Reinforcement Learning with Sparse Bellman Error Extrapolation for Infinite-Horizon Approximate Optimal Tracking

MAX Greene¹, PATRYK DEPTULA², SCOTT NIVISON³, WARREN DIXON¹

¹Dept. of Mechanical and Aerospace Engineering, Univ. of Florida
²The Charles Stark Draper Laboratory, Inc.
³Munitions Directorate, Air Force Research Laboratory

Under Review TAC, Fall 2020
Problem Formulation

**Dynamical System**

Given a control affine nonlinear dynamical system:

\[ \dot{x}(t) = f(x(t)) + g(x(t))u(t) \]

**Control Objective (Regulation Case)**

Design a controller, \( u(t) \), which minimizes a cost function:

\[ J(x, u) = \int_0^\infty (x(\tau)^T Qx(\tau) + u(\tau)^T Ru(\tau))d\tau \]

**Cost-to-Go**

Optimal value function:

\[ V^*(x) = \min_{u(\tau) \in U} \int_t^\infty (x(\tau)^T Qx(\tau) + u(\tau)^T Ru(\tau))d\tau \]
Hamilton Jacobi Bellman Equation

Hamilton Jacobi Bellman (HJB) equation:

\[ 0 = \nabla_x V^*(x) (f(x) + g(x)u^*(x)) + x^T Q x + u^*(x)^T R u^*(x) \]

Optimal Controller

From solving the HJB equation:

\[ u^*(x) = -\frac{1}{2} R^{-1} g(x)^T (\nabla_x V^*(x))^T \]

- Cannot solve HJB analytically
- Approximate the Value Function \((V^*)\)
  - Stone Weierstrass Theorem
  - Neural Networks
Approximate Optimal Solution

**Optimal Value Function and Optimal Control Policy:**

\[
V^*(x) = W^T \sigma(x) + \varepsilon(x) \quad u^*(x) = -\frac{1}{2} R^{-1} g(x)^T (\nabla_x \sigma(x)^T W + \nabla_x \varepsilon(x)^T)
\]

Unknown: Neural weights \( W \rightarrow \hat{W}_c, \hat{W}_a \)

\( \hat{W}_a \): Actor weight
\( \hat{W}_c \): Critic weight

**Value Function and Optimal Control Policy Approximation**

\[
\hat{V}(x, \hat{W}_c) = \hat{W}_c^T \sigma(x) \quad \hat{u}(x, \hat{W}_a) = -\frac{1}{2} R^{-1} g(x)^T (\nabla_x \sigma(x)^T \hat{W}_a)
\]

**Bellman Error (BE): Residual from HJB**

\[
\hat{\delta}(x, \hat{W}_c, \hat{W}_a) \triangleq \nabla_x \hat{V}(x, \hat{W}_c) \left( f(x) + g(x)\hat{u}(x, \hat{W}_a) \right) + \hat{u}(x, \hat{W}_a)^T R \hat{u}(x, \hat{W}_a) + x^T Q x
\]
Weight Update Laws using R-MBRL

\[ \dot{\hat{W}}_c(t) = -\eta_c \Gamma(t) \frac{\omega(t)}{\rho(t)} \hat{\delta} + \eta_c \sum_{j=1}^{N_j} \frac{\omega_i(t)}{\rho_i(t)} \hat{\delta}_i(t) \]

\[ \dot{\Gamma}(t) = \left( \lambda \Gamma(t) - \frac{\eta_c \Gamma(t) \omega(t) \omega^T(t) \Gamma(t)}{\rho(t)} \right) - \Gamma(t) \eta_c \sum_{j=1}^{N_j} \frac{\omega_i(t) \omega_i^T(t)}{\rho_i(t)} \hat{\delta}_i(t) \]

\[ \dot{\hat{W}}_a(t) = -\eta_c \left( \hat{W}_a(t) - \hat{W}_c(t) \right) - \eta_a \hat{W}_a(t) + \frac{\eta_c G_{\sigma i}^T(t) \hat{W}_a(t) \omega(t)^T}{4 \rho(t)} \hat{W}_c(t) \]

\[ + \left( \frac{\eta_c}{4 N_j} \sum_{i=1}^{N_j} \frac{G_{\sigma i}^T(t) \hat{W}_a(t) \omega_i(t)}{\rho_i(t)} \hat{\delta}_i(t) \right) \hat{W}_c(t) \]
• Separate operating domain
• Bellman error extrapolation contained to segment
• Smaller history stack
• Switches depending on region
• Introduces discontinuities
Simulation Results

• Linear Quadratic Tracking (LQT)

\[
\dot{x} = \begin{bmatrix}
-x_1 + x_2 \\
\frac{1}{2}x_1 - \frac{1}{2}x_2
\end{bmatrix} + \begin{bmatrix}
0 \\
1
\end{bmatrix} u \\
x_d = \begin{bmatrix}
4 \sin(t) \\
4 \cos(t) + 4 \sin(t)
\end{bmatrix}
\]

• Analytical solution known
• Non-sparse basis outside of box
• \(\sigma(\zeta) = [e_1^2, e_1 e_2, e_1 x_d, e_2^2, e_2 x_d, e_2 x_d^2]^T\)
• Sparse basis inside of box
• \(\sigma(\zeta) = [e_1^2, e_1 e_2, 0, 0, e_2^2, e_2 x_d, e_2 x_d^2]^T\)
• Dynamics approximated with neural network
Simulation Results

**NN System ID Weights**

**Control Policy**

**Critic/Actor Weights**

- $\hat{\theta}_1(t)$
- $\hat{\theta}_2(t)$
- $\hat{\theta}_3(t)$
- $\hat{\theta}_4(t)$

- Optimal Control Policy
- Estimated Optimal Control Policy

- $\hat{w}_c(t)$
- $\hat{w}_e(t)$

- Time (s)
- Time (s)
Simulation Results

<table>
<thead>
<tr>
<th></th>
<th>Standard Model-Based ADP</th>
<th>SS Model-Based ADP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median Computation Time (10 trials) (s)</td>
<td>120.40</td>
<td>25.90</td>
</tr>
<tr>
<td>Integral of Error ($\int_0^{150}</td>
<td></td>
<td>e(\tau)</td>
</tr>
<tr>
<td>5% Rise Time (s)</td>
<td>33.33</td>
<td>44.29</td>
</tr>
<tr>
<td>RMS Steady State Error (s)</td>
<td>$6.92 \cdot 10^{-3}$</td>
<td>$5.57 \cdot 10^{-3}$</td>
</tr>
</tbody>
</table>
Model-based Reinforcement Learning for Optimal Feedback Control of Switched Systems

Max Greene\textsuperscript{1}, Moad Abudia\textsuperscript{2}, Rushikesh Kamalapurkar\textsuperscript{2}, Warren Dixon\textsuperscript{1}

\textsuperscript{1}Dept. of Mech. and Aerospace Engineering, Univ. of Florida
\textsuperscript{2}Dept. of Mech. and Aerospace Engineering, Oklahoma State Univ.

To Appear, Conf. on Decision and Control (CDC), December 2020
• Theorem 1: Subsystem Stability Analysis
  
  \[ V_{L,i}(r_i, t) = V_i^*(x) + \frac{1}{2} \tilde{W}_{c,i}^T \Gamma_i^{-1} \tilde{W}_{c,i} + \frac{1}{2} \tilde{W}_{a,i}^T \tilde{W}_{a,i} \]
  
  \[ \dot{V}_{L,i}(r_i, t) \leq \frac{\Lambda_i}{\alpha_{2,i}} V_{L,i}(r_i, t) + l_i \]
  
  • System state \( x \), weight estimation errors \( (\tilde{W}_c, \tilde{W}_a) \), and control policy \( u(t) \) is Uniformly Ultimately Bounded

  • Exponential convergence to a region \( V_{L,i}(r_i, t) \leq \frac{2l_i \alpha_{3,i}^2}{\Lambda_i \alpha_{1,i}^2} \).
Theorem 2

• When switching from $i = 1 \rightarrow 2$, there is a jump between the multiple Lyapunov functions.

\[
V_{L,1}(r_1, t) = V_1^*(x) + \frac{1}{2} \tilde{W}_{c,1}^T \Gamma_1^{-1} \tilde{W}_{c,1} + \frac{1}{2} \tilde{W}_{a,1}^T \tilde{W}_{a,1}
\]

\[
V_{L,2}(r_2, t) = V_2^*(x) + \frac{1}{2} \tilde{W}_{c,2}^T \Gamma_2^{-1} \tilde{W}_{c,2} + \frac{1}{2} \tilde{W}_{a,2}^T \tilde{W}_{a,2}
\]

Switching causes discrete jumps in these values.

Scales by const. due to quadratic value fcn. assumption.

Largest UUB Region

"Jump"
Theorem 2:

The system consisting of a family of subsystems, each with control affine dynamics and a properly designed dwell-time, $\tau$, ensures that $x$, $\tilde{W}_{c,i}$ and $\tilde{W}_{a,i}$ $\forall i$ will converge to a neighborhood of the origin in the sense that $V_{L,i}(r_i, t) \leq V_{L,B}$ for all $t \geq T$; where $V_{L,B} \in \mathbb{R}$ is the maximum ultimate bound for all subsystems, and $T \in \mathbb{R}$ is the time required to reach the ultimate bound $V_{L,B}$; provided a minimum dwell-time $\tau^*$ is satisfied.
**F-16 longitudinal dynamics**

- [Stevens, Lewis, Johnson, 2016]

Explore further connection with Ben Dickenson (AFRL/RW), regarding reconfigurable aircraft munition that extend wings, retract wings

<table>
<thead>
<tr>
<th>Dynamic Model</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1, Unaltered Model</td>
<td>[ \dot{x} = \begin{bmatrix} -1 &amp; 0.9 &amp; -0.002 \ 0.8 &amp; -1.1 &amp; -0.2 \ 0 &amp; 0 &amp; -1 \end{bmatrix} x + \begin{bmatrix} 0 \ 0 \ 1 \end{bmatrix} u ]</td>
</tr>
<tr>
<td>Mode 2, Altered Model</td>
<td>[ \dot{x} = \begin{bmatrix} -0.8 &amp; 0.2 &amp; -0.01 \ 0.6 &amp; -1.3 &amp; -0.1 \ 0 &amp; 0 &amp; -1 \end{bmatrix} x + \begin{bmatrix} 0 \ 0 \ 1 \end{bmatrix} u ]</td>
</tr>
<tr>
<td>Mode 3, Altered Model</td>
<td>[ \dot{x} = \begin{bmatrix} -1 &amp; 0.5 &amp; -0.02 \ 0.9 &amp; -0.8 &amp; -0.4 \ 0 &amp; 0 &amp; -1 \end{bmatrix} x + \begin{bmatrix} 0 \ 0 \ 1 \end{bmatrix} u ]</td>
</tr>
</tbody>
</table>
• Switch between multiple dynamical systems
  • Arbitrary switching sequence
  • Satisfies minimum dwell-time condition

• Switching Sequence
  • {1,2,3,1,3,2}
Switched System ADP

![Graphs showing time evolution of state variables and value functions for Switched System ADP.]

- **Left Graph:**
  - \( \hat{W}_{1,1}(t) \) (red)
  - \( \hat{W}_{1,2}(t) \) (blue)
  - \( \hat{W}_{1,3}(t) \) (green)
  - Time range: 0 to 30 seconds

- **Right Graph:**
  - \( \hat{V}_n(x) \) (red)
  - \( V_n^*(x) \) (dashed blue)
  - Approximate Value Function
  - Optimal Value Function
  - Time range: 0 to 30 seconds

Incorporating feedback control schemes and graphical representations to visualize the system's performance over time.
Lyapunov-Based Real-Time and Iterative Adjustment of Deep Neural Networks

R. Sun, M. L. Greene, D. M. Le, Z. I. Bell, G. Chowdhary, W. E. Dixon

1Univ. of Florida, 1AFRL/RW, 2Univ. of Illinois Urbana-Champaign

Under Review, IEEE Control Systems Letters
DNN-Based Adaptive Control

Multiple Timescale Learning

- Offline & Real-time Data
  - Offline-Slow Learning
    - Inner-Layer Training
  - Real-time Learning
    - Analysis-based Outer-Layer Training

Reference Input

Closed-Loop Controller

Dynamic System

Output
• Van der Pol Oscillator
• Trained with 600s of simulation data
• Transient response is fast relative to the overall timescale
Trained on identical dynamics
Trained on similar dynamics (different coefficients) - transfer learning
No offline training. Inner-layer DNN weights are randomly initialized.