Reactive Anticipatory Robot Skills with Memory

Hakan Girgin, Julius Jankowski, Sylvain Calinon

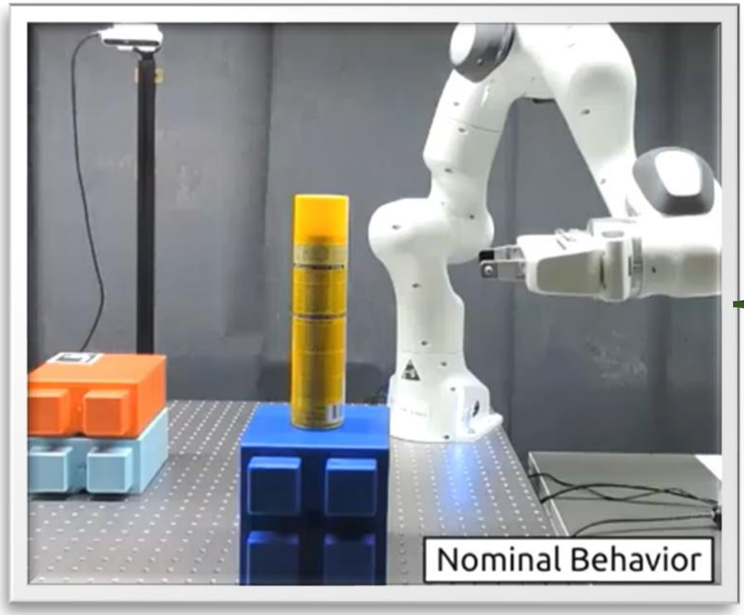
ISRR 2022

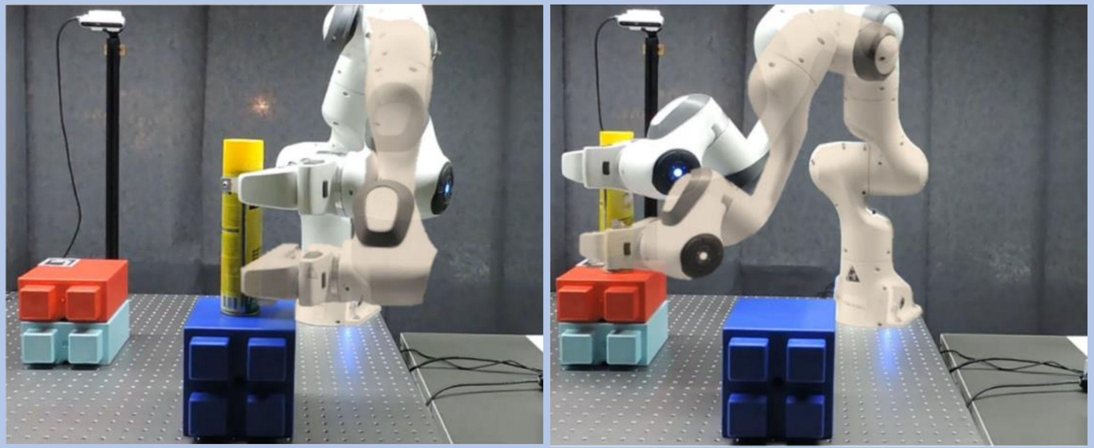
26.09.2022

idiap RESEARCH INSTITUTE

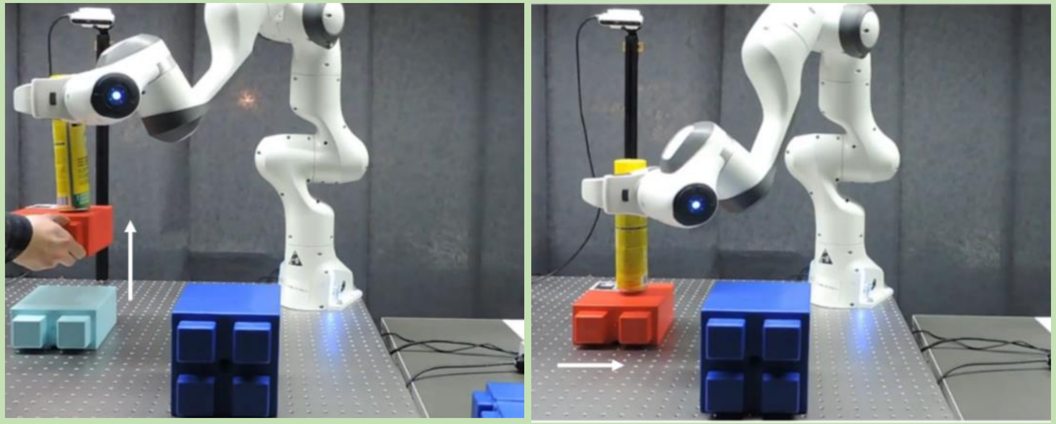
EPFL

Motivations





Memory feedback: time-correlations between states



Fast adaptation to changing objectives

Background: System Level Synthesis

Linear Quadratic Regulator (LQR)

$$\begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}^{0,0} & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}^{1,1} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{K}^{T,T} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}$$

System Level Synthesis (SLS)

$$\begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}^{0,0} & \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}^{1,0} & \boldsymbol{K}^{1,1} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{K}^{T,0} & \boldsymbol{K}^{T,1} & \boldsymbol{K}^{T,2} & \cdots & \boldsymbol{K}^{T,T-1} & \boldsymbol{K}^{T,T} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}$$

System Level Synthesis: Dynamics Formulation

Dynamics model

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{w}_t$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix} \quad \boldsymbol{w} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{w}_0 \\ \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_{T-1} \end{bmatrix}$$

Lifted vector form for an horizon T:

$$\boldsymbol{x} = \boldsymbol{Z}\boldsymbol{A}_d\boldsymbol{x} + \boldsymbol{Z}\boldsymbol{B}_d\boldsymbol{u} + \boldsymbol{w}$$

$$\boldsymbol{x} = \boldsymbol{S}_{\boldsymbol{x}}\boldsymbol{w} + \boldsymbol{S}_{\boldsymbol{u}}\boldsymbol{u} \qquad \boldsymbol{S}_{\boldsymbol{x}} = (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A}_d)^{-1}, \quad \boldsymbol{S}_{\boldsymbol{u}} = \boldsymbol{S}_{\boldsymbol{x}}\boldsymbol{Z}\boldsymbol{B}_d$$

$$\boldsymbol{A}_d := \text{blkdiag}\,(\boldsymbol{A}, \boldsymbol{A}, \ldots, \boldsymbol{A}) = \boldsymbol{I}_T \otimes \boldsymbol{A}$$

$$\boldsymbol{B}_d := \text{blkdiag}\,(\boldsymbol{B}, \boldsymbol{B}, \ldots, \boldsymbol{B}) = \boldsymbol{I}_T \otimes \boldsymbol{B}$$

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{0} & & & \\ \boldsymbol{I}_m & \boldsymbol{0} & & \\ \vdots & \ddots & \ddots & \\ \boldsymbol{0} & \cdots & \boldsymbol{I}_m & \boldsymbol{0} \end{bmatrix} = \boldsymbol{I}_m \otimes \boldsymbol{D}$$

$$\boldsymbol{D} = \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ 0 & 1 & 0 & & \\ \vdots & & \ddots & \ddots & \\ 0 & & \cdots & 1 & 0 \end{bmatrix}$$

System Level Synthesis: Closed-Loop Responses

In SLS, the optimization variables are closed-loop maps: $\mathbf{\Phi}_x \ \mathbf{\Phi}_u$

Closed Loop Dynamics:

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{Z}\boldsymbol{A}_d\boldsymbol{x} + \boldsymbol{Z}\boldsymbol{B}_d\boldsymbol{u} + \boldsymbol{w}, \\ &= \boldsymbol{Z}\boldsymbol{A}_d\boldsymbol{x} + \boldsymbol{Z}\boldsymbol{B}_d\boldsymbol{K}\boldsymbol{x} + \boldsymbol{w}, \\ &= (\boldsymbol{Z}\boldsymbol{A}_d + \boldsymbol{Z}\boldsymbol{B}_d\boldsymbol{K})\boldsymbol{x} + \boldsymbol{w} \end{aligned}$$

Controller:

$$\boldsymbol{u} = \boldsymbol{K}\boldsymbol{x}$$

$$\begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}^{0,0} & & & \\ \boldsymbol{K}^{1,1} & \boldsymbol{K}^{1,0} & & \\ \vdots & \ddots & \ddots & \\ \boldsymbol{K}^{T,T} & \cdots & \boldsymbol{K}^{T,1} & \boldsymbol{K}^{T,0} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}$$

Closed Loop Responses:

$$\boldsymbol{x} = \Big(\boldsymbol{I} - (\boldsymbol{Z}\boldsymbol{A}_d + \boldsymbol{Z}\boldsymbol{B}_d\boldsymbol{K})\Big)^{-1}\boldsymbol{w} = \mathbf{\Phi}_x\boldsymbol{w}$$

$$\boldsymbol{u} = \boldsymbol{K}\Big(\boldsymbol{I} - (\boldsymbol{Z}\boldsymbol{A}_d + \boldsymbol{Z}\boldsymbol{B}_d\boldsymbol{K})\Big)^{-1}\boldsymbol{w} = \mathbf{\Phi}_u\boldsymbol{w}$$

$$\boldsymbol{K} = \mathbf{\Phi}_u\mathbf{\Phi}_x^{-1}$$

System Level Synthesis: LQR

$$\begin{aligned}\min_{\boldsymbol{x},\boldsymbol{u}} \quad & \mathbb{E}[\|\boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{x}\|_2^2 + \|\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{u}\|_2^2] \\ \text{s.t.} \quad & \boldsymbol{x} = \boldsymbol{S}_{\boldsymbol{x}}\boldsymbol{w} + \boldsymbol{S}_{\boldsymbol{u}}\boldsymbol{u}\end{aligned}$$

$$\boldsymbol{x} = \boldsymbol{\Phi}_x \boldsymbol{w}$$

$$\boldsymbol{u} = \boldsymbol{\Phi}_u \boldsymbol{w}$$

$$\begin{aligned}\min_{\boldsymbol{\Phi}_x,\boldsymbol{\Phi}_u} \quad & \mathbb{E}[\|\boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{\Phi}_x\boldsymbol{w}\|_2^2 + \|\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{\Phi}_u\boldsymbol{w}\|_2^2] \\ \text{s.t.} \quad & \boldsymbol{\Phi}_x = \boldsymbol{S}_x + \boldsymbol{S}_u\boldsymbol{\Phi}_u, \\ & \boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u \in \mathcal{L}\end{aligned}$$

Q_{11}	Q_{12}	Q_{13}	Q_{14}
Q_{21}	Q_{22}	Q_{23}	Q_{24}
Q_{31}	Q_{32}	Q_{33}	Q_{34}
Q_{41}	Q_{42}	Q_{43}	Q_{44}

Time correlations

LQR - batch form

$$\begin{aligned} \min_{\boldsymbol{x}_t, \boldsymbol{u}_t} \quad & \sum_{t=0}^{T} \mathbb{E}[\boldsymbol{x}_t^\top \boldsymbol{Q}_t \boldsymbol{x}_t + \boldsymbol{u}_t^\top \boldsymbol{R}_t \boldsymbol{u}_t] \\ \text{s.t.} \quad & \boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{w}_t \end{aligned}$$

$$\boldsymbol{Q} = \text{blkdiag}(\boldsymbol{Q}_0, \boldsymbol{Q}_1, \dots, \boldsymbol{Q}_T)$$

$$\boldsymbol{R} = \text{blkdiag}(\boldsymbol{R}_0, \boldsymbol{R}_1, \dots, \boldsymbol{R}_T)$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix} \quad \boldsymbol{w} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{w}_0 \\ \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_{T-1} \end{bmatrix}$$

$$\boldsymbol{x} = \boldsymbol{S}_{\boldsymbol{x}}\boldsymbol{w} + \boldsymbol{S}_{\boldsymbol{u}}\boldsymbol{u} \qquad \begin{aligned} \boldsymbol{S}_{\boldsymbol{x}} &= (\boldsymbol{I} - \boldsymbol{Z}\boldsymbol{A}_d)^{-1}, \\ \boldsymbol{S}_{\boldsymbol{u}} &= \boldsymbol{S}_{\boldsymbol{x}}\boldsymbol{Z}\boldsymbol{B}_d \end{aligned}$$

$$\begin{aligned} \min_{\boldsymbol{x}, \boldsymbol{u}} \quad & \mathbb{E}[\|\boldsymbol{Q}^{\frac{1}{2}}\boldsymbol{x}\|_2^2 + \|\boldsymbol{R}^{\frac{1}{2}}\boldsymbol{u}\|_2^2] \\ \text{s.t.} \quad & \boldsymbol{x} = \boldsymbol{S}_{\boldsymbol{x}}\boldsymbol{w} + \boldsymbol{S}_{\boldsymbol{u}}\boldsymbol{u} \end{aligned}$$

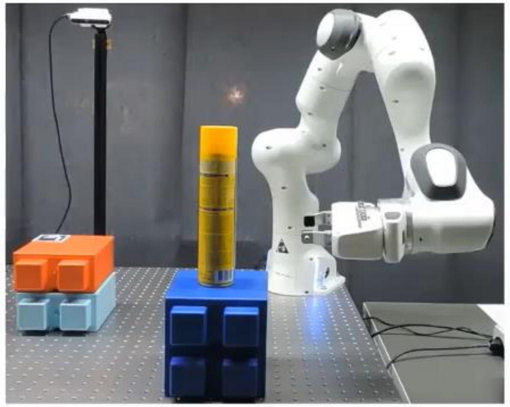
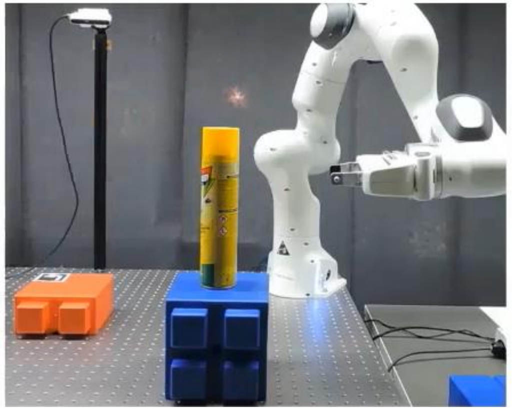
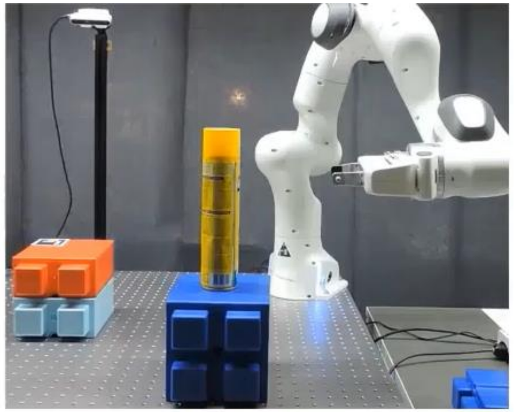
Contributions

extended SLS (eSLS)
Solves for linear quadratic tracking tasks

iterative SLS (iSLS)
Solves for nonlinear nonquadratic tasks

$$\boldsymbol{u} = \boldsymbol{K}\boldsymbol{x} + \boldsymbol{k}$$

Memory feedback and **time-correlations**	**Fast adaptation** to changing objectives

Extended System Level Synthesis (eSLS)

- Adding a feedforward part to the controller to solve tracking and via-point tasks. $u = Kx + k$

$$x = ZA_d x + ZB_d(Kx + k) + w,$$
$$= \Phi_x w + d_x$$
$$u = \Phi_u w + d_u$$

Convex optimization problem with analytical solution as in LQR

$$\min_{\Phi_x, \Phi_u, d_x, d_u} \quad \|\Phi_x w + d_x - x_d\|_Q^2 + \|\Phi_u w + d_u - u_d\|_R^2$$
$$\text{s.t.} \quad \Phi_x = S_x + S_u \Phi_u,$$
$$d_x = S_u d_u,$$
$$\Phi_x, \Phi_u \in \mathcal{L}$$

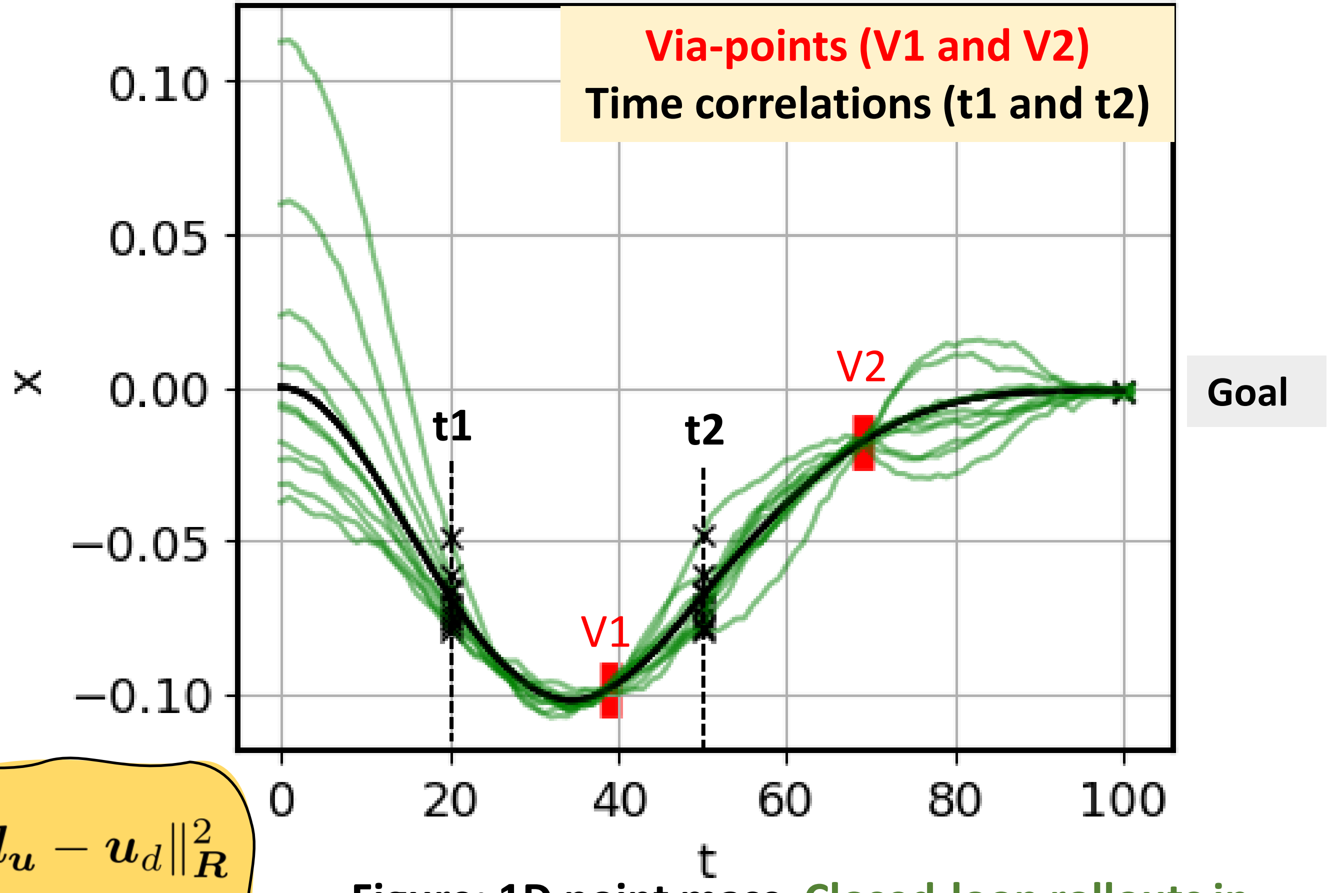
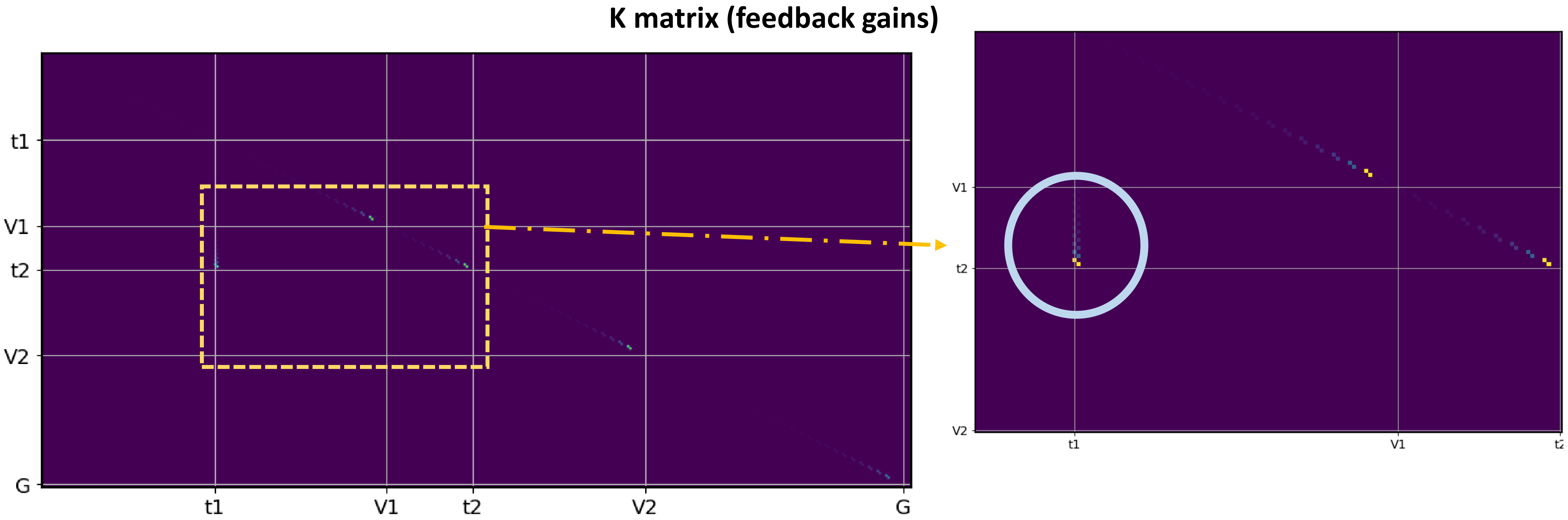


Figure: 1D point mass. Closed-loop rollouts in the presence of noise and starting from different initial states.

eSLS Feedback Gains



$$
\begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix} = \begin{bmatrix} \boldsymbol{K}^{0,0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \boldsymbol{K}^{1,0} & \boldsymbol{K}^{1,1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{K}^{T,0} & \boldsymbol{K}^{T,1} & \boldsymbol{K}^{T,2} & \cdots & \boldsymbol{K}^{T,T-1} & \boldsymbol{K}^{T,T} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}
$$

How to construct time correlations?

The correlations that we consider in this work are in the form $\boldsymbol{C}\boldsymbol{x}_{t_1} + \boldsymbol{c} \sim \boldsymbol{x}_{t_2}$, where $\boldsymbol{C}$ and $\boldsymbol{c}$ are the coefficient matrix and the vector, respectively.

$$
\begin{aligned}
c(\boldsymbol{x}_{t_1}, \boldsymbol{x}_{t_2}) &= (\boldsymbol{C}\boldsymbol{x}_{t_1} + \boldsymbol{c} - \boldsymbol{x}_{t_2})^\top \boldsymbol{Q}_\text{c} (\boldsymbol{C}\boldsymbol{x}_{t_1} + \boldsymbol{c} - \boldsymbol{x}_{t_2}),\\
&= \boldsymbol{x}_{t_1}^\top \underbrace{\boldsymbol{C}^\top \boldsymbol{Q}_\text{c} \boldsymbol{C}}_{\boldsymbol{Q}_{t_1}} \boldsymbol{x}_{t_1} + (\boldsymbol{x}_{t_2} - \boldsymbol{c})^\top \underbrace{\boldsymbol{Q}_\text{c}}_{\boldsymbol{Q}_{t_2}} (\boldsymbol{x}_{t_2} - \boldsymbol{c})\\
&\quad + \boldsymbol{x}_{t_1}^\top \underbrace{\boldsymbol{C}^\top \boldsymbol{Q}_\text{c}}_{-\boldsymbol{Q}_{t_1 t_2}} (\boldsymbol{x}_{t_2} - \boldsymbol{c}) - 2 (\boldsymbol{x}_{t_2} - \boldsymbol{c})^\top \underbrace{\boldsymbol{Q}_\text{c} \boldsymbol{C}}_{-\boldsymbol{Q}_{t_2 t_1}} \boldsymbol{x}_{t_1},\\
&= \begin{bmatrix} \boldsymbol{x}_{t_1} \\ \boldsymbol{x}_{t_2} - \boldsymbol{c} \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{Q}_{t_1} & \boldsymbol{Q}_{t_1 t_2} \\ \boldsymbol{Q}_{t_2 t_1} & \boldsymbol{Q}_{t_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{t_1} \\ \boldsymbol{x}_{t_2} - \boldsymbol{c} \end{bmatrix}.
\end{aligned}
$$

Q_{11}	Q_{12}	Q_{13}	Q_{14}
Q_{21}	Q_{22}	Q_{23}	Q_{24}
Q_{31}	Q_{32}	Q_{33}	Q_{34}
Q_{41}	Q_{42}	Q_{43}	Q_{44}

$$
\begin{aligned}
\min_{\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u} \quad & \mathbb{E}[\|\boldsymbol{Q}^{\frac{1}{2}} \boldsymbol{\Phi}_x \boldsymbol{w}\|_2^2 + \|\boldsymbol{R}^{\frac{1}{2}} \boldsymbol{\Phi}_u \boldsymbol{w}\|_2^2]\\
\text{s.t.} \quad & \boldsymbol{\Phi}_x = \boldsymbol{S}_x + \boldsymbol{S}_u \boldsymbol{\Phi}_u,\\
& \boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u \in \mathcal{L}
\end{aligned}
$$

Fast adaptation via a matrix-vector product

Feedback gains hold information only about **the precision of the task**

$$\boldsymbol{K} = \boldsymbol{\Phi}_u \boldsymbol{\Phi}_x^{-1}$$

- ***Feedforward terms*** *hold information* ***about the precisions and the desired states.***
- ***The feedforward term is a LINEAR function of the desired states.***

$$\begin{aligned}\boldsymbol{k} &= (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{S}_{\boldsymbol{u}})\boldsymbol{d}_{\boldsymbol{u}},\\ &= (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{S}_{\boldsymbol{u}})(\boldsymbol{S}_{\boldsymbol{u}}^{\top}\boldsymbol{Q}\boldsymbol{S}_{\boldsymbol{u}} + \boldsymbol{R})^{-1}(\boldsymbol{S}_{\boldsymbol{u}}^{\top}\boldsymbol{Q}\boldsymbol{x}_d + \boldsymbol{R}\boldsymbol{u}_d),\\ &= \boldsymbol{F}_{\boldsymbol{x}}\boldsymbol{x}_d + \boldsymbol{F}_{\boldsymbol{u}}\boldsymbol{u}_d,\end{aligned}$$

Algorithm 1: Extended System Level Synthesis

Solve for feedforward terms $\boldsymbol{d}_{\boldsymbol{u}} = (\boldsymbol{S}_{\boldsymbol{u}}^{\top}\boldsymbol{Q}\boldsymbol{S}_{\boldsymbol{u}} + \boldsymbol{R})^{-1}(\boldsymbol{S}_{\boldsymbol{u}}^{\top}\boldsymbol{Q}\boldsymbol{x}_d + \boldsymbol{R}\boldsymbol{u}_d)$

while $i < T$ **do** *Solve for feedback terms*

$$\hat{\boldsymbol{\Phi}}_u^i = -(\boldsymbol{S}_{\boldsymbol{u}}^{i\top}\boldsymbol{Q}^i\boldsymbol{S}_{\boldsymbol{u}}^i + \boldsymbol{R}^i)^{-1}\boldsymbol{S}_{\boldsymbol{u}}^{i\top}\boldsymbol{Q}^i\boldsymbol{S}_{\boldsymbol{x}}^i$$

$$\hat{\boldsymbol{\Phi}}_x^i = \boldsymbol{S}_{\boldsymbol{x}}^i + \boldsymbol{S}_{\boldsymbol{u}}^i\hat{\boldsymbol{\Phi}}_u^i$$

end

Compute the feedback and feedforward parts of the controller with
$\boldsymbol{K}=\boldsymbol{\Phi}_u\boldsymbol{\Phi}_x^{-1}$, $\boldsymbol{k}=(\boldsymbol{I}-\boldsymbol{K}\boldsymbol{S}_{\boldsymbol{u}})\boldsymbol{d}_{\boldsymbol{u}}$

Iterative System Level Synthesis (iSLS)

Closed Loop Responses:

$$\Delta \boldsymbol{x} = \Big(\boldsymbol{I} - (\boldsymbol{Z}\boldsymbol{A}_d + \boldsymbol{Z}\boldsymbol{B}_d\boldsymbol{K})\Big)^{-1}\Delta \boldsymbol{w} = \boldsymbol{\Phi}_x \Delta \boldsymbol{w}$$

$$\Delta \boldsymbol{u} = \boldsymbol{K}\Big(\boldsymbol{I} - (\boldsymbol{Z}\boldsymbol{A}_d + \boldsymbol{Z}\boldsymbol{B}_d\boldsymbol{K})\Big)^{-1}\Delta \boldsymbol{w} = \boldsymbol{\Phi}_u \Delta \boldsymbol{w}$$

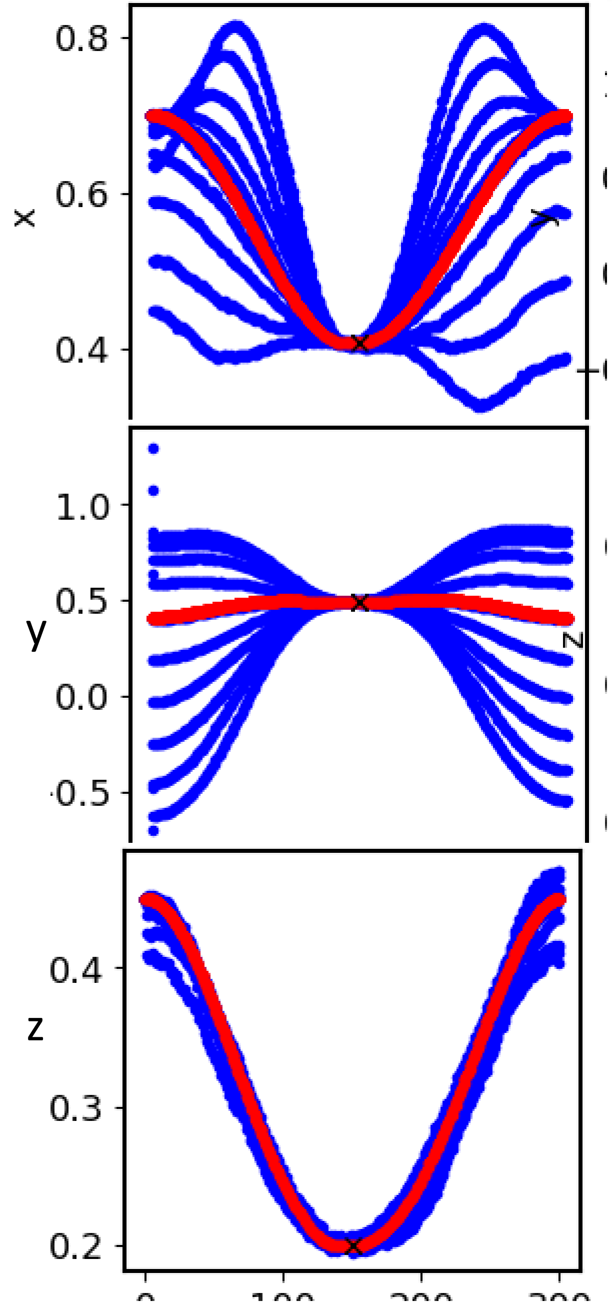
$$\begin{aligned} \min_{\boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u, \boldsymbol{d}_x, \boldsymbol{d}_u} \quad & \|\boldsymbol{\Phi}_x \Delta \boldsymbol{w} + \boldsymbol{d}_{\boldsymbol{x}} - \boldsymbol{x}_d\|^2_{\boldsymbol{Q}} + \|\boldsymbol{\Phi}_u \Delta \boldsymbol{w} + \boldsymbol{d}_{\boldsymbol{u}} - \boldsymbol{u}_d\|^2_{\boldsymbol{R}} \\ \text{s.t.} \quad & \boldsymbol{\Phi}_x = \boldsymbol{S}_x + \boldsymbol{S}_u \boldsymbol{\Phi}_u, \\ & \boldsymbol{d}_{\boldsymbol{x}} = \boldsymbol{S}_{\boldsymbol{u}} \boldsymbol{d}_{\boldsymbol{u}}, \\ & \boldsymbol{\Phi}_x, \boldsymbol{\Phi}_u \in \mathcal{L} \end{aligned}$$

Closed-form Newton step as in iLQR

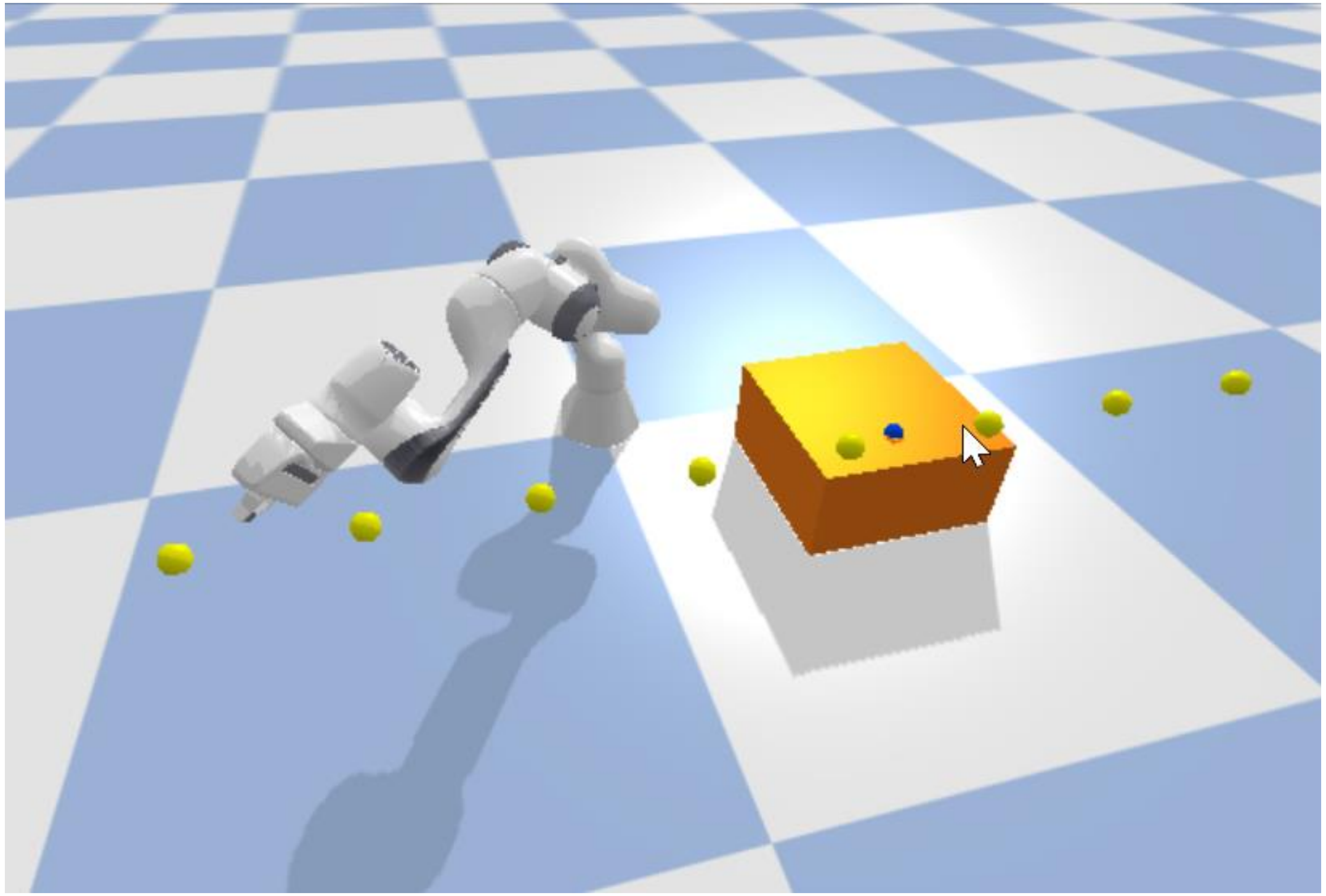
Algorithm 1: Iterative System Level Synthesis (iSLS)

Initialize the nominal state $\hat{\boldsymbol{x}}_t$ and control $\hat{\boldsymbol{u}}_t$;
Initialize the change in the cost Δc ;
Set a threshold τ ;
while $|\Delta c| > \tau$ **do** *Solve iSLS*
  Linearize the dynamics and quadratize the cost function around $\{\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{u}}_t\}_{t=0}^T$ to find $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}_{\boldsymbol{x}\boldsymbol{x}}$, $\boldsymbol{x}_d$ and $\boldsymbol{u}_d$;
  Solve (32) to find $\boldsymbol{K}$ and $\boldsymbol{k}$;
  Do line search to update $\boldsymbol{k}$ using the controller $\Delta \boldsymbol{u} = \boldsymbol{K}\Delta \boldsymbol{x} + \boldsymbol{k}$ and the dynamics model.
  Update Δc.
end

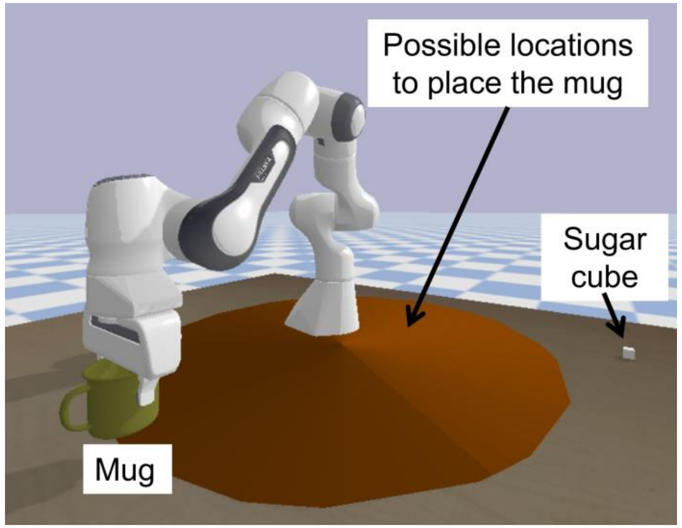
Simulation Study: Fast adaptation in iSLS

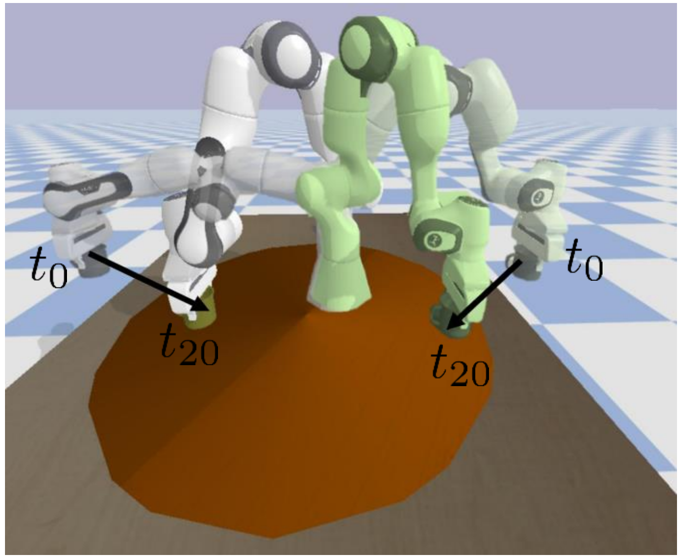


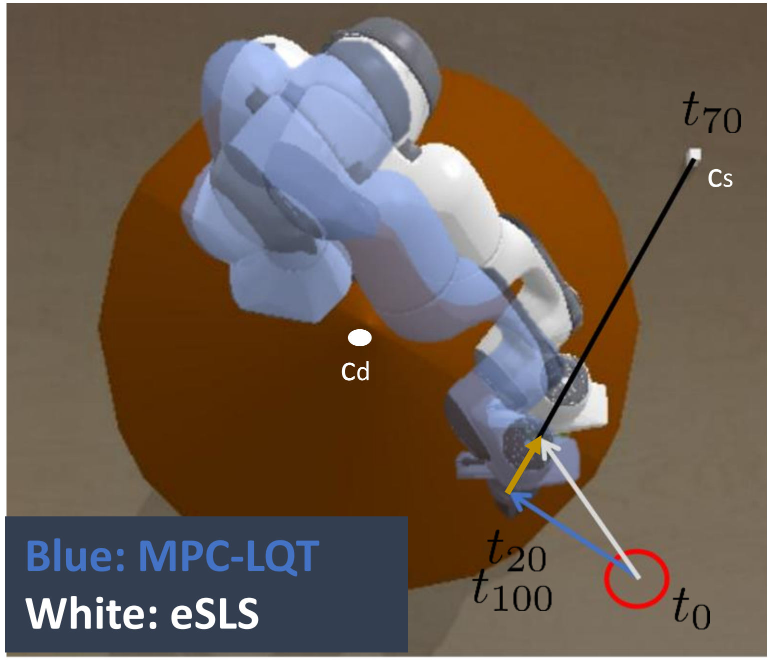
Cost is only passing through the viapoint (blue sphere) and correlating the first and the last timesteps. Thus, the controller achieves the task no matter what is the initial state (yellow spheres).



Simulation Study: Comparison eSLS to MPC-LQT







Geometrically: the length of {blue vector + yellow vector} > the length of white vector, meaning that the path is longer.

Numerically: the optimal control costs are:

eSLS : 474.3 ± 204.3

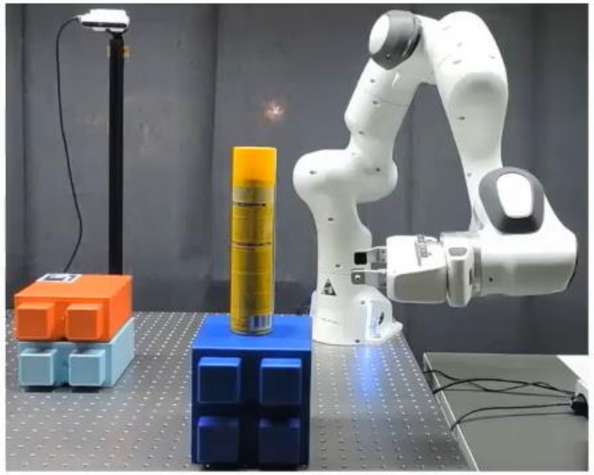
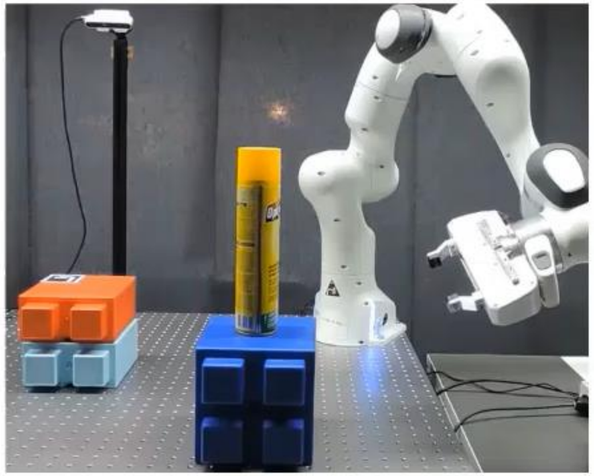
MPC-LQT : 1004.3 ± 464.2

$$c = \underbrace{\|\boldsymbol{x}_{20} - \boldsymbol{c}_d\|^2_{\boldsymbol{Q}_{20}}}_{\text{Place the mug}} + \underbrace{\|\boldsymbol{x}_{70} - \boldsymbol{c}_s\|^2_{\boldsymbol{Q}_{70}}}_{\text{Pick the sugar cube}} + \underbrace{\|\boldsymbol{x}_{20} - \boldsymbol{x}_{100}\|^2_{\boldsymbol{Q}_{20,100}}}_{\text{Place the sugar cube at timestep 70 wherever you placed the mug at timestep 20}} + 0.01\|\boldsymbol{u}\|$$

$$\boldsymbol{Q}_{20} = \mathrm{diag}(10^3, 10^3, 10^5, 10^5, 10^5, 10^5),\ \boldsymbol{Q}_{70} = 10^5 \times \boldsymbol{I}\ \boldsymbol{Q}_{20,100} = \mathrm{diag}(10^5 \times \boldsymbol{I}_3, \boldsymbol{0}_3)$$

Experimental Results

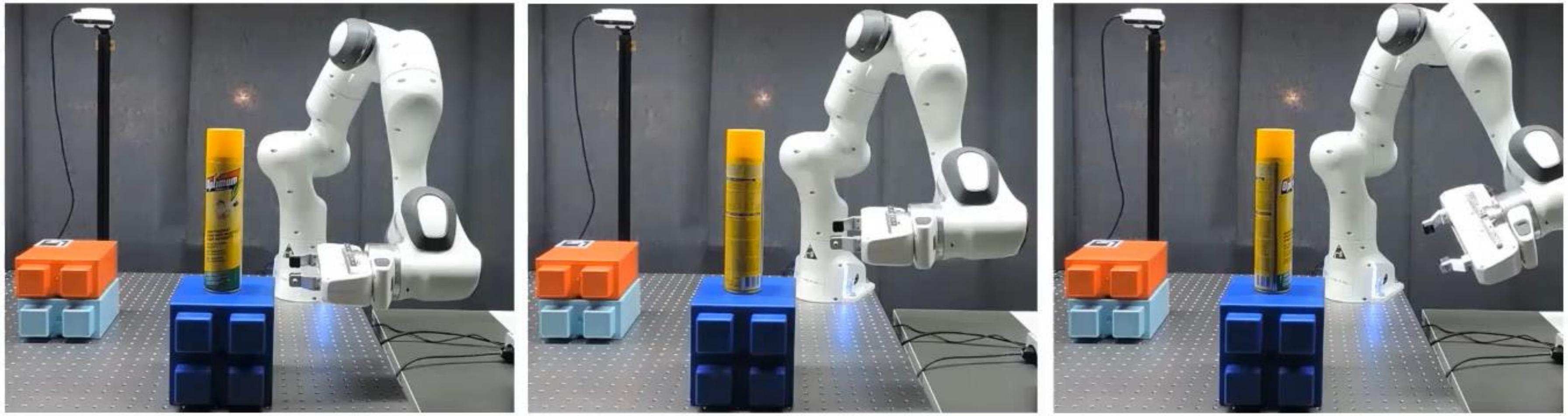
Memory feedback and **time-correlations**

Even a local controller can exploit memory feedback!

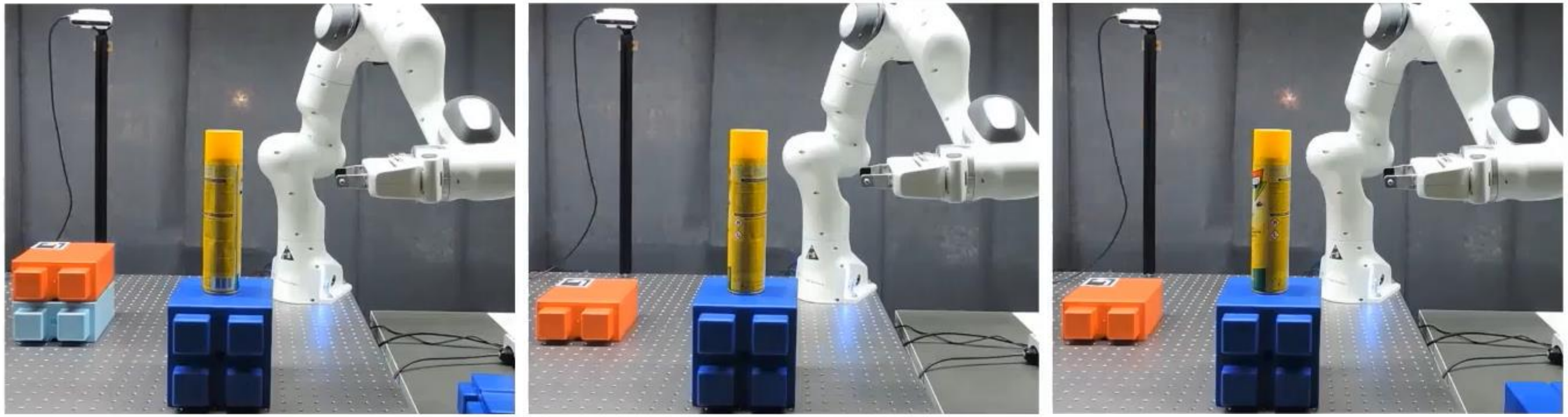
Experimental Results

Memory feedback and **time-correlations**



Even a local controller can exploit memory feedback!

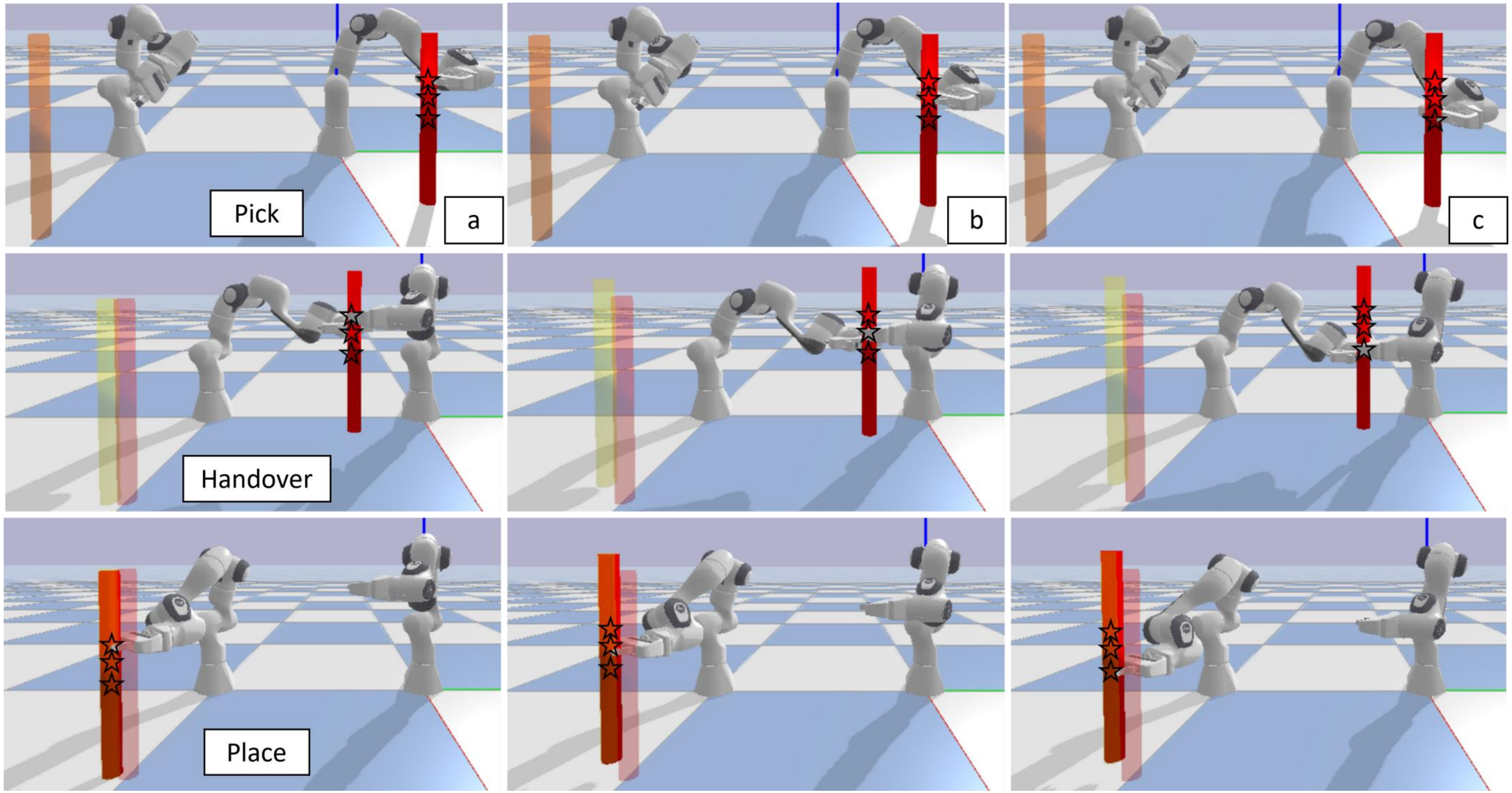
Fast adaptation to changing objectives



Local: without resolving the problem

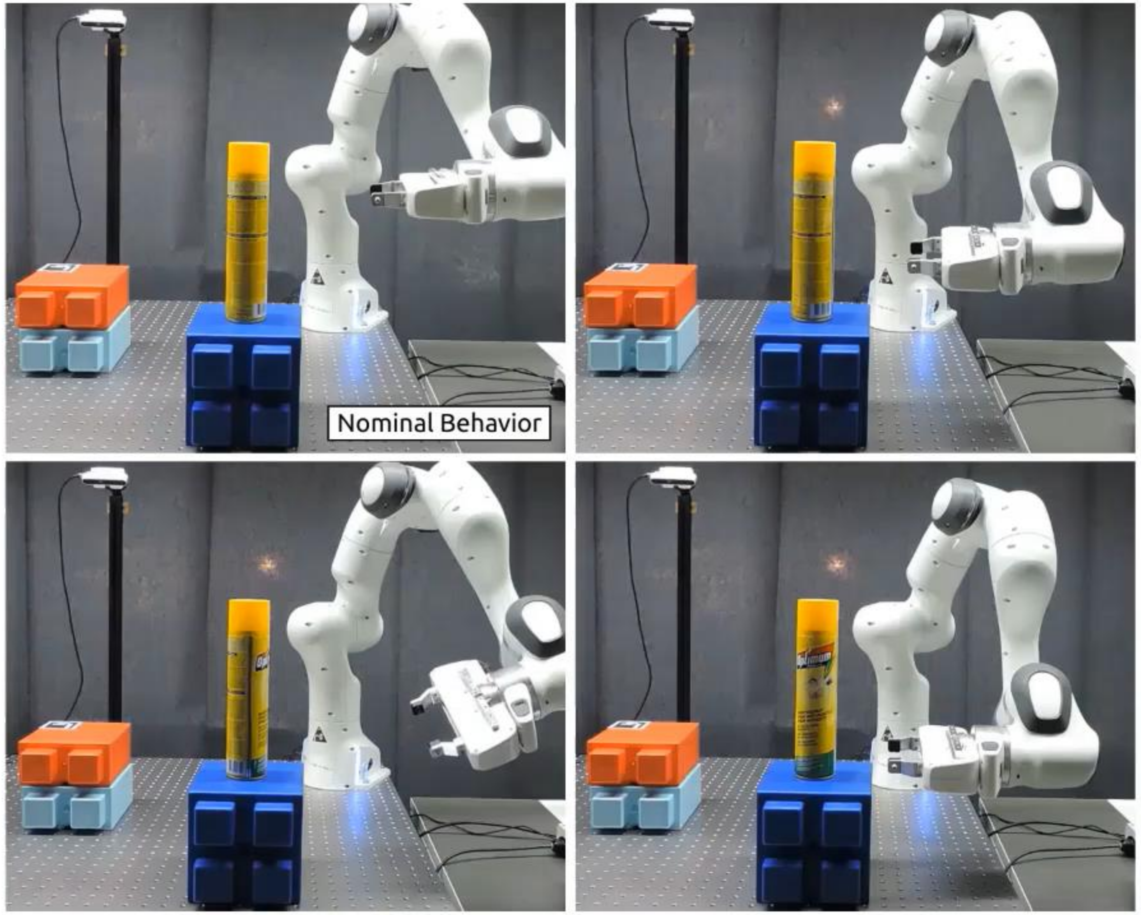
Fast: only a matrix-vector multiplication

Experimental Results: Bimanual Handover



Experimental Results: Bimanual Handover

Adapting to different initial configurations

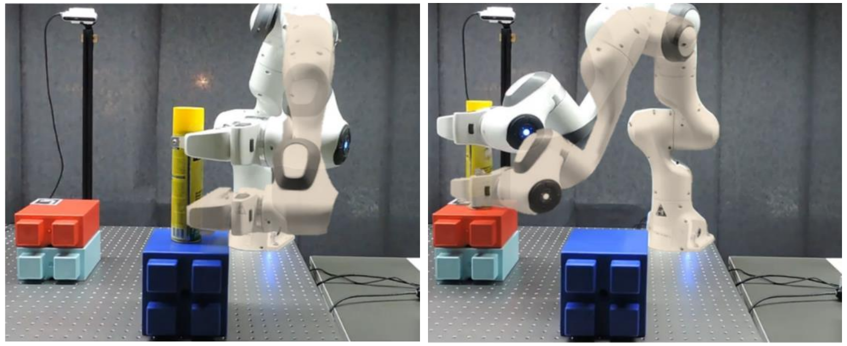


Discussion

$$u = Kx + k$$

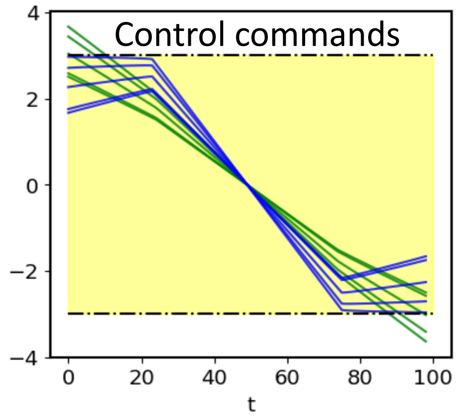
Memory feedback and **time-correlations**	**Fast adaptation** to changing objectives

- Learning time-correlations from demonstrations for **inverse optimal control**



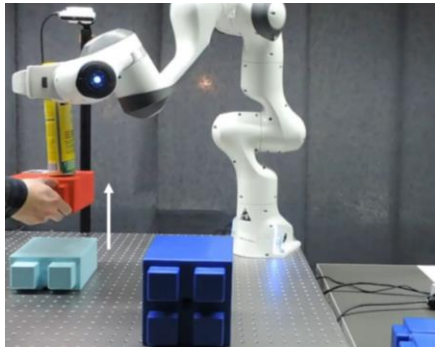
Time-correlations or independent variations?

- Increase the validity region of the local controller?

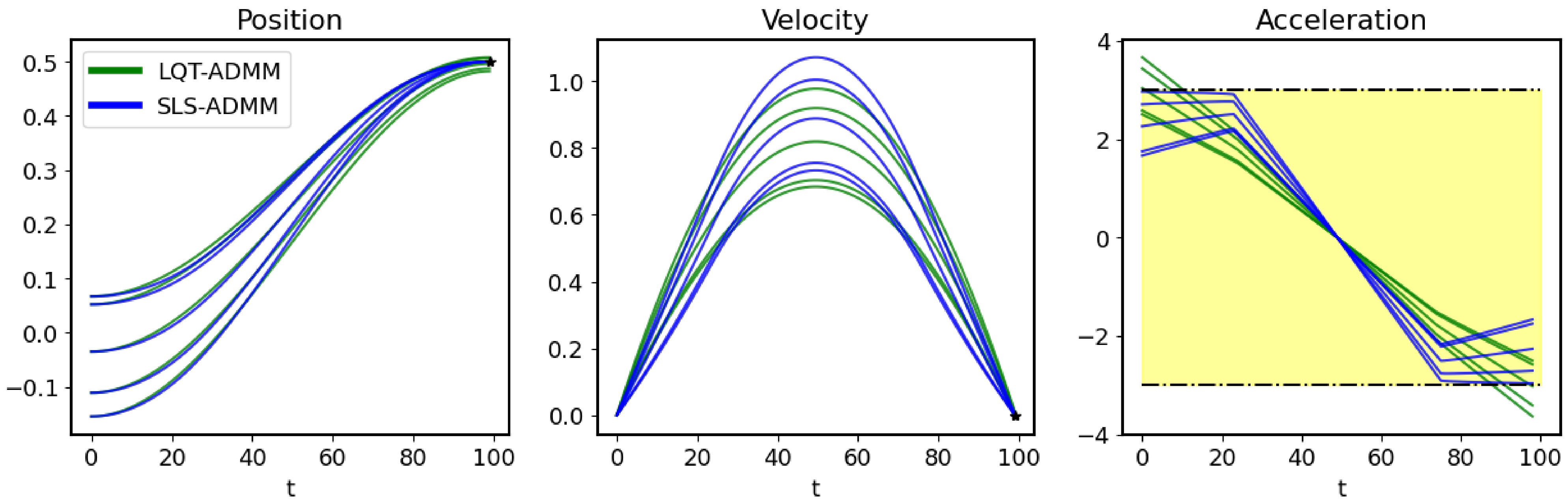


Robustly constrained iSLS

Warm-starting MPC?



Discussion: Robustly constrained SLS



Constraints: final position = 0.5, final velocity=0, -3 < acceleration < 3 **Dynamics:** double integrator

Acceleration bounds:

$$\boldsymbol{b}_u \leq \boldsymbol{u} \leq \boldsymbol{c}_u,$$

$$\boldsymbol{b}_u \leq \boldsymbol{\Phi}_u \boldsymbol{w} \leq \boldsymbol{c}_u,$$

Random variable

$$b_u^i \leq \boldsymbol{w}^\top \boldsymbol{\Phi}_u^i \leq c_u^i, \forall i$$

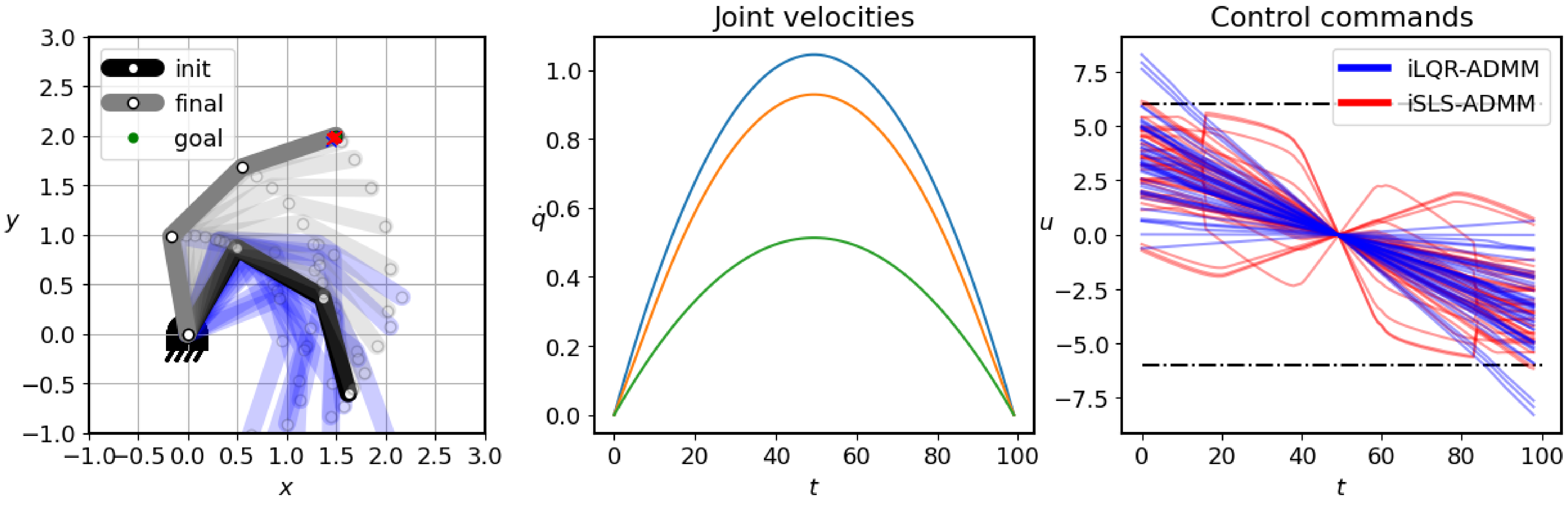
probability

Second order cone constraint

$$p(\boldsymbol{w}^\top \boldsymbol{\Phi}_u^i \leq c_u^i) \geq \eta \iff \boldsymbol{\mu}_{\boldsymbol{w}}^\top \boldsymbol{\Phi}_u^i + \Psi^{-1}(\eta) \|\boldsymbol{\Sigma}_{\boldsymbol{w}}^{1/2} \boldsymbol{\Phi}_u^i\|_2 \leq c_u^i$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix} \quad \boldsymbol{w} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{w}_0 \\ \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_{T-1} \end{bmatrix}$$

Discussion: Robustly constrained iSLS



Acceleration bounds:

$$\boldsymbol{b}_u \leq \boldsymbol{u} \leq \boldsymbol{c}_u,$$

$$\boldsymbol{b}_u \leq \boldsymbol{\Phi}_u \boldsymbol{w} \leq \boldsymbol{c}_u,$$

$$\boldsymbol{b}_u^i \leq \boldsymbol{w}^\top \boldsymbol{\Phi}_u^i \leq \boldsymbol{c}_u^i, \forall i$$

Random variable

probability

Second order cone constraint

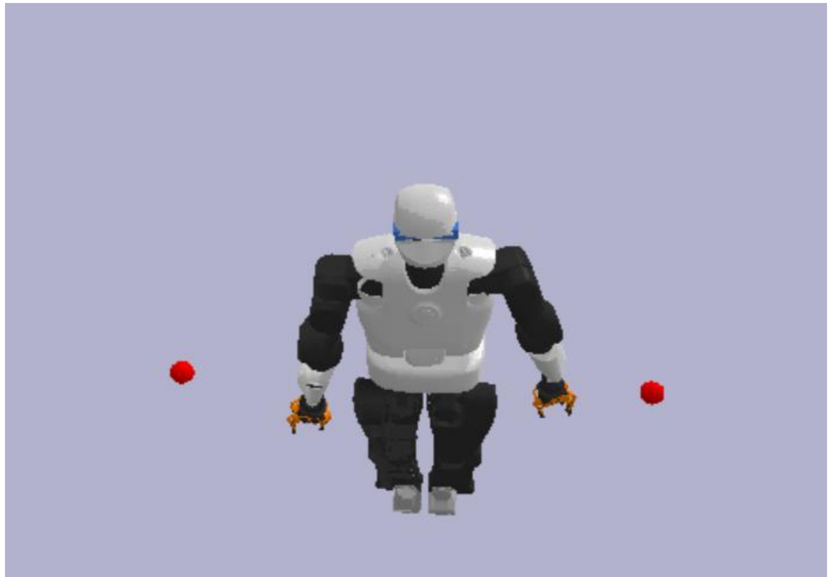
$$p(\boldsymbol{w}^\top \boldsymbol{\Phi}_u^i \leq \boldsymbol{c}_u^i) \geq \eta \iff \boldsymbol{\mu}_{\boldsymbol{w}}^\top \boldsymbol{\Phi}_u^i + \Psi^{-1}(\eta) \|\boldsymbol{\Sigma}_{\boldsymbol{w}}^{1/2} \boldsymbol{\Phi}_u^i\|_2 \leq \boldsymbol{c}_u^i$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_T \end{bmatrix}, \quad \boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_0 \\ \boldsymbol{u}_1 \\ \vdots \\ \boldsymbol{u}_T \end{bmatrix} \quad \boldsymbol{w} = \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{w}_0 \\ \boldsymbol{w}_1 \\ \vdots \\ \boldsymbol{w}_{T-1} \end{bmatrix}$$

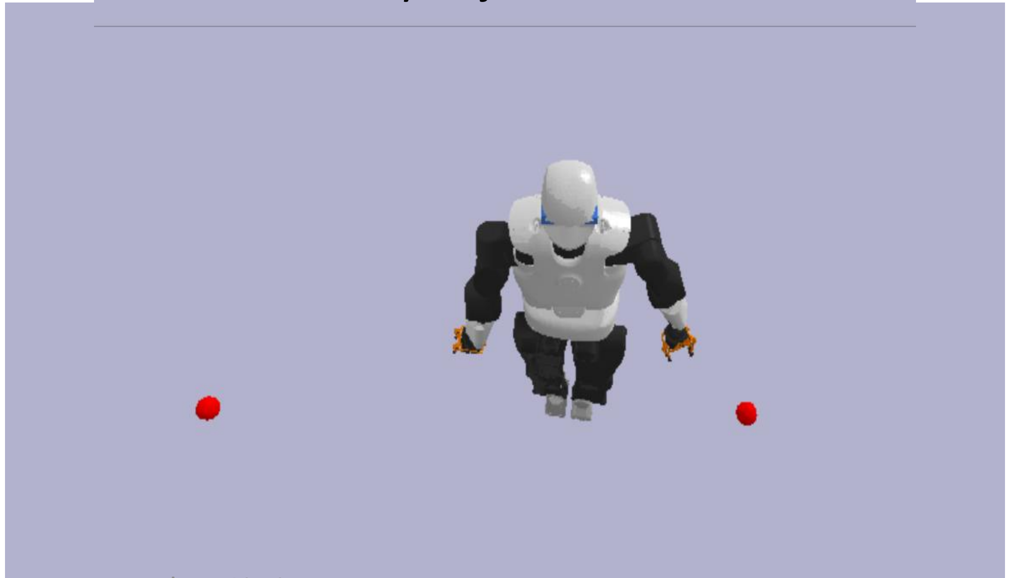
Nullspace in iSLS

- At each iteration of iSLS, we solve a problem of SLS, which can be exploited for the planning redundancy.
- If the nullspace exists, then it is only valid along the neighborhood of the nominal values.
- **Example:** Talos robot fixed base, bimanual operation.

Left hand reaching prioritized over right hand reaching.
Noisy dynamics simulations starting from different initial states.



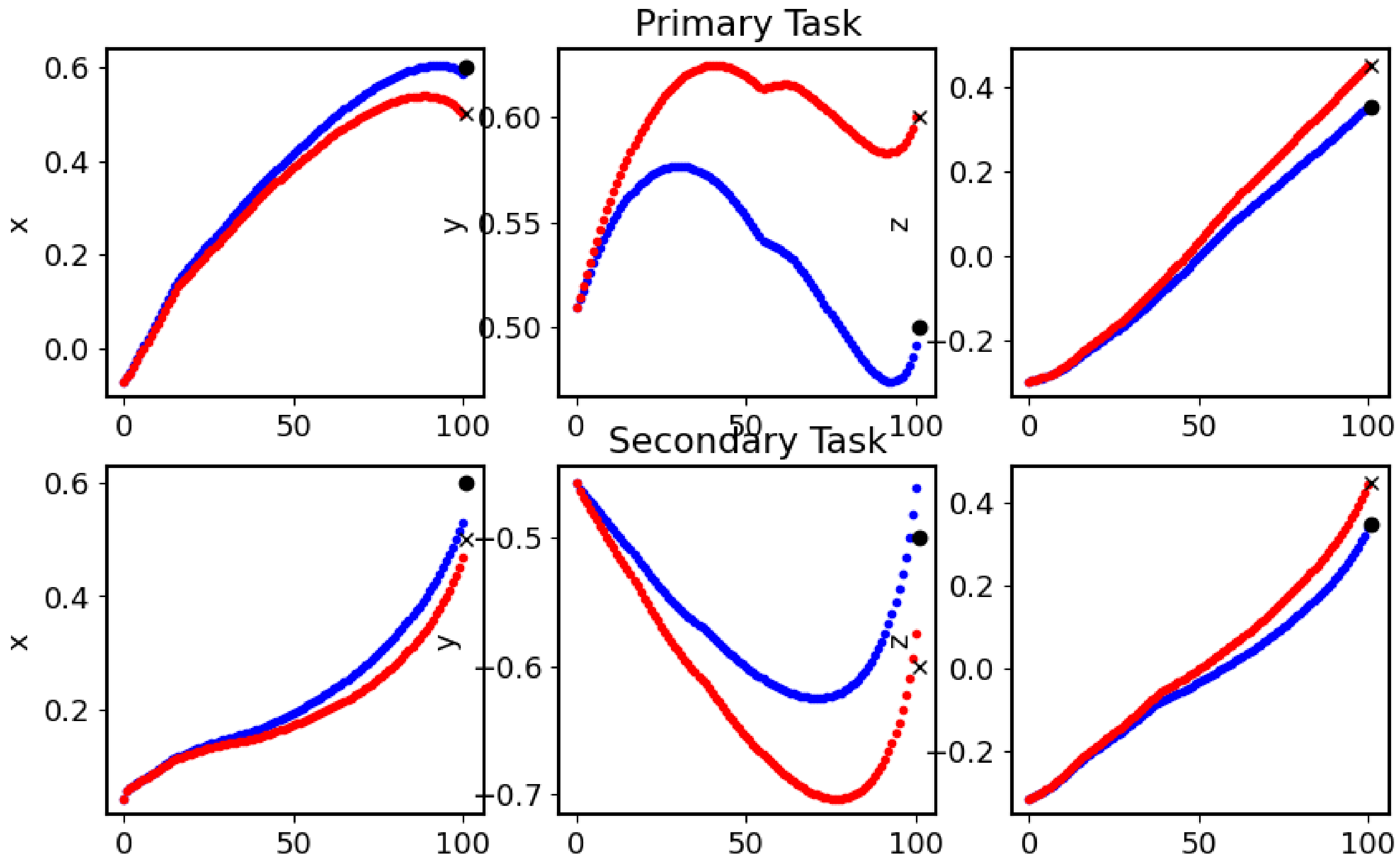
Secondary objective is reachable

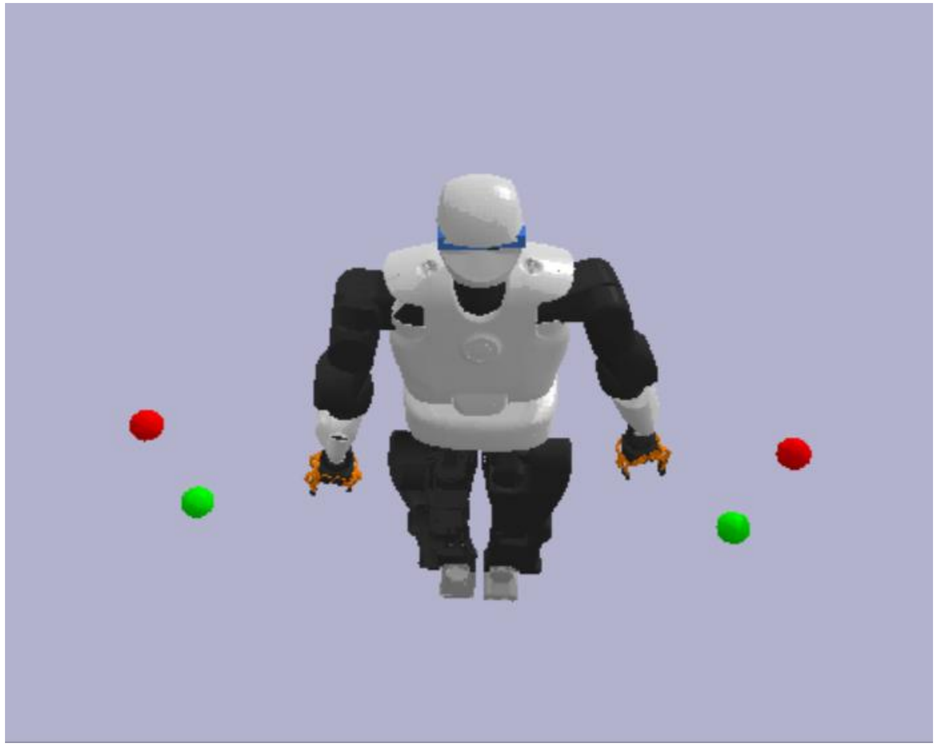


Secondary objective not reachable

Secondary point reachable – fast adaptation

First, we solved for hierarchical reaching for red points. (left hand > right hand)
No additional optimization step is required to achieve NEW green point reaching hierarchically.





Almost achieves the secondary task as well: showing that the fast adaptation makes the controller also aware of possible hierarchies in the task