A fresh look at a classical system identification algorithm

Necmiye Ozay
Electrical Engineering and Computer Science
University of Michigan

IPAM Workshop on
Intersections between Control, Learning and Optimization
February 24, 2020

Outline

- **Overview of some problems of interest at the intersection of learning, control, and optimization**

- **Sample complexity analysis of linear system identification (analysis of Ho-Kalman algorithm)**

Joint work with Samet Oymak, UC Riverside

- **Inverse constraint learning (a la inverse optimal control)**

Joint work with Glen Chou and Dmitry Berenson, Michigan

Outline

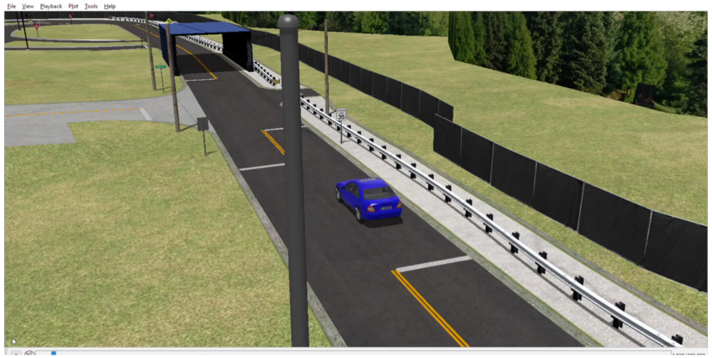
- **Overview of some problems of interest at the intersection of learning, control, and optimization**
- Sample complexity analysis of linear system identification (analysis of Ho-Kalman algorithm)

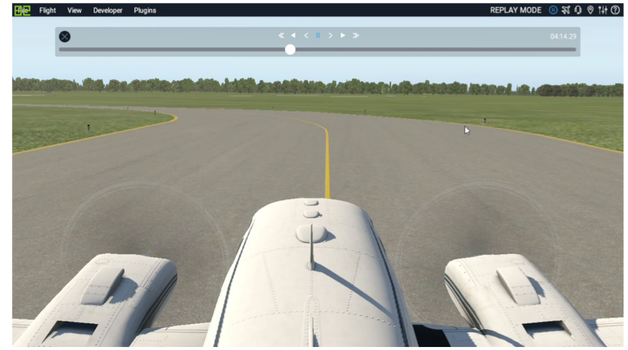
Joint work with Samet Oymak, UC Riverside

- Inverse constraint learning (a la inverse optimal control)

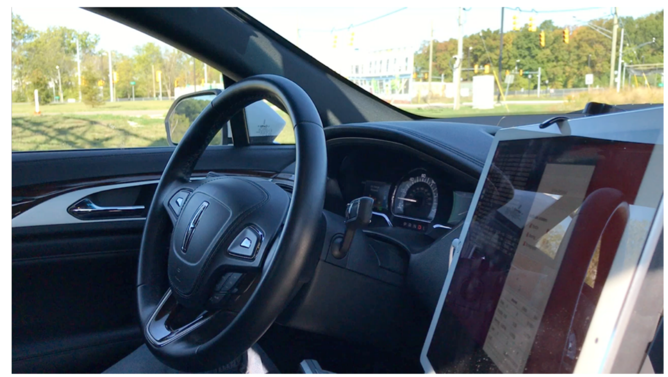

Joint work with Glen Chou and Dmitry Berenson, Michigan

Control of safety-critical autonomous systems

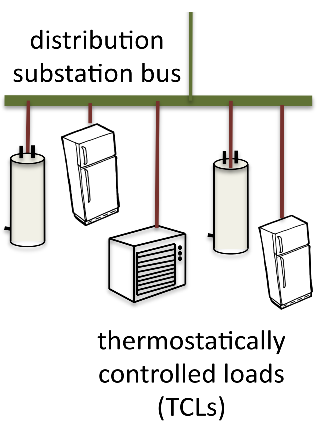


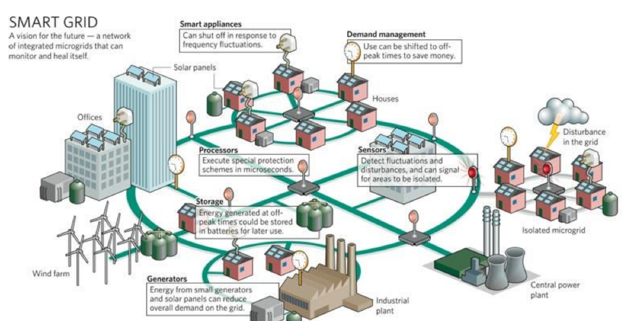


When I get adventurous

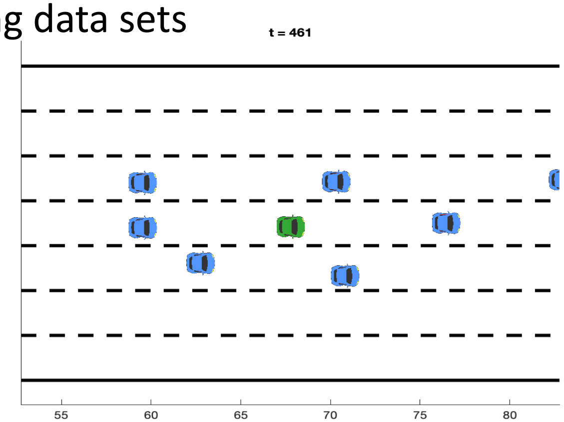


Domains where it is hard to get on-policy data





driving data sets



Data → Models → Control

$$x_{t+1} = f(x_t, u_t, w_t)$$

$$y_t = h(x_t, u_t, z_t)$$

$$w_t \in \mathcal{W}(x_t), z_t \in \mathcal{Z}(x_t)$$ Noise/uncertainty

$$u_t \in \mathcal{U}(x_t)$$ Control constraints

From data to model:

- System identification
- Model (in)validation
- Fault/anomaly detection

From model to control:

- Decision problems of the form:
  - Does there exist a control policy (of the form *C*) such that XXX holds?
    - YES (+ controller)
    - NO (+ certificate/proof of non-existence)

XXX: closed loop system is stable, has some inductive invariant (safe), achieves a task described by an automaton/temporal logic formula (reachability), etc.

Things that I like…

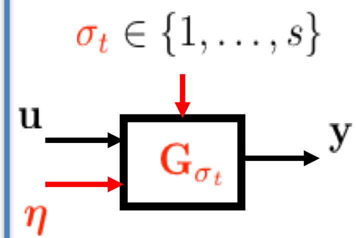
- **Models:**
  - I know how to ask (and in some cases answer) “does there exist a controller” type questions if I have a model
  - Ability to change control objective
  - If I get more data, I can check validity of my model (and therefore, of the controller)
    - Not so clear how to validate a policy (computed in a model-free way) against pre-collected data?

- **Constraints:** (rather than objective functions)
  - Specifying a task with a single reward/cost function is hard (?)
  - Why not append constraints to the objective function → nice additive/quadratic cost functions are not enough

Non-asymptotic analysis of linear system identification

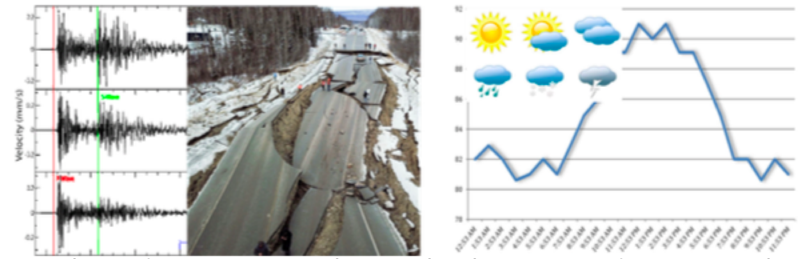
$$x_{t+1} = Ax_t + Bu_t$$
$$y_t = Cx_t + Du_t$$

With Samet Oymak, ACC 19, TAC (tbs)



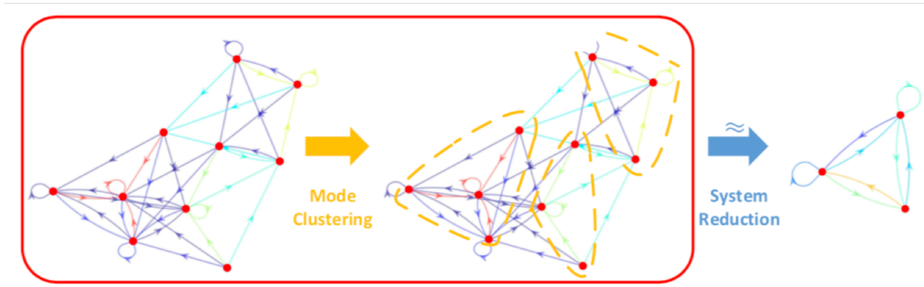
(Online) identification of hybrid/switched linear systems

$$y(t) = \sum_{i=1}^{n_a} a_i(\sigma_t)y(t-i) + \sum_{i=1}^{n_c} c_i(\sigma_t)u(t-i) + \eta(t)$$



Past work and recent results with Zhe Du and Laura Balzano

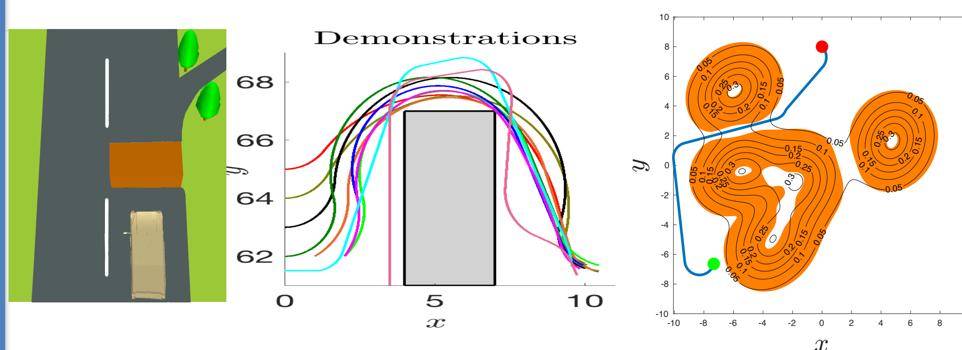
Learning "reduced models" for Markov jump linear systems



2/24/20

With Zhe Du and Laura Balzano, CAMSAP 19

Learning constraints from demonstrations



With Glen Chou and Dmitry Berenson, CORL 19, RA-L 20

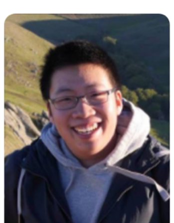
Outline

- Overview of some problems of interest at the intersection of learning, control, and optimization

- **Sample complexity analysis of linear system identification (analysis of Ho-Kalman algorithm)**

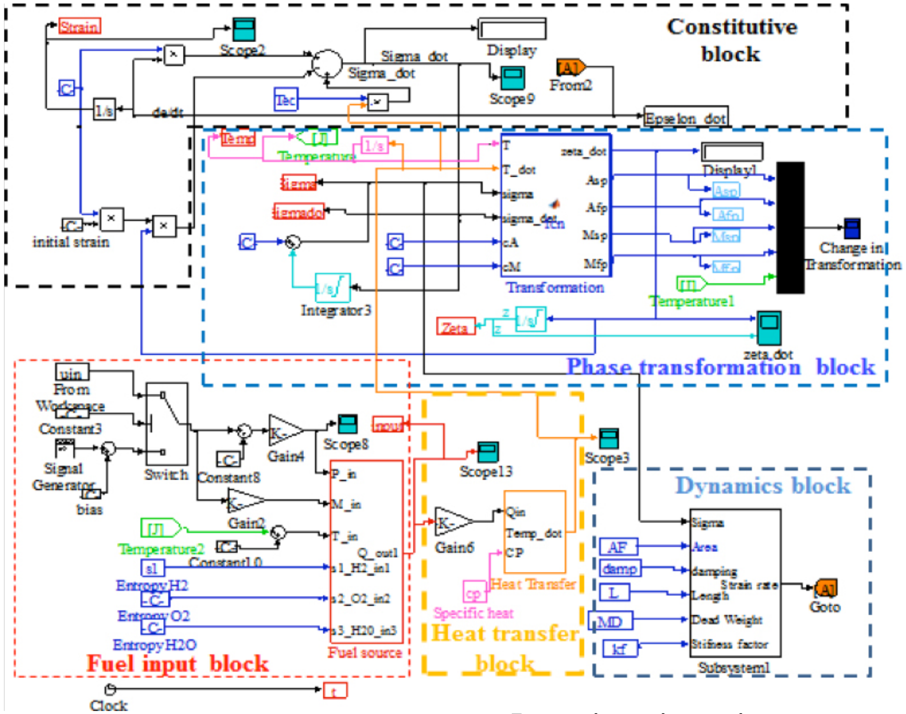
Joint work with Samet Oymak, UC Riverside

- Inverse constraint learning (a la inverse optimal control)



Joint work with Glen Chou and Dmitry Berenson, Michigan

Data → Models → Control



From: iopscience.iop.org

From: www.rsrit.com

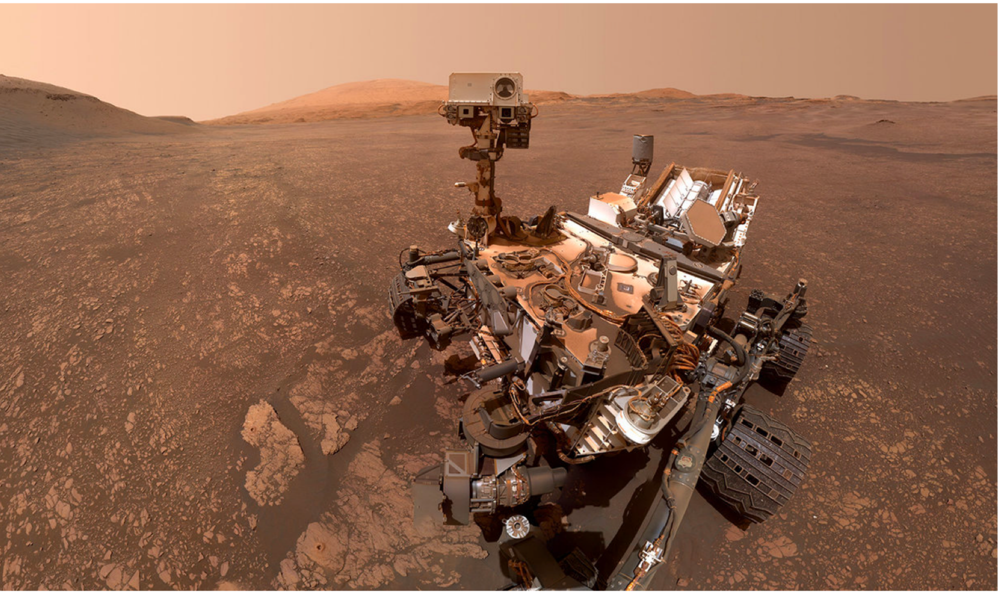
Complex simulation models/ability run lots of experiments

Big data

System identification → learning structured simple models

Models useful for (i) control design, (ii) fast simulations (iii) system monitoring, (iv) anomaly detection, etc.

Data → Models → Control



From: NASA

Exploring unknown environments

From: nbcnews.com

Handling unexpected failures

"Small" data

Online system identification ➔ learning models at run-time

↓

Adaptation

(repurposing, changing mission objectives)

LTI system identification

Given input/output data ($\{u_t, y_t\}_{t=0}^{N}$), find a model of the form:

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + Du_t + z_t$$

- **Asymptotic analysis:**
  - As the data size *N* goes to infinity and/or noise (w_t, z_t) level goes to zero, can we learn the system model?
- **Non-asymptotic analysis:**
  - Given finite amount of noisy data, how does the identification accuracy depend on the data size *N* and noise?
  - What can the best identification algorithm achieve in this case?

Existing results (incomplete list)

- **Asymptotic analysis:**
  - As the data size *N* goes to infinity and/or noise (w_t, z_t) level goes to zero, can we learn the system model?

    Textbook on sys id: [Ljung 99], *standard methods*: Ho-Kalman (Eigen Realization Algorithm-ERA), N4SID, etc.

- **Non-asymptotic analysis:**
  - Given finite amount of noisy data, how does the identification accuracy depend on the data size *N* and noise?
  - What can the best identification algorithm achieve in this case?

    Control theoretic methods: [Weyer et al. 99], [Vidyasagar & Karandikar 01], [Campi & Weyer 02], [Akcay 04], [Carè et al. 18], etc.

    Statistical machine learning methods: [Hardt et al. 16], [Dean et al. 17], [Hazan et al. 17], [Tu et al. 17], [Sarkar & Rakhlin 18], [Simchowitz et al. 18], etc.

Contributions

Understanding the noise sensitivity/robustness of Ho-Kalman algorithm and sample complexity analysis
→ provable guarantees on the accuracy of learned model

Some properties of Ho-Kalman (ERA):

- Learns from input-output data (no state measurement)
  - generically ill-posed
  - can only learn a canonical representation (i.e., learning is up to a similarity transformation)
- Single trajectory
  - requires carefully reasoning about dependencies
- Model class does not satisfy noise invertibility assumption

Ho-Kalman Algorithm

Given input/output data ($\{u_t, y_t\}_{t=0}^{N}$), find a model of the form:

$$x_{t+1} = Ax_t + Bu_t$$
$$y_t = Cx_t + Du_t$$

- Identification problem is **ill-posed**:
  - we can only learn up to a similarity transformation (change of basis)

Ho-Kalman Algorithm

Given input/output data ($\{u_t, y_t\}_{t=0}^{N}$), find a model of the form:

$$x_{t+1} = Ax_t + Bu_t$$
$$y_t = Cx_t + Du_t$$

- Identification problem is **ill-posed**:
  - we can only learn up to a similarity transformation (change of basis)

$$\tilde{x} = Px \Rightarrow \begin{aligned} \tilde{x}_{t+1} &= PAP^{-1}\tilde{x}_t + PBu_t \\ y_t &= CP^{-1}\tilde{x}_t + Du_t \end{aligned}$$

Ho-Kalman Algorithm

Given input/output data ($\{u_t, y_t\}_{t=0}^N$), find a model of the form:

$$x_{t+1} = Ax_t + Bu_t$$
$$y_t = Cx_t + Du_t$$

- Identification problem is **ill-posed**:
  - we can only learn up to a similarity transformation (change of basis).
  - we can only learn the controllable and observable part
- *Assume:* The system is controllable and observable

Ho-Kalman Algorithm

Given input/output data ($\{u_t, y_t\}_{t=0}^{N}$), find a model of the form:

$$x_{t+1} = Ax_t + Bu_t$$
$$y_t = Cx_t + Du_t$$

- Identification problem is **ill-posed**:
  - we can only learn up to a similarity transformation (change of basis).
  - we can only learn the controllable and observable part
- *Assume:* The system is controllable and observable
- Two step procedure:
  1. Estimate the Markov parameters of the system:

     $$D, CB, CAB, CA^2B, \ldots, CA^tB, \ldots$$

     Markov parameters are **invariant** to the choice of basis
  2. Estimate the “system matrices” from Markov parameters

Ho-Kalman Algorithm

Given input/output data ($\{u_t, y_t\}_{t=0}^N$), find a model of the form:

$$x_{t+1} = Ax_t + Bu_t$$
$$y_t = Cx_t + Du_t$$

- Identification problem is **ill-posed**:
  - we can only learn up to a similarity transformation (change of basis).
  - we can only learn the controllable and observable part
- *Assume:* The system is controllable and observable
- Two step procedure:
  1. Estimate the Markov parameters of the system:
  $$D, CB, CAB, CA^2B, \ldots, CA^tB, \ldots$$
  Markov parameters are **invariant** to the choice of basis
  2. **Estimate the "system matrices" from Markov parameters**

Ho-Kalman Algorithm

- Assume Markov parameters of the system are given:

$$D, CB, CAB, CA^2B, \ldots, CA^tB, \ldots$$

- Form the Hankel matrix of Markov parameters

H: hankel matrix

$$\underbrace{\begin{bmatrix} CB & CAB & CA^2B & \ldots & CA^{T_2}B \\ CAB & CA^2B & \iddots & \ldots & CA^{T_2+1}B \\ CA^2B & \iddots & \iddots & \ldots & \vdots \\ \vdots & \iddots & \iddots & \ldots & \vdots \\ CA^{T_1}B & \iddots & \iddots & \ldots & CA^{T_1+T_2}B \end{bmatrix}}_{} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{T_1} \end{bmatrix} \begin{bmatrix} B & AB & A^2B & \ldots & A^{T_2}B \end{bmatrix}$$

$$rank(H) = n$$

n: state-space dim

Ho-Kalman Algorithm

- Assume Markov parameters of the system are given:

$$D, CB, CAB, CA^2B, \dots, CA^tB, \dots$$

- Form the Hankel matrix of Markov parameters

H: hankel matrix

$$\begin{bmatrix} CB & CAB & CA^2B & \dots & CA^{T_2}B \\ CAB & CA^2B & \iddots & \dots & CA^{T_2+1}B \\ CA^2B & \iddots & \iddots & \dots & \vdots \\ \vdots & \iddots & \iddots & \dots & \vdots \\ CA^{T_1}B & \iddots & \iddots & \dots & CA^{T_1+T_2}B \end{bmatrix}$$

H^+ $\qquad$ H^-

$$H^+ = OQ \Longrightarrow H^- = OAQ$$

Ho-Kalman Algorithm

- Assume Markov parameters of the system are given:

$$D, CB, CAB, CA^2B, \ldots, CA^tB, \ldots$$

- Form the Hankel matrix of Markov parameters

H: hankel matrix

$$\begin{bmatrix} CB & CAB & CA^2B & \cdots & CA^{T_2}B \\ CAB & CA^2B & \iddots & \cdots & CA^{T_2+1}B \\ CA^2B & \iddots & \iddots & \cdots & \vdots \\ \vdots & \iddots & \iddots & \cdots & \vdots \\ CA^{T_1}B & \iddots & \iddots & \cdots & CA^{T_1+T_2}B \end{bmatrix}$$

H^+ $\quad$ H^-

$$H^+ = OQ \Longrightarrow H^- = OAQ$$

$$L \doteq \mathrm{SVD}_n(H^+) = U\Sigma V^\intercal$$

$$O \doteq U\Sigma^{1/2}, \ Q \doteq \Sigma^{1/2}V^\intercal$$

$$\bar{C} \doteq \text{first } m \text{ rows of } O$$

$$\bar{B} \doteq \text{first } p \text{ columns of } Q$$

$$\bar{A} \doteq O^\dagger H^- Q^\dagger$$

n: states
m: outputs
p: inputs

Ho-Kalman Algorithm

- Assume Markov parameters of the system are given:

$$D, CB, CAB, CA^2B, \ldots, CA^tB, \ldots$$

- Form the Hankel matrix of Markov parameters

$$H\text{: hankel matrix}$$

$$\begin{bmatrix} CB & CAB & CA^2B & \cdots & CA^{T_2}B \\ CAB & CA^2B & \iddots & \cdots & CA^{T_2+1}B \\ CA^2B & \iddots & \iddots & \cdots & \vdots \\ \vdots & \iddots & \iddots & \cdots & \vdots \\ CA^{T_1}B & \iddots & \iddots & \cdots & CA^{T_1+T_2}B \end{bmatrix}$$

H^+ $\quad$ H^-

Hankel singular values

$$L \doteq \mathrm{SVD}_n(H^+) = U\Sigma V^\intercal$$

$$O \doteq U\Sigma^{1/2},\ Q \doteq \Sigma^{1/2}V^\intercal$$

$$\bar{C} \doteq \text{first } m \text{ rows of } O$$

$$\bar{B} \doteq \text{first } p \text{ columns of } Q$$

$$\bar{A} \doteq O^\dagger H^- Q^\dagger$$

Balanced realization

$$H^+ = OQ \Longrightarrow H^- = OAQ$$

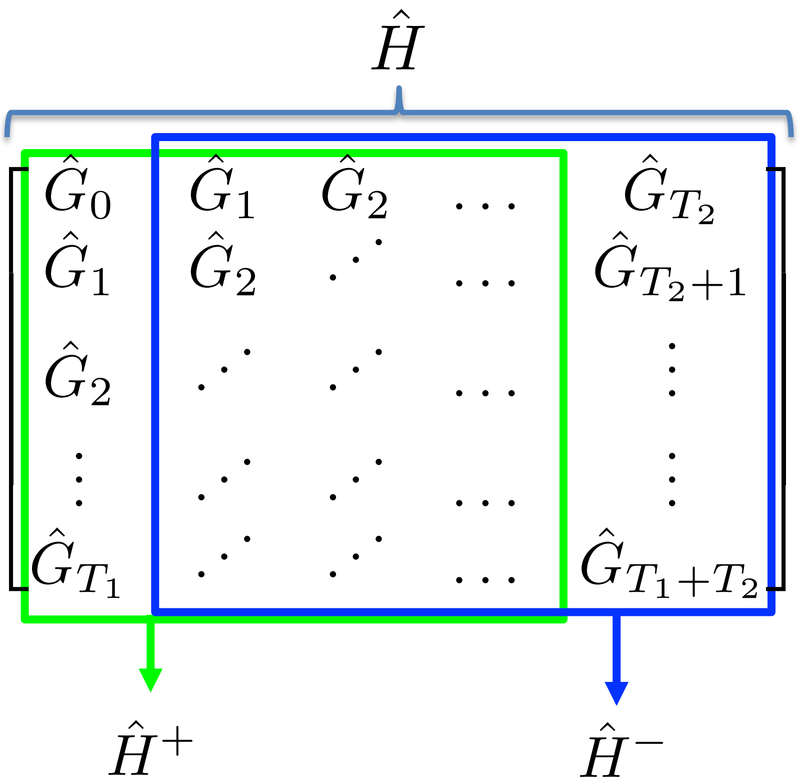
Ho-Kalman Algorithm

- **Estimated** Markov parameters:

$$G = [D, CB, CAB, \ldots, CA^T B]$$

$$\hat{G} = [\hat{G}_{-1}, \hat{G}_0, \hat{G}_1, \ldots, \hat{G}_T]$$

- Form the Hankel matrix:



$$\hat{L} \doteq \mathrm{SVD}_n(\hat{H}^+) = \hat{U}\hat{\Sigma}\hat{V}^\intercal$$

$$\hat{O} \doteq \hat{U}\hat{\Sigma}^{1/2}, \ \hat{Q} \doteq \hat{\Sigma}^{1/2}\hat{V}^\intercal$$

$$\hat{C} \doteq \text{first } m \text{ rows of } \hat{O}$$

$$\hat{B} \doteq \text{first } p \text{ columns of } \hat{Q}$$

$$\hat{A} \doteq \hat{O}^\dagger \hat{H}^- \hat{Q}^\dagger$$

Noise sensitivity of Ho-Kalman

- **Estimated** Markov parameters:

$$G = [D, CB, CAB, \ldots, CA^T B]$$
$$\hat{G} = [\hat{G}_{-1}, \hat{G}_0, \hat{G}_1, \ldots, \hat{G}_T]$$

- Given a bound on $||G - \hat{G}||$

how good are the other estimates?

Algorithm

$$\hat{L} \doteq \text{SVD}_n(\hat{H}^+) = \hat{U}\hat{\Sigma}\hat{V}^\intercal$$
$$\hat{O} \doteq \hat{U}\hat{\Sigma}^{1/2}, \ \hat{Q} \doteq \hat{\Sigma}^{1/2}\hat{V}^\intercal$$
$$\hat{C} \doteq \text{first } m \text{ rows of } \hat{O}$$
$$\hat{B} \doteq \text{first } p \text{ columns of } \hat{Q}$$
$$\hat{A} \doteq \hat{O}^\dagger \hat{H}^- \hat{Q}^\dagger$$

Lemma:

$$\|H - \hat{H}\| \leq \sqrt{\min\{T_1, T_2\}}\|G - \hat{G}\|$$
$$\|L - \hat{L}\| \leq 2\sqrt{\min\{T_1, T_2\}}\|G - \hat{G}\|$$

Theorem 1: Assume, $\|L - \hat{L}\| \leq \sigma_{min}(L)/2$. Then, there exists a unitary matrix *P* s.t.

$$\|\bar{C} - \hat{C}P\|_F \leq \sqrt{5n\|L - \hat{L}\|}$$
$$\|\bar{B} - P^\intercal \hat{B}\|_F \leq \sqrt{5n\|L - \hat{L}\|}$$
$$\|\bar{A} - P^\intercal \hat{A} P\|_F \leq \frac{14\sqrt{n}}{\sigma_{min}(L)}\left(2\|H^- - \hat{H}^-\| + \sqrt{\frac{\|L - \hat{L}\|}{\sigma_{min}(L)}}\|H^-\|\right)$$

Estimation of Markov parameters

Consider

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + Du_t + z_t$$

assume,

$$x_0 = 0,\ u_t \sim \mathcal{N}(0, \sigma_u^2 I_p),\ w_t \sim \mathcal{N}(0, \sigma_w^2 I_n),\ \text{and}\ z_t \sim \mathcal{N}(0, \sigma_z^2 I_m)$$

Then, cross-correlations of input and output give Markov parameters:

$$\mathbb{E}\left[\frac{y_t u_{t-k}^*}{\sigma_u^2}\right] = \begin{cases} D & \text{if} \quad k = 0, \\ CA^{k-1}B & \text{if} \quad k \geq 1. \end{cases}$$

Estimation of Markov parameters

Given input/output data ($\{u_t, y_t\}_{t=0}^{\bar{N}}$), from a process of the form:

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + Du_t + z_t$$

where

$$x_0 = 0,\ u_t \sim \mathcal{N}(0, \sigma_u^2 I_p),\ w_t \sim \mathcal{N}(0, \sigma_w^2 I_n),\ \text{and}\ z_t \sim \mathcal{N}(0, \sigma_z^2 I_m)$$

consider N subsequences of data of length T+1:

$$\begin{array}{cccccccc}
x_0, & x_1, & \ldots & x_T, & x_{T+1}, & \ldots & x_{\bar{N}-T} & \ldots & x_{\bar{N}} \\
u_0, & u_1, & \ldots & u_T, & u_{T+1}, & \ldots & u_{\bar{N}-T} & \ldots & u_{\bar{N}} \\
 & & & & & & & & \\
y_0, & y_1, & \ldots & y_T, & y_{T+1}, & \ldots & y_{\bar{N}-T} & \ldots & y_{\bar{N}}
\end{array}$$

Estimation of Markov parameters

Given input/output data ($\{u_t, y_t\}_{t=0}^{\bar{N}}$), from a process of the form:

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + Du_t + z_t$$

where

$$x_0 = 0,\ u_t \sim \mathcal{N}(0, \sigma_u^2 I_p),\ w_t \sim \mathcal{N}(0, \sigma_w^2 I_n),\ \text{and}\ z_t \sim \mathcal{N}(0, \sigma_z^2 I_m)$$

consider N subsequences of data of length T+1:

$$\begin{array}{cccccccc} x_0, & x_1, & \ldots & x_T, & x_{T+1}, & \ldots & x_{\bar{N}-T} & \ldots & x_{\bar{N}} \\ u_0, & u_1, & \ldots & u_T, & u_{T+1}, & \ldots & u_{\bar{N}-T} & \ldots & u_{\bar{N}} \\ & & & & & & & & \\ y_0, & y_1, & \ldots & y_T, & y_{T+1}, & \ldots & y_{\bar{N}-T} & \ldots & y_{\bar{N}} \end{array}$$

Estimation of Markov parameters

Given input/output data ($\{u_t, y_t\}_{t=0}^{\bar{N}}$), from a process of the form:

$$x_{t+1} = Ax_t + Bu_t + w_t$$

$$y_t = Cx_t + Du_t + z_t$$

where

$$x_0 = 0,\ u_t \sim \mathcal{N}(0, \sigma_u^2 I_p),\ w_t \sim \mathcal{N}(0, \sigma_w^2 I_n),\ \text{and}\ z_t \sim \mathcal{N}(0, \sigma_z^2 I_m)$$

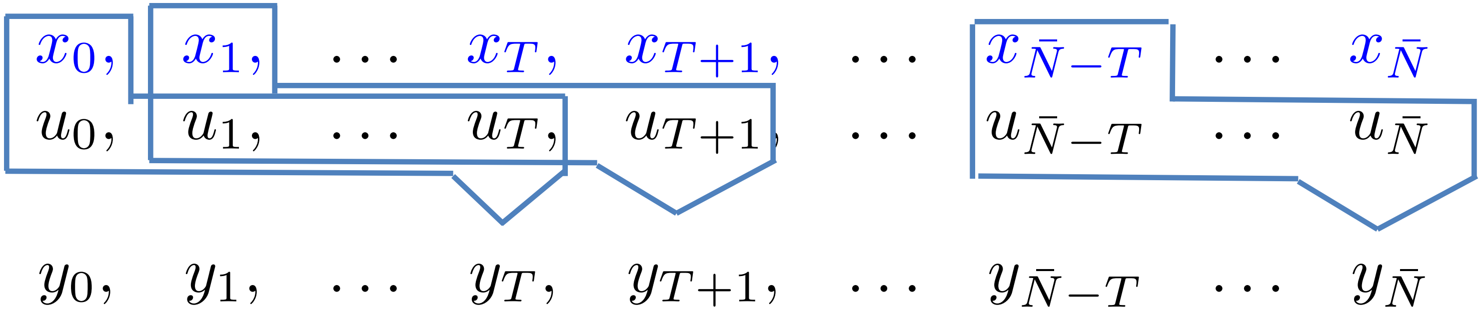
consider N subsequences of data of length T+1:

$$x_0,\quad x_1,\quad \dots\quad x_T,\quad x_{T+1},\quad \dots\quad x_{\bar{N}-T}\quad \dots\quad x_{\bar{N}}$$

$$u_0,\quad u_1,\quad \dots\quad u_T,\quad u_{T+1},\quad \dots\quad u_{\bar{N}-T}\quad \dots\quad u_{\bar{N}}$$

$$y_0,\quad y_1,\quad \dots\quad y_T,\quad y_{T+1},\quad \dots\quad y_{\bar{N}-T}\quad \dots\quad y_{\bar{N}}$$

Estimation of Markov parameters



for all $t \in \{T+1, \dots, \bar{N}\}$

$$
\begin{aligned}
y_t &= CA^T x_{t-T} + Du_t + \sum_{i=1}^{T} CA^{i-1} B u_{t-i} + \sum_{i=1}^{T} CA^{i-1} w_{t-i} + z_t \\
&= G\bar{\mathbf{u}}_t + F\bar{\mathbf{w}}_t + z_t + e_t
\end{aligned}
$$

Recall:

$G = [D, CB, CAB, \dots, CA^T B]$

Estimation of Markov parameters

$$x_0, \quad x_1, \quad \ldots \quad x_T, \quad x_{T+1}, \quad \ldots \quad x_{\bar{N}-T} \quad \ldots \quad x_{\bar{N}}$$

$$u_0, \quad u_1, \quad \ldots \quad u_T, \quad u_{T+1}, \quad \ldots \quad u_{\bar{N}-T} \quad \ldots \quad u_{\bar{N}}$$

$$y_0, \quad y_1, \quad \ldots \quad y_T, \quad y_{T+1}, \quad \ldots \quad y_{\bar{N}-T} \quad \ldots \quad y_{\bar{N}}$$

for all $t \in \{T+1, \ldots, \bar{N}\}$

$$y_t = CA^T x_{t-T} + Du_t + \sum_{i=1}^{T} CA^{i-1}Bu_{t-i} + \sum_{i=1}^{T} CA^{i-1}w_{t-i} + z_t$$

$$= G\bar{\mathbf{u}}_t + F\bar{\mathbf{w}}_t + z_t + e_t$$

e_t → Effect of x_{t-T} (characterize statistics and treat as noise)

$F\bar{\mathbf{w}}_t + z_t$: Noise terms

Recall:

$$G = [D, CB, CAB, \ldots, CA^T B]$$

Estimation of Markov parameters

$$x_0, \quad x_1, \quad \dots \quad x_T, \quad x_{T+1}, \quad \dots \quad x_{\bar{N}-T} \quad \dots \quad x_{\bar{N}}$$

$$u_0, \quad u_1, \quad \dots \quad u_T, \quad u_{T+1}, \quad \dots \quad u_{\bar{N}-T} \quad \dots \quad u_{\bar{N}}$$

$$y_0, \quad y_1, \quad \dots \quad y_T, \quad y_{T+1}, \quad \dots \quad y_{\bar{N}-T} \quad \dots \quad y_{\bar{N}}$$

for all $t \in \{T+1, \dots, \bar{N}\}$

$$y_t = CA^T x_{t-T} + Du_t + \sum_{i=1}^{T} CA^{i-1} B u_{t-i} + \sum_{i=1}^{T} CA^{i-1} w_{t-i} + z_t$$

$$= G\bar{\mathbf{u}}_t + F\bar{\mathbf{w}}_t + z_t + e_t$$

e_t → Effect of x_{t-T} (characterize statistics and treat as noise)

$F\bar{\mathbf{w}}_t + z_t$: Noise terms

Use least squares to estimate *G:*

$$\hat{G} = \arg\min_X \sum_{t=T}^{\bar{N}} ||y_t - X\bar{\mathbf{u}}_t||_2^2$$

How good is this estimate?

Sample complexity

Theorem:

Given input/output data ($\{u_t, y_t\}_{t=0}^{\bar{N}}$), from a process of the form:

$$x_{t+1} = Ax_t + Bu_t + w_t$$
$$y_t = Cx_t + Du_t + z_t$$

where *A* is stable and

$$x_0 = 0,\ u_t \sim \mathcal{N}(0, \sigma_u^2 I_p),\ w_t \sim \mathcal{N}(0, \sigma_w^2 I_n), \text{ and } z_t \sim \mathcal{N}(0, \sigma_z^2 I_m)$$

let $N \geq N_0 = cTq \log^2(2Tq) \log^2(2Nq)$ where $q = n + m + p$

and take $\bar{N} = N + T$

then with **very high** probability, we have

$$\|G - \hat{G}\| \leq \frac{\sigma_z + \sigma_e + \sigma_w \|F\|}{\sigma_u} \sqrt{\frac{N_0}{N}}$$

*Recall the error terms in least squares: $y_t = G\bar{\mathbf{u}}_t + F\bar{\mathbf{w}}_t + z_t + e_t$

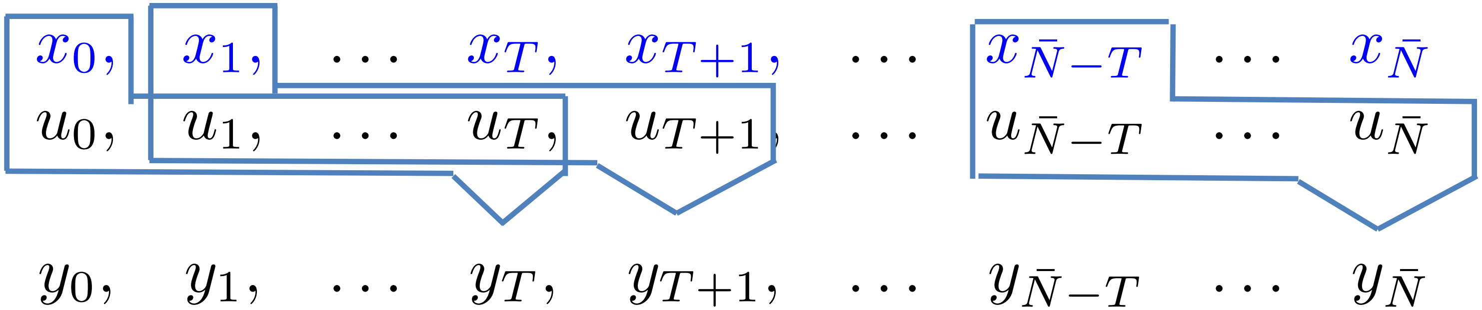
Combining it all...

Given any δ, ε, we can find a “tight” $\bar{N}$ such that if we have input/output data of length $\bar{N}$, with probability (1-δ), we can estimate the system matrices (of balanced realization) by accuracy at most ε.

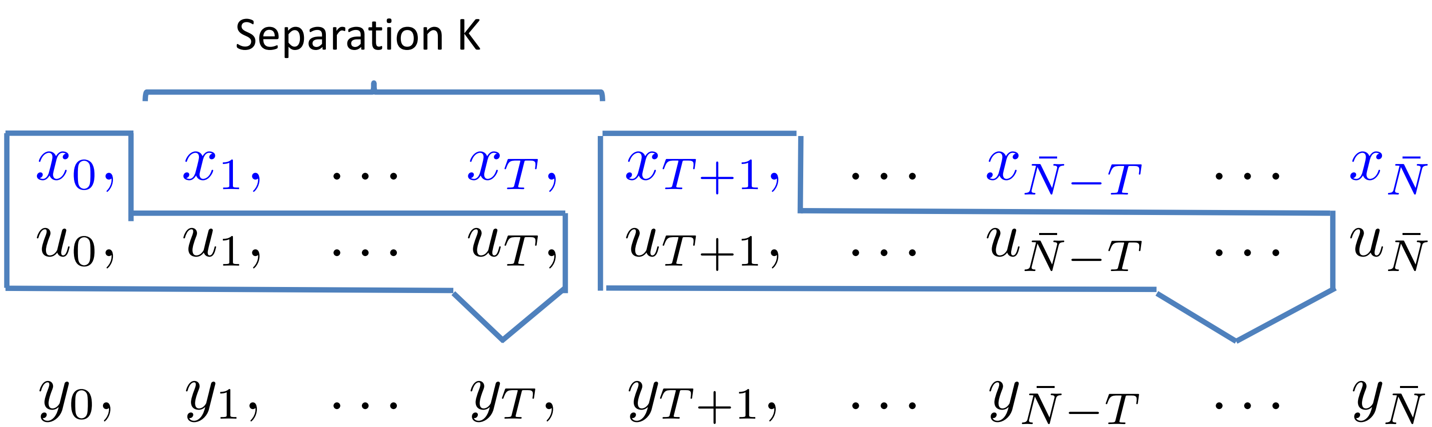
Similarly, given input/output data of length $\bar{N}$, and any δ, we can give a bound ε on the accuracy of the system matrix estimates that is valid with probability (1- δ).

Several applications/extensions: estimates of H-infinity norm error, system order, etc.

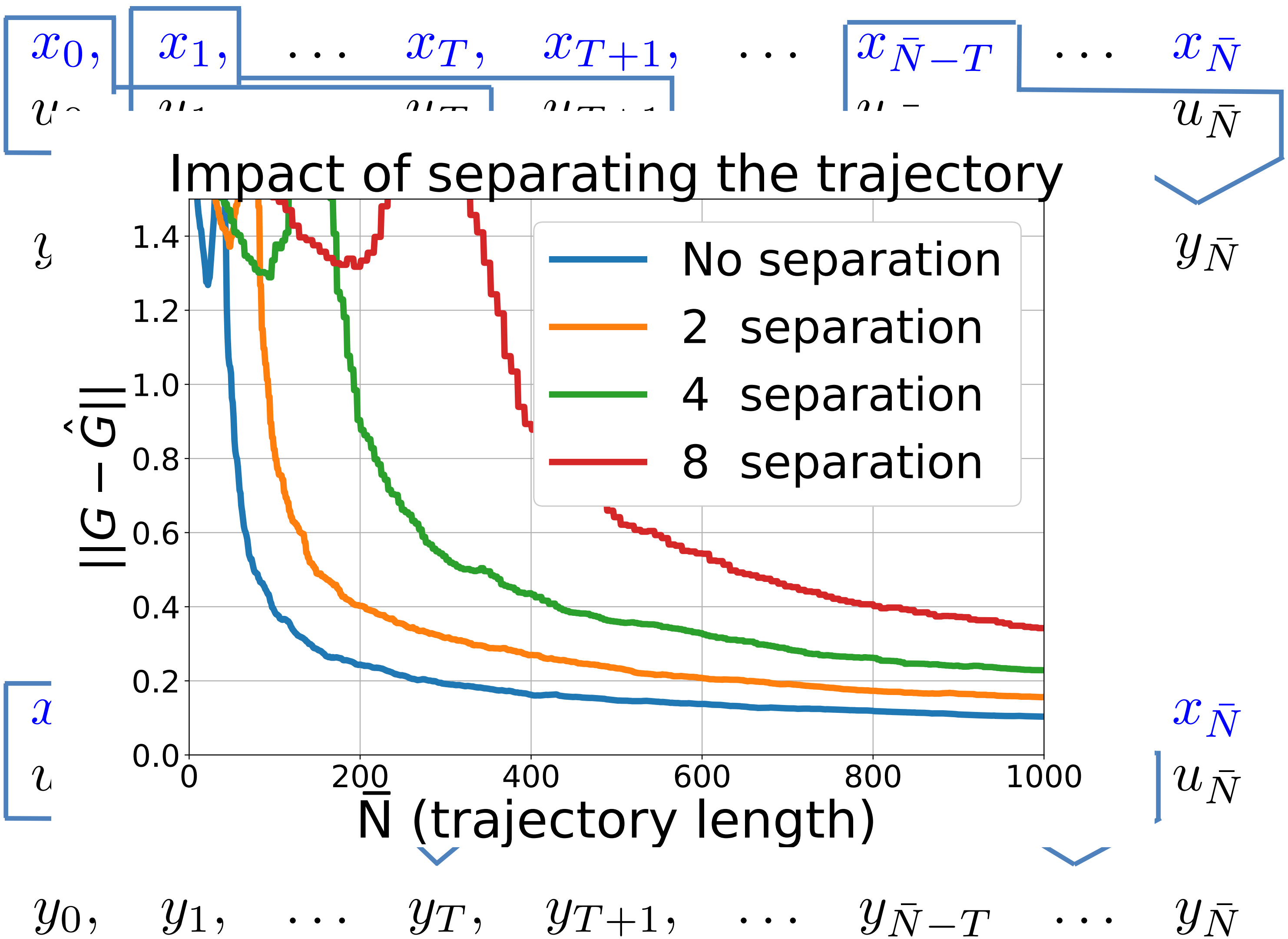
Numerical examples



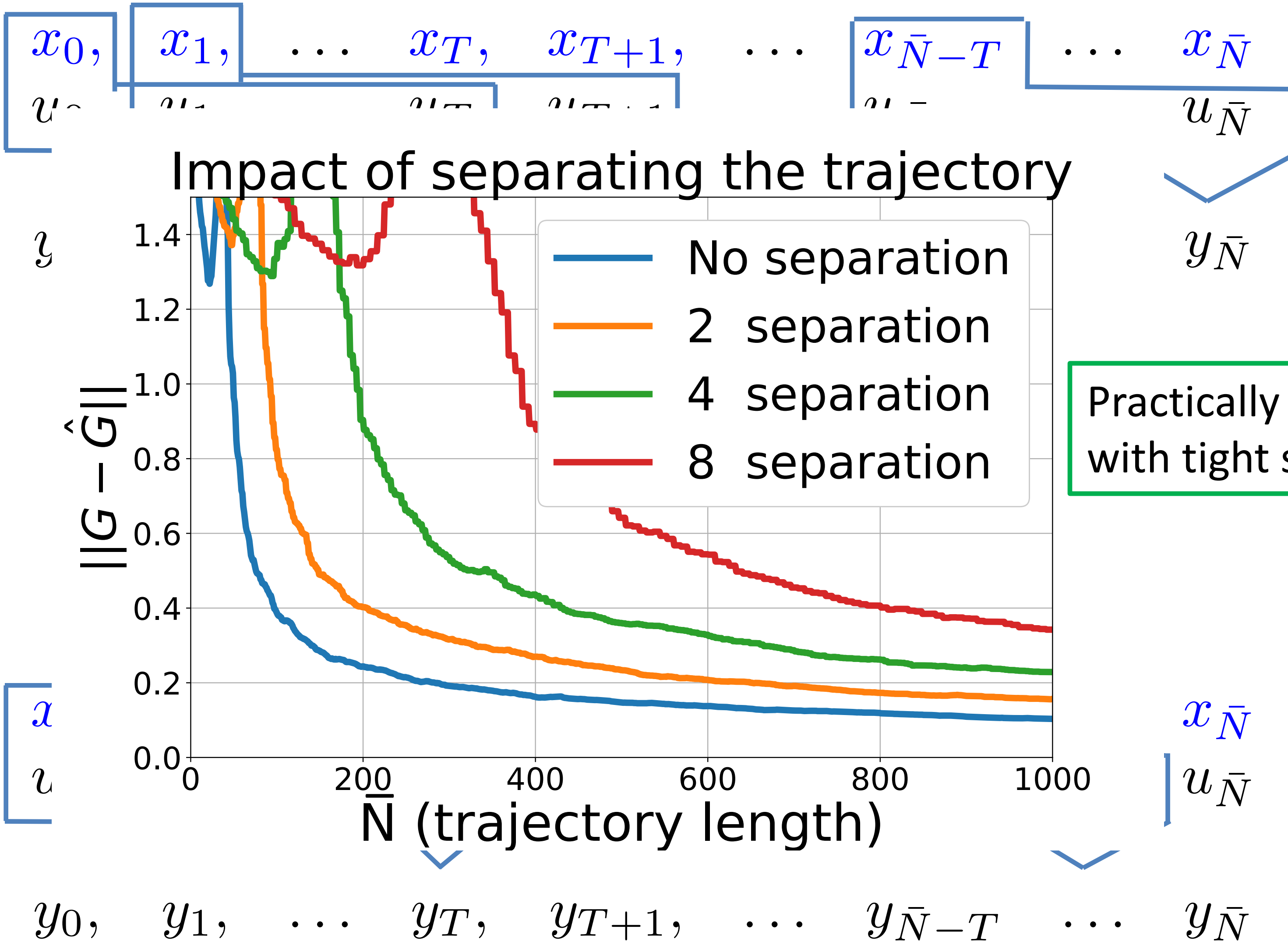
Statistically easier to analyze (but less "efficient") alternative:
a variant of the i.i.d. trajectory view point in [Oymak 19]



Numerical examples

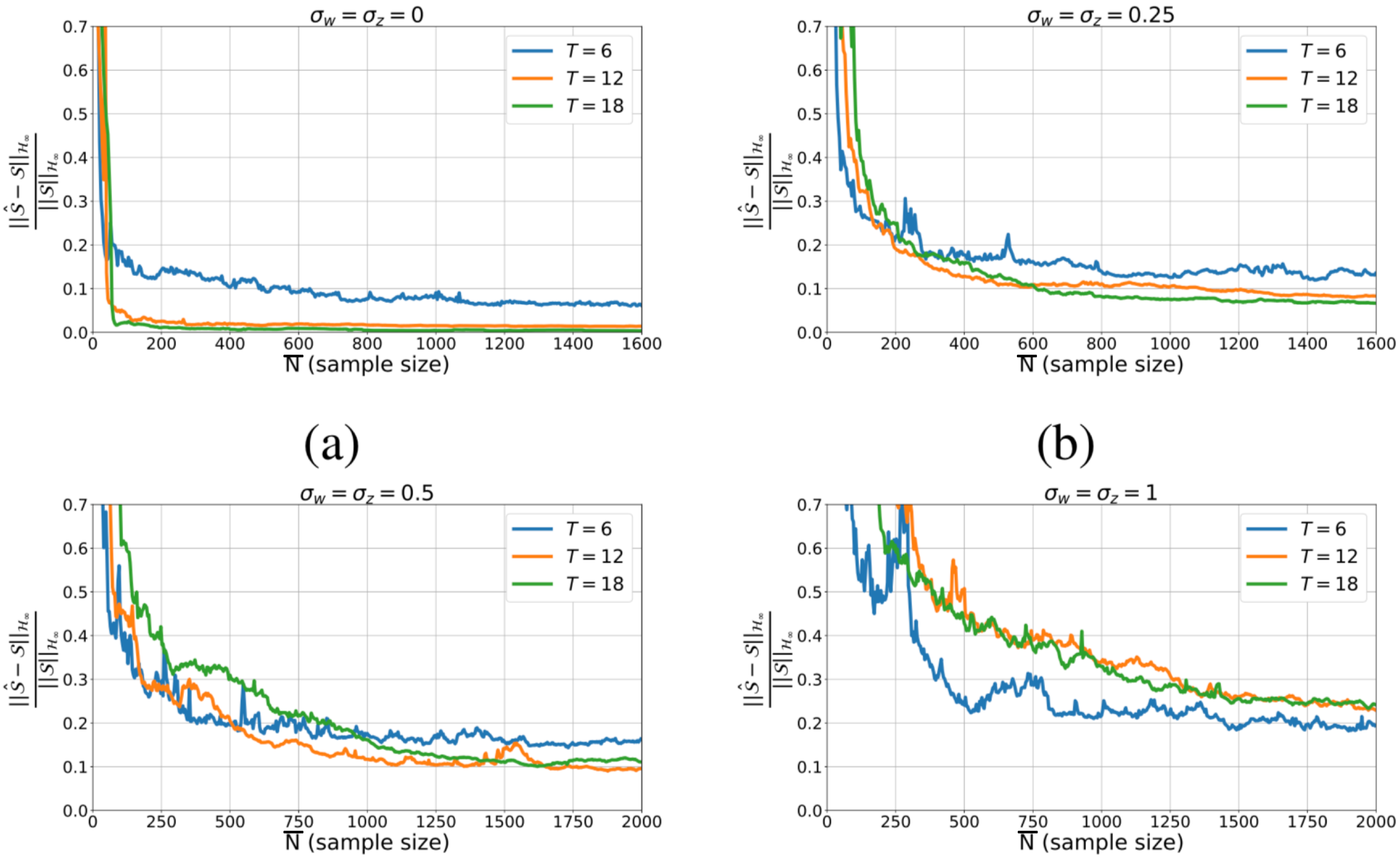


Numerical examples



Practically better use of data with tight statistical bounds!

Numerical examples



(a) (b) (c) (d)

Outline

- Overview of some problems of interest at the intersection of learning, control, and optimization

- Sample complexity analysis of linear system identification (analysis of Ho-Kalman algorithm)

Joint work with Samet Oymak, UC Riverside

- **Inverse constraint learning (a la inverse optimal control)**

Joint work with Glen Chou and Dmitry Berenson, Michigan

Inverse constraint learning

- Learning constraints from demonstrations
  - an alternative to inverse optimal control
  - Why constraints?
    1. Constraints are more modular and explainable
    2. [from some HRI person] Humans don’t think in terms of objective functions
    3. If we have constraints, we can use them for control design to provide system-level guarantees (safety & correctness)
- Challenge: learning only from positive examples

Inverse reinforcement learning: [Abbeel & Ng 04], [Englert et al. 17], [Johnson et al. 13], [Menner et al 19], [Ng & Russell 00], [Singh et al. 18], [Sadigh et al. 17]

Constraint learning: [Calinon & Billard 08], [Lin et al. '17], [Mehr et al '16], [Pais et al. 13], [Pérez-D'Arpino & Shah 17]

Temporal logic inference: [Bakhirkin et al. 18], [Bombara et al. 16], [Kong et al. 14], [Neider & Gavran 18], [Shah et al. '18], [Vazquez-Chanlatte et al. 18]

Demonstrations (from different tasks) with common constraints

Demonstrator knows:

- Cost function, all constraints

We know:

- Cost function, system dynamics, control constraints.

Demonstrator finds trajectory ξ^* that solves:

$$\begin{aligned} \underset{\xi}{\text{minimize}} \quad & c_{\text{task}}(\xi) \\ \text{subject to} \quad & \mathbf{g}(\xi, \theta) \leq \mathbf{0} \\ & \mathbf{g}_{\text{shared}}(\xi) \leq \mathbf{0}, \quad \mathbf{h}_{\text{shared}}(\xi) = \mathbf{0} \\ & \mathbf{g}_{\text{task}}(\xi) \leq \mathbf{0}, \quad \mathbf{h}_{\text{task}}(\xi) = \mathbf{0} \end{aligned}$$

known task-dependent cost (e.g. path length)

unknown shared constraint

known shared constraint (e.g. dynamics, control constraints)

known task-dependent constraint (e.g. start/goal state)

Learn from demonstrations $\{\xi_i^*\}_{i=1}^N$

Two approaches

Globally optimal demonstrations:

Key insight: Any trajectory satisfying the known constraints with lower cost than the demonstration must violate the unknown constraint.

→ Sample lower cost trajectories (can be done with a simulator if dynamics are not given in closed form)

Locally optimal demonstrations:

Key insight: Find a cost function and constraints that make the demonstrations satisfy the KKT optimality conditions.

$$\mathrm{KKT}(\xi)$$

Primal feasibility:	$\mathbf{g}_{\neg k}(\xi, \theta) \leq \mathbf{0}, \quad \mathbf{g}_k(\xi) \leq \mathbf{0}, \quad \mathbf{h}_k(\xi) = \mathbf{0}$
Lagrange multipliers:	$\boldsymbol{\lambda}_{\neg k} \geq \mathbf{0}, \quad \boldsymbol{\lambda}_k \geq \mathbf{0}$
Complementary slackness:	$\boldsymbol{\lambda}_{\neg k} \odot \mathbf{g}_{\neg k}(\xi, \theta) = \mathbf{0},$
	$\boldsymbol{\lambda}_k \odot \mathbf{g}_k(\xi, \theta) = \mathbf{0}$
Stationarity:	$\nabla_\xi c(\xi, \gamma) + \boldsymbol{\lambda}_k^\top \nabla_\xi \mathbf{g}_k(\xi) + \boldsymbol{\lambda}_{\neg k}^\top \nabla_\xi \mathbf{g}_{\neg k}(\xi, \theta)$
	$+ \boldsymbol{\nu}_k^\top \nabla_\xi \mathbf{h}_k(\xi) = \mathbf{0}$

Can be encoded as a mixed integer linear program for certain classes of constraint parameterizations

Constraint extraction problem

<u>Recover cost and constraints</u>:

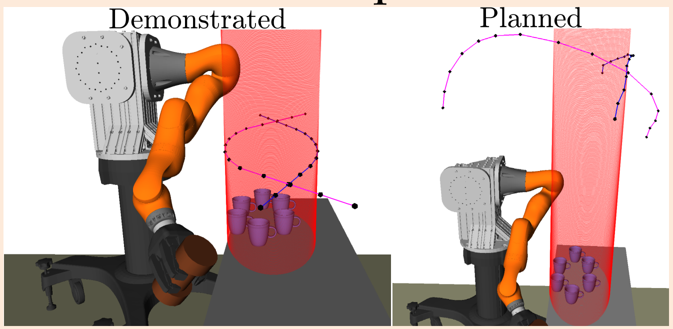
$$\text{find} \quad \gamma, \theta, \{\lambda_k^j, \lambda_{\neg k}^j, \nu_k^j\}_{j=1}^{N_{\text{demos}}}$$

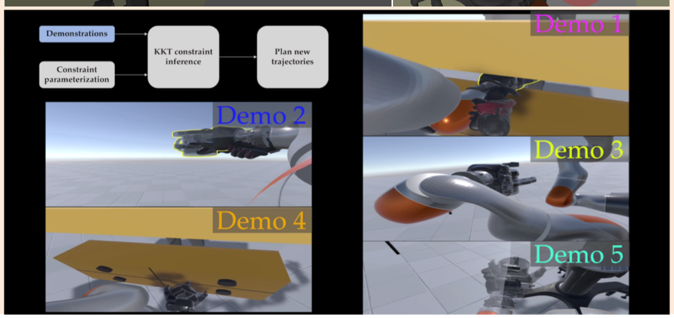
$$\text{subject to} \quad \{\text{KKT}(\xi^j)\}_{j=1}^{N_{\text{demos}}}$$

Finds a feasible constraint/cost parametrization → to extract volumes of guaranteed safe and unsafe regions, check infeasibility of negated constraint

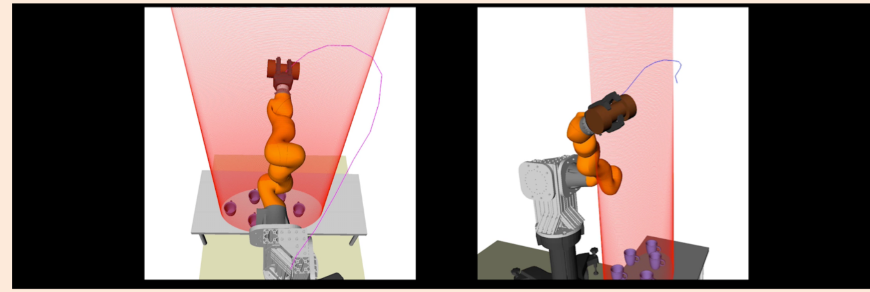
There is some geometric analysis of what is learnable (good parametrizations help).

7-DOF manipulation:

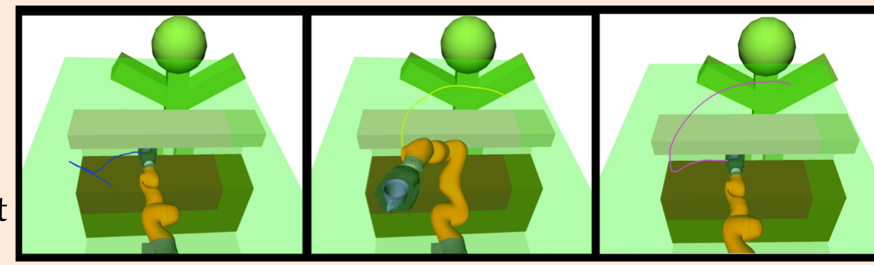




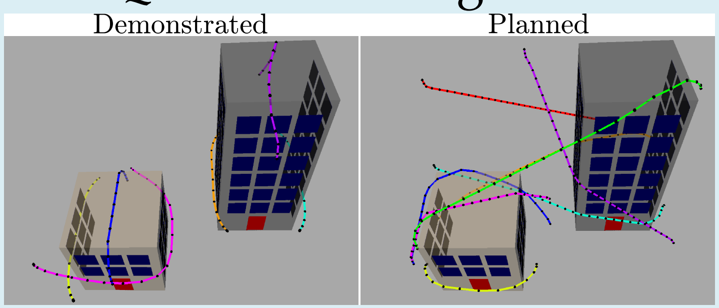
- Elliptical obstacle constraint
- 5 constraint parameters



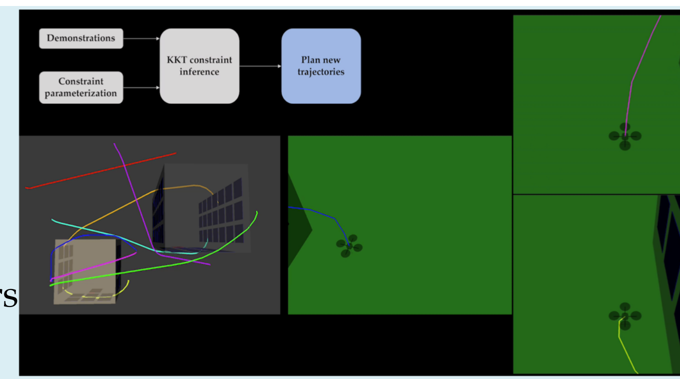
- Pose constraints
- Obstacle constraints
- 15 constraint parameters



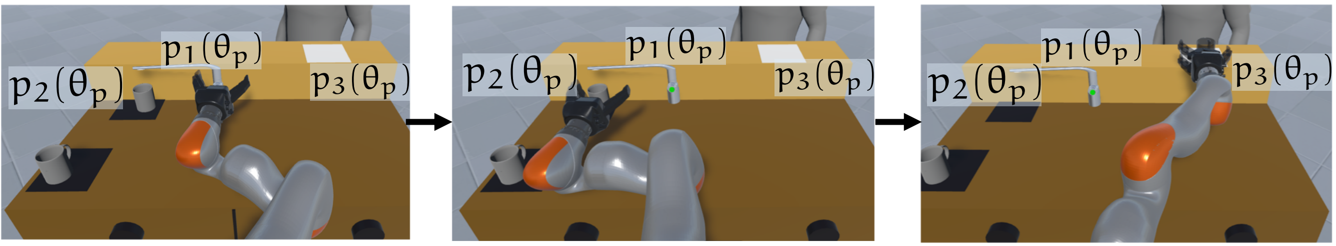
Quadrotor navigation:



- Angular velocity constraints
- Obstacle constraints
- Control input constraints
- Uncertain cost function
- 23 constraint parameters
- 6 cost function parameters



What if forward problem has time-varying constraints (e.g., linear temporal logic)?



$$\varphi(\theta_s, \theta_p) = (\neg p_2 \,\mathcal{U}\, p_1) \wedge (\neg p_3 \,\mathcal{U}\, p_2) \wedge \Diamond p_3$$

<u>*Case 1*</u>: Continuous local optimality, logical *feasibility*

specs for which the demonstration is feasible

specs for which the demonstration visits/avoids APs only if it is cost-advantageous

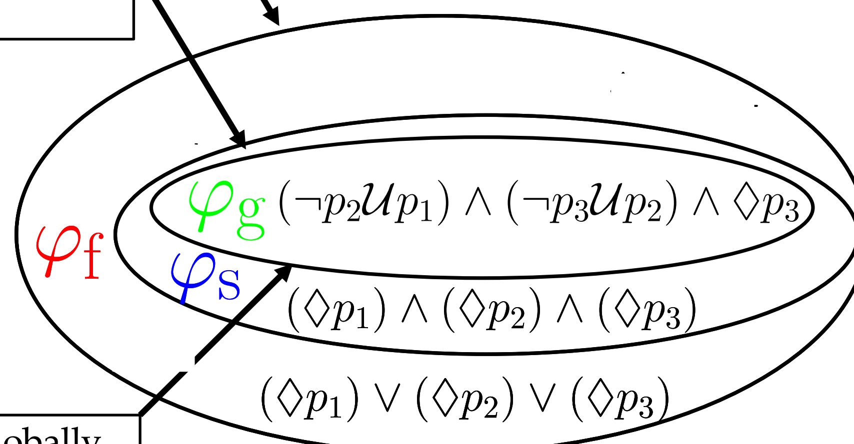
<u>*Case 2*</u>: Continuous local optimality, logical *spec-optimality*

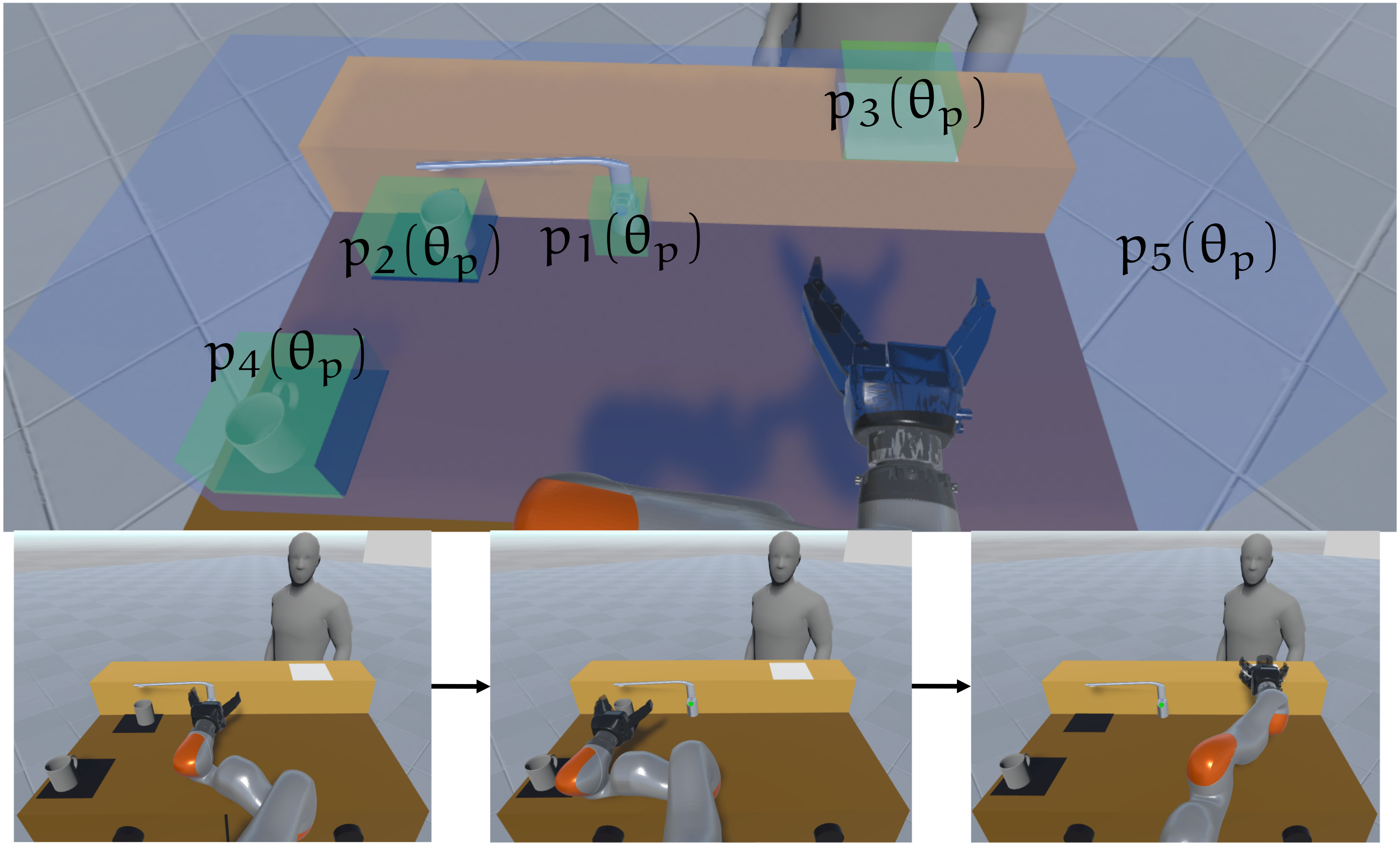
$$\hat{\mathbf{Z}}_{i,t}(\theta_p) \doteq \begin{cases} \neg \mathbf{Z}_{i',t'}(\theta_p) & i' = i, t' = t \\ \mathbf{Z}_{i',t'}(\theta_p) & \text{else} \end{cases}$$

either $\hat{\mathbf{Z}}_{i,t}(\theta_p) = 0$ or ξ is still locally-optimal after relaxing the constraints of p_i at time t

<u>*Case 3*</u>: Continuous global optimality, logical *global optimality*

specs for which the demonstration is globally optimal





$$\varphi(\theta_p, \theta_s) = (\neg p_2(\theta_p)\, \mathcal{U}\, p_1(\theta_p)) \wedge (\neg p_3(\theta_p)\, \mathcal{U}\, p_2(\theta_p)) \wedge \Diamond p_3(\theta_p)$$

Summary & Conclusions

System identification:

- Sensitivity/robustness analysis for Ho-Kalman algorithm
- End to end sample complexity results → guarantees on the learned system parameters from finite samples
- *Future directions:* analysis of other sys id algorithms, closing the loop, control for learning, unstable systems(?)

Inverse constraint learning:

- Optimality is a very strong prior for learning constraints: from simple constraints to complicated temporal tasks
- *Future directions:* more non-determinism in models or data