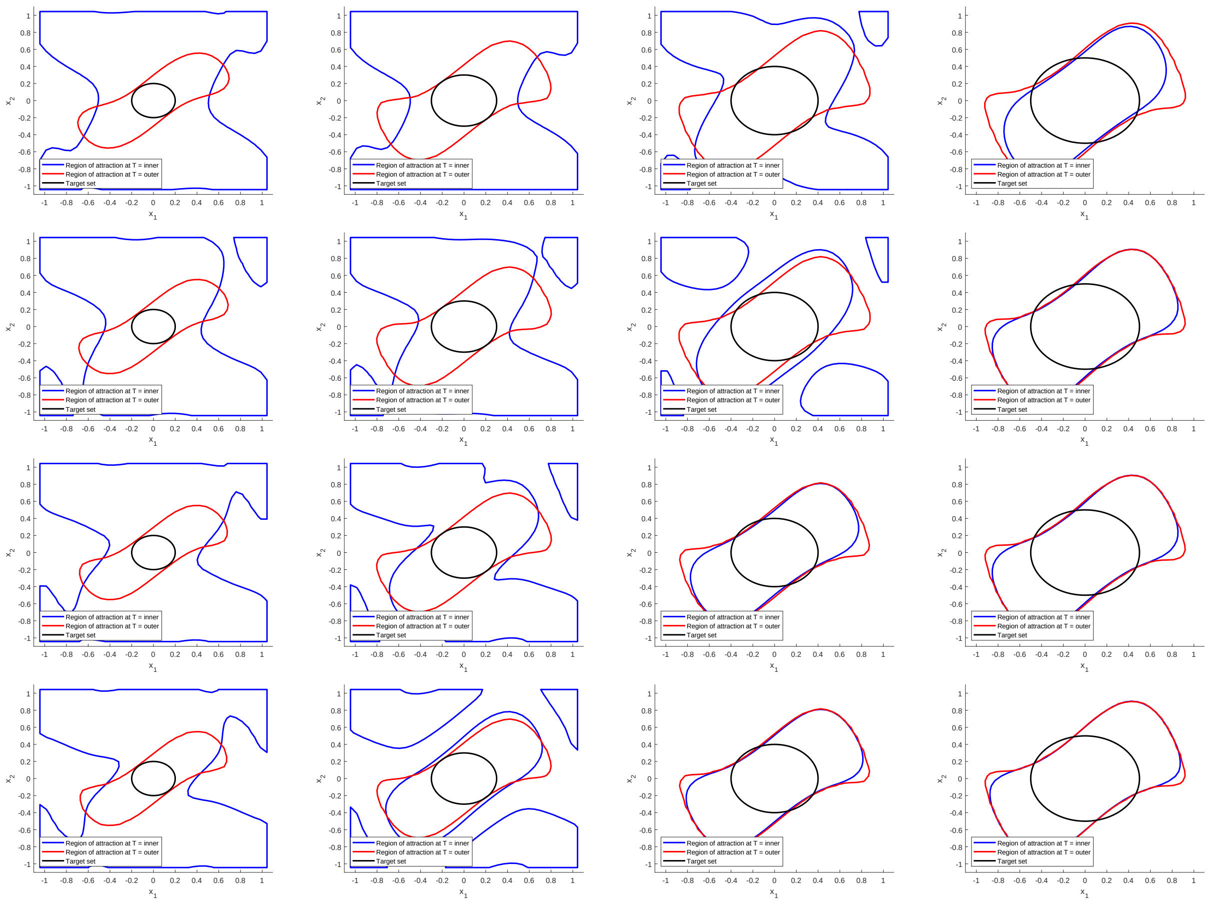


Figure 11: Plots of the inner (blue) and outer (red) ROA for the Van Der Pol oscillator for degrees in $\{12, 14, 16, 18\}$ (from top to bottom) and $\varepsilon \in \{0.2, 0.3, 0.4, 0.5\}$ (left to right): target set is in dark

While the outer approximations looks good for all the plots, the inner approximation is correct only for the higher polynomial degrees and the bigger target set. The radius ε of the target set must be at least 0.4 for the inner calculation to be relevant. Below, the problem is over-constrained for the given degrees and the polynomial w is very close to a constant 1. In fact, we have that

$$\forall \mathbf{x}, |w(\mathbf{x}) - 1| \lesssim 10^{-4}$$

for all the couples (δ, ε) where the plotting is unsatisfying, which gives us a practical criterion to determine, whether the results of the inner approximation are relevant or not.

Finally, for this particular problem, the degree must be at least 14 and the size of the ROA at least 10% of the feasible set[5] in order to have a relevant inner approximation. Such high degree of polynomials are not reachable with dimension 6 problems on a classical computer – degree 10 is already too high for a RAM memory of 16 GB – while for example the studied target set for the power converter was covering around 10^{-8} % of the feasible set. This certainly explains why the results for inner approximations on both the power converter and the triple synchronous machines were irrelevant.

[5] $0.4^2\pi/2.2^2 = 0.10$

5 Conclusion

Using the definition of the volume using measures enable to give the description the region of attraction of a dynamical system with an optimization problem. This problem can be strengthened and solved using the Lasserre hierarchy.

For the droop controlled power converter, the application of this theory provided satisfying results on the outer approximation of the region of attraction. On the other hand, the inner approximation gave no results at all.

The generalization of the code in order to create a toolbox for solving more easily the computation of the approximations of the region of attraction was successful. The SOStab toolbox is now in a working state for the calculation of outer approximations. On the other hand, the inner approximation is possible only for specific target and feasible sets and requires higher degree polynomials, so longer computations and bigger computer power.

References

[ApS, 2022] ApS, M. (2022). *The MOSEK optimization toolbox for MATLAB manual. Version 10.0.*

[Henrion and Korda, 2013] Henrion, D. and Korda, M. (2013). Convex computation of the region of attraction of polynomial control systems. *IEEE Transactions on Automatic Control*, 59(2):297–312.

[Josz et al., 2019] Josz, C., Molzahn, D. K., Tacchi, M., and Sojoudi, S. (2019). Transient stability analysis of power systems via occupation measures. In *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE.

[Korda et al., 2013] Korda, M., Henrion, D., and Jones, C. N. (2013). Inner approximations of the region of attraction for polynomial dynamical systems. volume 46, pages 534–539. Elsevier.

[Löfberg, 2009] Löfberg, J. (2009). Pre- and post-processing sum-of-squares programs in practice. *IEEE Transactions on Automatic Control*, 54(5):1007–1011.

[Putinar, 1993] Putinar, M. (1993). Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984.