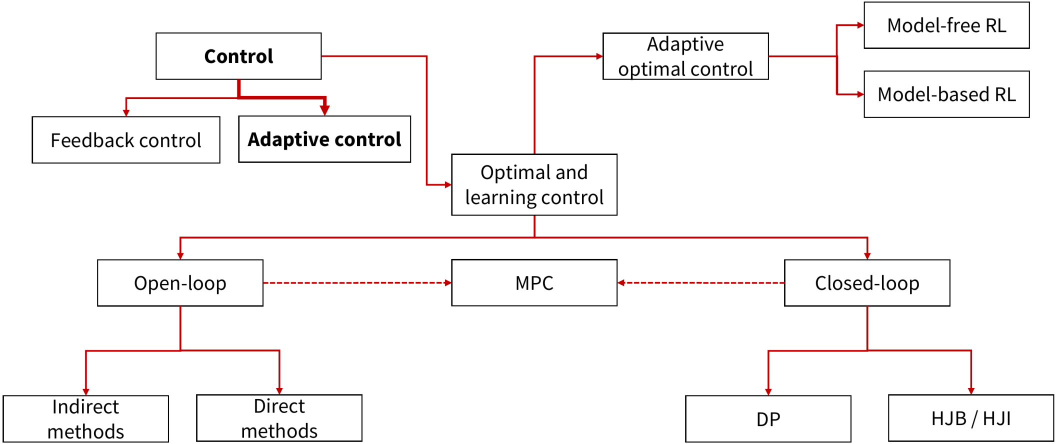
AA 203
Optimal and Learning-Based Control
System identification and adaptive control

Spencer M. Richards

Autonomous Systems Laboratory, Stanford University

May 1, 2023
(last updated May 3, 2023)

Course overview



Agenda

1. Learning objectives and settings

2. Linear-in-parameter models, persistent excitation, and Willems' fundamental lemma

3. Recursive least-squares algorithms

4. Adaptive control

Agenda

1. Learning objectives and settings
2. Linear-in-parameter models, persistent excitation, and Willems' fundamental lemma
3. Recursive least-squares algorithms
4. Adaptive control

Uncertain dynamical systems

Consider the discrete-time and continuous-time systems

$$\begin{aligned} x_{t+1} &= f(t, x_t, u_t) & \dot{x}(t) &= f(t, x(t), u(t)) \\ y_t &= h(t, x_t, u_t) & y(t) &= h(t, x(t), u(t)) \end{aligned}$$

where $x_t, x(t) \in \mathbb{R}^n$ is the state, $u_t, u(t) \in \mathbb{R}^m$ is the input, and $y_t, y(t) \in \mathbb{R}^d$ is the measured output.

We consider situations where at least one of f and h are, to some degree, *unknown*. Our goal is to use data measurements to improve control performance over time.

Handling uncertainty in dynamical systems

When uncertainties have only a small effect, *feedback* often adequately compensates for model error, e.g.,

- $\dot{x}(t) = f(t, x(t), u(t)) + \varepsilon(t)$, where f is known and $\varepsilon(t)$ is zero-mean, finite variance, uncorrelated noise.
- A quadrotor subject to *small* wind disturbances.

We can also try robust control approaches that consider average-case or worst-case disturbances, e.g.,

- $\dot{x}(t) = f(t, x(t), u(t)) + w(t)$, where f is known and a set $\mathcal{W}$ is known such that $w(t) \in \mathcal{W}$.
- A Harrier jump jet near hover during V/STOL.

The focus of this lecture is on observing state transitions to identify patterns and improve control, e.g.,

- $\dot{x}(t) = f(t, x(t), u(t))$, where f is wholly or partially *unknown*.
- An F-16 aircraft in high-speed flight subject to aerodynamic phenomena.

What can we learn?

We want to use measurements to improve control performance. Generally, there are two paradigms with which to approach this task.

- Use data to learn a better model, then use the model to improve the controller.
  - system identification
  - indirect adaptive control
  - model-based reinforcement learning
- Use data to directly improve the controller.
  - direct adaptive control
  - model-free reinforcement learning

Learning settings

In this course, we consider three settings describing how often we collect and learn from data.

- **Offline** We have access to data that has been collected previously, which we learn from prior to operation. This is a standard setting in *system identification*.
- **Online** We want to incrementally improve or re-optimize our controller in response to a stream of incoming data. This occurs in system identification, but it is more prominent in *adaptive control*.
- **Episodic** We interact with our environment in episodes, between which the system is reset. Learning and controller optimization can happen between episodes. This is a standard setting in *reinforcement learning*.

These are not exact categories – some overlap often occurs (e.g., train offline, fine-tune online).

This lecture will focus on *online* learning with *linear-in-parameter* models.

Agenda

Linear-in-parameter models

Consider the linear-in-parameter discrete-time model

$$y_t = A\phi(u_t),$$

where $y \in \mathbb{R}^d$ is the output, $u \in \mathbb{R}^m$ is the input, $A \in \mathbb{R}^{d \times p}$ is a matrix of *constant, unknown* parameters, and $\phi : \mathbb{R}^m \to \mathbb{R}^p$ is a *known* regressor function mapping inputs to *features*.

The input u_t does not necessarily correspond to just the control input in a dynamical system. For example, consider the LTI model

$$\underbrace{x_{t+1}}_{=:\text{``}y_t\text{''}} = Ax_t + Bu_t = \underbrace{\begin{bmatrix} A & B \end{bmatrix}}_{=:\text{``}A\text{''}} \underbrace{\begin{bmatrix} x_t \\ u_t \end{bmatrix}}_{=:\text{``}\phi(u_t)\text{''}},$$

where A and B are unknown. If we want to learn A and B, we can treat the current state and control input together as our "input", and measure the next state as our "output".

Linear-in-parameter models

Consider the linear-in-parameter discrete-time model

$$y_t = A\phi(u_t),$$

where $y \in \mathbb{R}^d$ is the output, $u \in \mathbb{R}^m$ is the input, $A \in \mathbb{R}^{d \times p}$ is a matrix of *constant, unknown* parameters, and $\phi : \mathbb{R}^m \to \mathbb{R}^p$ is a *known* regressor function mapping inputs to *features*.

Given data $\{(u_t, y_t)\}_{t=0}^{T-1}$ and with $z_t := \phi(u_t)$, we can estimate A as

$$\hat{A} \in \arg\min_A \sum_{t=0}^{T-1} \|y_t - Az_t\|_2^2 = \arg\min_A \|Y - AZ\|_\mathsf{F}^2.$$

where

$$Y := \begin{bmatrix} y_0 & y_1 & \cdots & y_{T-1} \end{bmatrix} \in \mathbb{R}^{m \times T}, \qquad Z := \begin{bmatrix} z_0 & z_1 & \cdots & z_{T-1} \end{bmatrix} \in \mathbb{R}^{p \times T}.$$

The stationarity condition for this optimization is the system of *normal equations*

$$\hat{A}ZZ^\mathsf{T} = YZ^\mathsf{T}.$$

Given data $\{(u_t, y_t)\}_{t=0}^{T-1}$ and with $z_t := \phi(u_t)$, we can estimate A as

$$\hat{A} \in \arg\min_A \sum_{t=0}^{T-1} \|y_t - Az_t\|_2^2 = \arg\min_A \|Y - AZ\|_{\mathsf{F}}^2.$$

The stationarity condition for this optimization is the system of *normal equations*

$$\hat{A}ZZ^{\mathsf{T}} = YZ^{\mathsf{T}},$$

where

$$ZZ^{\mathsf{T}} = \begin{bmatrix} z_0 & z_1 & \cdots & z_{T-1} \end{bmatrix} \begin{bmatrix} z_0^{\mathsf{T}} \\ z_1^{\mathsf{T}} \\ \vdots \\ z_{T-1}^{\mathsf{T}} \end{bmatrix} = \sum_{t=0}^{T-1} z_t z_t^{\mathsf{T}} \succeq 0.$$

If $ZZ^{\mathsf{T}} \succ 0$, then $\hat{A} = YZ^{\mathsf{T}}(ZZ^{\mathsf{T}})^{-1}$. A necessary and sufficient condition for this is that Z has full row rank, i.e., $\operatorname{rank} Z = p$. We need $T \geq p$ for this!

Persistent excitation

A sequence $z_{0:T-1} := \{z_t\}_{t=0}^{T-1} \subset \mathbb{R}^p$ is *persistently exciting* if

$$H_1(z_{0:T-1}) := \begin{bmatrix} z_0 & z_1 & \cdots & z_{T-1} \end{bmatrix} \in \mathbb{R}^{p \times T}$$

has full row rank, or equivalently that $H_1(z_{0:T-1})H_1(z_{0:T-1})^\mathsf{T} \succ 0$.

Define the *Hankel matrix* of depth $N \leq T$ for u by

$$H_N(z_{0:T-1}) := \begin{bmatrix} z_0 & z_1 & \cdots & z_{T-N} \\ z_1 & z_2 & \cdots & z_{T-N+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_{N-1} & z_N & \cdots & z_{T-1} \end{bmatrix} \in \mathbb{R}^{pN \times (T-N+1)}$$

Then z is *persistently exciting of order* N if $\operatorname{rank} H_N(z_{0:T-1}) = pN$, or equivalently that $H_N(z_{0:T-1})H_N(z_{0:T-1})^\mathsf{T} \succ 0$.

Willems' fundamental lemma for LTI systems

Consider the discrete-time LTI system

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \end{aligned}$$

with state $x_t \in \mathbb{R}^n$, input $u_t \in \mathbb{R}^m$, and output $y_t \in \mathbb{R}^d$.

Suppose (A, B) is controllable and we have measured a single input-output trajectory $\{(\bar{u}_t, \bar{y}_t)\}_{t=0}^{T-1}$, where $\bar{u}_{0:T-1}$ is persistently exciting of order $n + N$.

Then $\{(u_t, y_t)\}_{t=0}^{T-1}$ is an input-output trajectory as well if and only if there exists a vector $v \in \mathbb{R}^{T-N+1}$ such that

$$\begin{bmatrix} \mathrm{vec}(u_{0:T-1}) \\ \mathrm{vec}(y_{0:T-1}) \end{bmatrix} = \begin{bmatrix} H_N(\bar{u}_{0:T-1}) \\ H_N(\bar{y}_{0:T-1}) \end{bmatrix} v$$

where $\mathrm{vec}(u_{0:T-1}) \in \mathbb{R}^{mT}$ is the column vector formed from stacking $\{u_t\}_{t=0}^{T-1}$ in order.

This expresses all input-output trajectories in terms of a single persistently exciting trajectory. This construction is *model-free*, since it does not explicitly form (A, B, C, D).

Agenda

1. Learning objectives and settings
2. Linear-in-parameter models, persistent excitation, and Willems' fundamental lemma
3. Recursive least-squares algorithms
4. Adaptive control

Recursive least-squares estimation in discrete-time

Consider the linear-in-parameter discrete-time model

$$y_t = \Phi(u_t)a$$

where $y \in \mathbb{R}^d$ is the output, $u \in \mathbb{R}^m$ is the input, $a \in \mathbb{R}^p$ is a vector of *constant, unknown* parameters, and $\Phi : \mathbb{R}^m \to \mathbb{R}^{d \times p}$ is a *known* regressor function mapping inputs to *features*.

We can always re-write $y = A\phi(u)$ in this form, since

$$y = \operatorname{vec} y = \operatorname{vec}(A\phi(u)) = \underbrace{(\phi(u)^\mathsf{T} \otimes I_m)}_{=:\Phi(u)} \underbrace{\operatorname{vec} A}_{=:a},$$

where $\otimes$ is the Kronecker product and we have used that $\operatorname{vec}(ABC) = (C^\mathsf{T} \otimes A) \operatorname{vec} B$ for any conformable matrices A, B, C.

Recursive least-squares estimation in discrete-time

Consider the linear-in-parameter discrete-time model

$$y_t = \Phi(u_t)a$$

where $y_t \in \mathbb{R}^d$ is the output, $u_t \in \mathbb{R}^m$ is the input, $a \in \mathbb{R}^p$ is a vector of *constant, unknown* parameters, and $\Phi : \mathbb{R}^m \to \mathbb{R}^{d\times p}$ is a *known* regressor function mapping inputs to *features*.

Suppose we measure $\{(u_t, y_t)\}_{t=0}^{\infty}$ online as a stream of incoming data. We initialize $\hat{a}_0$ and $P_0 \succ 0$, and define the time-varying estimate

$$\hat{a}_{t+1} = \arg\min_a \left(\|a - \hat{a}_0\|^2_{P_0^{-1}} + \sum_{k=0}^{t} \|y_k - \Phi(u_k)a\|_2^2 \right)$$

where $\|a - \hat{a}_0\|^2_{P_0^{-1}} := (a - \hat{a})^\mathsf{T} P_0^{-1}(a - \hat{a}_0)$ acts as a *regularizer*.

The stationarity condition for this optimization is

$$P_0^{-1}(a - \hat{a}_0) + \sum_{k=0}^{t-1} \Phi(u_k)^\mathsf{T}(\Phi(u_k)\hat{a}_{t+1} - y_t) = 0$$

Recursive least-squares estimation in discrete-time

Suppose we measure $\{(u_t, y_t)\}_{t=0}^{\infty}$ online as a stream of incoming data. We initialize $\hat{a}_0$ and $P_0 \succ 0$, and define the time-varying estimate

$$\hat{a}_{t+1} = \arg\min_a \left(\|a - \hat{a}_0\|_{P_0^{-1}}^2 + \sum_{k=0}^{t} \|y_k - \Phi(u_k)a\|_2^2, \right)$$

where $\|a - \hat{a}_0\|_{P_0^{-1}}^2 := (a - \hat{a})^\mathsf{T} P_0^{-1}(a - \hat{a}_0)$ acts as a *regularizer*.

The stationarity condition for this optimization is

$$P_0^{-1}(\hat{a}_{t+1} - \hat{a}_0) + \sum_{k=0}^{t} \Phi(u_k)^\mathsf{T}(\Phi(u_k)\hat{a}_{t+1} - y_k) = 0.$$

Define $P_{t+1}^{-1} := P_0^{-1} + \sum_{k=0}^{t} \Phi(u_k)^\mathsf{T}\Phi(u_k)$ and rearrange to get

$$P_{t+1}^{-1}\hat{a}_{t+1} = P_0^{-1}\hat{a}_0 + \sum_{k=0}^{t} \Phi(u_k)^\mathsf{T} y_k.$$

Recursive least-squares estimation in discrete-time

Define $P_{t+1}^{-1} := P_0^{-1} + \sum_{k=0}^{t} \Phi(u_k)^\mathsf{T}\Phi(u_k)$ and rearrange to get

$$P_{t+1}^{-1}\hat{a}_{t+1} = P_0^{-1}\hat{a}_0 + \sum_{k=0}^{t} \Phi(u_k)^\mathsf{T} y_k.$$

From $P_{t+1}^{-1} = P_t^{-1} + \Phi(u_t)^\mathsf{T}\Phi(u_t)$ and the Woodbury formula

$$(A + UBV)^{-1} = A^{-1} - A^{-1}U(B^{-1} + VA^{-1}U)^{-1}VA^{-1},$$

we have

$$P_{t+1} = P_t - \underbrace{P_t\Phi(u_t)^\mathsf{T}\left(I + \Phi(u_t)P_t\Phi(u_t)^\mathsf{T}\right)^{-1}}_{=:K_t}\Phi(u_t)P_t = (I - K_t\Phi(u_t))P_t$$

where K_t is the *Kalman gain*.

Recursive least-squares estimation in discrete-time

We have $P_{t+1}^{-1}\hat{a}_{t+1} = P_0^{-1}\hat{a}_0 + \sum_{k=0}^{t} \Phi(u_k)^\mathsf{T} y_k$ and $P_{t+1} = (I - K_t\Phi(u_t))P_t$, so

$$\begin{aligned}
\hat{a}_{t+1} &= P_{t+1}\left(P_0^{-1}\hat{a}_0 + \sum_{k=0}^{t} \Phi(u_k)^\mathsf{T} y_k\right) \\
&= (I - K_t\Phi(u_t))P_t\left(\Phi(u_t)^\mathsf{T} y_t + P_0^{-1}\hat{a}_0 + \sum_{k=0}^{t-1} \Phi(u_k)^\mathsf{T} y_k\right) \\
&= (I - K_t\Phi(u_t))\big(P_t\Phi(u_t)^\mathsf{T} y_t + \hat{a}_t\big)
\end{aligned}$$

Use the fact that $K_t = (I - K_t\Phi(u_t))P_t\Phi(u_t)^\mathsf{T}$ to get $\hat{a}_{t+1} = \hat{a}_t + K_t(y_t - \Phi(u_t)\hat{a}_t)$, so overall

$$\begin{aligned}
K_t &= P_t\Phi(u_t)^\mathsf{T}\big(I + \Phi(u_t)P_t\Phi(u_t)^\mathsf{T}\big)^{-1} \\
\hat{a}_{t+1} &= \hat{a}_t + K_t(y_t - \Phi(u_t)\hat{a}_t) \\
P_{t+1} &= (I - K_t\Phi(u_t))P_t
\end{aligned}$$

with user-specified initial conditions $\hat{a}_0 \in \mathbb{R}^p$ and $P_0 \succ 0$.

Recursive least-squares estimation in continuous-time

Consider the linear-in-parameter continuous-time model

$$y(t) = \Phi(u(t))a,$$

where $y(t) \in \mathbb{R}^d$ is the output, $u(t) \in \mathbb{R}^m$ is the input, $a \in \mathbb{R}^p$ is a vector of *constant, unknown* parameters, and $\Phi : \mathbb{R}^m \to \mathbb{R}^{d\times p}$ is a *known* regressor function mapping inputs to *features*.

We initialize $\hat{a}(0) \in \mathbb{R}^p$ and $P(0) \succ 0$, and set our estimate $\hat{a}(t)$ such that

$$\hat{a}(t) = \arg\min_a \left(\|a - \hat{a}(0)\|^2_{P(0)^{-1}} + \int_0^t \|y(s) - \Phi(u(s))a\|_2^2 \, ds. \right)$$

The stationarity condition for this optimization is

$$P(0)^{-1}(\hat{a}(t) - \hat{a}(0)) + \int_0^t \Phi(u(s))^\mathsf{T}(\Phi(u(s))\hat{a}(t) - y(s)) \, ds = 0.$$

Recursive least-squares estimation in continuous-time

We initialize $\hat{a}(0)$ and set our estimate $\hat{a}(t)$ such that

$$\hat{a}(t) = \arg\min_a \left(\|a - \hat{a}(0)\|^2_{P(0)^{-1}} + \int_0^t \|y(s) - \Phi(u(s))a\|_2^2 \, ds. \right)$$

The stationarity condition for this optimization is

$$P(0)^{-1}(\hat{a}(t) - \hat{a}(0)) + \int_0^t \Phi(u(s))^\mathsf{T}(\Phi(u(s))\hat{a}(t) - y(s)) \, ds = 0.$$

Set $\frac{d}{dt}(P(t)^{-1}) = \Phi(u(t))^\mathsf{T}\Phi(u(t))$ with initial condition $P(0)^{-1}$ and rearrange to get

$$P(t)^{-1}\hat{a}(t) = P(0)^{-1}\hat{a}(0) + \int_0^t \Phi(u(s))^\mathsf{T} y(s) \, ds.$$

Differentiate with respect to t and rearrange to get

$$\dot{\hat{a}}(t) = P(t)\Phi(u(t))^\mathsf{T}(y(t) - \Phi(u(t))\hat{a}(t)).$$

Recursive least-squares estimation in continuous-time

Overall, the parameter update law is

$$\dot{\hat{a}}(t) = P(t)\Phi(u(t))^\mathsf{T}(y(t) - \Phi(u(t))\hat{a}(t))$$
$$\frac{d}{dt}(P(t)^{-1}) = \Phi(u(t))^\mathsf{T}\Phi(u(t))$$

with user-specified initial conditions $\hat{a}(0) \in \mathbb{R}^p$ and $P(0) \succ 0$.

Since $P(t)P(t)^{-1} = I$, we must have

$$\frac{d}{dt}\Big(P(t)P(t)^{-1}\Big) = \dot{P}(t)P(t)^{-1} + P(t)\frac{d}{dt}(P(t)^{-1}) = 0 \implies \dot{P}(t) = -P(t)\frac{d}{dt}(P(t)^{-1})P(t).$$

So we can use the more convenient update

$$\dot{\hat{a}}(t) = P(t)\Phi(u(t))^\mathsf{T}(y(t) - \Phi(u(t))\hat{a}(t))$$
$$\dot{P}(t) = -P(t)\Phi(u(t))^\mathsf{T}\Phi(u(t))P(t)$$

with user-specified initial conditions $\hat{a}(0) \in \mathbb{R}^p$ and $P(0) \succ 0$.

Agenda

1. Learning objectives and settings
2. Linear-in-parameter models, persistent excitation, and Willems' fundamental lemma
3. Recursive least-squares algorithms
4. Adaptive control

Adaptive control

So far, we have looked at recursive estimators that, for persistently excited linear-in-parameter systems, ensure estimated parameter values converge to those of the true system.

Suppose we knew a controller $u(t) = \pi(t, x(t), a)$ that could stabilize the system

$$\dot{x}(t) = \Phi(t, x(t), u(t))a,$$

e.g., ensure $\lim_{t\to\infty} x(t) = 0$ for all $x(0)$.

If we do not know a, we could simultaneously update an estimate $\hat{a}(t)$ to minimize model error while applying the controller $u(t) = \pi(t, x(t), \hat{a}(t))$. This is sometimes referred to as *model identification adaptive control (MIAC)*.

However, even if $\lim_{t\to\infty} \hat{a}(t) = a$, there is no guarantee that $u(t) = \pi(t, x(t), \hat{a}(t))$ will stabilize the system, since intermediate values of $\hat{a}(t)$ can be arbitrarily poor estimates of a.

Adaptive control focuses on stable concurrent learning and control, such that adaptation of $\hat{a}(t)$ over time alongside feedback $u(t) = \pi(t, x(t), \hat{a}(t))$ is still guaranteed to stabilize the system.

Certainty-equivalent adaptive control with matched uncertainty

Let $\dot{x} = f(x) + B(x)(u + \Phi(x)a)$, where $a \in \mathbb{R}^p$ is a vector of unknown parameters.

Suppose we know a radially unbounded, positive-definite function $\bar{V} : \mathbb{R}^n \to \mathbb{R}$ and a controller $u = \bar{\pi}(x)$ such that

$$\nabla V(x)^{\mathsf{T}}(f(x) + B(x)\bar{\pi}(x)) \leq -\rho(x)$$

for some positive-definite $\rho : \mathbb{R}^n \to \mathbb{R}$. That is, $u = \bar{\pi}(x)$ would stabilize the system if $a = 0$, which we could prove with $\dot{\bar{x}} = f(\bar{x}) + B(\bar{x})\bar{\pi}(\bar{x})$ and Lyapunov function $\bar{V}$.

For the uncertain system, we propose the controller

$$u = \bar{\pi}(x) - \Phi(x)\hat{a}(t),$$

with parameter estimate $\hat{a}(t)$. If $\hat{a}(t) = a$, then the second term of this controller would cancel out $\Phi(x)a$ and leave us with the stable nominal dynamics.

For this reason, the uncertain term $\Phi(x)a$ is referred to as a *matched uncertainty*. The concept of using a fixed estimate $\Phi(x)\hat{a}(t)$ in feedback on the true system as if its "correct" is referred to as *certainty equivalent control*.

Certainty-equivalent adaptive control with matched uncertainty

How do we update $\hat{a}(t)$ over time such that the certainty-equivalent controller $u = \bar{\pi}(x) - \Phi(x)\hat{a}(t)$ stabilizes $\dot{x} = f(x) + B(x)(u + \Phi(x)a)$?

Consider the augmented Lyapunov candidate function

$$V(x, \hat{a}) := \bar{V}(x) + \frac{1}{2}\|\hat{a} - a\|^2_{\Gamma^{-1}},$$

for some *adaptation gain* $\Gamma \succ 0$. Then for $u = \bar{\pi}(x) - \Phi(x)\hat{a}$ we have

$$\begin{aligned}
\dot{V}(x, \hat{a}) &= \nabla\bar{V}(x)^\mathsf{T}(f(x) + B(x)(\bar{\pi}(x) - \Phi(x)(\hat{a} - a))) + (\hat{a} - a)^\mathsf{T}\Gamma^{-1}\dot{\hat{a}} \\
&= \nabla\bar{V}(x)^\mathsf{T}(f(x) + B(x)\bar{\pi}(x)) - \nabla\bar{V}(x)^\mathsf{T}B(x)\Phi(x)(\hat{a} - a) + (\hat{a} - a)^\mathsf{T}\Gamma^{-1}\dot{\hat{a}} \\
&= \underbrace{\nabla\bar{V}(x)^\mathsf{T}(f(x) + B(x)\bar{\pi}(x))}_{\leq -\rho(x)} - (\hat{a} - a)^\mathsf{T}\left(\Phi(x)^\mathsf{T}B(x)^\mathsf{T}\nabla\bar{V}(x) - \Gamma^{-1}\dot{\hat{a}}\right)
\end{aligned}$$

If we choose the adaptation law

$$\dot{\hat{a}} = \Gamma\Phi(x)^\mathsf{T}B(x)^\mathsf{T}\nabla\bar{V}(x),$$

then $\dot{V}(x, \hat{a})$ is a Lyapunov function for the closed-loop system with controller $u = \bar{\pi}(x) - \Phi(x)\hat{a}$, which ensures $\lim_{t\to\infty} x(t) = 0$.

Certainty-equivalent adaptive control with matched uncertainty

If we choose the adaptation law

$$\dot{\hat{a}} = \Gamma \Phi(x)^\mathsf{T} B(x)^\mathsf{T} \nabla \bar{V}(x),$$

then $\dot{V}(x, \hat{a})$ is a Lyapunov function for the closed-loop system with controller $u = \bar{\pi}(x) - \Phi(x)\hat{a}$, which ensures $\lim_{t\to\infty} x(t) = 0$.

In fact, this choice of adaptation law ensures

$$\frac{d}{dt} V(x, \hat{a}) \bigg|_{\dot{x}=f(x)+B(x)(\bar{\pi}(x)-\Phi(x)(\hat{a}-a))} = \frac{d}{dt} \bar{V}(x) \bigg|_{\dot{x}=f(x)+B(x)\bar{\pi}(x)},$$

i.e., the Lyapunov "energy" of the adaptive system evolves in the same manner as in the nominal dynamics, so any stability analysis "carries over" to the adaptive system.

Overall, we achieved closed-loop stability, without any guarantees on how well $\hat{a}(t)$ estimates a!

Adaptive control learns on a "need-to-know" basis to cancel $\Phi(x)a$ in closed-loop, rather than to estimate a in open-loop (Slotine and Li, 1991; Richards et al., 2021, 2023).

Next class

Reinforcement learning

References

S. M. Richards, N. Azizan, J.-J. Slotine, and M. Pavone. Adaptive-control-oriented meta-learning for nonlinear systems. In *Robotics: Science and Systems*, 2021. URL https://arxiv.org/abs/2204.06716.

S. M. Richards, N. Azizan, J.-J. Slotine, and M. Pavone. Control-oriented meta-learning. *Int. Journal of Robotics Research*, 2023. URL https://arxiv.org/abs/2103.04490. In press.

J.-J. E. Slotine and W. Li. *Applied Nonlinear Control*. 1991.