Adaptive Control and Interconnections with Learning

Anuradha Annaswamy

Active-adaptive Control Laboratory
Department of Mechanical Engineering

Massachusetts Institute of Technology

* Collaborators: Peter Fisher, Johannes Autenrieb, Anubhav Guha, Joey Gaudio, Heather Hussain, Eugene Lavretsky, Mike Bolender

Sponsors: Boeing, AFRL
Outline

• Introduction
• Adaptive Control & Intersection with learning
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning
• Reinforcement learning
  – Analytical Guarantees
    • Neural network-based control (NeurIPS 1995)
    • Safe-model based RL with stability guarantee (NeurIPS 2017)
• Synergies between AC and RL
  – Example 1: Quad rotor
  – Example 2: Adaptive LQR
Outline

• Introduction
  • Adaptive Control & Intersection with learning
    – Basics of adaptive control
    – Parameter learning
    – Imperfect learning

• Reinforcement learning
  – Analytical Guarantees
    • Neural network-based control (NeurIPS 1995)
    • Safe-model based RL with stability guarantee (NeurIPS 2017)

• Synergies between AC and RL
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
Control Goals in Complex Dynamic Systems

- Stability
- Robustness
- Safety
- Optimality
- Accommodate Uncertainties
- Meet Constraints
From Water Clocks and Flyball Governors

James Watt’s steam governor (1788): As the speed of the prime mover increases, the central spindle of the governor speeds up causing the two masses on lever arms to move up, pulls down a thrust bearing, closing a throttle valve.

From 3rd century BC. The hour indicator ascends as water flows in. A series of gears rotate a cylinder to correspond to the temporal hours.
To Control of Physical Systems

Aerospace Control
Process Control
Automotive Control
Robotics
Control in Biological Systems

to name a few...
And Various Success Stories Along the Way!

Auto-tuners for PID Controllers – K.J. Astrom and T. Hägglund
Advanced Tension Control in Steel Rolling Mills – T. Parisini and L. Ciani
Advanced Energy Solutions for Power Plants – V. Havlena
Performance Monitoring for Mineral Processing – C. Aldrich
Advanced Control for the Cement Industry – E. Gallestey
Cross-Direction Control of Paper Machines – G. Stewart
Ethylene Plantwide Control and Optimization – J. Lu and R. Nath
Automated Collision Avoidance Systems – C. Tomlin and H. Erzberger
Digital Fly-by-Wire Technology – C. Philippe
Nonlinear Multivariable Flight Control – J. Bosworth and D. Enns
Robust Adaptive Control for the Joint Direct Attack Munition – K.A. Wise and E. Lavretsky
Control for Industrial Spacecraft – M. Whorton
H-infinity Control for Telecommunication Satellites – C. Philippe
Control for Formula One! – M. Smith
Active Safety Control for Automobiles – L. Glielmo
Automated Manual Transmissions – L. Iannelli
Mobile Robot-Enabled Smart Warehouses – R. D’Andrea
Digital Printing Control: Print Shop in a Box – L.K. Mestha
Trip Optimizer for Railroads – D. Eldredge and P. Houpt
Coordinated Ramp Metering for Freeways – M. Papageorgiou
Control in Mobile Phones – B. Bernhardsson
Improved Audio Reproduction with Control Theory – Y. Yamamoto
To Cyber-physical Systems

A Nexus of the digital and physical worlds
To Societal-scale Challenges*

* https://ieeecss.org/control-societal-scale-challenges-road-map-2030
Real-time Control

What is the role of adaptation?
Outline

• Introduction

• Adaptive Control & Intersection with learning
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning

• Reinforcement learning
  – Analytical Guarantees
    • Neural network-based control (NeurIPS 1995)
    • Safe-model based RL with stability guarantee (NeurIPS 2017)

• Synergies between AC and RL
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
How does an adaptive controller work?

Design provably correct learning algorithms to generate $\hat{\Theta}, u$ in real-time

$$u = C_1(\omega, \hat{\Theta}, t)$$
$$\dot{\Theta} = C_2(\omega, \hat{\Theta}, t)$$
Error Models - Two types of errors

- \( e \): Performance error \( (\text{ex. } \hat{x} - x; \, x - x_m) \)
  - can be measured, needs to be reduced

- \( \tilde{\theta} \): Parameter error \( (\text{ex. } \hat{\theta} - \theta) \)
  - Unknown, can be adjusted – Learning Rule

Goal: Determine error models and learning rules
A Simple Linear Regression

System Model:
\[ y = \phi^T \theta \]

Build an estimator:
\[ \hat{y} = \phi^T \hat{\theta} \]

Performance error:
\[ e = \hat{y} - y \]

Parameter error:
\[ \tilde{\theta} = \hat{\theta} - \theta \]

Error Model:
\[ J = e^2 \]
\[ \dot{\theta} = -\eta \nabla J \]

Simplest Gradient Descent

\[ \dot{\hat{\theta}} = -\eta e \phi \]

Error Model 1*

Distinctions: Gradient-descent leads to instability!

\[ e = W(s)[\phi^T \tilde{\theta}] \]

Real-time data \( \phi \)  \( \rightarrow \)  \( \tilde{\theta} \)  \( \rightarrow \)  Performance error \( e \)

Suppose \( J = e'^2 \)

\[ \frac{\partial J}{\partial \tilde{\theta}} = 2e' x_p \]

Don‘t have access to \( e' \)

* HP Whitaker, An Adaptive System for Control of the Dynamics Performances of Aircraft and Spacecraft, Inst Aeronautical Services, 1959 - MIT Rule

Use a Stability Framework

\[ e = W(s)[\phi^T \tilde{\theta}] \]

\[ \dot{\tilde{\theta}} = -\nabla(e^2) \]
\[ \rightarrow \text{Instability} \]

\[ V = e^2 + \|\tilde{\theta}\|^2 \]
\[ \dot{V} = 2\tilde{\theta}^T \dot{\theta} + 2e\dot{e} \]
\[ \dot{\theta} = -e\phi \]
\[ \dot{V} = -2e^2 \leq 0 \quad (\text{for a class of } W(s)) \]

* HP Whitaker, An Adaptive System for Control of the Dynamics Performances of Aircraft and Spacecraft, Inst Aeronautical Services, 1959 - MIT Rule


Princeton Workshop on Optimization, Learning, and Control, June 2024
Error Model 1 (stability framework)

System Model:
\[ \theta^T \phi y \]

Build an estimator:
\[ \hat{y} = \phi^T \hat{\theta} \]

Error Model:
\[ \tilde{\theta}^T \phi e \]

Stability Framework:
\[ V = \|\tilde{\theta}\|^2 \]
\[ \dot{V} = 2\tilde{\theta}^T \dot{\tilde{\theta}} = -2\eta\tilde{\theta}^T e\phi = -2\eta e^2 \leq 0 \]

Performance error:
\[ e = \hat{y} - y \]

Parameter error:
\[ \tilde{\theta} = \hat{\theta} - \theta \]

\[ J = e^2 \]

\[ \tilde{\theta} = \theta(t) - \theta^* \]

May not look like this

\[ \dot{\theta} = -\eta e\phi \]

\[ \pm \frac{\partial J}{\partial \theta} \text{ always} \]
Adaptive Control of a High-order System

- Plant + adaptive controller
  \[
  \dot{x}_p = A_m x_p + B_p \tilde{\theta}^T(t) x_p + B_p r
  \]

- Reference Model: Produces a target \(x_m\)
  \[
  e = x_p - x_m
  \]

- Dynamic Error Model:
  \[
  \dot{\theta} = -\Gamma (e^T P B_p) x_p
  \]

\[
V = e^T Pe + \text{Trace}\left(\tilde{\theta}^T \Gamma^{-1}\tilde{\theta}\right) \quad \Rightarrow \quad \dot{V} = -e^T Q e \leq 0
\]

\(\rightarrow\) Global Stability \quad Barbalat’s Lemma \(\rightarrow\)

\[
\lim_{t \to \infty} e(t) = 0
\]
Adaptive Control with Partial Observations

**Adaptive law:**

\[ \dot{\theta}(t) = -\Gamma e(t)\omega(t) \quad (n^* = 1) \]

\[ = -\Gamma \frac{\varepsilon(t)\zeta(t)}{1 + \zeta^T(t)\zeta(t)} \quad (n^* \geq 2) \]

\[ \rightarrow \text{Global Stability} \quad \text{Barbalat’s Lemma} \rightarrow \lim_{t \to \infty} e(t) = 0 \]
Adaptive Nonlinear Control

• Unknown nonlinear system: \( \dot{x} = f(x) + g(x)u \)

• Suppose \( f(x) = \theta^T \phi(x), g(x) \) known

• Feedback linearization*: \( u = g^{-1}(x)(-\alpha x + \theta^T \phi(x)) \)

• Adaptive nonlinear control: \( u = g^{-1}(x)(-\alpha x + \hat{\theta}^T \phi(x)) \)

\[ \dot{\theta} = C_2(\hat{\theta}, \phi, e) \]

• Suitable functions \( C_2 \) and performance error \( e \) can be found.

• Requires knowledge of \( \phi(x) \)

\[ f(x) = \theta^T \phi(x) \]: A linear single-layer network

* Krstic, Kanellakopoulos, Kokotovic, Nonlinear and Adaptive Control Design, 1995
Results (experimental)*

Robustness Tools*

- Integration with a Baseline
- Guarantees to unknown actuator dynamics
  \[ W(s) = G_{act}(s)W_p(s) \]
  
  (i) \( G_{act}(s) = e^{-s\tau} \) (time delay)  
  
  (ii) \( G_{act}(s) = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \) (unmodeled)

- Addition of regularization, projection, or dead-zone

Accommodation of Input Constraints

Model the effect of saturation as an additive disturbance

Modify the performance error

\[ \dot{\theta} = -\Gamma (e_u^T P B_p) x_p \]

Parametric Uncertainties, Saturation effects

*Papers with Karason, Gaudio, Smith, Slocum, Evesque, Rumsey, Dowling, Lavretsky since 1994

Princeton Workshop on Optimization, Learning, and Control, June 2024
Adaptive Control in Propulsion Systems

- Uncontrolled case: 163–167 dB
- Nominal: 150 dB reduction (delivered energy saving)
- Adaptive: [Graph showing comparison]

Princeton Workshop on Optimization, Learning, and Control, June 2024
Results (High Performance Aircraft)

Not limited elevator rate reaches -105°/s (physically infeasible)

Limited elevator rate does not pass specified -60°/s
Accommodation of State Constraints

**Safety:**
Guaranteeing that $x(t)$ stays within a set $\mathcal{C}$ for any $t \geq 0$.

**Performance:**
Guaranteeing that $x(t)$ stays bounded and converges to a desired $r(t)$.

Mission Goals

Real-time uncertainties

Plant

$u$

$x_p$

Control inputs

Princeton Workshop on Optimization, Learning, and Control, June 2024
An integrative adaptive controller

- Adaptive controller accommodates uncertainties and magnitude limits.
- Constraints are met using a control barrier function (CBF) for a reference model and an error-based relaxation (EBR).

**Example 1:** Obstacle avoidance

A First-order Example
(50% loss of effectiveness)

EBR: Error-based Relaxation

Waypoints specified
Example 2: Validation Using SDRT

50% loss of effectiveness in 2 of the 4 motors

No Adaptation

With Adaptation

SDRT: Simulink Desktop Real-time
Example 2: Validation Using VR-Goggles

50% loss of effectiveness in 2 of the 4 motors

Command

6-DOF Quadrotor

States

CBF + no adaptation

CBF + adaptation
Adaptive Control Systems in Production

Unmanned Flight Control Applications

Improved performance and reliability

- Flight Test

Courtesy Boeing Company
Outline

• Introduction

• Adaptive Control & Intersection with learning
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning

• Reinforcement learning
  – Analytical Guarantees
    • Neural network-based control (NeurIPS 1995)
    • Safe-model based RL with stability guarantee (NeurIPS 2017)

• Synergies between AC and RL
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
Learning and Adaptation

\[ \dot{x} = f(x, u, \Theta) \]
\[ y = g(x, u, \Theta) \]

- \( e \): Performance error (ex. \( \hat{x} - x; x - x_m \))
- \( \tilde{\Theta} \): Parameter error (ex. \( \hat{\Theta} - \Theta \))

Learning: \( \hat{\Theta}(t) \rightarrow \Theta \)
The Effect of Parameter Learning

- State space (All initial states are on the unit circle) 
  $$(\omega_n = 0.2, \zeta = 0.7)$$

- Estimated plant’s state space 
  $$(\omega_n = 0.1, \zeta = 0.35)$$

50% changes in $\theta$

250% changes in state space ($x$)

Real-time decisions are important

Parameter learning is important

Leads to Robustness Guarantees
Uniform Asymptotic Stability of $\tilde{\theta} = 0$ in

$$\dot{\tilde{\theta}} = -[\phi(t)\phi^T(t)]\tilde{\theta}$$

• Necessary and Sufficient Condition: $\phi$ is persistently exciting*:

$$\text{There exists } T > 0 \text{ and } \alpha > 0 \text{ such that }$$

$$\int_t^{t+T} \phi(t)\phi^T(t)dt \geq \alpha I \quad \forall t \geq t_0$$

* Morgan and Narendra, SIAM J. Optimization, Vol. 15, No. 1, pg. 5-24, 1977
Examples of Persistent Excitation

**Definition:**
\[
\sum_{i=k}^{k+\Delta T-1} \phi_i \phi_i^T \geq \alpha I, \forall k \geq k_0
\]

For all \(t_0 \geq 0\),
\[
\phi_k = \begin{cases} 
v, & k \in [t_0, t_0 + \Delta T/2] \\
w, & k \in [t_0 + \Delta T/2, t_0 + \Delta T] 
\end{cases}
\]

\[\sum_{i=k}^{k+\Delta T-1} \phi_i \phi_i^T \text{ is full rank}\]

\[\sum_{i=k}^{k+\Delta T-1} \phi_i \phi_i^T \text{ is not full rank}\]
Outline

• Introduction

• Adaptive Control & Intersection with learning
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning

• Reinforcement learning
  – Analytical Guarantees
    • Neural network-based control (NeurIPS 1995)
    • Safe-model based RL with stability guarantee (NeurIPS 2017)

• Synergies between AC and RL
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
Guarantees with Imperfect Learning

• Performance and Learning are conflicting objectives

• Effect of imperfect learning on performance has to be accommodated.

• With persistent excitation, $\tilde{\theta}(t) \to 0 \Rightarrow \text{Learning!}$

• Without persistent excitation, guaranteed performance can be ensured.
Guarantees with Imperfect Learning

Stable Subspace

Unstable Subspace

\[ \theta^* = \theta_1 \]

\[ \theta_2 \]

Adaptive Control

\[ \theta_1 + \Delta \]

\[ u \]

Performance goal: \( \text{Min}(e) \)

Imperfect learning, yet safe performance

Open Loop Unstable Short Period Dynamics

Effect of Different Adaptive Parameters

\[ \theta_1 + \Delta \]

With \( \theta_1 \)

Adaptive, \( \theta_2 \)

\[ \alpha \]

\[ \alpha_d \]
Imperfect Learning can lead to bursting phenomena*

Outline

• Introduction
• Adaptive Control & Intersection with learning
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning
• Reinforcement learning
  – Analytical Guarantees
    • Neural network-based control (NeurIPS 1995)
    • Safe-model based RL with stability guarantee (NeurIPS 2017)
• Synergies between AC and RL
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
• Unknown nonlinear system: \[ \dot{x} = f(x, u) \]

• Minimize infinite-horizon cost: \[ J(u; x_0) = \int_0^\infty \gamma^t r(x, u) dt \]

• Optimal cost-to-go: \[ V^*(x_0) = \inf_u J(u; x_0) \]

• Hamilton-Jacobi-Bellman Equation: \[ 0 = \inf_a \{ \partial_x V^*(x)f(x, a) + r(x, a) \} \]

• Optimal control policy: \[ \mu^*(x) = \arg\inf_a \{ \partial_x V^*(x)f(x, a) + r(x, a) \} \]

• Reinforcement Learning: Learn the Policy/Value Function

What happens when \( f(x, u) \) is uncertain?
Universal Approximation Theorem

For $\epsilon > 0$, $\exists N, w^*$ s.t. $|y_a - y| \leq \epsilon$ $\forall x \in X$

Estimate weights $w_j$ using back-propagation

$\hat{\theta} = -\Gamma \nabla_{\theta} L_t(\theta)$

(From YouTube, Why do Nnets work?)
Approximate Policy Iteration for Optimal Control*

• Use neural networks to approximate $H_k$, $V_k$ and $\mu_k$

$$\hat{H}(w_k, x, u) \approx H_k(x, u), \hat{V}(c_k, x) \approx V_k(x), \quad \hat{\mu}(\theta_k, x) \approx \mu_k(x)$$

• Use an iterative procedure:

  • For a sampling period $[t_k, t_{k+1}]$, collect data $(x, u)$ and cost $r(x, u)$
  • Policy evaluation: Given $\theta_k$, solve for $w_k$ and $c_k$ from HJB equation

$$0 = \int_{t_k}^{t_{k+1}} \hat{H}(w_k, x, \hat{\mu}(\theta_k, x)) dt, \quad \hat{V}(c_k, x) = \int_{t_k}^{t_{k+1}} (\hat{H}(w_k, x, u) - r(x, u)) dt$$

  • Policy improvement: Update $\theta_{k+1} = \inf_{\theta} \hat{H}(w_k, x, \hat{\mu}(\theta, x))$

• Special case: Single layer network reduces to a linear regression

• Estimation accuracy depends on persistent excitation
Outline

• Introduction

• Adaptive Control & Intersection with learning
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning

• Reinforcement learning
  – Analytical Guarantees
    • Neural network-based control (NeurIPS 1995)
    • Safe-model based RL with stability guarantee (NeurIPS 2017)

• AC-RL: Our recent approach
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
Neural Networks Based Control*

Neural control

\[ u = N(x; W) \]

A Lyapunov-centric approach for training the weights:

\[ V = x^T P x \quad P = P^T > 0 \]

\[ \dot{V} = 2x^T P f(x, N(x; W)) \]

\[ \dot{V}_d: \text{desired target for } \dot{V} \quad \text{ex. } \dot{V}_d = -x^T Q x \]

Goal: \( x(t) \to 0 \) as \( t \to \infty \) for any initial condition.

\[ \dot{x} = f(x, u), y = h(x) \]

\[ u = \gamma(x, t) \]

Neural Networks Based Control (contd.)

\[ V = x^T P x \quad P = P^T > 0 \]

\[ \dot{V} = 2x^T Pf(x, N(x; W)) \]

\[ \dot{V}_d = -x^T Q x \]

Training Procedure

\[ \min J \triangleq \min_w \frac{1}{2} \sum_{i=1}^{M} \left( \max\{0, \Delta V_{e,i}\} \right)^2 \]

\[ \frac{\partial J}{\partial w_j} = \sum_{i \in A} \frac{\partial J}{\partial N(y_i; W)} \frac{\partial N(y_i; W)}{\partial w_j} = 2 \sum_{i \in A} \Delta V_{e,i} x_i^T P \frac{\partial f(x_i, u_i)}{\partial u} \frac{\partial N(y_i; W)}{\partial w_j} \]

\[ \Delta V_e = \dot{V} - \dot{V}_d \]

Results

\[ x_t = f(x_{t-1}, u_{t-1}) = \begin{bmatrix} x_{1t-1} (1 + x_{2t-1}) + x_{2t-1} (1 - u_{t-1} + u_{t-1}^2) \\ x_{1t-1}^2 + 2x_{2t-1} + u_{t-1} (1 + x_{2t-1}) \end{bmatrix} \]

(a) Linear Control     (b) Correction from Neural control: N(x)-Kx

Closed loop responses from (a) linear control and (b) neural control

Safe model-based RL with analytical guarantees*

\[
\begin{align*}
    u &= \gamma(x, t) \\
    x_{t+1} &= h(x_t, u_t) + g(x_t, u_t)
\end{align*}
\]

Goal: Design the input so that \( x(t) \to x_{cmd} \) with an optimal input \( u \).

Optimal policy

\[
u = \pi_n(x)
\]

\[
\pi_n, c_n = \arg\max_{\pi \in \Pi_L, c \in \mathbb{R}_>0} c, \text{ such that for all } x \in \mathcal{V}(c) \cap \mathcal{X}_\tau: (x, \pi(x)) \in D_n
\]

\[
D_n = \{(x, u) \in \mathcal{X}_\tau \times \mathcal{U} | u_n(x, u) - v(x) < -L_{\Delta v} \tau\}
\]

\[
\mathcal{V}(c) = \{x \in \mathcal{X} \setminus \{0\} | v(x) \leq c\}, c > 0
\]

\([\mathcal{X}_\tau]: \text{discretization of } \mathcal{X}\)

Optimal policy

\[ u = \pi_n(x) \]

\[
\begin{align*}
\pi_n, c_n &= \arg\max_{\pi \in \Pi_L, c \in \mathbb{R}_{>0}} c, \quad \text{such that for all } x \in \mathcal{V}(c) \cap \mathcal{X}_\tau: (x, \pi(x)) \in \mathcal{D}_n \\
\mathcal{D}_n &= \{(x, u) \in \mathcal{X}_\tau \times \mathcal{U} | u_n(x, u) - v(x) < -L_{\Delta v}\tau\}
\end{align*}
\]

\[
\mathcal{V}(c) = \{x \in \mathcal{X} \setminus \{0\} | v(x) \leq c\}, \quad c > 0
\]

\[ \mathcal{X}_\tau : \text{discretization of } \mathcal{X} \]

An implementable training procedure

\[
\pi_n = \arg\min_{\pi_\theta \in \Pi_L} \sum_{x \in \mathcal{X}_\tau} r(x, \pi_\theta(x)) + \gamma J_{\pi_\theta} (\mu_{n-1}(x, \pi_\theta(x)) + \lambda (u_n(x, \pi_\theta(x)) - v(x) + L_{\Delta v}\tau)
\]

**Theorem 2.** Under Assumptions 1 and 2 with \( L_{\Delta v} := L_v L_f (L_{\pi} + 1) + L_v \), let \( \mathcal{X}_\tau \) be a discretization of \( \mathcal{X} \) such that \( \|x - [x]_\tau\|_1 \leq \tau \) for all \( x \in \mathcal{X} \). If, for all \( x \in \mathcal{V}(c) \cap \mathcal{X}_\tau \) with \( c > 0 \), \( u = \pi(x) \), and for some \( n \geq 0 \) it holds that \( u_n(x, u) < v(x) - L_{\Delta v}\tau \), then \( v(f(x, \pi(x))) < v(x) \) holds for all \( x \in \mathcal{V}(c) \) with probability at least \( 1 - \delta \) and \( \mathcal{V}(c) \) is a region of attraction for (1) under policy \( \pi \).
Outline

• Introduction
• Adaptive Control & Intersection with learning
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning
• Reinforcement learning
  – Analytical Guarantees
    • Neural network-based control (NeurIPS 1995)
    • Safe-model based RL with stability guarantee (NeurIPS 2017)
• Synergies between AC and RL
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
An online policy: AC-RL

- Idea: Use RL for training offline for optimal policy, based on nominal information
  \[
  \dot{x}_r = f_r(x_r, u_r) = f_r(x_r, \pi(x_r)) \\
  \dot{x} = f(x, u)
  \]
- AC-RL: Augment with AC to correct in real-time
  \[
  u = u_r + f(e, \hat{\Theta}, \ldots) \\
  \dot{\hat{\Theta}} = \Gamma \nabla L_{\hat{\Theta}}(e)
  \]
- Can be shown to be stable*
- Leads to \( \lim_{t \to \infty} \|e(t)\| = 0 \)

AC-RL: Details

- $\dot{x}_r = Ax_r + B[u_r + g(x_r)] \rightarrow$ RL
  - $u_r = \pi(x_r)$
  - Assume $g(x) = \Theta_n \Phi_n(x)$
  - Assume $\exists \Theta_l$ s.t. $A_H = A + B\Theta_l$ Hurwitz

- True system:
  $\dot{x} = Ax + B\Lambda[u + \Lambda^{-1}\Theta_n \Phi_n(x)]$

- Adaptive Law: $\rightarrow$ AC
  - $u = \hat{\Theta}(t)\Phi(t)$
  - $\dot{\varepsilon} = -\gamma B^T P e \Phi^T$
  - $\dot{\hat{\Theta}} = -\beta (\hat{\Theta} - \varepsilon) N_t$
  - $N_t = 1 + \mu \Phi^T \Phi$
  - $\mu \geq \frac{2\gamma}{\beta} \|PB\|_F^2$

- Feedback Linearizable
- $\Phi_n, \Theta_n$ account for unknown nonlinear and linear components of dynamics

- $\Phi = \begin{bmatrix} u_r + g(x_r) + \Theta_l e \\ \Phi_n(x) \end{bmatrix}$
- $\Phi$ contains the nonlinear basis of the dynamics, and a term incorporating the RL input and reference tracking error
Outline

• Introduction
• Adaptive Control & Intersection with learning
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning
• Reinforcement learning
  – Analytical Guarantees
    • Neural network-based control \textit{(NeurIPS 1995)}
    • Safe-model based RL with stability guarantee \textit{(NeurIPS 2017)}
• Synergies between AC and RL
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
Example 1 (Optimality): Quadrotor *

- Autonomous landing of quadrotor on a moving platform
- Parameter uncertainties (25%)
- Loss of Effectiveness (50-75%)
- Success:
  - $|\Delta z| \leq 5cm$ \textbf{and}
  - $|\Delta xy| \leq 25cm$ \textbf{and}
  - $|\phi|, |\theta| \leq 10^\circ$ \textbf{and}
  - $|v_{xy}| \leq 50cm/s$ \textbf{and}
  - $|v_z| \leq 10cm/s$
- Failure:
  - $\Delta z \leq 0$ \textbf{or}
  - Timeout (complete in 10 sec)
- Goal: Succeed ASAP

Quadrotor: Land on a moving platform

With 50% Loss of Effectiveness mid-flight

<table>
<thead>
<tr>
<th>RL Success Rate</th>
<th>AC-RL Success Rate</th>
<th>LOE</th>
</tr>
</thead>
<tbody>
<tr>
<td>94%</td>
<td>95%</td>
<td>0%</td>
</tr>
<tr>
<td>71%</td>
<td>81%</td>
<td>10%</td>
</tr>
<tr>
<td>28%</td>
<td>47%</td>
<td>25%</td>
</tr>
<tr>
<td>4%</td>
<td>11%</td>
<td>50%</td>
</tr>
<tr>
<td>0%</td>
<td></td>
<td>75%</td>
</tr>
</tbody>
</table>

**Significant improvement with AC-RL over pure RL**

Quadrotor crashes

Why is AC-RL successful?

* Quadrotor crashes

Main feature that gives AC-RL an edge
Outline

• **Introduction**

• **Adaptive Control & Intersection with learning**
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning

• **Reinforcement learning**
  – Analytical Guarantees
    • Neural network-based control *(NeurIPS 1995)*
    • Safe-model based RL with stability guarantee *(NeurIPS 2017)*

• **AC-RL: Our recent approach**
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
Example 2: Adaptive LQR

• AC-RL: learn *nominally optimal* controller offline via RL, then use AC online to stabilize with parametric uncertainties

• Controller found online is *stabilizing* but not necessarily *optimal* for the true unknown dynamics

• The next step: how do we learn and converge to an optimal controller online *while still guaranteeing stability*?

• To address this, we simply the problem with linear time-invariant dynamics and a quadratic cost function...
General Structure of Adaptive LQR

1. Stability relies on excitation to stabilize!
2. Initial controller must be stabilizing
3. Requires difficult optimization

Search for Nearby Optimistic Params

Calculate New Feedback Gain

Controller

Plant

Cond($V_t, t$)

$V_t := \sum^{t-1}_{\tau=0} \phi_\tau \phi_\tau^T$, $\phi_t := [x_t^T, u_t^T]^T$

Can we develop an algorithm that requires neither 1, 2, nor 3?

Princeton Workshop on Optimization, Learning, and Control, June 2024
New Approach: Leverage Direct AC

\[ V_t := \sum_{\tau=0}^{t-1} \phi_\tau \phi_\tau^T, \quad \phi_t := [x_t^T, u_t^T]^T \]

\[ x_{t+1} = A_x x_t + B_u u_t + \eta_{t+1} \]
Simulation Setup

- Two dynamical systems are simulated:
  - Marginally unstable Laplacian system: $A_* = I_{3 \times 3} + \begin{bmatrix} * & * & 0 \\ * & * & * \\ 0 & * & * \end{bmatrix}, \quad B_* = I_{3 \times 3}$
    - Starred elements are uncertain parameters
    - Academic system for the purpose of illustration
  - 6 DOF quadrotor with actuator LOE: $A_* = I + \Delta t A_c, \quad B_* = \Delta t B_c \text{diag}(\epsilon)$
    - LOE vector $\epsilon \in \mathbb{R}^4$ is unknown
    - Need to keep all angles small to maintain linearization

- Four algorithms are compared:
  - Baseline: optimal controller with $u_t = K_{opt} x_t, \quad K_{opt} = \text{dlqr}(A_*, B_*, Q, R)$
  - Certainty equivalence controller with decaying excitation*
  - SLS controller with FIR relaxation†
  - Our algorithm

- Measure optimality with regret
An Academic Example

- Marginally stable Laplacian dynamics
- Initial feedback gain is destabilizing due to large uncertainty

More optimal

State magnitude stays smaller
6 DOF Quadrotor with Actuator LOE

- 6DOF quadrotor (linear, discrete-time) with slight turbulence
- Front rotor has 50% LOE

More optimal
Pitch angle stays smaller
Summary

• **Adaptive Control & Intersection with learning**
  – Basics of adaptive control
  – Parameter learning
  – Imperfect learning

• **Reinforcement learning**
  – Analytical Guarantees
    • Neural network-based control (NeurIPS 1995)
    • Safe-model based RL with stability guarantee (NeurIPS 2017)

• **Synergies between AC and RL**
  – Example 1: Quadrotor
  – Example 2: Adaptive LQR
Integrative adaptive control architectures

AC-RL

Outer-loop

AC-Calibrated CBF

Inner-loop

Adaptation

Real-time Decision

Uncertainties

State-constraints

Input constraints

Safety-critical Systems

Mission goals

Learning

Princeton Workshop on Optimization, Learning, and Control, June 2024
Some Takeaways

• Real-time Control of Safety-critical Systems
  – Adapt first – requires a stability+adaptive control framework
  – Guarantees with imperfect learning are essential
  – Learning comes with hindsight

• Towards fully autonomous systems
  – Real-time decision making tools – with guarantees
  – Integration with other building blocks

• “Control for Learning” needs to be addressed
  – For decision-making under fast time-scales
For Further Reading


Thank you!

aanna@mit.edu