Verifiable Data Driven Control

M. Sznaier

Robust Systems Lab
Electrical and Computer Engineering
Northeastern University

Verifiable, Control-Oriented Learning On The Fly MURI Kickoff Meeting
Motivation: survive unexpected changes in dynamics.

Formalization: control of unknown dynamics based on data collected from a single execution and side information.

Challenges
- Lack of training data (single execution)
- Learning while acting
- Taking provably correct actions.
Motivation: survive unexpected changes in dynamics.

Formalization: control of unknown dynamics based on data collected from a single execution and side information.

Challenges
- Lack of training data (single execution)
- Learning while acting
- Taking provably correct actions.

Humans can do it!
Big Picture: Control Oriented Learning on the Fly

- Three Stages:
- Stayin’ Alive
Big Picture: Control Oriented Learning on the Fly

- **Three Phases**

- **Stayin’ Alive**
  - Maintain viability by avoiding a known bad set
  - Myopic control (Topcu & Ornik)
  - Alternative: Rantzer’s dual control?
  - Use this phase to learn about the plant

![Diagram showing a bad set and a system's trajectory](image)
Three Phases

Stayin’ Alive
- Maintain viability by avoiding a known bad set
- Myopic control (Topcu & Ornik)
- Alternative: Rantzer’s dual control?
- Use this phase to learn about the plant

Verifiable Data Driven Control
- Find a good robust control invariant set and associated control action (Sznaier & Dai)
Big Picture: Control Oriented Learning on the Fly

- **Three Phases**

- **Stayin’ Alive**
  - Maintain viability by avoiding a known bad set
  - Myopic control (Topcu & Ornik)
  - Alternative: Rantzer’s dual control?
  - Use this phase to learn about the plant

- **Verifiable Data Driven Control**
  - Find a good robust control invariant set and associated control action (Sznaier & Dai)

- **Learning for Performance**
  - Keep learning while DD keeps you safe
  - Use these models to optimize performance
Data Driven Control
Design a switched stabilizing controller $C(P_i)$ based solely on experimental data.
From data to control (LTI)

- Find the consistency set $T(y)$:
  - All plants that could have generated the data
From data to control (LTI)

- **Find the consistency set** $T(y)$:
  - All plants that could have generated the data

- **Pick a plant in** $T(y)$
  - Ideally the Chebyshev center, but any will do
From data to control (LTI)

- Find the consistency set $T(y)$:
  - All plants that could have generated the data

- Pick a plant in $T(y)$
  - Ideally the Chebyshev center, but any will do

- Bound the worst case error and design a robust controller
From data to control (LTI)

- **Find the consistency set** $T(y)$:
  - All plants that could have generated the data

- **Pick a plant in** $T(y)$
  - Ideally the Chebyshev center, but any will do

- **Bound the worst case error and design a robust controller**

- **Works, but**
  - Conservative (!)
  - Computational issues
  - Hard to extend beyond LTI
Switched Data Driven Control:

- **Given:**
  - Prior: the plant admits a model of the form
    \[
    x_{k+1}^i = A_i x_k^i + B_i u_k + w_k, \quad \|w_k\|_\infty \leq \epsilon, \quad i \in \{1, \ldots, s\}
    \]
  - Experimental data (labeled):
    \[
    \left\{ u_k, x_k^i, x_{k+1}^i \right\}_{k=0}^T
    \]

- Find a switched controller so that \( A_{\sigma_t} + B_{\sigma_t} F_{\sigma_t} \) is G.A.S for all switching sequences and for all plants that could have generated the experimental data
Switched Data Driven Control:

- **Given:**
  - Prior: the plant admits a model of the form
    \[ x_{k+1}^i = A_i x_k^i + B_i u_k + w_k, \quad \|w_k\|_\infty \leq \epsilon, \ i \in \{1, \ldots, s\} \]
  - Experimental data (labeled):
    \( \{u_k, x_k^i, x_{k+1}^i\}_{k=0}^T \)

- **Consistency set is a collection of polytopes:**
  \[ \mathcal{P}_i = \{A_i, B_i : \|x_{t_k+1}^i - A_i x_{t_k}^i - B_i u_{t_k}^i\|_\infty \leq \epsilon \text{ for all } t_k^i \in \mathcal{K}_i\} \]
Switched Data Driven Control:

- Find a switched controller $F_i$ that stabilizes the set $\{P_i\}$ for any arbitrary switching

All $(A_i, B_i)$ compatible with the available information live here
Detour: control of switched systems:

\[ x_{k+1} = A_i x_k + B_i u_k \]

can be stabilized by a switched linear controller

There exist

a) A full row rank matrix \( T \)

b) Matrices \( U_i, P_i \) such that

\[ A_i T + B_i U_i = TP_i, \]
\[ \|P\|_{\ell_1 \to \ell_1} < 1 \]

\[ Blanchini \ et. \ al., \ 2009 \]
Switched Data Driven Control as a Robust Optimization.

- Find matrices $T, U_i, P_i$ such that
  \[
  A_i T + B_i U_i = T P_i, \\
  \|P\|_{\ell_1 \to \ell_1} < 1
  \]

  for all matrices in the polytopes $\{P_i\}$
Switched Data Driven Control as a Robust Optimization.

- Find matrices $T, U_i, P_i$ such that
  \[
  A_i T + B_i U_i = TP_i,
  \]
  \[
  \|P\|_{\ell_1 \rightarrow \ell_1} < 1
  \]

  for all matrices in the polytopes $\{P_i\}$

- A very challenging non-convex min-max problem

- Main result: solvable via Farkas’ Lemma + SoS
Switched Data Driven Control as a Robust Optimization.

- Find matrices $V, U_i, P_i$ such that

$$A_i V + B_i U_i = V H_i, \quad \|H\|_{\ell_1 \to \ell_1} \leq d < 1$$

for all matrices in the polytopes $\{P_i\}$

- A non-convex min-max problem, but solvable using SoS (via Farkas’ Lemma)
Extended Farkas Lemma

Lemma: Consider two polytopes of the form $P_N = \{x : Nx \leq n\}$ and $P_M = \{x : Mx \leq m\}$, where $M, N$ are matrices and $m, n$ are vectors. Then $P_N \subseteq P_M$ if and only if there exists a matrix $Y$ with non-negative entries such that

\[
Y N = M \\
Y n \leq m
\]
Data Driven Control:

- **Fact:** There exist a controller $F_i$ that stabilizes iff there exist $Y^i \geq 0, F_i, V$ such that:

  $Y^i M(x^i, u^i) = [(S(V^{-T} \otimes V) \ S(F_i^T \otimes V)]$

  $Y^i \begin{bmatrix} x^i + \epsilon 1 \\ -x^i + \epsilon 1 \end{bmatrix} \leq d^i < 1$

  $\forall S \doteq \begin{bmatrix} s_1^T & 0_{1 \times n} & \ldots & 0 \\ 0_{1 \times n} & s_2^T & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1 \times n} & 0_{1 \times n} & \ldots & s_n^T \end{bmatrix}, \ s_j = \pm 1$

*Dai and S., CDC’18*
Data Driven Control:

- Q: how do we solve these:

\[
Y^i M(x^i, u^i) = \left[ (S(V^{-T} \otimes V) \quad S(F_i^T \otimes V) \right]
\]

\[
Y^i \begin{bmatrix} x^i + \epsilon 1 \\ -x^i + \epsilon 1 \end{bmatrix} \leq d^i < 1
\]

\[
\forall S \doteq \begin{bmatrix} s_1^T & 0_{1 \times n} & \ldots & 0 \\ 0_{1 \times n} & s_2^T & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1 \times n} & 0_{1 \times n} & \ldots & s_n^T \end{bmatrix}, \quad s_j = \pm 1
\]
Q: how do we solve these:

\[ Y^i M(x^i, u^i) = [(S(V^{-T} \otimes V) \ S(F_i^T \otimes V)] \]

\[ Y^i \begin{bmatrix} x^i + \epsilon 1 \\ -x^i + \epsilon 1 \end{bmatrix} \leq d^i < 1 \]

\[ \forall S = \begin{bmatrix} s_1^T & 0_{1 \times n} & \ldots & 0 \\ 0_{1 \times n} & s_2^T & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1 \times n} & 0_{1 \times n} & \ldots & s_n^T \end{bmatrix}, \ s_j = \pm 1 \]

A: equivalent to checking feasibility of a semi-algebraic set.

Convex relaxations via SoS/Moments (P-satz)
Data Driven Control:

- **Q:** how do we solve these:

\[
Y^i M(x^i, u^i) = \left[ (S(V^{-T} \otimes V) \quad S(F_i^T \otimes V) \right] \\
Y^i \begin{bmatrix} x^i + \epsilon 1 \\ -x^i + \epsilon 1 \end{bmatrix} \leq d^i < 1
\]

- **A:** equivalent to checking feasibility of a semi-algebraic set.

A **very large** SDP: \( \mathcal{O}\left( [n^2 + N_s(n^2 + 3n + 1)]^2 \right) \) variables (!!)
Exploiting sparsity:

- The underlying CSG has chordal sparsity
  - Implication: the “big” SD and rank constraints decouple into a collection of smaller ones
The underlying CSG has chordal sparsity
- Implication: the “big” SD and rank constraints decouple into a collection of smaller ones
Exploiting sparsity:

- The underlying CSG has chordal sparsity
  - Implication: the “big” SD and rank constraints decouple into a collection of smaller ones

For three third order systems: \(~ 220\) variables down from \(4300\)
Simple Example:

\[
\begin{bmatrix}
\dot{X} \\
\ddot{X} \\
\dot{v}
\end{bmatrix} = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & \frac{4K \sin(\theta)}{M} \\
0 & 0 & -\omega
\end{bmatrix} \begin{bmatrix}
X \\
\dot{X} \\
v
\end{bmatrix} + \begin{bmatrix}
0 \\
0 \\
\omega
\end{bmatrix} u_Z
\]

\[A_1 = \begin{bmatrix}
1 & 0.1 & 0.1913 \\
0 & 1 & 3.083 \\
0 & 0 & 0.2231
\end{bmatrix}, \quad B_1 = \begin{bmatrix}
0.1063 \\
2.87 \\
0.7769
\end{bmatrix} \quad \text{(System 1)}
\]

\[A_2 = \begin{bmatrix}
1 & 0.1 & 0.3769 \\
0 & 1 & 6.073 \\
0 & 0 & 0.2231
\end{bmatrix}, \quad B_2 = \begin{bmatrix}
0.2094 \\
5.653 \\
0.7769
\end{bmatrix} \quad \text{(System 2)}
\]

\[F_1 = \begin{bmatrix}
-0.2580 \\
-0.2123 \\
-0.8111
\end{bmatrix} \quad \text{(Gain 1)}
\]

\[F_2 = \begin{bmatrix}
-0.1527 \\
-0.1238 \\
-0.9908
\end{bmatrix} \quad \text{(Gain 2)}
\]

Quanser Qball-X4
Simple Example:

State trajectories under arbitrary switching
Future work:

- Extension to non-linear dynamics
  - Alternatives: manifold embeddings or polynomial liftings
  - Non convex, but good relaxations available

- Combine with U. Topcu’s Myopic Control
  - Use Rantzer’s dual Lyapunov formulation to maintain viability until invariance can be certified

- All of these reduce to polynomial optimization
  - Use low complexity relaxations (A. Ahmadi)
  - Use randomized linear algebra