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 \( \gamma_1 \): gain from \( y_2 \) to \( y_1 \) and \( \gamma_2 \): 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

$\dot{x} = f(x, u)$, 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
\[ \dot{x} = f(x,u), \text{ with unknown } f \]

Bellman’s Dynamic Programming is **not** applicable, because of

- Curse of dimensionality  
  (Bellman, 1959)
- Curse of modeling  
  (Bertsekas, 1996)
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
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.
Maximizing the cumulative reward, through

1) Exploration (finding better policies).

2) Agent-environment interaction.

Figure: Reinforcement Learning (Minsky, 1954).
Many papers

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).

*Figure*: Reinforcement Learning (Minsky, 1954).
<table>
<thead>
<tr>
<th>Year</th>
<th>Event Description</th>
<th>Inventor(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1952</td>
<td>Dynamic programming</td>
<td>Bellman</td>
</tr>
<tr>
<td>1954</td>
<td>Reinforcement learning</td>
<td>Minsky</td>
</tr>
<tr>
<td>1960</td>
<td>DP algorithms (VI, PI) for MDPs</td>
<td>Bellman; Howard</td>
</tr>
<tr>
<td>1960s</td>
<td>Positive &amp; Negative DP</td>
<td>Blackwell; Strauch</td>
</tr>
<tr>
<td>1968</td>
<td>RL + approximate DP</td>
<td>Werbos</td>
</tr>
<tr>
<td>1983</td>
<td>Actor-critic algorithm</td>
<td>Barto</td>
</tr>
<tr>
<td>1984</td>
<td>TD-learning</td>
<td>Sutton</td>
</tr>
<tr>
<td>1989</td>
<td>Q-learning</td>
<td>Watkins</td>
</tr>
<tr>
<td>1990s</td>
<td>Neuro-DP</td>
<td>Bertsekas</td>
</tr>
<tr>
<td>2010s</td>
<td>Adaptive DP</td>
<td>Lewis</td>
</tr>
<tr>
<td>2013</td>
<td>Abstract DP</td>
<td>Bertsekas</td>
</tr>
</tbody>
</table>

2012 – Robust Adaptive DP (here)

Continuous-time ADP: still in its infancy
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:

- \( u^* = -K^*x \),
- \( J(x_0; u^*) = x_0^T P^* x_0 \), 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)
Assume the knowledge of an initial stabilizing policy $K_0$.

From

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

Integral RL equation.

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

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

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

$B$ 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)

\[
x^T(t)P_jx(t) = \int_t^{t+\delta t} x^T(Q + K_j^TRK_j)x \, d\tau + x^T(t + \delta t)P_jx(t + \delta t),
\]

\[K_{j+1} = R^{-1}B^TP_j.\]

\[B\] is required. \(x(t)\) is generated by \(u = -K_jx\) (on-policy).

Integral RL equation.

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

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

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

\[u' = -K_jx + \xi \text{ or } u' = -K_0x + \xi\]
On-line Off-policy PI-based ADP algorithm

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

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

(II)

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

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

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

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

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

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

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

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

Jiang & ZPJ, 2012

1) Full rank of $\Theta_k$

$\Rightarrow$ unique solution of (II)

(due to exploration noise $\xi$)

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

3) Stability + suboptimality without $\xi$. 
Question:

Can we remove the assumption on the knowledge of an initial, stabilizing policy $K_0$, when the system dynamics are not 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)

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

\[
\dot{M} = A^T M + M A - M B R^{-1} B^T M + Q, \quad M(0) = M^T(0) > 0
\]

Stochastic Approximation:

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

where

- \( Z_t \) is a projection term;
- \( \epsilon_t \) 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 \( \theta^* \). (Kushner-Yin, 2003)
\[ \{B_p\}_{q=0}^\infty : B_q \subseteq B_{q+1}, \lim_{q \to \infty} B_q = \{P \in \mathbb{R}^{n \times n} : P^T = P \geq 0\} \]

\[ \{\epsilon_k\}_{k=0}^\infty : \epsilon_k > 0, \lim_{k \to \infty} \epsilon_k = 0, \sum_{k=0}^\infty \epsilon_k = \infty. \epsilon > 0 \text{ is a threshold} \]

**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 \( \tilde{P}_{k+1} \not\in B_q \) then

\[ P_{k+1} \leftarrow P_0 \cdot q \leftarrow q + 1 \]

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

else \( P_{k+1} \leftarrow \tilde{P}_{k+1} \)

\( k \leftarrow k + 1 \)

- Convergence: \( P_k \to P^* \)
- Stability: If \( Q > \epsilon I_n \), then \( A - B \hat{R}^* \) is Hurwitz, where \( \hat{R}^* = R^{-1} B^T \hat{P}^* \), 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

$$x^T(t + \delta t)P_k(t + \delta t) = \int_t^{t+\delta t} x^TH_kx\,ds + 2\int_t^{t+\delta t} u^TRK_kx\,ds + x^T(t)P_kx(t). \tag{1}$$

where $H_k = A^TP_k + P_kA$.

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)

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

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

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

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

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

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

$k \leftarrow k + 1$

Figure: Bounded learning trajectory.
How about general nonlinear systems?

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

with unknown dynamics \( f \)?

- No distinction between linear and nonlinear problems
- First solution to adaptive/nonlinear optimal control
(Bian, Jiang & ZPJ, 2014)

Nonaffine system: \( \dot{x} = f(x, u), \quad J(x_0; u) = \int_0^\infty r(x, u) dt. \)

HJB equation:
\[
0 = \min_{\nu \in \mathbb{R}^m} \{ \partial_x V^*(x)f(x, \nu) + r(x, \nu) \}, \quad V^*(0) = 0, \\
\mu^*(x) = \arg \min_{\nu \in \mathbb{R}^m} \{ \partial_x V^*(x)f(x, \nu) + r(x, \nu) \}
\]

Policy iteration

1. Policy evaluation: \( \partial_x V_j(x)f(x, \mu_j(x)) + r(x, \mu_j(x)) = 0, V_j(0) = 0. \)

2. Policy improvement: \( \mu_{j+1}(x) = \arg \min_{\nu \in \mathbb{R}^m} \{ \partial_x V^*(x)f(x, \nu) + r(x, \nu) \}, \forall x \in \mathbb{R}^n \)

- **Convergence:** If \( \mu_0 \) is admissible, \( V_j \to V^*, \mu_j \to \mu^* \).
- **Stability:** \( \mu_j \) is stabilizing and, \( J(x_0; u_j(x)) < \infty \)
Using basis function approximation, we have for all \( x \in \mathbb{A} \) and \( v \in \mathbb{U} \),

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

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

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

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

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

\[
\frac{1}{M} \sum_{k=1}^{M} \Theta_k^j \Theta_k^j \geq \gamma I_{2N}, \quad \Theta_k^j \in \mathbb{R}^{1 \times 2N}
\]

is the vector of input-state data

The ADP algorithm:

1. Apply \( u_0(t) \) to the system. \( j \leftarrow 0 \).

2. Policy evaluation:

\[
[\hat{\omega}^j, \hat{\epsilon}^j]^T = -\left( \sum_{k=1}^{M} \Theta_k^j \Theta_k^j \right)^{-1} \sum_{k=1}^{M} \Theta_k^j \int_{t_{k-1}}^{t_k} r(x, \mu_j(x)) \, dt.
\]

3. Policy update:

\[
\hat{l}^{j+1} = \arg \min_{\{l|l \theta(x) \in \mathcal{U}\}} \{ \hat{\epsilon}^j \psi(x, l \theta(x)) + r(x, l \theta(x)) \}, \hat{\mu}_j = \hat{l}^j \theta.
\]

Convergence on \( \mathcal{A} \):

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

Bian, Jiang & ZPJ, 2014
Question 2:

How to learn suboptimal controllers with guaranteed robustness to dynamic uncertainties?
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!
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 \(\Delta\) 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


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 $\gamma_1 \circ \gamma_2 < \text{Id}$, then, the overall system is globally asymptotically stable at the origin.

**Challenge**

How to achieve gain assignment $\gamma_2$ via i/o data and ADP?
Special Case: Linear Gain Assignment

Lemma (Gain assignment): Let \( u = -K^* x \) be the optimal control policy of system \( \text{S1} \) 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 \( \text{S2} \), we have

\[
\dot{V} \leq -\gamma |y|^2 + |w|^2
\]

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

**Mechanical Dynamics:**  
[P. Kundur et al. 1994]  
\[
\frac{d^2 \delta_i}{dt^2} = -\frac{D_i}{2H_i} \frac{d\delta_i}{dt} + \frac{\omega_0}{2H_i} \left( P_{mi} - P_{ei} \right) \quad i = 1, 2
\]

**Governor Dynamics:**  
\[
dP_{mi} = \frac{1}{T_i} \left[ -P_{mi} + u_i \right] \quad i = 1, 2
\]

**Active Power:**  
\[
P_{e1} = E_1 E_2 \left( B_{12} \sin \delta_{12} + G_{12} \cos \delta_{12} \right) + E_1 \frac{V_s}{x_{ds}} \sin \delta_1
\]
\[
P_{e2} = E_1 E_2 \left( B_{21} \sin \delta_{21} + G_{21} \cos \delta_{21} \right)
\]

**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
**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
➢ 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
2. Robust autonomous driving with humans in the loop
The speed of the leading vehicle is \( v_0 = v^* + v_0^{\text{amp}} \sin(\omega_f t) \) with amplitude \( v_0^{\text{amp}} = 5 [m/s] \), frequency \( \omega_f = 1 [rad/s] \) and \( v^* = 15 [m/s] \).
We also test the cut-out scenario:

Vehicle #2 changes the lane
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.
“Flappy Bird” with RL using RADP-based Learning Controller
- **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.
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.

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

\[ I \ddot{\theta} = -mgl \cos \theta + \eta + T_m \]
Human Motor Control

Original System Model

\[
\dot{\eta} = -\frac{1}{\tau_N} \eta + T_m \\
I \ddot{\theta} = -mg l \cos \theta + \eta + T_m
\]

State and input transformation

\[
x_1 = \theta - \theta_0 \\
x_2 = \dot{\theta} \\
w = \eta - \frac{\tau_N mg l \cos \theta_0}{\tau_N + 1} \sin - Ix_2
\]

Transformed system

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

Cost

\[
J = \int_0^\infty \left( 100x_1^2(t) + x_2^2(t) + u^2(t) \right) dt
\]

Yu Jiang/ZPJ, 2013
A sensorimotor control problem (Cont’d)

**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]
• 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