On the Sample Complexity of Stabilizing LTI Systems on a Single Trajectory

Yang Hu[1], Adam Wierman[2], Guannan Qu[3]

[1] Harvard University [2] California Institute of Technology [3] Carnegie Mellon University

36th Conference on Neural Information Processing Systems

Problem Setting

We consider a noiseless LTI system

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

where:

- $x_t \in \mathbb{R}^n$ is the ***state***, and $u_t \in \mathbb{R}^m$ is the ***control input***.
- ***Dynamical matrices*** $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are unknown.
- The learner is allowed to learn about the system by interacting with it on a ***single trajectory***.
  - The initial state is sampled from the unit hyper-sphere surface in $\mathbb{R}^n$ uniformly at random.
  - At each time step t, the learner is allowed to observe x_t and freely determine u_t.
- The goal of the learner is to learn a ***stabilizing controller***.

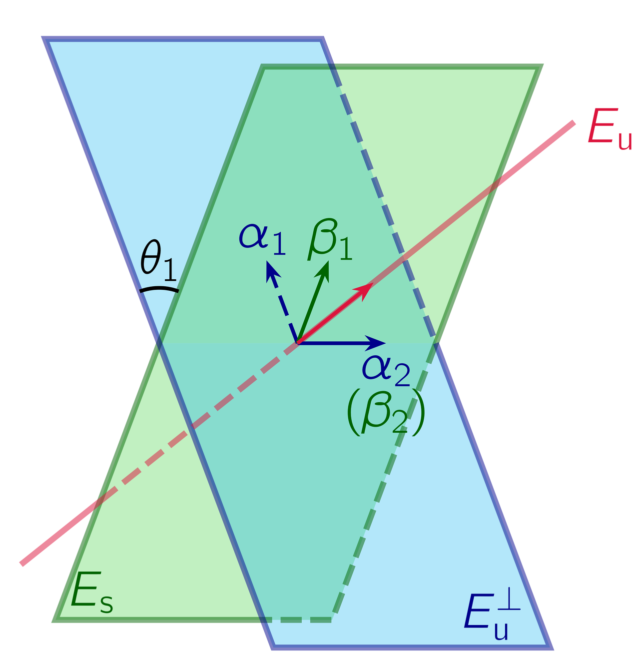
We make the following assumptions on the system.

Assumption 1 (Spectral Property). *A is diagonalizable with **instability index** k and distinct eigenvalues $\lambda_1, \cdots, \lambda_n$ satisfying $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_k| > 1 > |\lambda_{k+1}| \geq \cdots \geq |\lambda_n|$.*

Assumption 2 (Initialization). *The initial state of the system is sampled uniformly at random on the unit hyper-sphere surface in $\mathbb{R}^n$.*

Assumption 3 ((ν,σ)-Strong Controllability). *The system is **(ν,σ)-strongly controllable**; i.e., $\sigma_{\min}(C_\nu) > \sigma$, where $C_\nu := [A^{\nu-1}B \;\; A^{\nu-2}B \;\; \cdots \;\; AB \;\; B]$ is the ν-step controllability matrix.*

Subspace Decomposition



Under Assumption 1, define the subspaces:

- the ***eigenspace*** $E_i = \mathrm{span}(v_i)$ for an eigenvector v_i corresponding to λ_i, define
- the ***unstable subspace*** $E_u := \bigoplus_{i\leq k} E_i$;
- the ***stable subspace*** $E_s := \bigoplus_{i>k} E_i$.

Suppose the unstable subspace E_u and its orthogonal complement $E_u^\perp$ are spanned by ***orthonormal*** columns of $P_1 \in \mathbb{R}^{n\times k}$ and $P_2 \in \mathbb{R}^{n\times(n-k)}$, respectively, namely

$$E_u = \mathrm{col}(P_1),\; E_u^\perp = \mathrm{col}(P_2).$$

Write $P = [P_1 \; P_2]$ as a shorthand. Let $\Pi_1 := P_1P_1^\top$ and $\Pi_2 = P_2P_2^\top$ be the ***orthogonal projectors*** onto E_u and $E_u^\perp$, respectively.

Under such subspace decomposition, we shall also decompose A as

$$AP = P\begin{bmatrix} M_1 & \Delta \\ & M_2 \end{bmatrix} \Leftrightarrow M := \begin{bmatrix} M_1 & \Delta \\ & M_2 \end{bmatrix} = P^{-1}AP.$$

- The top-left block M_1 represents the action of A on the unstable subspace.
- ***This is the "small part" that leads to instability which we want to eliminate.***
- The matrices M_1 (and P_1), as compared to A, is much smaller in size and thus ***takes much fewer samples to learn***.

How to learn about the unstable subspace? A critical observation is that, when we recursively apply A to a state vector, the stable component of it will shrink, and the unstable component will stretch. Therefore, after letting the system run in open loop for sufficiently many heat-up steps, ***the state vector is automatically pushed to approach*** E_u (see the top right figure).

We propose a novel algorithm to learn to stabilize an unknown linear time-invariant (LTI) system on a single trajectory. It uses $O(k \log n)$ samples to stabilize a system with instability index k, which is sublinear in n when $k \ll n$.

Learn $E_u = \mathrm{col}(P_1)$.

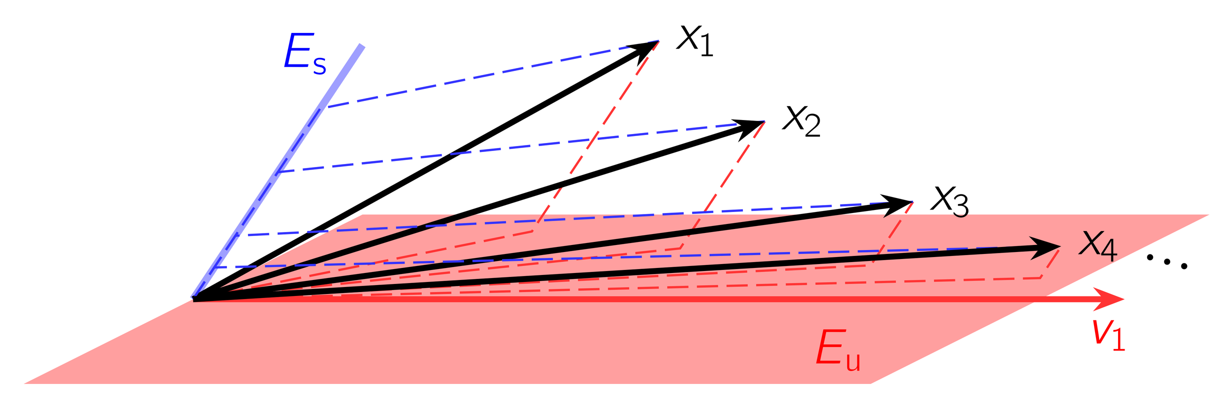
↓

Learn dynamics (A, B) "restricted" to E_u.

↓

Stabilize the "restricted" system every τ steps.

Scan the QR code to download the full paper



System state evolution in open loop.

Algorithm Design

Algorithm 1 LTS_0: Learning a τ-hop Stabilizing Controller

Stage 1: learn the unstable subspace of A.
Run the system in open loop for t_0 steps for initialization.
Run the system in open loop for k more steps and let $D \leftarrow [x_{t_0+1} \;\cdots\; x_{t_0+k}]$.
Calculate $\hat{\Pi}_1 \leftarrow D(D^\top D)^{-1}D^\top$.
Calculate the top k (normalized) eigenvectors $\hat{v}_1, \cdots \hat{v}_k$ of $\hat{\Pi}_1$, and let $\hat{P}_1 \leftarrow [\hat{v}_1 \;\cdots\; \hat{v}_k]$.
Stage 2: approximate M_1 on the unstable subspace.
Solve the least squares $\hat{M}_1 \leftarrow \arg\min_{M_1\in\mathbb{R}^{k\times k}} \mathcal{L}(M_1) := \sum_{t=t_0+1}^{t_0+k} \|\hat{P}_1^\top x_{t+1} - \hat{M}_1\hat{P}_1^\top x_t\|^2$.
Stage 3: restore B_τ for τ-hop control.
for $i \leftarrow 1, \cdots, k$ **do**
  Let the system run in open loop for ω time steps.
  Run for τ more steps with initial $u_{t_i} = \alpha\|x_{t_i}\|e_i$, where $t_i = t_0 + k + i\omega + (i-1)\tau$.
Let $\hat{B}_\tau \leftarrow [\hat{b}_1 \cdots \hat{b}_k]$, where the i^{th} column $\hat{b}_i \leftarrow \frac{1}{\alpha\|x_{t_i}\|}(\hat{P}_1^\top x_{t_i+\tau} - \hat{M}_1^\tau \hat{P}_1^\top x_{t_i})$.
Stage 4: construct a τ-hop stabilizing controller K.
Construct the τ-hop stabilizing controller $\hat{K} \leftarrow -\hat{B}_\tau^{-1}\hat{M}_1^\tau\hat{P}_1^\top$.

τ-hop control. The intuition of stabilizing the unstable component only holds when the state is close enough to the unstable subspace. Luckily, the stable component automatically vanishes over time, so we design the controller to eliminate the unstable component ***only "once in a while"***.

The closed-loop dynamics with the τ-hop controller can be written as

$$\tilde{y}_{s+1} = \begin{bmatrix} M_1^\tau + P_1^\top A^{\tau-1}B\hat{K}_1\hat{P}_1^\top P_1 & \Delta_\tau + P_1^\top A^{\tau-1}B\hat{K}_1\hat{P}_1^\top P_2 \\ P_2^\top A^{\tau-1}B\hat{K}_1\hat{P}_1^\top P_1 & M_2^\tau + P_2^\top A^{\tau-1}B\hat{K}_1\hat{P}_1^\top P_2 \end{bmatrix} \begin{bmatrix} \tilde{y}_{1,s} \\ \tilde{y}_{2,s} \end{bmatrix}.$$

Performance of the Algorithm

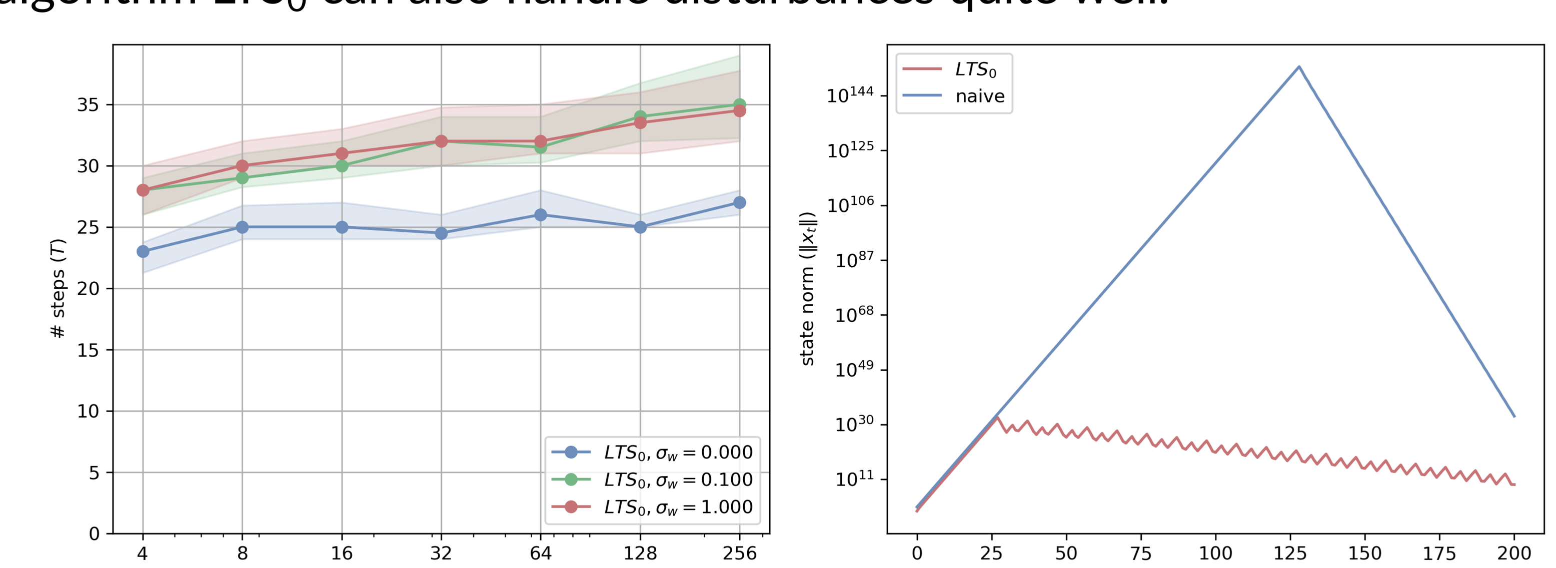
Theoretical guarantee. We show the following Main Theorem to provide theoretical performance guarantee for LTS_0 in noiseless LTI systems.

Theorem 1 (Main Theorem). *Given a noiseless LTI system $x_{t+1} = Ax_t + Bu_t$ subject to Assumptions 1, 2 and 3, and additionally $|\lambda_1|^2|\lambda_{k+1}| < |\lambda_k|$, by running LTS_0 with parameters*

$$\tau = O(1),\; \omega = O(\ell \log k),\; \alpha = O(1),\; t_0 = O(k \log n)$$

that terminates within $t_0 + k(1+\omega+\tau) = O(k \log n)$ time steps, the closed-loop system is exponentially stable with probability $1 - O(k^{-\ell})$ over the initialization of x_0 for any $\ell \in \mathbb{N}$. Here the big-O notation only shows dependence on k and n and hides dependency on instance-specific parameters.

Experimental results. Though for clarity of exposition our Main Theorem does not contain disturbances, we show by numerical experiments that our algorithm LTS_0 can also handle disturbances quite well.



(a) Running steps of LTS_0 (b) State norms along one trajectory