Sparse moment-sum-of-squares relaxations of nonlinear dynamical systems with guaranteed convergence

Corbinian Schlosser[1], Milan Korda[1,2]
1: LAAS-CNRS Toulouse,
2: Faculty of Electrical Engineering, Czech Technical University in Prague

Brainstorming polynomial optimization day
September 18, 2020

Definitions

Given a dynamical system $\dot{x} = f(x)$ on $X \subset \mathbb{R}^n$ compact we denote the solution/flow map by $\varphi_t(\cdot)$.

Definitions

Given a dynamical system $\dot{x} = f(x)$ on $X \subset \mathbb{R}^n$ compact we denote the solution/flow map by $\varphi_t(\cdot)$.

Definition (Maximum positively invariant (MPI) set)

Set of initial conditions $x_0 \in X$, such that the solutions $\varphi_t(x_0)$ stay in X for all $t \in \mathbb{R}_+$, is called the MPI set.

A LP for the MPI set

In Korda et al. (2014) the following linear program for computing the MPI set is proposed

$$
\begin{aligned}
d^* := \quad & \inf \int_X w(x)\, d\lambda(x) \\
\text{s.t.} \quad & v \in \mathcal{C}^1(\mathbb{R}^n), w \in \mathcal{C}(X) \\
& \beta v(x) - \nabla v \cdot f(x) \geq 0 \quad \text{for } x \in X \\
& w \geq 0 \\
& w - v - 1 \geq 0
\end{aligned}
$$

A LP for the MPI set

In Korda et al. (2014) the following linear program for computing the MPI set is proposed

$$
\begin{aligned}
d^* := \quad & \inf \int_X w(x)\, d\lambda(x) \\
\text{s.t.} \quad & v \in \mathcal{C}^1(\mathbb{R}^n), w \in \mathcal{C}(X) \\
& \beta v(x) - \nabla v \cdot f(x) \geq 0 \quad \text{for } x \in X \\
& w \geq 0 \\
& w - v - 1 \geq 0
\end{aligned}
$$

Theorem

It holds $d^* = \lambda(MPI)$ and $w^{-1}([1, \infty)) \supset MPI$ for all feasible (v, w).

The corresponding hierarhy of SDPs

The corresponding semidefinite programs (SDPs) have the following form

$$\begin{aligned} d_k^* := \quad & \inf \int_X w(x)\, d\lambda(x) \\ \text{s.t.} \quad & v, w \in \mathbb{R}[x]_k \\ & \beta v - \nabla v \cdot f = \mathrm{SoS}_k \\ & w = \mathrm{SoS}_k \\ & w - v - 1 = \mathrm{SoS}_k \end{aligned}$$

The corresponding hierarhy of SDPs

The corresponding semidefinite programs (SDPs) have the following form

$$\begin{aligned} d_k^* := \quad & \inf \int_X w(x)\, d\lambda(x) \\ \text{s.t.} \quad & v, w \in \mathbb{R}[x]_k \\ & \beta v - \nabla v \cdot f = \mathrm{SoS}_k \\ & w = \mathrm{SoS}_k \\ & w - v - 1 = \mathrm{SoS}_k \end{aligned}$$

Theorem

It holds $d_k^* \to d^* = \lambda(MPI)$ and $w_k^{-1}([1, \infty)) \supset MPI$ for all feasible (v_k, w_k).

The corresponding hierarhy of SDPs

The corresponding semidefinite programs (SDPs) have the following form

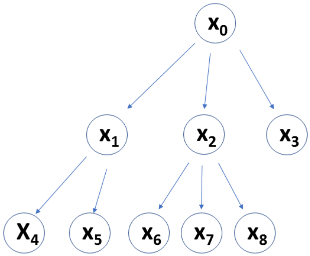
$$\begin{aligned} d_k^* := \quad & \inf \int_X w(x)\, d\lambda(x) \\ \text{s.t.} \quad & v, w \in \mathbb{R}[x]_k \\ & \beta v - \nabla v \cdot \mathbf{f} = \mathrm{SoS}_k \\ & w = \mathrm{SoS}_k \\ & w - v - 1 = \mathrm{SoS}_k \end{aligned}$$

Theorem

It holds $d_k^* \to d^* = \lambda(MPI)$ and $w_k^{-1}([1,\infty)) \supset MPI$ for all feasible (v_k, w_k).

Sparse structures we are considering

We consider functions f whose action can be represented by

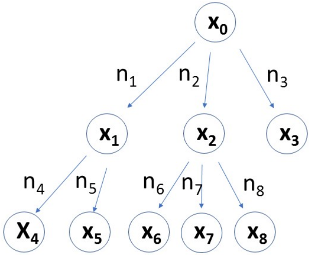


Complexity based on the graph structure of f

We measure complexity by the number of variables in a SDP (for a fixed degree $k \in \mathbb{N}$).

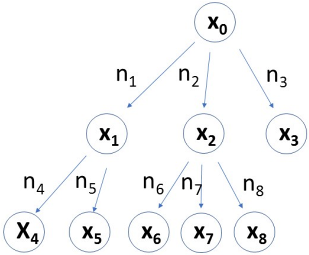
Complexity based on the graph structure of f

We measure complexity by the number of variables in a SDP (for a fixed degree $k \in \mathbb{N}$).
Hence the graph of the dynamics has to include that information



Complexity based on the graph structure of f

We measure complexity by the number of variables in a SDP (for a fixed degree $k \in \mathbb{N}$).
Hence the graph of the dynamics has to include that information



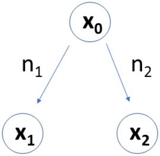
.

Theorem

There exists a convergent hierarchy of sparse SDPs with the size of the largest SoS multiplier containing N variables, where N is the longest weighted path in the graph.

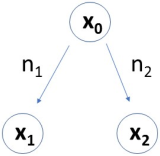
Basic sparse structure

The previous structure results by induction on the number of "cherries" and the "cherry structure" from Chen et al. (2018)



Basic sparse structure

The previous structure results by induction on the number of "cherries" and the "cherry structure" from Chen et al. (2018)



$$
\begin{aligned}
\dot{x}_1 &= f_1(x_1, x_0) \quad \text{on } \mathbb{R}^{n_1} \\
\dot{x}_2 &= f_2(x_2, x_0) \quad \text{on } \mathbb{R}^{n_2} \\
\dot{x}_0 &= f_0(x_0) \quad \text{on } \mathbb{R}^{n_0}
\end{aligned}
$$

with induced subsystems

$$
\dot{(x_1, x_0)} = (f_1, f_0)(x_1, x_0)\ ,\ \dot{(x_2, x_0)} = (f_2, f_0)(x_2, x_0)\ ,\ \dot{x}_0 = f(x_0)
$$

Main principles

Main principles

1. Sparse description of the sets of interest

Main principles

1. Sparse description of the sets of interest
2. Finding a corresponding (sparse) formulation as solution of a linear optimization problem

Main principles

1. Sparse description of the sets of interest
2. Finding a corresponding (sparse) formulation as solution of a linear optimization problem
3. Exploit sparsity computationally

Sparse description of the MPI set

Assumption: $X = X_1 \times X_2 \times X_0$

Sparse description of the MPI set

Assumption: $X = X_1 \times X_2 \times X_0$

Lemma (Gluing along the x_c component)

Let $X = X_1 \times X_2 \times X_0$ and the dynamical system be sparse (in the sense as above) and let $M_+^{(1)}$ and $M_+^{(2)}$ denote the MPI sets for the subsystems on (x_1, x_0) and (x_2, x_0) then the MPI set M_+ of the whole system is given by

$$\{(x_1, x_2, x_0) \in X : (x_1, x_0) \in M_+^{(1,0)}, (x_2, x_0) \in M_+^{(2,0)}\}.$$

Interpretation to remember

Sparsity allows to compute the objects separately for the subsystems and then gluing them together along the coupling component x_0.

Interpretation to remember

Sparsity allows to compute the objects separately for the subsystems and then gluing them together along the coupling component x_0.

Corollary

There exists a sparse LP and corresponding sparse SDPs, i.e. the constraints almost separate into functions only on the variables (x_1, x_0) *and* (x_2, x_0)*. More precisely for the SDP; the SoS multiplier are functions only in the variables* (x_1, x_0) *and* (x_2, x_0) *respectively.*

Interpretation to remember

Sparsity allows to compute the objects separately for the subsystems and then gluing them together along the coupling component x_0.

Corollary

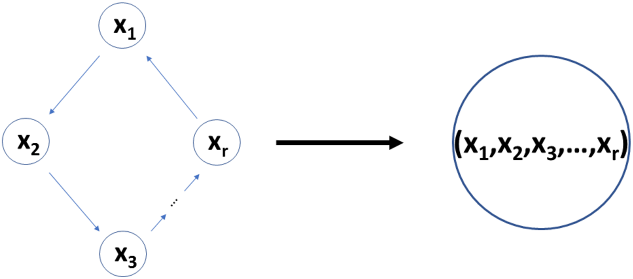
There exists a sparse LP and corresponding sparse SDPs, i.e. the constraints almost separate into functions only on the variables (x_1, x_0) *and* (x_2, x_0)*. More precisely for the SDP; the SoS multiplier are functions only in the variables* (x_1, x_0) *and* (x_2, x_0) *respectively.*

Remark

Now induction on the number of cherries and the cherry structure proves the main theorem.

More general graphs

Circles are treated as one node because no reduction by our method is possible

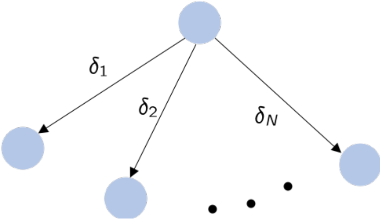


A hard example

The graph of the dynamics considered in Tacchi et al. (2019) is given by

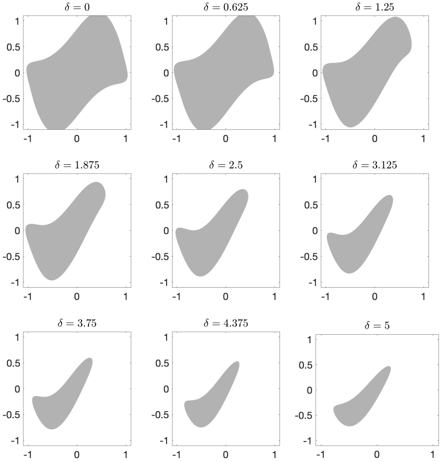
Numerical example

For a Van der Pol oscillator with cherry structure of the form

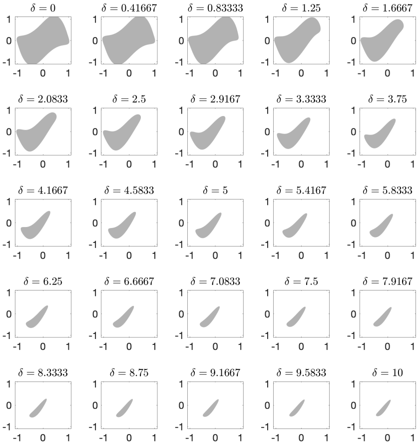


Numerical example

We get



for degree $k = 8$ and $N = 9$ and total dimension 20 and



for degree $k = 8$ and $N = 25$ and total dimension 52.

Conclusion

Contribution of this work

1. Can be applied in a similar way for the region of attraction and global attractors
2. First sparse method to approximate the MPI set, global attractors and the region of attraction with guaranteed convergence

Conclusion

Contribution of this work

1. Can be applied in a similar way for the region of attraction and global attractors
2. First sparse method to approximate the MPI set, global attractors and the region of attraction with guaranteed convergence

Outlook and perspectives

1. Extending to sparse control systems
2. Improve/increase exploiting sparse structures
3. Exploit polynomial structure of f and not only focusing sparse coupling