Online Learning for Control Systems

Athindran Ramesh Kumar

Princeton University
Advisor: Prof. Peter Ramadge
arkumar@princeton.edu

April 10, 2021

Overview

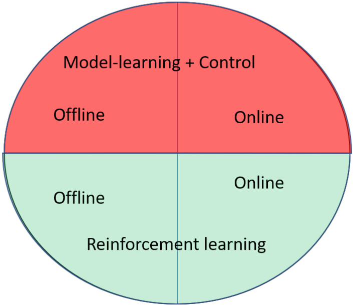
Introduction

Learning and Control - Different Paradigms

Traditional Approach

1. Obtain a model and nominal parameters from first principles
2. Use control design methods

Learning + Control



When do we need learning in control?

Inadequate first-principles model

- Parameter estimates inaccurate
- Drift in system parameters
- Unmodeled dynamics - Common in non-rigid bodies
- Changes in the system

Possible solutions

- A - Robust control with inaccurate model - too conservative
- B - Offline model learning + Control - System Identification
- C - Online model learning + Control
- D - Reinforcement learning - Directly learn a control law

B - Offline model learning and Control

- Perturb the system with informative signals and identify parameters
- Extensively studied for linear systems[1]
- Non-linear system identification studied more recently[2]
- Similar techniques currently being explored in model-based RL

Recent Important Progress

1. End-to-end guarantees for learning+LQR[3]
2. Practical advances in model-based RL for controlling robotic systems

[1] Lennart Ljung (2001). "System identification". In: *Wiley Encyclopedia of Electrical and Electronics Engineering*.

[2] Johan Schoukens and Lennart Ljung (2019). "Nonlinear System Identification: A User-Oriented Road Map". In: *IEEE Control Systems Magazine* 39.6, pp. 28–99.

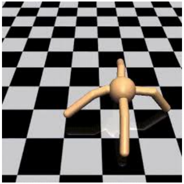
[3] Sarah Dean et al. (2019). "On the sample complexity of the linear quadratic regulator". In: *Foundations of Computational Mathematics*, pp. 1–47.

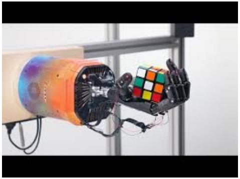
D - Reinforcement Learning

- Directly learn to control by parametrizing the policy or value function
- Initially - model-free. Models coming into practice now.

Recent Important Progress

- Policy optimization for LQR and mixed H-2/H-inf control[4],[5]





4 Maryam Fazel et al. (2018). "Global convergence of policy gradient methods for the linear quadratic regulator". In: *arXiv preprint arXiv:1801.05039*.

5 Kaiqing Zhang, Bin Hu, and Tamer Basar (2019). "Policy optimization for H2 linear control with Hinf robustness guarantee: Implicit regularization and global convergence". In: *arXiv preprint arXiv:1910.09496*.

C - Online Model Learning and Control

- Refine parameters of the model online
- Update control strategy on the refined model

Recent Important Progress

- Regret bound for online prediction using spectral filtering[6]
- Regret bound for online control with adversarial robustness[7]
- Boosting for learning control systems[8]
- Control with learning on the fly[9]

---

[6] Elad Hazan, Holden Lee, et al. (2018). "Spectral filtering for general linear dynamical systems". In: *Advances in NeurIPS*, pp. 4634–4643.

[7] Naman Agarwal, Brian Bullins, et al. (2019). "Online control with adversarial disturbances". In: *arXiv preprint arXiv:1902.08721*.

[8] Naman Agarwal, Nataly Brukhim, et al. (2019). "Boosting for Dynamical Systems". In: *arXiv preprint arXiv:1906.08720*.

[9] Charlie Fefferman et al. (2019). "Control with Learning on the Fly: First Toy Problems". Seminar in ORFE, Princeton University.

Role of Models

Model - description of the input-output behavior of the system

Advantages of Models

- More sample efficient learning
- Safer - consequence of sample efficiency
- Can incorporate prior information

Disadvantages of Models

- Inaccurate model - hinder exploring and finding better global strategies

Role of Models

Linear Models

- Play a big role in control theory
- Local linearization
- Simple testbed for new methods and to prove guarantees

Power of machine learning

- Non-linearities play a big role, e.g Neural Networks
- Kernels and Feature maps incorporate prior information

Can machine learning provide a principled method of dealing with difficult to model systems?

Online Learning for Model Identification

Learning Linear State Space Models

$$x_{t+1} = \mathbf{A}x_t + \mathbf{B}u_t + w_t \quad (1)$$

$$y_t = \mathbf{C}x_t + \mathbf{D}u_t + n_t \quad (2)$$

- $x_t \in \mathbb{R}^n$ - state
- $u_t \in \mathbb{R}^m$ - input
- $y_t \in \mathbb{R}^k$ - output

- $w_t \in \mathbb{R}^n$ - disturbance
- $n_t \in \mathbb{R}^k$ - noise

$\mathbf{A} \in \mathbb{R}^{n\times n}$,$\mathbf{B} \in \mathbb{R}^{n\times m}$, $\mathbf{C} \in \mathbb{R}^{k\times n}$, $\mathbf{D} \in \mathbb{R}^{k\times m}$
The system is stable if $\rho(\mathbf{A}) < 1$

Method 1 - EM algorithm

- Originally invented by Dempster,Laird and Rubin in 1977
- First applied to linear systems by Shumway and Stoffer 1982
- Most complete version of the method discussed in 1996[10]

Pros

- Very efficient and easy to implement
- E step and M step are individually optimal in some sense

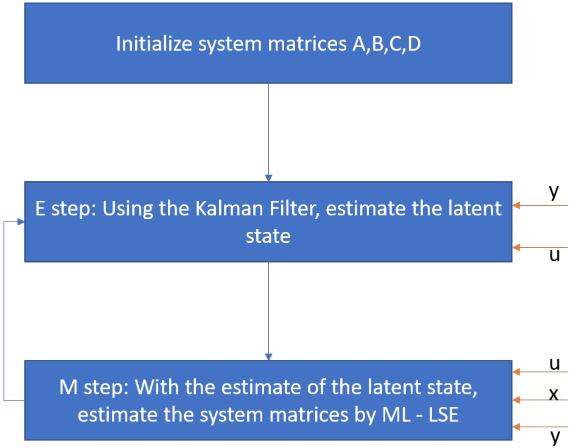
Cons

- Both steps together will probably converge to a local optimum

[10] Zoubin Ghahramani and Geoffrey E Hinton (1996). *Parameter estimation for linear dynamical systems.* Tech. rep. Technical Report CRG-TR-96-2, University of Toronto, Dept. of Computer Science.

Method 1 - EM algorithm

Method for learning the state-space model directly



Method 2 - Subspace identification

- Started out in the 1960's.
- Pioneered by Van Overschee, De moor, Verhagen in the late 1980's.
- Robust SSID algorithm[11] - culmination of all the ideas
- Naive implementations do not work well
- Pros: Works well if implemented with all bells and whistles
- Cons: Batch algorithm, complicated, computationally expensive

[11] Peter Van Overschee and BL De Moor (2012). *Subspace identification for linear systems: Theory—Implementation—Applications.* Springer Science & Business Media.

Method 2 - Subspace identification

$$y_t^r = \begin{bmatrix} y_{t-r} \\ y_{t-r+1} \\ \vdots \\ y_t \end{bmatrix} \quad u_t^r = \begin{bmatrix} u_{t-r} \\ u_{t-r+1} \\ \vdots \\ u_t \end{bmatrix}$$

$$\begin{aligned} \mathbf{Y} &= \begin{bmatrix} y_1^r & y_2^r & \dots & y_N^r \end{bmatrix} \\ \mathbf{U} &= \begin{bmatrix} u_1^r & u_2^r & \dots & u_N^r \end{bmatrix} \\ \mathbf{X} &= \begin{bmatrix} x_1 & x_2 & \dots & x_N \end{bmatrix} \end{aligned}$$

The following relationship holds with $\mathcal{O}_r$ - extended observability matrix,

$$\mathbf{Y} = \mathcal{O}_r \mathbf{X} + S_r \mathbf{U} + \underbrace{\mathbf{V}}_{\text{Noise terms}} \tag{3}$$

$$\mathcal{O}_r = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{r-1} \end{bmatrix} \qquad S_r = \begin{bmatrix} D & 0 & \dots & 0 \\ CB & D & \dots & 0 \\ \vdots & \vdots & & \vdots \\ CA^{r-2}B & CA^{r-3}B & \dots & CB \end{bmatrix}$$

Method 2 - Subspace identification

Define $\mathbb{P}^{\perp}_{\mathbf{U}^T} = \mathbf{I} - \mathbf{U^T}\left(\mathbf{UU^T}\right)^{-\mathbf{1}}\mathbf{U}$, a projection operator

1. Form $G = \frac{1}{N}\mathbf{Y}\mathbb{P}^{\perp}_{\mathbf{U}^T}\Phi^T$
2. Select W_1 and W_2,form $\hat{G}$,then perform SVD

$$\hat{G} = W_1 G W_2 = USV^T \approx U_d S_d V_d^T \tag{4}$$

   Different choices of W_1 and W_2 - MOESP,N4SID,IVM,CVA
3. Select R - an arbitrary full rank matrix and form the observability matrix $\mathcal{O}_R = W_1^{-1} U_d R$
4. Estimate $\hat{A}$, $\hat{C}$ from the observability matrix
5. Estimate $\hat{B}$, $\hat{D}$ by linear regression

Method 3 - EKF with augmented state

1. Kalman filter invented in 1960[12]
2. Define hyperstate - unknown system matrices included in the state
3. Estimate both the state and the system matrices using EKF

$$
\begin{aligned}
x_{t+1} &= A_t x_t + B_t u_t + w_t \\
A_{t+1} &= A_t + n_{A,t} \\
B_{t+1} &= B_t + n_{B,t} \\
C_{t+1} &= C_t + n_{C,t} \\
y_t &= C_t x_t
\end{aligned}
$$

Advantage: Can get uncertainty estimates. Disadvantage: Does not work very well for systems that are not fully observable.

[12] Rudolph Kalman (1960). "E. 1960. A new approach to linear filtering and prediction problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82, pp. 35–45.

Learning Input-Output models

- Learn a mapping from input to output without modeling the state

$$\underbrace{\det\left(z\mathbf{I}-\mathbf{A}\right)}_{\text{degree n polynomial}} \; y = \underbrace{\mathbf{C}\mathrm{adj}\left(z\mathbf{I}-\mathbf{A}\right)\mathbf{B}}_{\text{matrix of polynomials each of degree at most n-1}} \; u \qquad (5)$$

Hints towards an autoregressive prediction model.
$\beta_0, \beta_1, \ldots \beta_n$ - coefficients of the characteristic polynomial.

$$\begin{aligned}\mathbf{C}\mathrm{adj}(z\mathbf{I}-\mathbf{A})\mathbf{B}u &= \mathbf{C}\det(z\mathbf{I}-\mathbf{A})\left(z\mathbf{I}-\mathbf{A}\right)^{-1}\mathbf{B}u \\ &= \mathbf{C}\left(\sum_{i=0}^{n}\beta_i z^i\right) z^{-1}\left(\sum_{j=0}^{\infty}\mathbf{A}^j\mathbf{z}^{-j}\right)\mathbf{B}u\end{aligned}$$

Method 1 - ARX models

Let $p = i - j$.

$$\mathbf{C}\mathrm{adj}(\mathbf{zI} - \mathbf{A})\mathbf{B}u = \sum_{p=1}^{n}\sum_{i=p}^{n} \mathbf{C}\beta_i \mathbf{A}^{i-p}\mathbf{B}\mathbf{z}^{p-1}u + \sum_{p=-\infty}^{0} A^{-p}\mathbf{C}\underbrace{\sum_{i=0}^{n}\beta_i \mathbf{A}^i}_{0}\mathbf{B}\mathbf{z}^{p-1}u$$

To conclude,

$$\sum_{j=0}^{n} \beta_j y_{t+j} = \sum_{p=1}^{n} \mathbf{P}_p u_{t+p-1} \tag{6}$$

ARX model - coefficients can be learnt using least squares
Input-output description without the state.

Method 2 - Spectral Filtering

- Recent work by Hazan et al.[13]
- Input output mapping decomposed into a projection onto a space spanned by the eigenvectors of a particular Hankel matrix
- Eigenvectors are called "wave filters"
- Prove regret bounds in the online case for identification
- Translates to generalization bounds in the batch case
- Main result of prior work - asymptotic consistency

[13] Elad Hazan, Karan Singh, and Cyril Zhang (2017). "Learning linear dynamical systems via spectral filtering". In: *Advances in NeurIPS*, pp. 6702–6712.

Method 2 - Spectral Filtering

ARX model - β_j are chosen to be the coefficients of the characteristic polynomial,

$$\sum_{i=0}^{n} \beta_i y_{t-i} = \sum_{j=0}^{n-1} \mathbf{P_j} u_{t-j} \tag{7}$$

In spectral filtering, choose β_j to be coefficients of the polynomial with roots given by the phases of the eigenvalues of $\mathbf{A}$
Define approximation error

$$\delta_t = \sum_{i=0}^{n} \beta_i y_{t-i} - \sum_{j=0}^{n-1} \mathbf{P_j} u_{t-j} \tag{8}$$

Method 2 - Spectral Filtering

For $\mathbf{a} \in \mathbb{R}^T$, define

$$
\begin{aligned}
a^{(\omega)} &:= (a_j \omega^j)_{1 \le j \le T} \\
a^{(\cos,\theta)} &:= (a_j \cos(j\theta))_{1 \le j \le T} \\
a^{(\sin,\theta)} &:= (a_j \sin(j\theta))_{1 \le j \le T}
\end{aligned}
$$

δ_t can be well approximated using the wave filters

$$
\begin{aligned}
\delta_t \approx \sum_{w=1}^{W} \sum_{h=1}^{k} & M(w,h,:,:)\sigma_h^{\frac{1}{4}} \left(\phi_{sf,h}^{(\cos,2\pi\frac{w}{W})} \circledast u\right) \\
&+ N(w,h,:,:)\sigma_h^{\frac{1}{4}} \left(\phi_{sf,h}^{(\sin,2\pi\frac{w}{W})} \circledast u\right) \qquad (9)
\end{aligned}
$$

Numerical Comparison

Experimental Setup

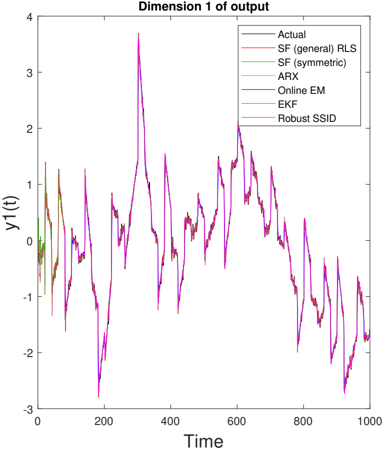
- System is **time-invariant**
- Experiment 1 - Toy system fully observable $m = 1, n = 3, k = 3$
- Experiment 2,3 - $m = 3, n = 10, k = 5$
- $\mathbf{B}, \mathbf{C}$ iid Gaussian
- Inputs block gaussian signals and gaussian random noise
- Signal level 0.5, Noise level 0.05.

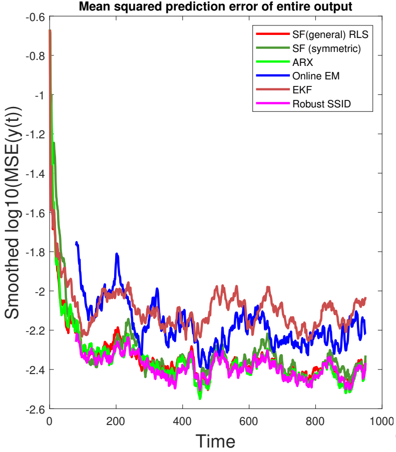
Metrics

- Prediction error
- Runtime

Experiment 1

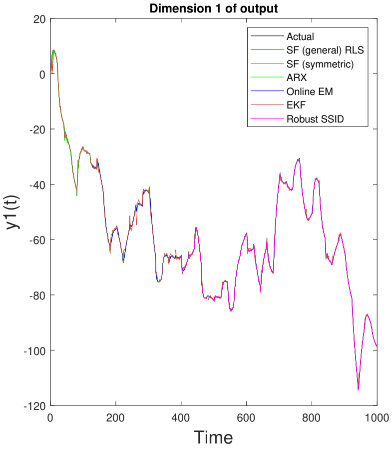
Experiment 1 - Simple toy system. 3-dimensional single input fully observable. Everything works!

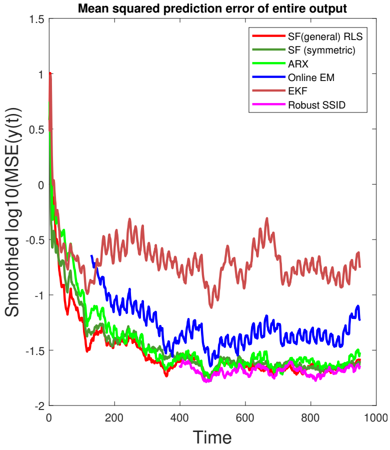




Experiment 2

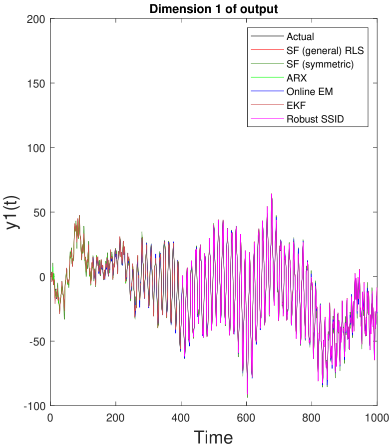
Experiment 2 - $\mathbf{A} = \mathrm{diag}([0.1, 0.2, \ldots, 0.99])$, $m = 3$, $n = 10$, $k = 5$,

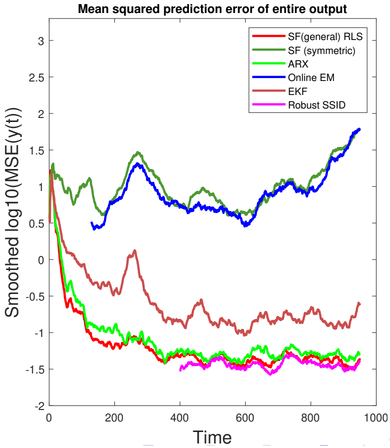




Experiment 3

Experiment 3 - **A** block diagonal matrix with 5 rotation matrices
$m = 3, \; n = 10, \; k = 5$





Runtime and Model Order Comparison

- Runtime and MSE for experiment 3 - true system order 10

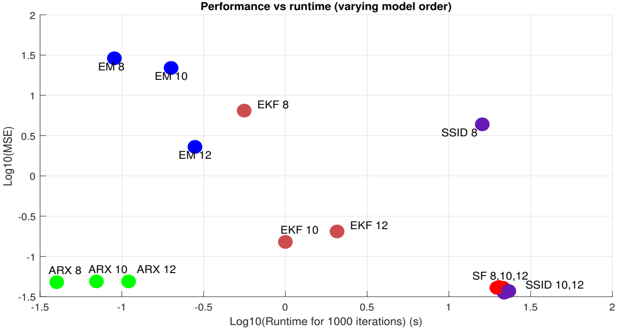


Figure: Performance vs Runtime comparison with different model orders

Conclusions from experiments

Trend similar over multiple seeds and system parameters.

Which optimization algorithm to use?

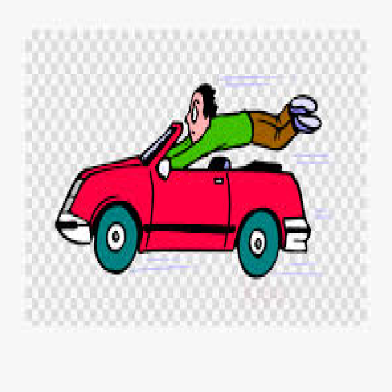
- Small problems: RLS
- Large problems: Only option GD with manually tuned step-size

Which identification algorithm to use?

- ARX - very efficient and sufficiently accurate for most problems
- High accuracy - SSID for small problems and SF for large problems
- Input-output models if system order unknown
- EKF - State-space control design methods available
- EKF - Estimate of uncertainty useful for robust control

Online Control with Models

How do we learn to drive?



Play it safe until we understand how the car behaves.

- Start with a conservative controller
- Transition to a aggressive controller - based on current model uncertainty
- Will a convex combination of controllers work?
- Tune the weights

Convex combination of controllers

For this section, assume state is fully measurable

Bad news

- Spectral radius non-convex non-smooth
- Stability not guaranteed

Good news - still a lot of structure in the problem for SISO systems

$$\begin{aligned} K_3 &= \alpha K_1 + (1-\alpha)K_2 \\ L_3(i\omega) &= K_3(i\omega I - A)^{-1}B \\ &= \alpha L_1 + (1-\alpha)L_2 \qquad (10) \\ p_3(z) &= \det(zI - A) + K_3 \mathrm{adj}(zI - A)B \\ &= \alpha p_1(z) + (1-\alpha)p_2(z) \\ &= \alpha p_1(z)\left(1 + \frac{1-\alpha}{\alpha}\frac{p_2(z)}{p_1(z)}\right) \qquad (11) \end{aligned}$$

Root Locus and Nyquist Plot

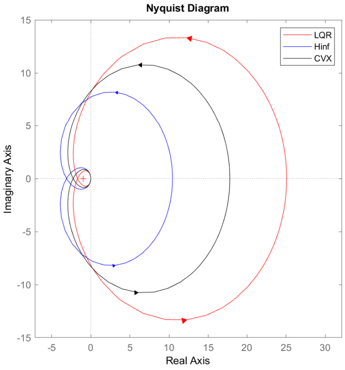


Figure: Nyquist plot of CVX control

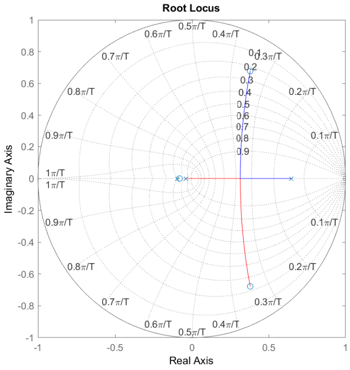


Figure: Root locus plot varying α

Stability of time-varying convex combination

Controller switching

Dwell time - maintain stability switching between stabilizing controllers[14]

Relevant literature on convex combination

- Conditions for stability of convex polytope of polynomials[15]
- Condition for schur stability of convex polytope of polynomials[16]

Our approach

Gradient-based to constrain the controllers to be stabilizing

[14] José C Geromel and Patrizio Colaneri (2006). "Stability and stabilization of discrete time switched systems". In: *International Journal of Control* 79.07, pp. 719–728.

[15] Stanisław Białas (2004). "A necessary and sufficient condition for stability of the convex combination of polynomials". In: *Control and Cybernetics* 33.4, pp. 589–597.

[16] Juergen E Ackermann and B Ross Barmish (1988). "Robust Schur stability of a polytope of polynomials". In: *IEEE transactions on automatic control* 33.10, pp. 984–986.

Our Method

- A_t, B_t be the system at time t
- $K_t = (1-\alpha_t)K_1 + \alpha_t K_2$
- Estimates A_{et}, B_{et} with $\triangle_{A,t}, \triangle_{B,t}$ the errors in the estimates
- v_K, u_K eigenvectors of $(A_{et} - B_{et}K_{t-1})^T$ and $A_{et} - B_{et}K_{t-1}$ for the eigenvalue with the maximum radius

$$
\begin{aligned}
\rho_t &\approx \left|\lambda_1(A_{et} - B_{et}K_{t-1}) + D_A(\lambda_1)[\triangle_{At}] + D_B(\lambda_1)[\triangle_{Bt}] + \frac{d\lambda_1}{d\alpha}(\triangle\alpha_t)\right| \\
&\approx \left|\lambda_1(A_{et} - B_{et}K_{t-1}) + \frac{v_K^H \triangle_{At} u_K}{v_K^H u_K} + \frac{v_K^H \triangle_{Bt} K_{t-1} u_K}{v_K^H u_K} \right. \\
&\left. + \frac{v_K^H B_{et}(K_2 - K_1)u_K}{v_K^H u_K}(\alpha_t - \alpha_{t-1})\right| \\
&\lesssim \rho_{t-1} + s_K\|\triangle_{A,t}\|_2 + s_K\|K_{t-1}\|_2\|\triangle_{B,t}\|_2 + s_\alpha(\alpha_t - \alpha_{t-1})
\end{aligned}
$$

Our Method

Therefore,

$$\rho_t \leq \rho_{t-1} + s_K \|\triangle_{A,t}\|_2 + s_K \|K_{t-1}\|_2 \|\triangle_{B,t}\|_2 + s_\alpha(\alpha_t - \alpha_{t-1}) \qquad (12)$$

where $s_K = \frac{\|v_K^H\| \|u_K\|}{|v_K^H u_K|}$, $s_\alpha = \mathrm{Re}\left(\frac{\bar{\lambda}_K}{|\lambda_K|} \frac{v_K^H B_{et}(K_2 - K_1) u_K}{v_K^H u_K}\right)$

Let $\|\triangle_{A,t}\|_2 \leq \delta_{At}$ and $\|\triangle_{B,t}\|_2 \leq \delta_{Bt}$

- Compute an aggressive controller K_1 (LQR) and a robust controller K_2 (H^∞) at a lower frequency
- At each time perform the following update (η_t - learning rate):

$$\begin{aligned} \rho_c &= \rho_{t-1} + s_K \delta_{At} + s_K \|K_{t-1}\| \delta_{Bt} \\ \alpha_t &= \alpha_{t-1} + \eta_t s_\alpha (\rho_d - \rho_c) \\ K_t &= (1 - \alpha_t) \times K_1 + \alpha_t \times K_2 \end{aligned}$$

Experiments

All the system parameters chosen randomly

Experiment 1 - $n = 3$ $m = 1$ $k = 1$

- Artificially drift A_{et} from A_0 to A where $\|A_0 - A\| = 0.6$ slowly

Experiment 2 - $n = 3,\ m = 1,\ k = 1$

- Introduce learning and use $\|A_{e0} - A_0\| = 0.7$
- A_t is drifting slowly over time

Experiment 3 - $n = 4,\ m = 2,\ k = 3$

- Introduce learning and use $\|A_{e0} - A_0\| = 0.6$
- A_t is drifting slowly over time

Experiment 1

Experiment 1 $n = 3$ $m = 1$ $k = 1$

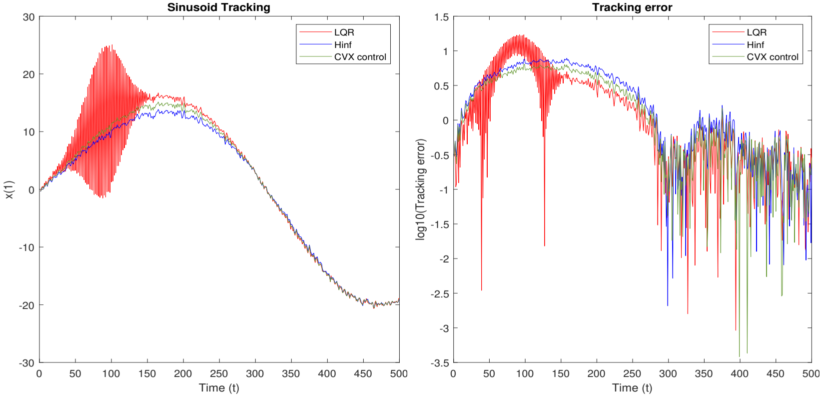


Figure: Left: Tracking of sinusoid with disturbance. Right: Tracking error

Experiment 1

Experiment 1 $n = 3$ $m = 1$ $k = 1$

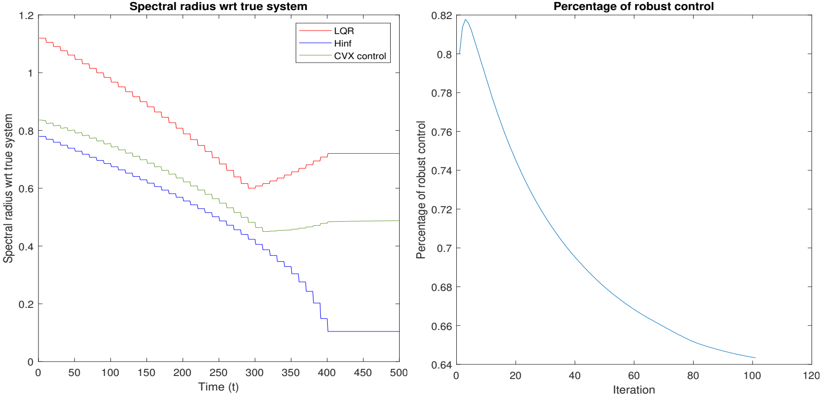


Figure: Left: Spectral Radius of the three control strategies. Right: Percentage of robust control

Experiment 2

Experiment 2 - $n = 3$, $m = 1$, $k = 1$

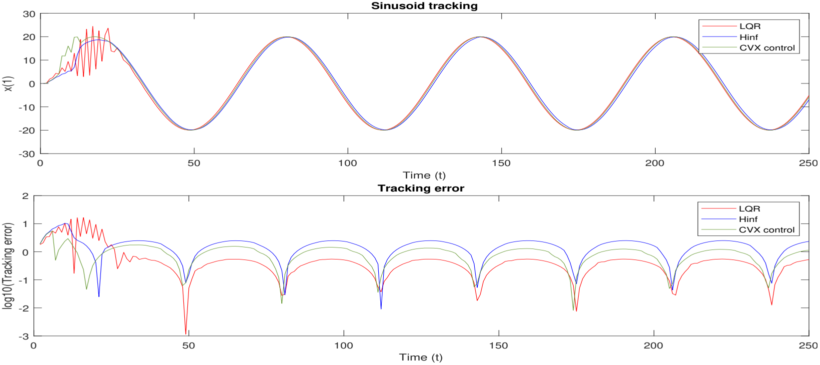


Figure: Left: Reference tracking of a sinusoid with online learning. No disturbance added Right: Tracking error

Experiment 3

Experiment 3 - $n = 4$, $m = 2$, $k = 3$

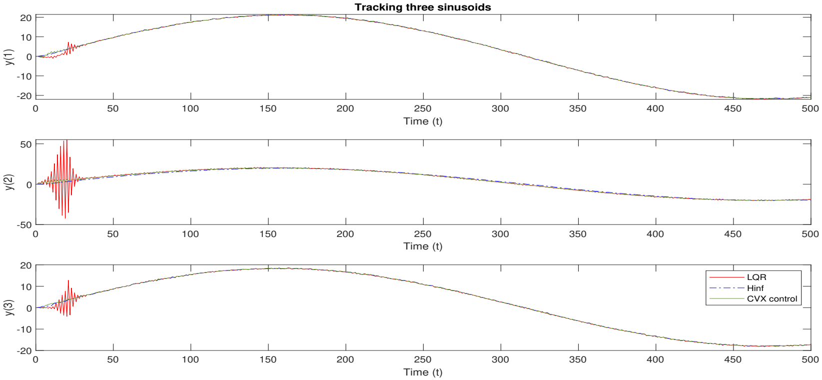


Figure: Reference tracking of three sinusoids with disturbance added and online learning

Way Forward

- Can we prove a guarantee that the optimization prevents escape out of the space of stabilizing controllers?
- Recent results on policy optimization for LQR[17][18]
- Can we transfer from a robust conservative controller to an aggressive controller while constraining ourselves to the space of stabilizing controllers using policy optimization?
- Spectral radius - difficult choice of objective function
- Investigate benefits and disadvantages.

[17] Kaiqing Zhang, Bin Hu, and Tamer Basar (2019). "Policy optimization for H2 linear control with Hinf robustness guarantee: Implicit regularization and global convergence". In: *arXiv preprint arXiv:1910.09496*.

[18] Maryam Fazel et al. (2018). "Global convergence of policy gradient methods for the linear quadratic regulator". In: *arXiv preprint arXiv:1801.05039*.

Avenues for further research

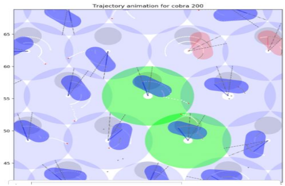
More realistic systems

- Actuator saturation
- Order of the system unknown - can change with time
- State not available for feedback - highly noisy measurements
- Non-linear systems
- Prove guarantees at least under some idealized assumptions

One Application

Telescope Fiber Positioning

- 2304 cobra fibers in a telescope
- Move all the fibers to destined locations quickly
- Avoid collisions
- Motors highly stochastic and non-linear



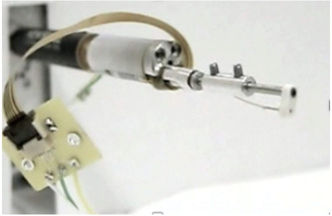


Figure: Telescope fiber positioning

Zero-shot learning to control of the simple pendulum

- Model non-linearity as a time-varying linearity.
- Can stabilize the system in first attempt without an accurate model.

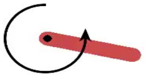


Figure: OpenAI Gym undamped

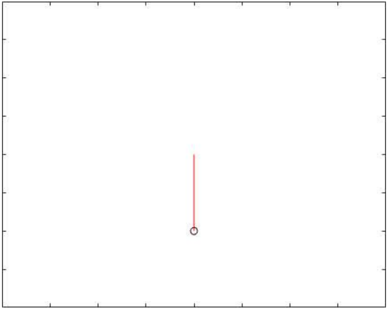


Figure: Damping and disturbance

References I

Ackermann, Juergen E and B Ross Barmish (1988). "Robust Schur stability of a polytope of polynomials". In: *IEEE transactions on automatic control* 33.10, pp. 984–986.

Agarwal, Naman, Nataly Brukhim, et al. (2019). "Boosting for Dynamical Systems". In: *arXiv preprint arXiv:1906.08720*.

Agarwal, Naman, Brian Bullins, et al. (2019). "Online control with adversarial disturbances". In: *arXiv preprint arXiv:1902.08721*.

Białas, Stanisław (2004). "A necessary and sufficient condition for stability of the convex combination of polynomials". In: *Control and Cybernetics* 33.4, pp. 589–597.

Dean, Sarah et al. (2019). "On the sample complexity of the linear quadratic regulator". In: *Foundations of Computational Mathematics*, pp. 1–47.

References II

Fazel, Maryam et al. (2018). "Global convergence of policy gradient methods for the linear quadratic regulator". In: *arXiv preprint arXiv:1801.05039*.

Fefferman, Charlie et al. (2019). "Control with Learning on the Fly: First Toy Problems". Seminar in ORFE, Princeton University.

Geromel, José C and Patrizio Colaneri (2006). "Stability and stabilization of discrete time switched systems". In: *International Journal of Control* 79.07, pp. 719–728.

Ghahramani, Zoubin and Geoffrey E Hinton (1996). *Parameter estimation for linear dynamical systems*. Tech. rep. Technical Report CRG-TR-96-2, University of Toronto, Dept. of Computer Science.

Hazan, Elad, Holden Lee, et al. (2018). "Spectral filtering for general linear dynamical systems". In: *Advances in NeurIPS*, pp. 4634–4643.

References III

Hazan, Elad, Karan Singh, and Cyril Zhang (2017). "Learning linear dynamical systems via spectral filtering". In: *Advances in NeurIPS*, pp. 6702–6712.

Kalman, Rudolph (1960). "E. 1960. A new approach to linear filtering and prediction problems". In: *Transactions of the ASME–Journal of Basic Engineering* 82, pp. 35–45.

Ljung, Lennart (2001). "System identification". In: *Wiley Encyclopedia of Electrical and Electronics Engineering*.

Schoukens, Johan and Lennart Ljung (2019). "Nonlinear System Identification: A User-Oriented Road Map". In: *IEEE Control Systems Magazine* 39.6, pp. 28–99.

Shumway, Robert H and David S Stoffer (1982). "An approach to time series smoothing and forecasting using the EM algorithm". In: *Journal of time series analysis* 3.4, pp. 253–264.

References IV

Van Overschee, Peter and BL De Moor (2012). *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media.

Zhang, Kaiqing, Bin Hu, and Tamer Basar (2019). "Policy optimization for H2 linear control with Hinf robustness guarantee: Implicit regularization and global convergence". In: *arXiv preprint arXiv:1910.09496*.

Online Learning and Regret

Algorithm 1 Paradigm of online learning

while $t \leq T$ **do**
  Observe x_t
  Make prediction $\hat{y}_t(x_t) \in \hat{Y}_t$
  Observe y_t (can be adversarial)
  Suffer loss $l_t(x_t, y_t, \hat{y}_t)$ (can be adversarial)
  t=t+1
end while

Regret

$$\sum_{t=1}^{T} \left(l_t(x_t, y_t, \hat{y}_t) - \min_{y^* \in Y_t} l_t(x_t, y_t, y^*) \right) \tag{13}$$

Linear Quadratic Regulator (LQR)

Discrete-Time

$$\min_{u_1,u_2\dots} J = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} \left(x_t^T \mathbf{Q} x_t + u_t^T \mathbf{R} u_t\right) \tag{14}$$

Static linear feedback control law optimal

$$(\mathbf{A}-\mathbf{BK})^T\mathbf{P}(\mathbf{A}-\mathbf{BK}) - \mathbf{P} + \mathbf{Q} + \mathbf{K}^T\mathbf{RK} = 0$$
$$\mathbf{K} = (\mathbf{R}+\mathbf{B}^T\mathbf{PB})^{-1}\mathbf{B}^T\mathbf{PA}$$

Continous time

$$\min_{u_t} J = \lim_{T\to\infty} \frac{1}{T} \int_{t=0}^{T} \left(x_t^T \mathbf{Q} x_t + u_t^T \mathbf{R} u_t\right) dt \tag{15}$$

$$(\mathbf{A}-\mathbf{BK})^T\mathbf{P} + \mathbf{P}(\mathbf{A}-\mathbf{BK}) + \mathbf{Q} + \mathbf{K}^T\mathbf{RK} = 0$$
$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$$

Kalman Filter

- Optimal estimate of state x given y and u
- $\mathbf{P}_t^{t_1}$ - covariance of $\mathbf{x}_t$ conditioned on the first t_1 inputs and outputs
- Let R_w and R_n be the covariance of the Gaussian noise terms

$$\begin{aligned}
\mathbf{x}_t^{t-1} &= \mathbf{A}x_{t-1}^{t-1} + \mathbf{B}u_t \\
\mathbf{P}_t^{t-1} &= \mathbf{A}\mathbf{P}_{t-1}^{t-1}\mathbf{A}^T + R_w \\
\mathbf{K}_t &= \mathbf{P}_t^{t-1}\mathbf{C}^T\left(\mathbf{C}\mathbf{P}_t^{t-1}\mathbf{C}^T + \mathbf{R}_n\right)^{-1} \\
\mathbf{x}_t^t &= \mathbf{x}_t^{t-1} + \mathbf{K}_t\left(\mathbf{y}_t - \mathbf{C}x_t^{t-1} - \mathbf{D}u_t\right) \\
\mathbf{P}_t^t &= \mathbf{P}_t^{t-1} - \mathbf{K}_t\mathbf{C}\mathbf{P}_t^{t-1}
\end{aligned}$$

Expectation Maximization

- Estimate parameters in the presence of underlying hidden state
- Parameters represented by θ
- Gaussian disturbance and noise

E step

$$Q(\theta|\theta^t) = \mathbb{E}_{x_t|y;\theta_{t-1},u}\left[\log P(y_t, x_t; \theta, u)\right]$$

$x_t|y;\theta_{t-1}, u$ is Gaussian - estimated by the Kalman filter

M step

$$\theta_{t+1} = \arg\max_{\theta} Q(\theta|\theta^t)$$

θ appears linearly with gaussian noise... LSE

Subspace identification

Assume $\hat{\mathbf{A}}$ and $\hat{\mathbf{C}}$ are known

$$\hat{y}\left(t|\mathbf{B},\mathbf{D}\right) = \hat{\mathbf{C}}\left(z\mathbf{I} - \hat{\mathbf{A}}\right)^{-1}\mathbf{B}u(t) + \mathbf{D}u(t)$$

$$\hat{y}\left(t\right) = \psi(t)\begin{bmatrix}\mathrm{Vec}(\mathbf{B})\\ \mathrm{Vec}(\mathbf{D})\end{bmatrix}$$

How do we get the observability matrix?

$$\mathbf{Y} = \mathcal{O}_r\mathbf{X} + S_r\mathbf{U} + \underbrace{\mathbf{V}}_{\text{Noise terms}}$$

$$\mathbf{P}^{\perp}_{\mathbf{U}^T} = \mathbf{I} - \mathbf{U}^{\mathbf{T}}\left(\mathbf{U}\mathbf{U}^{\mathbf{T}}\right)^{-\mathbf{1}}\mathbf{U}$$

$$\mathbf{Y}\mathbf{P}^{\perp}_{\mathbf{U}^T} = \mathcal{O}_r\mathbf{X}\mathbf{P}^{\perp}_{\mathbf{U}^T} + \mathbf{V}\mathbf{P}^{\perp}_{\mathbf{U}^T}$$

Subspace identification - Estimating observability matrix

How do we get rid of the noise term? Try to correlate it with another suitable matrix Φ.

$$\Phi = \begin{bmatrix} \phi_s(1) & \phi_s(2) & \dots & \phi_s(N) \end{bmatrix}$$

$$G = \frac{1}{N}\mathbf{Y}\mathbf{P}^{\perp}_{\mathbf{U}^T}\Phi^T = \mathcal{O}_r \frac{1}{N}\mathbf{X}\mathbf{P}^{\perp}_{\mathbf{U}^T}\Phi^T + \frac{1}{N}\mathbf{V}\mathbf{P}^{\perp}_{\mathbf{U}^T}\Phi^T = \mathcal{O}_r \tilde{T}_N + V_N$$

We want:

$$\lim_{N\to\infty} V_N = 0$$
$$\lim_{N\to\infty} \tilde{T}_N = \tilde{T}$$

Following choice of Φ works

$$\phi_s(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-s_1) \\ u(t-1) \\ \vdots \\ u(t-s_2) \end{bmatrix}$$

Spectral Filtering - Symmetric dynamics matrix

- Real eigenvalues
- Initial state is assumed to be 0
- w.lo.g $\mathbf{A}$ can be assumed to be diagonal
- $\mathbf{c}_l$ be the l^{th} column of $\mathbf{C}$ and $\mathbf{b}_l$ be the l^{th} row of $\mathbf{B}$
- Let $\mu(\alpha) = \left[\alpha^{i-1}(1-\alpha)\right]$ be a T dimensional vector

$$y_t - y_{t-1} = (\mathbf{CB} - \mathbf{D})u_{t-1} + \sum_{i=1}^{T} \mathbf{C}\left(\mathbf{A}^i - \mathbf{A}^{i-1}\right)\mathbf{B}u_{t-i-1} + \mathbf{D}u_t \quad (16)$$

$$= (\mathbf{CB} - \mathbf{D})u_{t-1} + \sum_{i=1}^{T} \mathbf{C} \sum_{l=1}^{d} \left(\alpha_l^i - \alpha_l^{i-1}\right) e_l e_l^T \mathbf{B}u_{t-i-1} + \mathbf{D}u_t \quad (17)$$

$$= (\mathbf{CB} - \mathbf{D})u_{t-1} + \sum_{l=1}^{d} \left(c_l b_l^T \mu(\alpha_l) \circledast u\right) + \mathbf{D}u_t \quad (18)$$

Spectral Filtering - Symmetric dynamics matrix

$$y_t - y_{t-1} = (\mathbf{CB} - \mathbf{D})u_{t-1} + \sum_{l=1}^{d} c_l b_l^T \left(\mu(\alpha_l) \circledast u\right) + \mathbf{D}u_t \tag{19}$$

Find a basis for representation of the vectors with structure $\mu(\alpha)$
Define a matrix Z such that:

$$Z_{ij} = \int_{\alpha=0}^{1} \mu(\alpha)_i \mu(\alpha)_j d\alpha = \frac{2}{(i+j)^3 - (i+j)} \tag{20}$$

Eigenvectors of Z denoted by $\phi_{sf,i}$

$$\mathbf{y}_t - \mathbf{y}_{t-1} \tag{21}$$

$$= (\mathbf{CB} - \mathbf{D})u_{t-1} + \sum_{f=1}^{k} \sum_{l=1}^{d} c_l b_l^T \langle \mu(\alpha_l), \phi_{sf,f} \rangle \left(\phi_{sf,f} \circledast u\right) + \mathbf{D}u_t \tag{22}$$

Spectral Filtering - General

Eigenvalues of $\mathbf{A}$ can be complex. According to the ARX model, if β_j are chosen to be the coefficients of the characteristic polynomial,

$$\sum_{i=0}^{d} \beta_i y_{t-i} = \sum_{j=0}^{d} \mathbf{P}_j u_{t-j} \tag{23}$$

Instead, let us choose β_j to be the coefficients of the polynomial with roots given by the phases of the eigenvalues of $\mathbf{A}$.
Define approximation error

$$\delta_t = \sum_{i=0}^{d} \beta_i y_{t-i} - \sum_{j=0}^{d} \mathbf{P}_j u_{t-j} \tag{24}$$

Spectral Filtering - General

For $\mathbf{a} \in \mathbb{R}^T$, define $a^{(\omega)} := (a_j \omega^j)_{1 \le j \le T}$,
$a^{(\cos,\theta)} := (a_j \cos(j\theta))_{1 \le j \le T}$, $a^{(\sin,\theta)} := (a_j \sin(j\theta))_{1 \le j \le T}$
δ_t can be well approximated using the wave filters. Let ω_l be the actual phases of $\mathbf{A}$. Phase Quantization.

$$\delta_t \approx \sum_{l=1}^{d} \sum_{h=1}^{k} M_l^{'}(h,:,:)\sigma_h^{\frac{1}{4}} \left(\phi_{sf,h}^{(\omega_l)} \circledast u\right) \tag{25}$$

$$\approx \sum_{w=1}^{W} \sum_{h=1}^{k} M(w,h,:,:)\sigma_h^{\frac{1}{4}} \left(\phi_{sf,h}^{(e^{2\pi i \frac{w}{W}})} \circledast u\right) \tag{26}$$

$$\approx \sum_{w=1}^{W} \sum_{h=1}^{k} M(w,h,:,:)\sigma_h^{\frac{1}{4}} \left(\phi_{sf,h}^{(\cos,2\pi \frac{w}{W})} \circledast u\right) \tag{27}$$

$$+ N(w,h,:,:)\sigma_h^{\frac{1}{4}} \left(\phi_{sf,h}^{(\sin,2\pi \frac{w}{W})} \circledast u\right) \tag{28}$$

Policy Optimization

$$\min_{u_t} J = \sum_{t=1}^{T} c_t$$
$$x_{t+1} = f(x_t, u_t)$$

Parameterize policy as $u_t = \pi_\theta(x_t)$

Policy Gradient

$$\theta_{t+1} = \theta_t - \left(\sum_{t=1}^{T} c_t \sum_{t'=1}^{T} \nabla_\theta \ln(\pi_\theta(x_{t'})) \right)$$

Natural Policy Gradient

$$\theta_{t+1} = \theta_t - \mathbf{G}_\theta^{-1} \nabla_\theta J$$

where $\mathbf{G}_\theta = \mathbb{E}(\nabla_\theta \ln \pi_\theta(x_t) \nabla_\theta^T \ln \pi_\theta(x_t))$ is the fisher information matrix

Inputs in system ID experiment

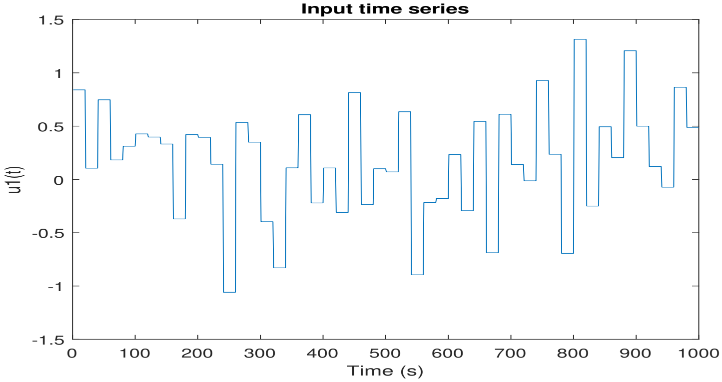


Figure: Sample input for the system identification experiment