Learning-Based Control:

A Theory based on Robust Adaptive Dynamic Programming

Zhong-Ping Jiang

Control and Networks (CAN) Lab
New York University (NYU)

Small-Gain Theory: Robust Nonlinear Control Design

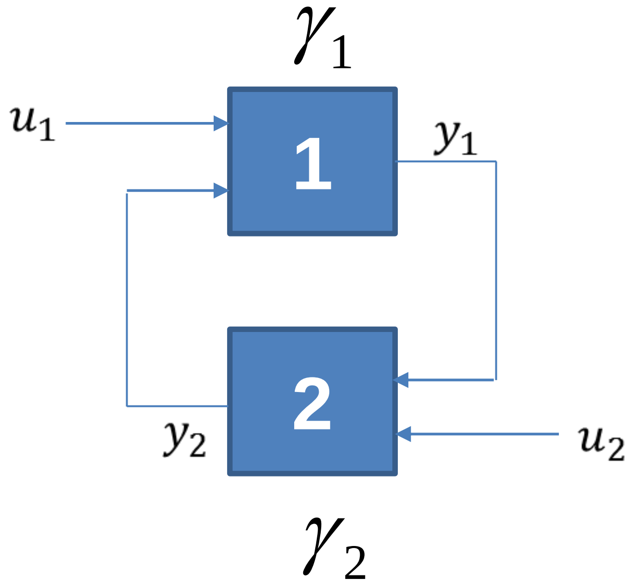
Plenary lecture @2003 Chinese Control Conference

Tolstoy(托尔斯泰):
All happy families are alike.

Why so?

because they all satisfy the small-gain condition!

$\gamma_1 \circ \gamma_2 < Id$ implies "network stability".



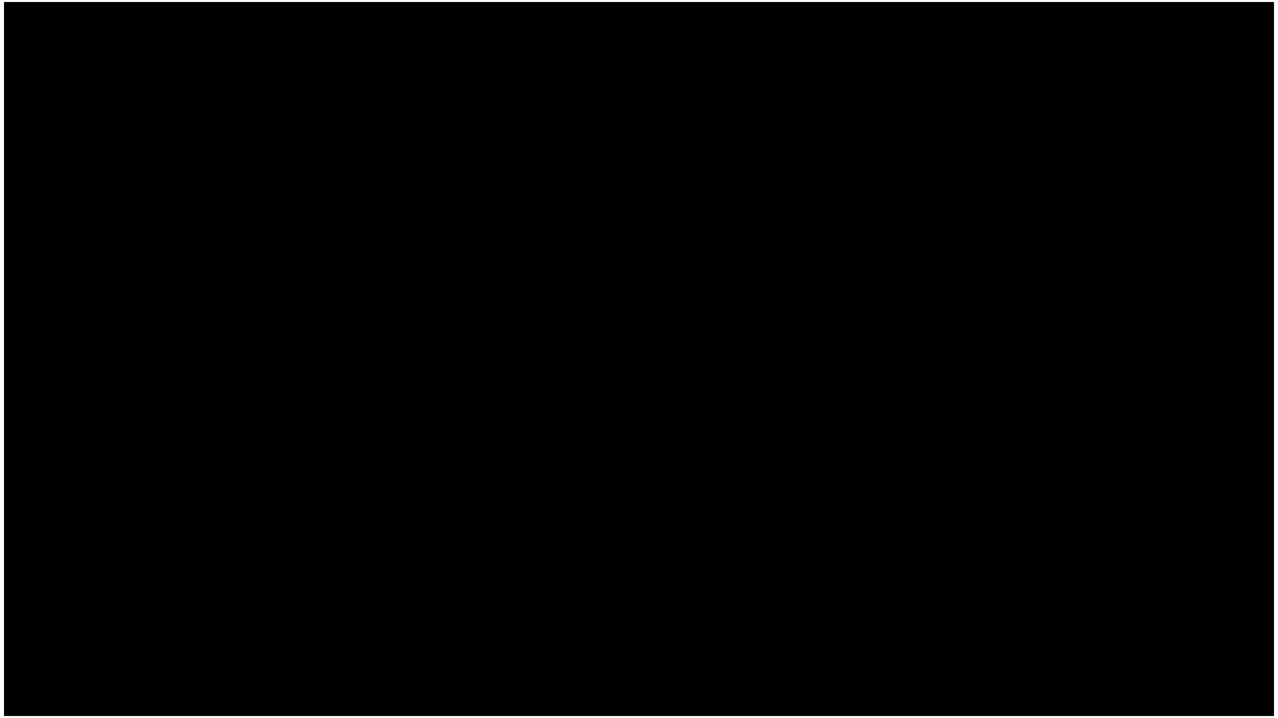
where γ_1 : gain from y_2 to y_1 and γ_2 : gain from y_1 to y_2

Small-Gain Theory 2

第二集

Learning-Based Control

Why Learning-Based Control Theory?



Data and Learning-Based Control

Use i/o data to learn better (adaptive/optimal) controllers in the absence of exact model knowledge

- **Rapid response**
- **Stability/robustness/safety guarantee**
- **Optimality for reduced energy consumption**

Adaptive Optimal Control Problem

How to solve $\min J(x_0;u)=\int_{t_0}^{\infty} r(x,u)dt$

subject to

$$\dot{x}=f\left(x,u\right),\ \text{with unknown}\ f$$

Model-based approach

- **For linear systems, many published papers by several authors:**

  Guo/Duncan/Pasik-Duncan, Bitmead, Kumar, HF Chen, etc

- **For nonlinear systems,**
  "Almost" None

Limitations of Dynamic Programming (DP)

How to solve $\min J(x_0;u) = \int_{t_0}^{\infty} r(x,u)\,dt$

subject to

$\dot{x} = f\left(x,u\right)$, with unknown f

Bellman's Dynamic Programming is <u>not</u> applicable, because of

- Curse of dimensionality (Bellman, 1959)
- Curse of modeling (Bertsekas, 1996)

Outline

- Data-Driven Learning-based Control Theory: Why?
- Robust Adaptive Dynamic Programming
  - Adaptive LQR for continuous-time linear systems
  - Extensions: nonlinear and robust
- Application:

  Connected and Autonomous Vehicles
- Conclusions and Future Work

NYU

Why Data-driven Learning-based Control?

- "System modeling is expensive, time consuming, and inaccurate." **(Frank Lewis @ASCC'09)**

- **Brought together "stability" and "reinforcement learning" (for c-t systems)**

- **"Adaptive Dynamic Programming" (ADP):**

An active research area, integrating reinforcement learning (RL) and controls to remove the curses of dimensionality and of modeling.

Reinforcement Learning (RL)

Model-free approach

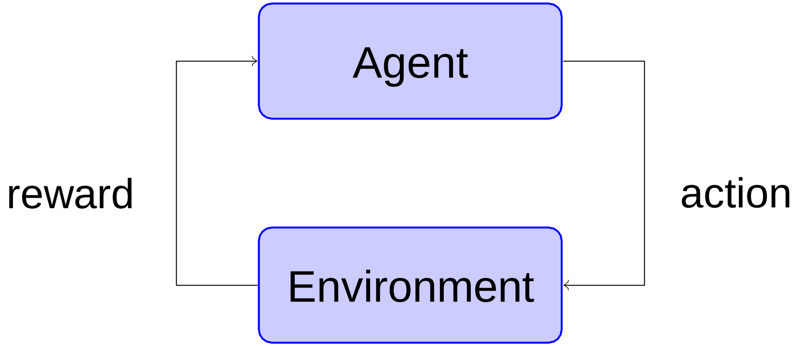


Figure: Reinforcement Learning (Minsky, 1954).

Maximizing the cumulative reward, through

1) Exploration (finding better policies).

2) Agent-environment interaction.

Many papers

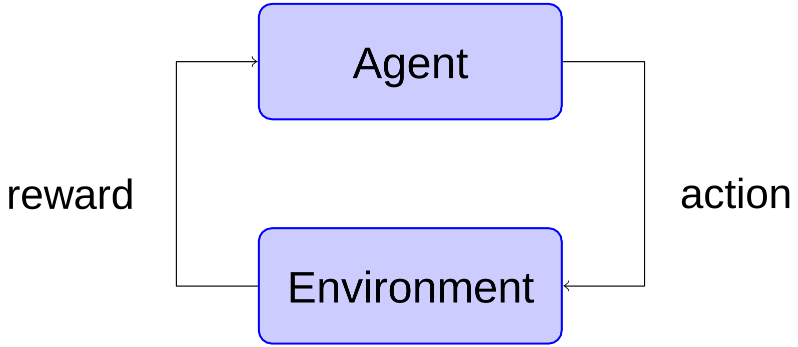


Figure: Reinforcement Learning (Minsky, 1954).

1. CS perspective (Barto, Dayan, Sutton, Watkins, Kaelbling, Littman, Doya).
   Finite state/action space; early ideas.

2. OR perspective (Bertsekas, Tsitsiklis, Van Roy, Nedic, Borkar, Powell).
   Countable state/action space; advanced convergence analysis (stochastic approximation, function approximation).

NYU

The history of DP and RL

1952	Dynamic programming	(Bellman)
1954	Reinforcement learning	(Minsky)
1960	DP algorithms (VI, PI) for MDPs	(Bellman; Howard)
1960s	Positive & Negative DP	(Blackwell; Strauch)
1968	RL + approximate DP	(Werbos)
1983	Actor-critic algorithm	(Barto)
1984	TD-learning	(Sutton)
1989	Q-learning	(Watkins)
1990s	Neuro-DP	(Bertsekas)
2010s	Adaptive DP	(Lewis)
2013	Abstract DP	(Bertsekas)

2012 – Robust Adaptive DP (here)

Continuous-time ADP: still in its infancy

Outline

- Data-Driven Learning-based Control Theory: Why?
- Robust Adaptive Dynamic Programming
  - Adaptive LQR for continuous-time linear systems
  - Extensions: nonlinear and robust
- Application:

  Connected and Autonomous Vehicles
- Conclusions and Future Work

LTI system $\dot{x} = Ax + Bu,\ x(0) = x_0$

When (A, B) are **unknown**, find an “i/s data-driven” linear control policy

$u = -Kx$

that minimizes $J = \int_0^\infty (x^T Q x + u^T R u) dt = x_0^T P x_0$

where $Q = Q^T \geq 0,\ R = R^T > 0,\ (A, B)$ is controllable, and $(A, Q^{1/2})$ is observable.

Linear-quadratic regulator (LQR)

Optimal Controller:

- $u^* = -K^* x,$
- $\mathcal{J}(x_0; u^*) = x_0^T P^* x_0,\ with\ P^* = P^{*T} > 0$

Algebraic Riccati equation: $A^T P^* + P^* A - P^* B R^{-1} B^T P^* + Q = 0, \quad K^* = R^{-1} B^T P^*.$

Question 1:

How to learn suboptimal controllers, from i/s data, that converge to the (unknown) optimal controller?

1. Policy Iteration (PI)
2. Value Iteration (VI)

Learning-based Policy Iteration

Assume the knowledge of an initial stabilizing policy K_0

From

$$\int_t^{\infty}\left(x^T Q x+u^T R u\right)d\tau=\int_t^{t+\delta t}\left(x^T Q x+u^T R u\right)d\tau+\int_{t+\delta t}^{\infty}\left(x^T Q x+u^T R u\right)d\tau$$

Integral RL equation.

ADP for partially unknown linear systems (Lewis et al., 2009)

$$x^T(t)P_j x(t)=\int_t^{t+\delta t} x^T\left(Q+K_j^T R K_j\right)x d\tau+x^T(t+\delta t)P_j x(t+\delta t),$$

$$K_{j+1}=R^{-1}B^T P_j.$$

B is required. $x(t)$ is generated by $u=-K_j x$ (on-policy).

Under mild conditions,

$$P_j \to P^*,\quad K_j \to K^*.$$

Learning-based Policy Iteration

ADP for partially unknown linear systems (Lewis et al., 2009)

$$x^T(t)P_jx(t) = \int_t^{t+\delta t} x^T(Q + K_j^T RK_j)x d\tau + x^T(t+\delta t)P_jx(t+\delta t),$$

$$K_{j+1} = R^{-1}B^T P_j.$$

B is required. $x(t)$ is generated by $u = -K_j x$ (on-policy).

Integral RL equation.

ADP for fully unknown linear systems (Jiang & ZPJ, 2012)

$$x^T(t)P_jx(t) = \int_t^{t+\delta t} \left(x^T(Q + K_j^T RK_j)x - 2(K_{j+1}x)^T R(u' + K_jx)\right) d\tau + x^T(t+\delta t)P_jx(t+\delta t),$$

B is not required. x is generated by $u = u'$ (off-policy). Usually, we choose

$u' = -K_jx + \xi$ or $u' = -K_0x + \xi$

On-line Off-policy PI-based ADP algorithm

Jiang & ZPJ, 2012

Collecting i/s data over $[t_i, t_{i+1}]$, $i = 0, \ldots, l-1$,

$$\Theta_k \begin{bmatrix} \hat{P}_k \\ vec(K_{k+1}) \end{bmatrix} = \Xi_k \qquad (**)$$

$$\Theta_k = \left[\delta_{xx} - 2I_{xx}\left(I_n \otimes K_k^T R\right) - 2I_{xu}\left(I_n \otimes R\right)\right] \in \mathbb{R}^{l \times \left(\frac{n(n+1)}{2} + nm\right)},$$

$$\Xi_k = -I_{xx} vec\left(Q + K_k^T R K_k\right),$$

For $P \in \mathbb{R}^{n \times n}$ and $x \in \mathbb{R}^n$,

$$\delta_{xx} = [\bar{x}(t_1) - \bar{x}(t_0), \bar{x}(t_2) - \bar{x}(t_1), \cdots, \bar{x}(t_l) - \bar{x}(t_{l-1})]^T \in \mathbb{R}^{l \times \frac{n(n+1)}{2}},$$

$$I_{xx} = \left[\int_{t_0}^{t_1} x \otimes x d\tau, \int_{t_1}^{t_2} x \otimes x d\tau, \cdots, \int_{t_{l-1}}^{t_l} x \otimes x d\tau\right]^T \in \mathbb{R}^{l \times n^2},$$

$$I_{xu} = \left[\int_{t_0}^{t_1} x \otimes u d\tau, \int_{t_1}^{t_2} x \otimes u d\tau, \cdots, \int_{t_{l-1}}^{t_l} x \otimes u d\tau\right]^T \in \mathbb{R}^{l \times nm},$$

$$\bar{x} = [x_1^2, x_1 x_2, \cdots, x_1 x_n, x_2^2, x_2 x_3, \cdots, x_{n-1} x_n, x_n^2,]^T \in \mathbb{R}^{\frac{n(n+1)}{2}},$$

$$\hat{P} = [p_{11}, 2p_{12}, \cdots, 2p_{1n}, p_{22}, 2p_{23}, \cdots, 2p_{2n}, p_{nn}]^T \in \mathbb{R}^{\frac{n(n+1)}{2}}.$$

1) Full rank of Θ_k

$\Rightarrow$ unique solution of (**)

(due to exploration noise ξ)

2) $P_k \to P^*$, $K_k \to K^*$ as $k \to \infty$.

3) Stability + suboptimality

without ξ.

Learning-based Value Iteration

Question:

Can we <u>remove</u> the assumption on the knowledge of an initial, stabilizing policy K_0, when the system dynamics are <u>not</u> known?

Yes!

Generalize & apply the “Value Iteration” (VI) method to continuous-time dynamical systems.

History of Value Iteration (VI)

Value iteration:

- 1959 — VI for MDPs (Bellman)
- 1960 — The name of Value iteration was introduced (Howard)
- 1995 — VI for DT linear systems. (Lancaster & Rodman)
- 2015 — VI for DT nonlinear systems (Bertsekas, Lewis, …)

Policy iteration:

- 1960 — PI for MDPs (Howard)
- 1969 — PI for CT linear systems. (Kleinman)
- 1976 — PI for DT linear systems. (Bertsekas)
- 1995 — PI for CT affine nonlinear systems (Beard & Saridis)
- 2014 — PI for CT nonaffine nonlinear systems (Bian, ZPJ, etc)
- 2015 — PI for DT nonlinear systems (Bertsekas, D. Liu, Lewis, …)

- ➢ VI is more difficult.
- ➢ It is still an open problem to develop VI for continuous-time systems.
- ➢ We give a VI by combining DMRE and stochastic approximation theory.

VI for Linear-quadratic regulator (LQR)

Bian & ZPJ, Automatica, 2016

Continuous-time VI: $\lim_{t\to\infty} M(t) = P^*$, where

$$\dot{M} = A^T M + MA - MBR^{-1}B^T M + Q, \qquad M(0) = M^T(0) > 0$$

Stochastic Approximation:

$$\theta_{t+1} = \theta_t + \epsilon_t(g(\theta_t) + \delta M_t) + Z_t$$

where

- Z_t is a projection term;
- ϵ_t is the step size;
- $\{\delta M_t\}$ is a sequence of i.i.d random variables, $E[\delta M_t] = 0, Var[\delta M_t] < \infty$;
- $g(\cdot)$ is measurable and locally Lipschitz.

Convergence: $\theta_t \to \theta^*$ with probability 1,

$\dot{\theta} = g(\theta)$ is asymptotically stable at θ^*. *(Kushner-Yin, 2003)*

VI for Linear-quadratic regulator (LQR)

$\{B_p\}_{q=0}^{\infty}$: $B_q \subseteq B_{q+1}, \lim_{q\to\infty} B_q = \{P \in \mathbb{R}^{n\times n}: P^T = P \geq 0\}$

$\{\epsilon_k\}_{k=0}^{\infty}$: $\epsilon_k > 0, \lim_{k\to\infty} \epsilon_k = 0, \sum_{k=0}^{\infty} \epsilon_k = \infty$. $\varepsilon > 0$ is a threshold

---

Algorithm 1 SA-based continuous-time VI algorithm (Bian & ZPJ, 2016):

Choose $P_0 = P_0^T > 0.\, k, q \leftarrow 0.$

Loop

$\tilde{P}_{k+1} \leftarrow P_k + \epsilon_k(A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q)$

if $\tilde{P}_{k+1} \notin B_q$ **then**

$P_{k+1} \leftarrow P_0.\, q \leftarrow q + 1$

else if $\frac{|\tilde{P}_{k+1} - P_k|}{\epsilon_k} < \varepsilon$ **then return** $\hat{P}^* = P_k$

else $P_{k+1} \leftarrow \hat{P}_{k+1}$

$k \leftarrow k + 1$

- Convergence: $P_k \to P^*$
- Stability: If $Q > \varepsilon I_n$, then $A - B\hat{K}^*$ is Hurwitz, where $\hat{K}^* = R^{-1} B^T \hat{P}^*$, and $\hat{P}^*$ is obtained from the VI algorithm.

On-line Off-policy ADP algorithm (Bian & ZPJ, 2016):

Solve H_k and K_k from

$$x^T(t+\delta t)P_k(t+\delta t) = \int_t^{t+\delta t} x^T H_k x ds + 2\int_t^{t+\delta t} u^T R K_k x ds + x^T(t) P_k x(t). \quad \textbf{(1)}$$

where $H_k = A^T P_k + P_k A$.

Continuous-time VI-based ADP algorithm

Choose $P_0 = P_0^T > 0. k, q \leftarrow 0.$
Apply a locally bounded input u to the system.
Loop

Solve (H_k, K_k) from (1)

$\tilde{P}_{k+1} \leftarrow P_k + \epsilon_k(A^T P_k + P_k A - P_k B R^{-1} B^T P_k + Q)$

$= P_k + \epsilon_k(H_k - K_k^T R K_k + Q)$

if $\tilde{P}_{k+1} \notin B_q$ **then**

$P_{k+1} \leftarrow P_0. q \leftarrow q + 1$

else if $\frac{|\tilde{P}_{k+1} - P_k|}{\epsilon_k} < \varepsilon$ **then return** $\hat{P}^* = P_k$

else $P_{k+1} \leftarrow \hat{P}_{k+1}$

$k \leftarrow k + 1$

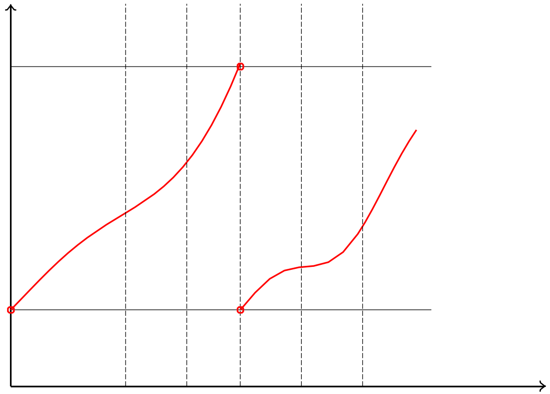


Figure: Bounded learning trajectory.

How about general nonlinear systems?

$$\dot{x} = f(x,u), \qquad \min \quad \mathcal{J}(x_0;u) = \int_0^\infty r(x,u)dt\,.$$

with unknown dynamics *f* ?

- **No distinction between linear and nonlinear problems**
- **First solution to adaptive/nonlinear optimal control**

Policy iteration for nonaffine systems

(Bian, Jiang & ZPJ, 2014)

Nonaffine system:

$$\dot{x} = f(x,u), \qquad \mathcal{J}(x_0;u) = \int_0^\infty r(x,u)dt\,.$$

HJB equation:

$$0 = \min_{v\in\mathbb{R}^m}\{\partial_x V^*(x) f(x,v) + r(x,v)\}, \qquad V^*(0) = 0,$$

$$\mu^*(x) = \arg\min_{v\in\mathbb{R}^m}\{\partial_x V^*(x) f(x,v) + r(x,v)\}$$

Policy iteration

1. Policy evaluation: $\partial_x V_j(x) f\left(x, \mu_j(x)\right) + r\left(x, \mu_j(x)\right) = 0, V_j(0) = 0.$
2. Policy improvement: $\mu_{j+1}(x) = \arg\min_{v\in\mathbb{R}^m}\{\partial_x V^*(x) f(x,v) + r(x,v)\}, \forall x \in \mathbb{R}^n$

- Convergence: If μ_0 is admissible, $V_j \to V^*, \mu_j \to \mu^*$.
- Stability: μ_j is stabilizing and, $\mathcal{J}(x_0; u_j(x)) < \infty$

NYU

Basis function approximation

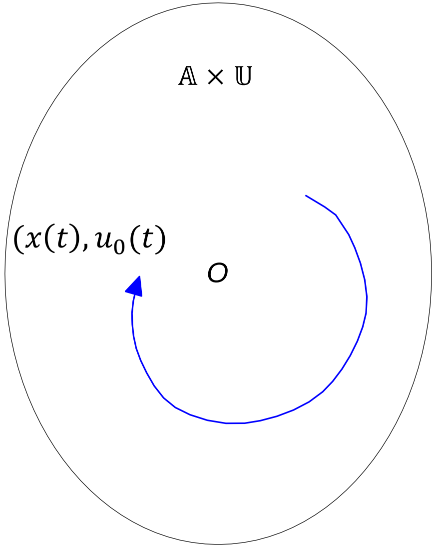
Using basis function approximation, we have for all $x \in \mathbb{A}$ and $v \in \mathbb{U}$,

$$V_j(x) = \sum_{i=1}^{N} \hat{w}_i^j \phi_i(x) + e_\phi^j(x),$$

$$\partial_x V_j(x) f(x,v) = \sum_{i=1}^{N} \hat{c}_i^j \psi_i(x,v) + e_\psi^j(x,v),$$

$$\mu_j(x) = \sum_{i=1}^{N} \hat{l}_i^j \theta_i(x) + e_\theta^j(x)$$

Linear-like problem



- $\{\phi_i\}_{i=1}^N, \{\psi_i\}_{i=1}^N, and\ \{\theta_i\}_{i=1}^N,$
  with $\phi_i: \mathbb{R}^n \to \mathbb{R}, \psi_i: \mathbb{R}^n \times \mathbb{R}^m \to R,\ and\ \theta_i: \mathbb{R}^n \to \mathbb{R}^m,$
  are three sets of linearly independent and continuous functions;
- $e_\phi^j, e_\psi^j, and\ e_\theta^j$ are the approximation errors.

ADP algorithm: Semiglobal stabilization

Assumption (Persistent excitation (PE))

For all $\{\hat{\mu}_j\}_{j=0}^{\infty}$, there exist $\bar{M} > 0$ and $\gamma > 0$, such that for all $M \geq \bar{M}$,

$\frac{1}{M}\sum_{k=1}^{M} {\Theta_k^j}^T \Theta_k^j \geq \gamma I_{2N}$, $\Theta_k^j \in \mathbb{R}^{1\times 2N}$ is the vector of input-state data

The ADP algorithm:

1. Apply $u_0(t)$ to the system. $j \leftarrow 0$.
2. Policy evaluation: $$[\hat{w}^j, \hat{c}^j]^T = -\left(\sum_{k=1}^{M} {\Theta_k^j}^T \Theta_k^j\right)^{-1} \sum_{k=1}^{M} {\Theta_k^j}^T \int_{t_{k-1}}^{t_k} r\left(x, \hat{\mu}_j(x)\right) dt.$$
3. Policy update: $$\hat{l}^{j+1} = \arg\min_{\{l \mid l\theta(x)\in\mathbb{U}\}} \{\hat{c}^j \psi(x, l\theta(x)) + r(x, l\theta(x))\}, \hat{\mu}_j = \hat{l}^j\theta.$$

Convergence on $\mathbb{A}$:

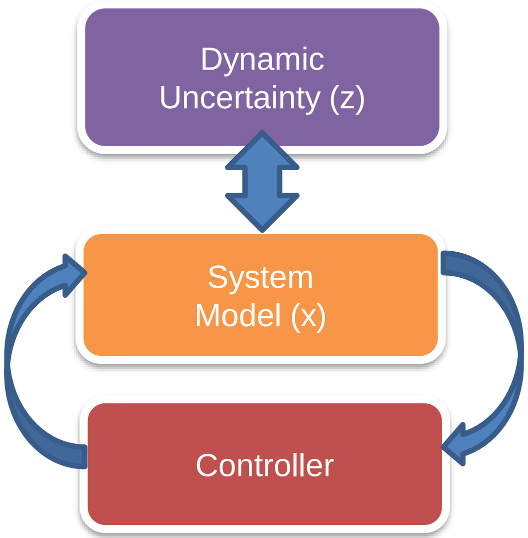
$$\lim_{N\to\infty} \left|\sum_{i=1}^{N} \hat{l}_i^j \theta_i(x) - \mu_j(x)\right| = 0, \quad \lim_{N\to\infty} \left|\sum_{i=1}^{N} \hat{w}_i^j \phi_i(x) - V_j(x)\right| = 0.$$

Bian, Jiang & ZPJ, 2014

Question 2:

How to learn suboptimal controllers with guaranteed robustness to dynamic uncertainties?

Robust Adaptive Dynamic Programming



dim(z, x) unknown,

with possibly huge dim(z)

Dynamic uncertainties:

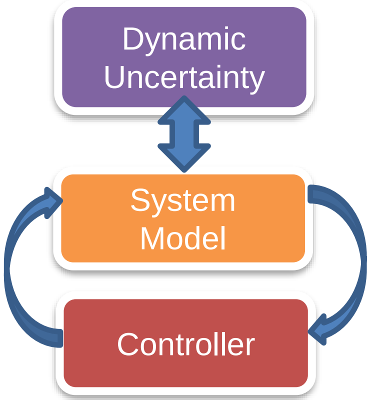
- Mismatch between model and plant
- Observation errors
- Subsystems in large-scale networks
- Model reduction

Note: Previous ADP algorithms assume the system order is known!

RADP: Robust Adaptive Dynamic Programming

Jiang & ZPJ; TNNLS, 2013

For illustration, consider partially linear composite systems with "**dynamic uncertainty**".

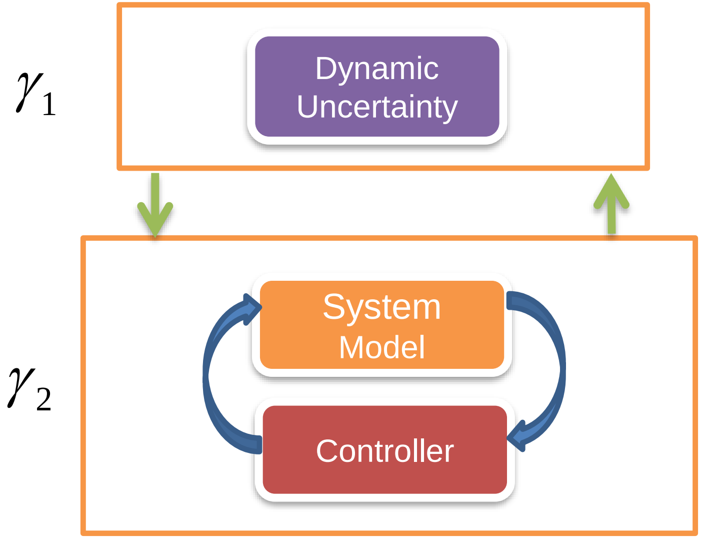


$$\dot{w} = q(w, y)$$
$$\dot{x} = Ax + B[u + E\Delta(w, y)]$$
$$y = Cx$$

where A, B, C, E are unknown matrices, q and Δ are unknown locally Lipschitz functions vanishing at the origin.

Challenge: How to learn robust/adaptive nonlinear optimal controllers via real-time and partial-state information?

RADP for partially linear composite systems



Input-to-state stability (ISS) and *Input-to-output stability* (IOS) [Sontag 1989], [Sontag & Wang 1995].

The state-space *nonlinear small-gain theorem* proposed in [Jiang, Teel, & Praly 1994] is an important tool for network stability and control.

A simplified version of the small-gain theorem: If $\gamma_1 \circ \gamma_2 < \mathrm{Id}$, then, the overall system is globally asymptotically stable at the origin.

Challenge

How to achieve gain assignment γ_2 *via* i/o data and ADP?

Special Case: Linear Gain Assignment

Jiang & ZPJ; TNNLS, 2013

$$\text{S1}: \begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases} \qquad \text{S2}: \begin{cases} \dot{x} = Ax + B(u + Ew) \\ y = Cx \end{cases}$$

Lemma (Gain assignment): Let $u = -K^* x$ be the optimal control policy of system S1 and assume the weighting matrices satisfying $Q > \gamma C^T C$ and $R^{-1} > EE^T$. Then, there exists a continuously differentiable, positive definite and radially unbounded function $V(x)$, such that along the solutions of S2, we have

$$\dot{V} \le -\gamma |y|^2 + |w|^2$$

Remark: The constant $\gamma > 0$ can be arbitrarily assigned by choosing appropriate weighting matrices Q and R, without knowing A and B.

Linear Gain Assignment

Jiang & ZPJ; TNNLS, 2013

Assumption: There exist a continuously differentiable, positive definite and radially unbounded function W and two constants $c_1, c_2 \geq 0$, such that

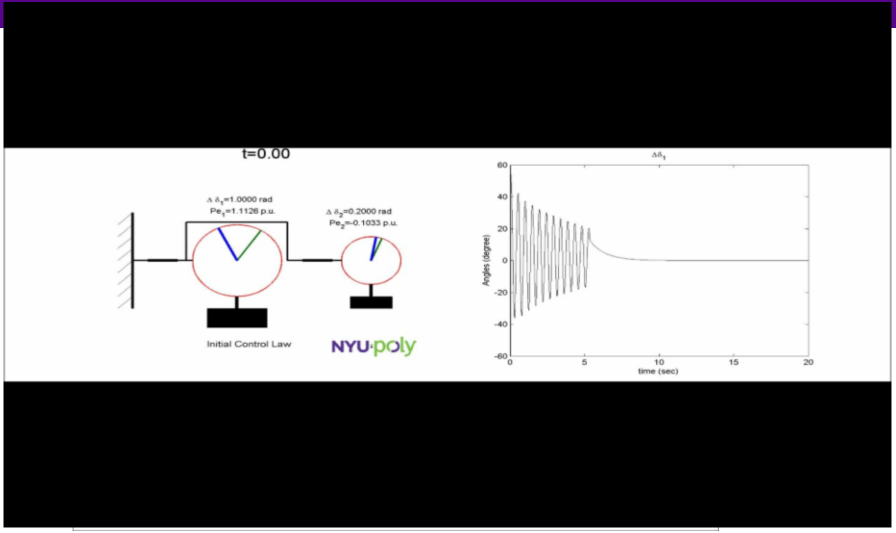
$$\frac{\partial W}{\partial z} q(w, y) \leq -c_1 |\Delta(w, y)|^2 + c_2 |y|^2$$

Lemma (Global Stabilization): Under mild assumptions, the overall system is globally asymptotically sable under the control policy $u = -K^* x$

if the following small-gain condition holds:

$$\frac{1}{\gamma} \frac{c_2}{c_1} < 1$$

Application to a Power System



Control Challenges:
1. Unknown dynamics
2. Locally available state variables
3. Prevent oscillation

Robust-ADP Approach:
1. Online learning
2. Partial state feedback
3. Stability and Suboptimality

Mechanical Dynamics: [P. Kundur et al. 1994]

$$\frac{d^2\delta_i}{dt^2} = -\frac{D_i}{2H_i}\frac{d\delta_i}{dt} + \frac{\omega_0}{2H_i}\left(P_{mi} - P_{e_i}\right) \qquad i = 1, 2$$

Governor Dynamics:

$$\frac{dP_{mi}}{dt} = \frac{1}{T_i}[-P_{mi} + u_i] \qquad i = 1, 2$$

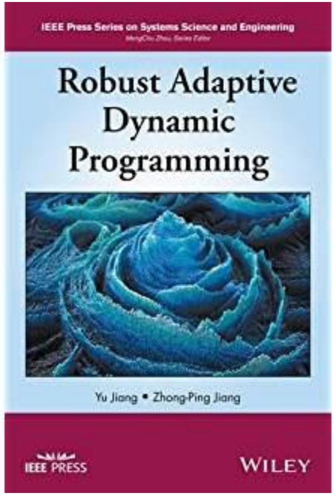
Active Power:

$$P_{e1} = E_1 E_2 \left(B_{12}\sin\delta_{12} + G_{12}\cos\delta_{12}\right) + E_1\frac{V_s}{x_{ds}}\sin\delta_1$$

$$P_{e2} = E_1 E_2 \left(B_{21}\sin\delta_{21} + G_{21}\cos\delta_{21}\right)$$

NYU

RADP: Robust Adaptive Dynamic Programming



Recent extensions:

- **Value iteration (c-t)**
- **Output feedback ADP**
- **Adaptive/optimal output regulation via ADP**
- **ADP for multi-agent systems**
- **Stochastic systems**

Tools:

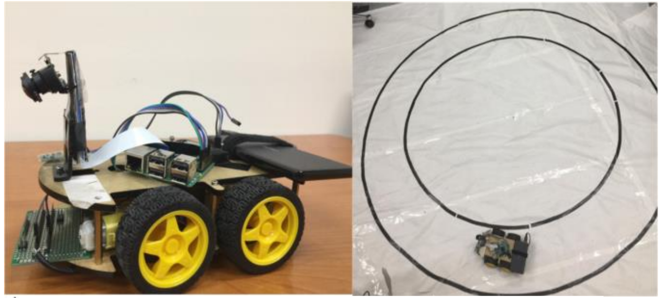
Semiglobal ADP, Global ADP, Decentralized ADP,
with applications in electric power systems, human motor control

Outline

- Data-Driven Learning-based Control Theory: Why?
- Robust Adaptive Dynamic Programming
  - Adaptive LQR for continuous-time linear systems
  - Extensions: nonlinear and robust
- Application:

  Connected and Autonomous Vehicles
- Conclusions and Future Work

1. Reinforcement Learning for Vision-Based Lateral Control



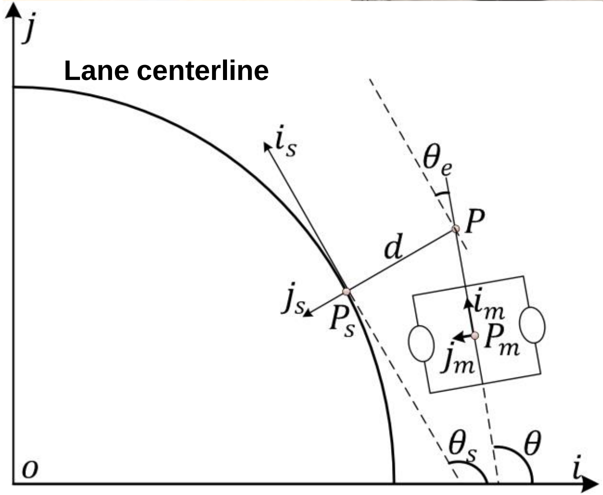
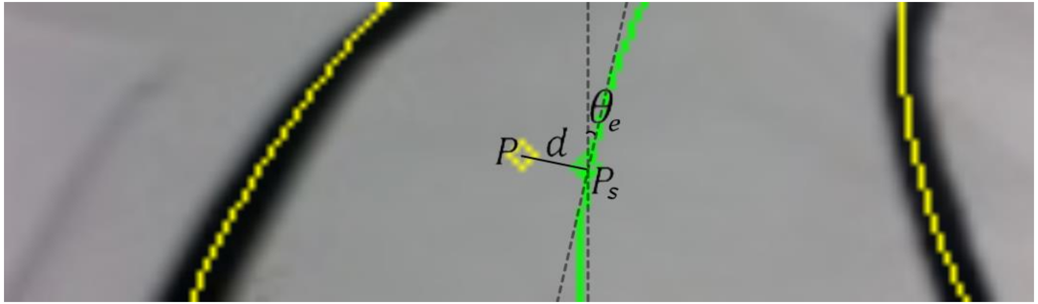


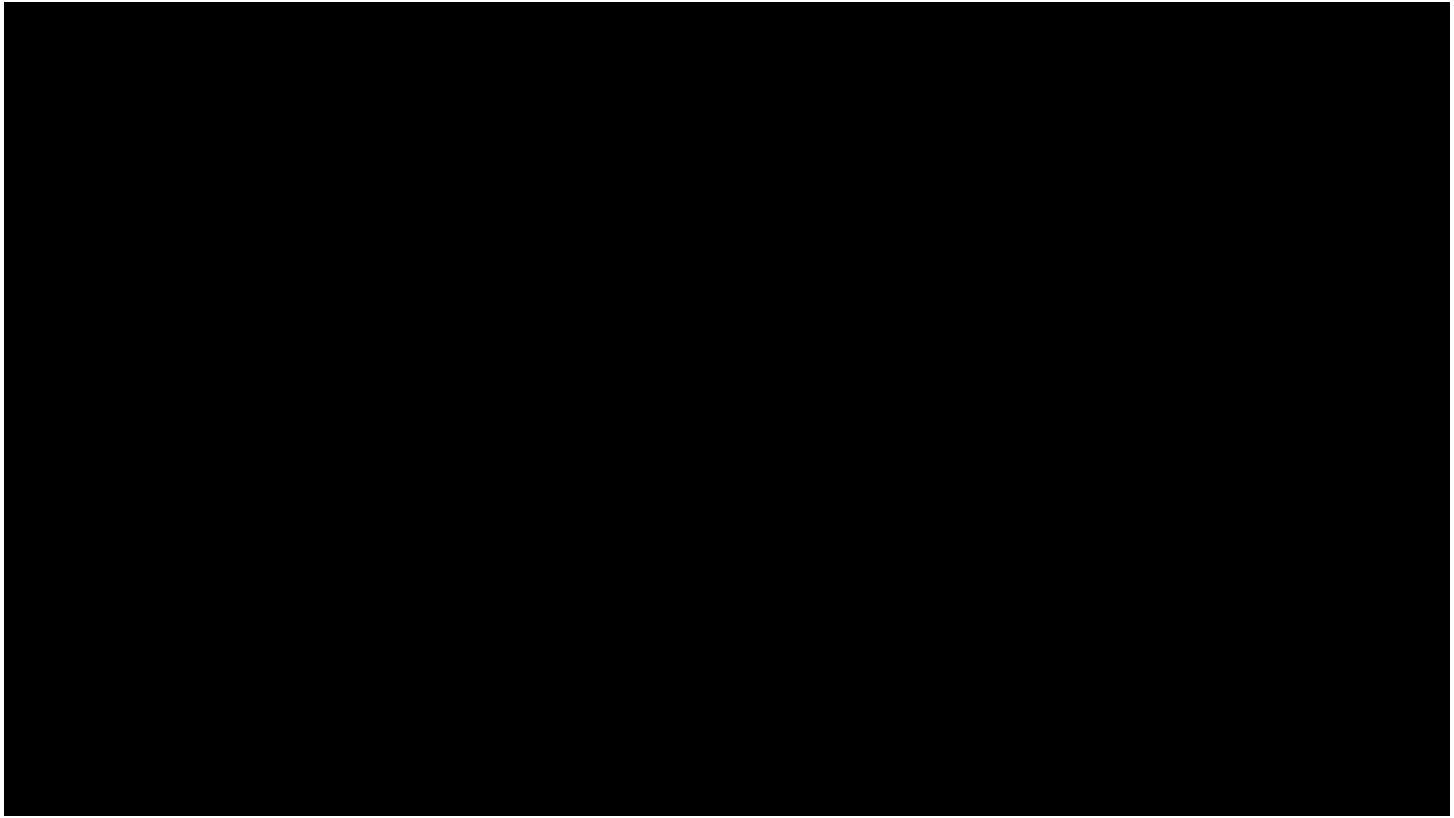
Image before and after processing

(a) Raw image

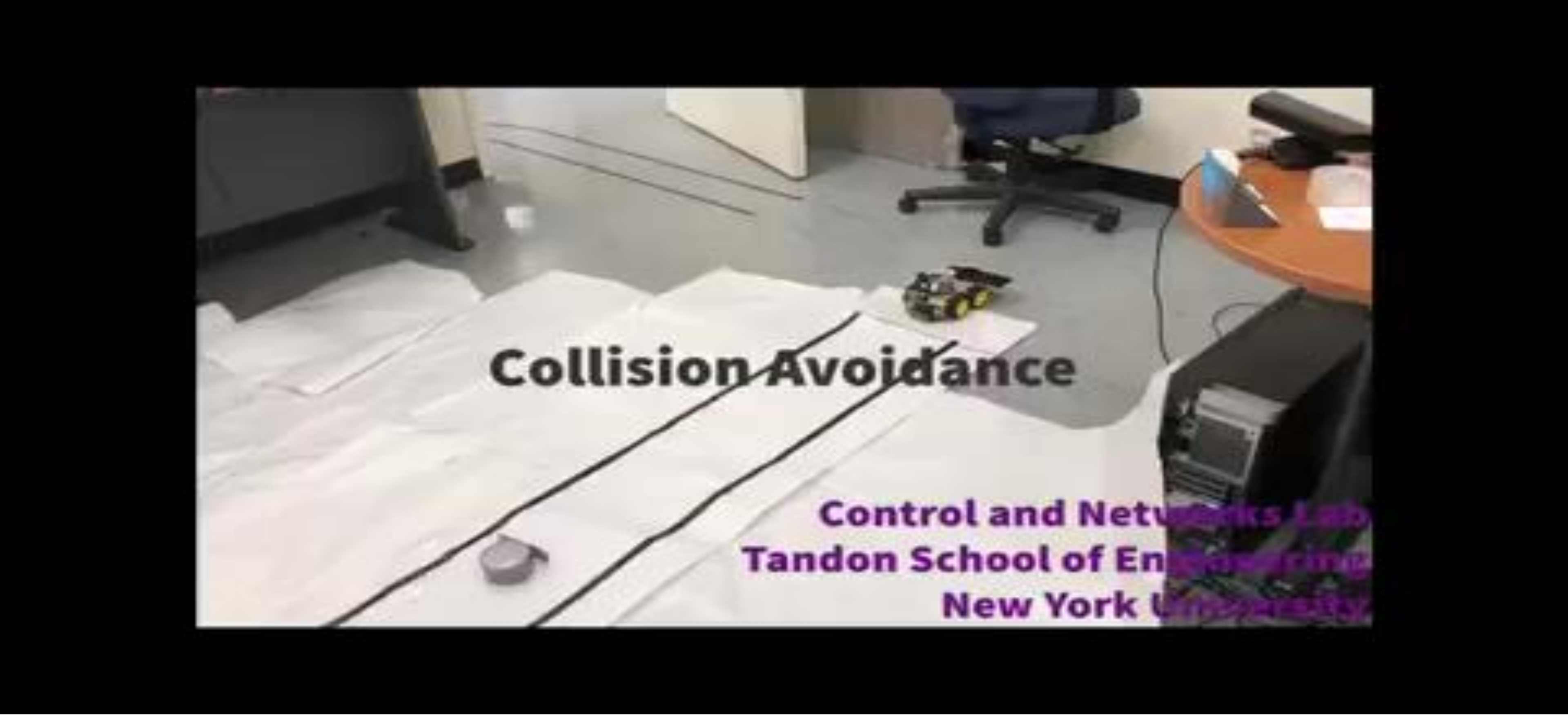


(b) Processed image including detected lane boundaries, lane centerline and $\xi = [d,\ \theta_e]^T$.

Learning behavior



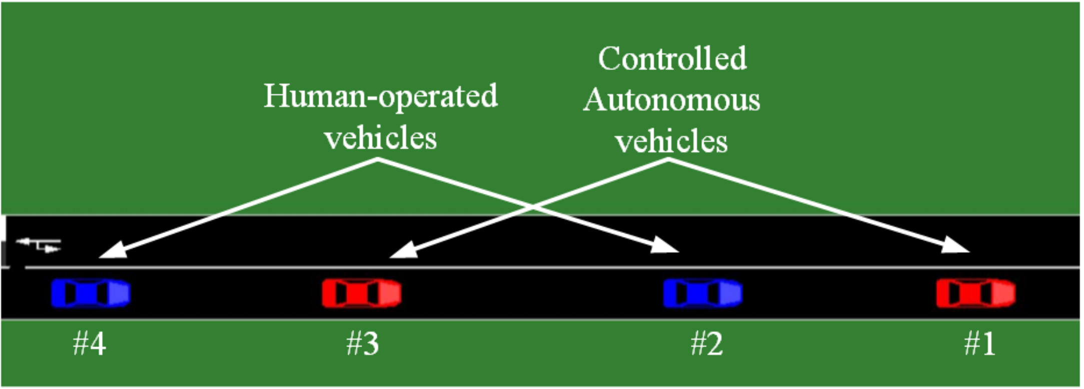
Car Following and Lane Keeping

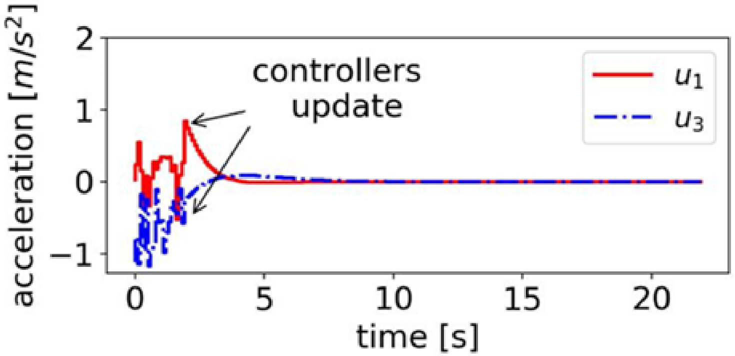


Control and Networks Lab, New York University

Microscopic Simulation in SUMO

2. Robust autonomous driving with humans in the loop



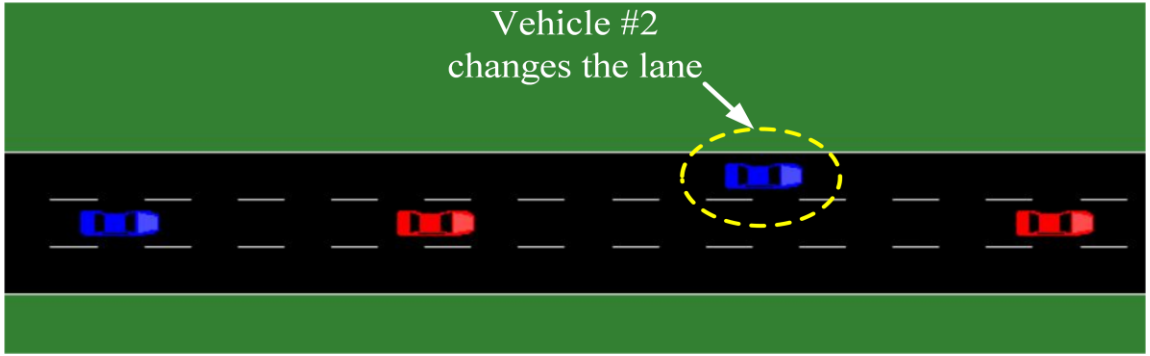


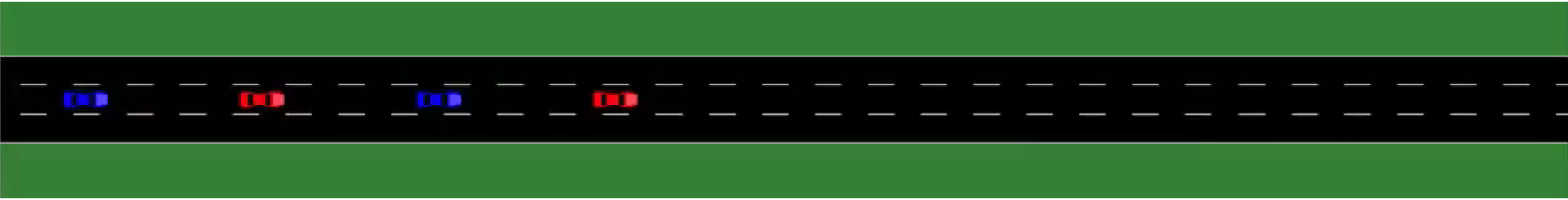
Robustness Evaluation

The speed of the leading vehicle is $v_0 = v^* + v_0^{\mathrm{amp}} \sin(\omega_f t)$ with amplitude $v_0^{\mathrm{amp}} = 5\ [m/s]$, frequency $\omega_f = 1\ [rad/s]$ and $v^* = 15\ [m/s]$.

Other scenarios: merging and splitting

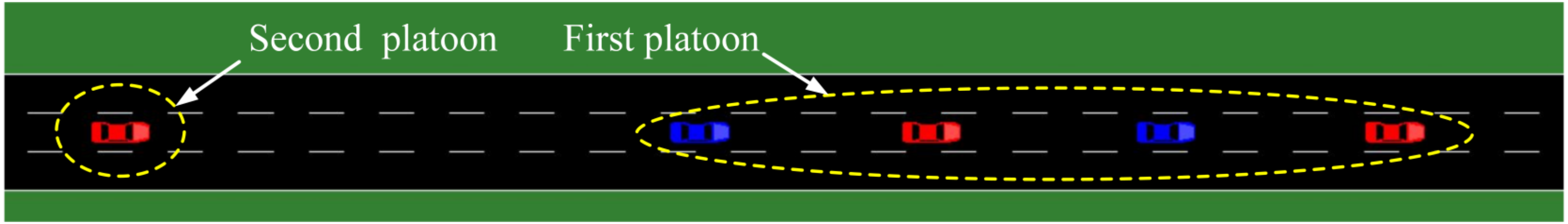
We also test the cut-out scenario:

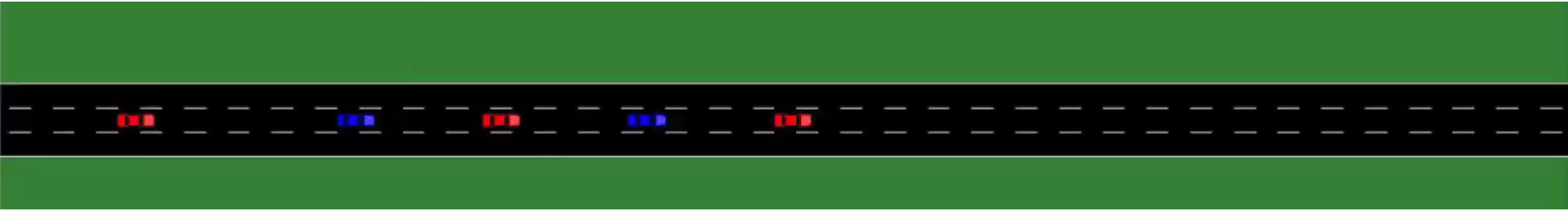




Other scenarios: merging and splitting

The merging of two platoons

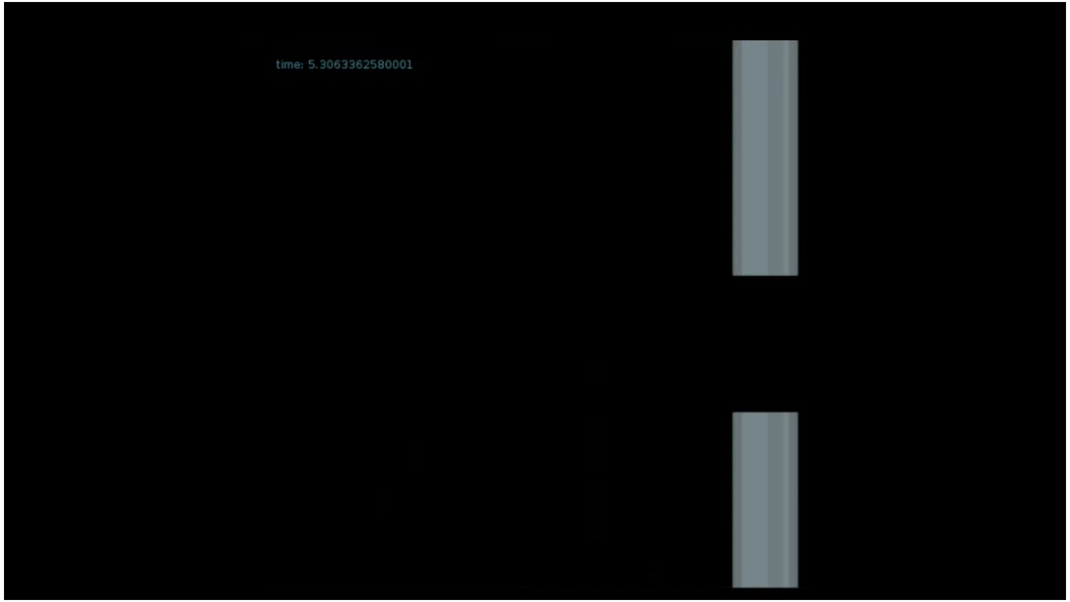




In both cases, after learning, the learned controllers can stabilize the new platoon as wanted.

3. Data-efficient Reinforcement Learning

“Flappy Bird” with RL using RADP-based Learning Controller



Conclusions

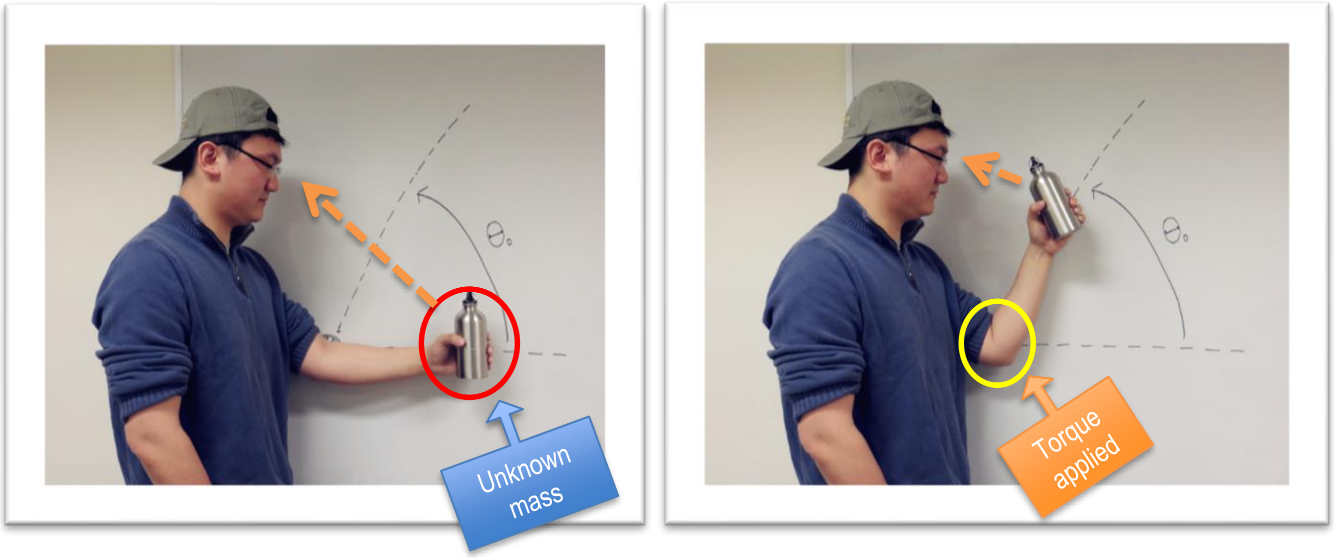
- **Learning-based nonlinear control is a promising field, yet still in its infancy.**
- **RADP (Robust Adaptive Dynamic Programming) for data-driven, learning-based robust/adaptive optimal control design.**
- **Validations via applications to power systems and CAVs.**

下集预告

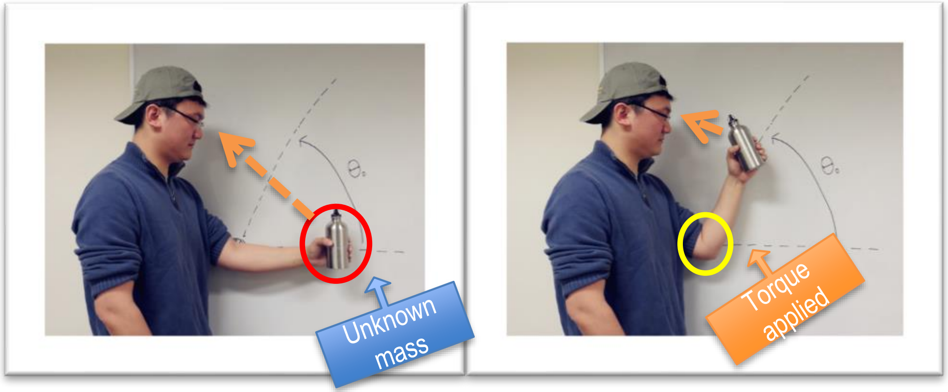
Future Work: Human Motor Control

- **Is RADP a computational mechanism of human motor control?**

- **A case study: reaching problem**



- **Is RADP a computational mechanism of human motor control?**
- **A case study: reaching problem**



Sensorimotor Control:

- One of the most common activities in daily lives
- Highly stereotyped trajectories have been reported
- Still unclear how the trajectories are formulated
- Research in this area may be helpful for better understanding related diseases.

$$\dot{\eta} = -\frac{1}{\tau_N}\eta + T_m$$

$$I\ddot{\theta} = -mgl\cos\theta + \eta + T_m$$

Original System Model

$$\dot{\eta} = -\frac{1}{\tau_N}\eta + T_m$$
$$I\ddot{\theta} = -mgl\cos\theta + \eta + T_m$$

State and input transformation

$$x_1 = \theta - \theta_0$$
$$x_2 = \dot{\theta}$$
$$w = \eta - \frac{\tau\_Nmgl\cos\theta_0}{\tau_N + 1}\sin - Ix_2$$

Transformed system

$$\dot{w} = -\frac{1+\tau_N}{\tau_N}w + Ix_2$$
$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \frac{2mgl}{I}\sin\left(\frac{x_1}{2}\right)\sin\left(\frac{x_1}{2} + \theta_0\right) + \frac{1}{I}(u + Ix_2 + w)$$

Cost

$$J = \int_0^\infty (100x_1^2(t) + x_2^2(t) + u^2(t))dt$$

Yu Jiang/ZPJ, 2013

A sensorimotor control problem (Cont'd)

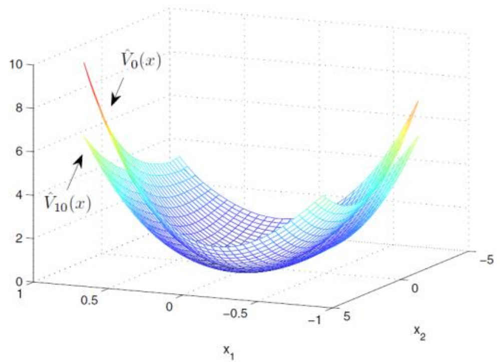


Figure: Cost functions

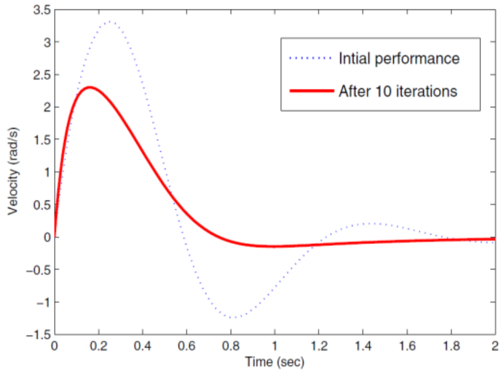
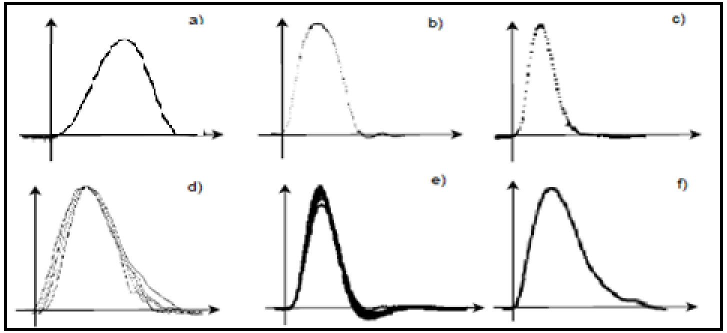


Figure: Velocity profiles



Experimental observations

a) [Harris & Wolpert, Nature, 1998]

b) [Morasso, Exp Brain Research, 1981]

c) [Abend et al, Brain: A Journal of Neurology, 1982]

d) [Atkeson et al. J of Neuroscience, 1985]

e) [Cooke, Neurobiology of Aging, 1989]

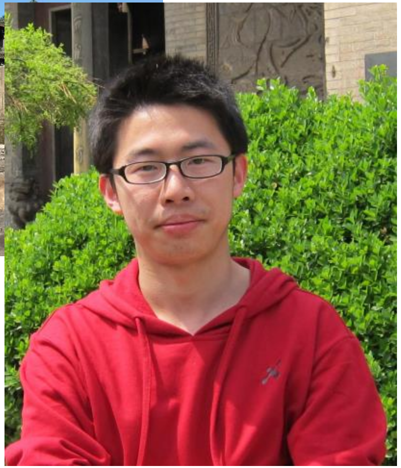
f) [Flash et al, J of Neuroscience, 1985]

NYU

Acknowledgements

- **Frank Lewis and his students for collaboration on ADP**

- **My students:**



Thank YOU for your attention!

Please send me your comments and feedback:
zjiang@nyu.edu