SAPO

REACHABILITY COMPUTATION AND PARAMETER SYNTHESIS OF POLYNOMIAL DYNAMICAL SYSTEMS

Tommaso Dreossi
UC Berkeley
April 18, 2017
SAPO

A tool for:

- **Formal analysis** of dynamical systems
- Reachability computation
- Parameter synthesis

Dynamical systems:

\[ x_{k+1} = f(x_k, p) \]
**SAPO**

A tool for:

- **Formal analysis** of dynamical systems
- Reachability computation
- Parameter synthesis

Dynamical systems:

\[ x_{k+1} = f(x_k, p) \]

- **Reachability**: Compute reachable states from a set of initial conditions
SAPO

A tool for:

- **Formal analysis** of dynamical systems
- Reachability computation
- Parameter synthesis

Dynamical systems:

\[ x_{k+1} = f(x_k, p) \]

- **Reachability**: Compute reachable states from a set of initial conditions
- **Parameter Synthesis**: Find parameters s.t. a property is satisfied
Overview

Reachability computation:

- Linear systems - hundreds of variables \([\text{FLGD}^{+}11, \text{KV}00]\)
- Nonlinear systems - low dimensions \((\approx 10)\) \([\text{CÁS}13, \text{KGCC}15]\)

Parameter synthesis:

- Analytic (scalability issues)
- Simulation-based (not formal/exhaustive) \([\text{Don}10, \text{MMB}03]\)
OVERVIEW

Reachability computation:
- Linear systems - hundreds of variables \([\text{FLGD}^{+}11, \text{KV00}]\)
- Nonlinear systems - low dimensions \((\approx 10)\) \([\text{CÁS}13, \text{KGCC}15]\)

Parameter synthesis:
- Analytic (scalability issues)
- Simulation-based (not formal/exhaustive) \([\text{Don}10, \text{MMB}03]\)

Sapo:
- Polynomial dynamical systems (nonlinear)
- Infinite compact sets (for both states and parameters)
**Overview**

Reachability computation:

- Linear systems - hundreds of variables [FLGD\textsuperscript{11}, KV00]
- Nonlinear systems - low dimensions (\(\approx 10\)) [CÁS13, KGCC15]

Parameter synthesis:

- Analytic (scalability issues)
- Simulation-based (not formal/exhaustive)[Don10, MMB03]

**Sapo:**

- **Polynomial** dynamical systems (nonlinear)
- **Infinite** compact sets (for both states and parameters)
- Boxes, parallelotopes, and parallelotope bundles
- Synthesize parameter sets using STL
- **Bernstein coefficients** (Efficient symbolic computation)
Overview

Roadmap

1 Reachability Analysis
   1 Bernstein coefficients for polynomials
   2 Box, parallelotope, parallelotope bundle

2 Parameter Synthesis
   1 Problem formalization via STL
   2 Synthesis algorithm

3 Conclusion
   1 Tool overview
1 Reachability Analysis
   1 Bernstein coefficients for polynomials
   2 Box, parallelootope, parallelootope bundle

2 Parameter Synthesis
   1 Problem formalization via STL
   2 Synthesis algorithm

3 Conclusion
   1 Tool overview
**Problem**

Given:
- dynamical system $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$
- set $X_0 \subset \mathbb{R}^n$

compute the reachable sets up to time $T \in \mathbb{N}$
**Reachability**

**Problem**

Given:
- dynamical system $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$
- set $X_0 \subset \mathbb{R}^n$

compute the reachable sets up to time $T \in \mathbb{N}$

- Nonlinear set transformations (nonconvexity)
## Reachability

### Problem

Given:
- dynamical system \( f : \mathbb{R}^n \rightarrow \mathbb{R}^n \)
- set \( X_0 \subset \mathbb{R}^n \)

compute the reachable sets up to time \( T \in \mathbb{N} \)

- **Nonlinear set transformations** (nonconvexity)
- **Idea**: Over-approximate sets with simpler objects (polytopes)
Reachability

$X_0 \equiv Dx \leq c \ (D, c: \text{template and offset})$

$c'_j \geq \max_{x \in X_i} D_j f(x)$
Reachability

\[ X_0 \equiv D x \leq c \ (D, c: \text{template and offset}) \]

\[ c'_j \geq \max_{x \in X_i} D_j f(x) \]

Nonlinear optimization problem
Bernstein Polynomials

Power basis

\[ \pi(x) = \sum_{i \leq d} a_i x^i \]

Bernstein basis

\[ \pi(x) = \sum_{i \leq d} b_i B_{d,i}(x) \]

\[ b_i = \sum \binom{i}{j} \binom{d}{j} a_j \]

Range enclosure property

For all \( x \in [0, 1] \):

\[ \min b_i \leq \pi(x) \leq \max b_i \]
Bernstein Polynomials

Power basis
\[ \pi(x) = \sum_{i \leq d} a_i x^i \]

Bernstein basis
\[ \pi(x) = \sum_{i \leq d} b_i B_{d,i}(x) \]
\[ b_i = \sum \binom{i}{j} \frac{d!}{j!(d-j)!} a_j \]

Range enclosure property
For all \( x \in [0, 1]^n \) : \( \min b_i \leq \pi(x) \leq \max b_i \)
**Bernstein Polynomials**

**Power basis**

\[ \pi(x) = \sum_{i \leq d} a_i x^i \]

**Bernstein basis**

\[ \pi(x) = \sum_{i \leq d} b_i B_{d,i}(x) \]

\[ b_i = \sum \binom{i}{j} \frac{\binom{d}{j}}{\binom{d}{d}} a_j \]

**Range enclosure property**

For all \( x \in [0, 1]^n \): \( \min b_i \leq \pi(x) \leq \max b_i \)

How to generalize to other domains?
Box

\[ \mathbb{V}([0,1]^n) \rightarrow f(V([0,1]^n)) \]

\[ [0,1]^n \rightarrow X_0 \]

\[ D_1 x \leq c_1 \]
\[ D_2 x \leq c_2 \]
\[ D_3 x \leq c_3 \]
\[ D_4 x \leq c_4 \]

\[ f(X_0) \rightarrow X_1 \]

\[ D_j x \leq c_j' \]
\[ c_j' \geq \max_{x \in [0,1]^n} D_j f(v(x)) \]

\[ c_j' \leftarrow \text{maximum Bernstein coefficient of } D_j f(v(x)) \]
Parallelootope

- More generic set
- More flexibility
- Precision improvement
Parallelotope

- More generic set
- More flexibility
- Precision improvement

Combining different sets? (Boxes + Parallelotopes)
**Bundle**

Polytopes as intersection of parallelotopes

**Definition**

A *bundle* $B = \{P_1, \ldots, P_b\}$ is a finite set of parallelotopes s.t. $Q = \bigcap_{i=1}^{n} P_b$.
**Bundle**

Polytopes as intersection of parallelotopes

**Definition**

A **bundle** $B = \{P_1, \ldots, P_b\}$ is a finite set of parallelotopes s.t. $Q = \bigcap_{i=1}^{n} P_b$

**Theorem**

*For any polytope $Q$ there exists a bundle $B = \{P_1, \ldots, P_b\}$ s.t.*

$$Q = \bigcap_{i=1}^{n} P_b$$
**Bundle**

![Diagram showing P₁, Q, P₂, f(P₁), f(Q), and f(P₂) with a function f mapping from P₁ and P₂ to their respective images under f.](image)
**Bundle**

![Diagram showing the bundle concept with sets and functions](image)

- **$P_1$** and **$P_2$** are sets with a function $f$ mapping them to $f(P_1)$ and $f(P_2)$.
- The image also shows the parameter synthesis process with a dotted line $D_j$ and a modified line $c'_j$.
Case Studies

Reachability

SIR epidemic model (3d)

(a) Box-based
(b) Bundle-based

Quadcopter (17d)

(a) Height (h)  (b) Vertical speed (w)  (c) Controller height (h_I)

<table>
<thead>
<tr>
<th>Model</th>
<th>Vars</th>
<th>Steps</th>
<th>Dirs/Temps</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Van der Pol</td>
<td>2</td>
<td>300</td>
<td>4/6</td>
<td>1.45</td>
</tr>
<tr>
<td>Rössler</td>
<td>3</td>
<td>250</td>
<td>5/3</td>
<td>0.42</td>
</tr>
<tr>
<td>SIR</td>
<td>3</td>
<td>300</td>
<td>5/3</td>
<td>0.79</td>
</tr>
<tr>
<td>Lotka-Volterra</td>
<td>5</td>
<td>500</td>
<td>7/3</td>
<td>51.52</td>
</tr>
<tr>
<td>Phosphorelay</td>
<td>7</td>
<td>200</td>
<td>10/3</td>
<td>8.13</td>
</tr>
<tr>
<td>Quadcopter</td>
<td>17</td>
<td>300</td>
<td>18/2</td>
<td>7.65</td>
</tr>
</tbody>
</table>
Roadmap

1. Reachability Analysis
   1. Bernstein coefficients for polynomials
   2. Box, parallelotope, parallelotope bundle

2. Parameter Synthesis
   1. Problem formalization via STL
   2. Synthesis algorithm

3. Conclusion
   1. Tool overview
Parameter Synthesis

\[ g_0(x) \geq 0 \]
\[ g_1(x) \geq 0 \]
\[ g_2(x) \geq 0 \]

1) How to express time-dependent properties over traces?
2) How to synthesize \( P_s \)?
**Parameter Synthesis**

\[ g_0(x) \geq 0 \]

\[ g_1(x) \geq 0 \]

\[ g_2(x) \geq 0 \]

1) How to express time-dependent properties over traces?

2) How to synthesize \( P_s \)?
Parameter Synthesis

Definition (STL)

\[ \varphi := \top \mid f(x_1[t], \ldots, x_n[t]) > 0 \mid \varphi \land \psi \mid \varphi \lor \psi \mid \varphi \ U_{[a,b]} \psi \]

Problem

Given:

- dynamical system \( f(x, p) \)
- set \( X_0 \), parameter set \( P \), STL specification \( \varphi \)

find \( P_s \subseteq P \) such that \( f(X_0, P_s) \models \varphi \)
Synthesis Algorithm

Atomic Predicate

Rewrite safe condition:

\[ X_{i+1} = f(X_i, P_s) \]

Valid if \( g(X_{i+1}) < 0 \)

\( g(f(X_i, P_s)) < 0 \)
Synthesis Algorithm

\( \land, \lor, \cup \)

\[ \varphi_1 \land \varphi_2(\lor) \]

1. Solve the problem for \( \varphi_1 \rightarrow P_1 \)
2. Solve the problem for \( \varphi_2 \rightarrow P_2 \)
3. Return the \( \rightarrow P_1 \cap P_2(\cup) \)

\[ \varphi_1 \cup_{[a,b]} \varphi_2 \text{ (cases on } [a, b]) \]

1. \( \varphi_1 \cup_{[0,0]} \varphi_2 \)
2. \( \varphi_1 \cup_{[0,b]} \varphi_2 \text{ (} b > 0 \text{)} \)
3. \( \varphi_1 \cup_{[a,b]} \varphi_2 \text{ (} a, b > 0 \text{)} \)
## Case Studies

### Parameter Synthesis

**SIR epidemic model (3d, 2 param)** \( G_{[50,100]}(i \leq 0.44) \)

(a) Synthesized parameters  
(b) Constrained evolution

<table>
<thead>
<tr>
<th>Model</th>
<th>Vars</th>
<th>Params</th>
<th>Spec</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIR</td>
<td>3</td>
<td>2</td>
<td>( G_{[50,100]}(i \leq 0.44) )</td>
<td>0.42</td>
</tr>
<tr>
<td>Influenza</td>
<td>4</td>
<td>2</td>
<td>( G_{[0,50]}(i \leq 0.43) )</td>
<td>2.12</td>
</tr>
<tr>
<td>Ebola</td>
<td>5</td>
<td>4</td>
<td>((q &gt; 0.04)U_{[10,15]}(i &gt; 0.27))</td>
<td>0.0007</td>
</tr>
</tbody>
</table>
Roadmap

1. Reachability Analysis
   1. Bernstein coefficients for polynomials
   2. Box, parallelotope, parallelotope bundle

2. Parameter Synthesis
   1. Problem formalization via STL
   2. Synthesis algorithm

3. Conclusion
   1. Tool overview
SAPO

OVERVIEW

Sapo Core

- Dynamical System
- Base converter
- Paralleloptope
- STL

1) Improved matrix method
2) Symbolic coefficients

Bundle

Linear System
**Conclusion**

- **Sapo**: C++ tool for reachability analysis and parameter synthesis of polynomial dynamical systems
- Source + VM: [https://github.com/dreossi/sapo](https://github.com/dreossi/sapo)

**Applications:**

- System biology: SIR, SARS, Influenza, Ebola (3-7d)
- Population growth: Honeybees nest choice (5d)
- Robotics: Quadcopter drone (17d)
- Approved by RE Committee

**Future works:**

- Parallelization (bundles easy to parallelize)
- From parameter to input synthesis (controller)
- Hybrid automata verification
References I


Bernstein Expansion

**Basis**

**Power basis**

\[ \pi(x, p) = \sum_{i \leq d} a_i(p)x^i \]

**Bernstein basis**

\[ \pi(x, p) = \sum_{i \in I} b_i(p)B_{d, i}(x) \]

\[ B_{d, i}(x) = B_{d_1, i_1}(x_1) \ldots B_{d_n, i_n}(x_n) \]

\[ B_{d, i}(x) = \binom{d}{i} x^i (1 - x)^{d - i} \]
**Lemma ([DT12])**

Let $C_\pi : \mathbb{R}^n \to \mathbb{R}$ be the piecewise linear function defined by the Bernstein control points of the polynomial $\pi : \mathbb{R}^n \to \mathbb{R}$, with respect to the box $[0, 1]^n$. For all $x \in [0, 1]^n$

$$| \pi(x) - C_\pi(x) | \leq \max_{x \in [0,1]^n; i,j \in \{1,\ldots,n\}} | \partial_i \partial_j \pi(x) |$$

(1)

where $| \cdot |$ is the infinity norm on $\mathbb{R}^n$. 