Lecture – 21

Approximate Dynamic Programming (ADP), Adaptive Critic (AC) and Single Network Adaptive Critic (SNAC) Designs

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Outline

• Approximate Dynamic Programming
• Adaptive Critic (AC) Design
• Single Network Adaptive Critic (SNAC) Design
• AC vs. SNAC: A Comparison Study
Approximate Dynamic Programming

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore

Approximate Dynamic Programming: Discrete-time Framework

Problem:
Find an admissible control $U_k$ (at time $t_k$) which minimizes a “meaningful” cost function:

$$J = \sum_{k=0}^{N-1} \Psi(X_k, U_k)$$

subjected to the constraint of system dynamics:

$$X_{k+1} = f(X_k, U_k)$$

and appropriate boundary conditions

(Note: $N \to \infty$ leads to infinite horizon problem)
Approximate Dynamic Programming:
Necessary Conditions of Optimality

- Write: \( J_k = \sum_{i=0}^{N-1} \Psi_k(X_i, U_i) = \Psi_k + J_{k+1} \)

- Define: \( \lambda_k \triangleq \frac{\partial J_k}{\partial X_k} \)

- Optimal Control Equation: \( \frac{\partial J_k}{\partial U_k} = 0 \)

Approximate Dynamic Programming:
Necessary Conditions of Optimality

\[
\frac{\partial J_k}{\partial U_k} = \left( \frac{\partial \Psi_k}{\partial U_k} \right) + \left( \frac{\partial J_{k+1}}{\partial U_k} \right) X_{k+1} = f(X_k, U_k)
\]

\[
= \left( \frac{\partial \Psi_k}{\partial U_k} \right) + \left( \frac{\partial X_{k+1}}{\partial U_k} \right) \left( \frac{\partial J_{k+1}}{\partial X_{k+1}} \right) \lambda_{k+1}
\]

Optimal Control Equation: \( \left( \frac{\partial \Psi_k}{\partial U_k} \right) + \left( \frac{\partial X_{k+1}}{\partial U_k} \right) \lambda_{k+1} = 0 \)
Approximate Dynamic Programming: Necessary Conditions of Optimality

- **Costate Equation:**
  \[
  \lambda_k = \left( \frac{\partial J}{\partial X_k} \right) = \left( \frac{\partial \Psi}{\partial X_k} \right) + \left( \frac{\partial J_{k+1}}{\partial X_k} \right) \\
  X_{k+1} = f(X_k, U_k) \\
  U_k = \Phi(X_k)
  \]

- **Costate Equation on Optimal Path:**
  \[
  \lambda_k = \left( \frac{\partial \Psi}{\partial X_k} \right) + \left( \frac{\partial x_{k+1}}{\partial X_k} \right) T \lambda_{k+1}
  \]

Summary: Necessary Conditions of Optimality

- **State Equation:**
  \[
  X_{k+1} = f(X_k, U_k)
  \]

- **Costate Equation:**
  \[
  \lambda_k = g(X_k, \lambda_{k+1}, U_k)
  \]

- **Optimal Control Equation:**
  \[
  U_k = \Phi(X_k, \lambda_{k+1})
  \]

- **Boundary Conditions:** 
  **TPBVP** (split)
References


Adaptive Critic Design for Infinite Time Regulator Problems

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore
Feed Forward Neural Networks

**Principle:** If presented with “proper” input-output data, training (error minimization) algorithms exist, so that after training the network can “capture” the underlying functional relationship.

**Theorem:** A three layer network with “sufficient” number of neurons in the hidden layer can approximate any continuous function with arbitrarily small error bound in compact (closed and bounded) set.

Adaptive Critic Methodology:

**Philosophy**

Action network leads to the optimal control solution (after mutually consistent training of both networks)
Adaptive Critic Methodology: Advantages

- Applicable for Nonlinear problems (without any linear/quasi-linear approximations)
- Solution for a large number of initial conditions
  Feedback optimal control in the domain of interest
- Feasible computational load
  (unlike dynamic programming)
- Self-contained methodology
- Real-time control

Synthesis of Action Network

Assumption: Critic Network is Optimal

```
Action Network

State Equation

Critic Network

Optimal Control Equation

X_t \rightarrow U_t

X_{t+1} \rightarrow \lambda_{t+1}

U^* \rightarrow U_t
```
**Synthesis of Critic Network**

**Assumptions:** Action Network is Optimal, Critic Network is optimal at $t_n$

![Diagram of Synthesis of Critic Network]

**Initialization of Networks:**

**Pre-training**

- Linearize the Problem
- Formulate a LQR Problem and Obtain the Solution
  
  \[
  X_{k+1} = A_d X_k + B_d U_k
  \]
  
  \[
  J = \sum_{i=1}^{\infty} (X_i^T Q_d X_k + U_i^T R_d U_k)
  \]
  
  \[
  \lambda_k = S_d X_k \quad U_k = -K_d X_k
  \]

- Train the networks
Initialization of Networks: An Alternative Approach

- Obtain the control solution from other techniques (like “dynamic inversion”)
- For a known state, assume this solution as “optimal” and compute the associated costate
- Train the networks based on these control and costate solutions

Single Network Adaptive Critic (SNAC) for Infinite Time Regulator Problems

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore
**Single-Network Adaptive Critic (SNAC) Design**

Assumption: The optimal control equation is explicitly (symbolically) solvable for control in terms of state and costate

- Retains all the advantages of adaptive-critic synthesis
- Elimination of training of action networks
- Elimination of iterative training between action and critic networks

**Synthesis of Critic Network**

Assumption: Critic Network is optimal at $t_{oa}$

[Diagram of the Synthesis of Critic Network]
Initialization of Networks: Pre-training

- Linearize the Problem and Obtain LQR Solution
  \[ \lambda_k = S_d X_k \quad U_k = -K_d X_k \]
- Obtain the relationship between \( X_k \) and \( \lambda_{k+1} \)
  \[ \lambda_{k+1} = S_d X_{k+1} = S_d (A_d X_k + B_d U_k) \]
  \[ = S_d (A_d - B_d K_d) X_k \]
  \[ = \tilde{S}_d X_k \]
- Train the networks

AC vs. SNAC: A Comparison Study

Prof. Radhakant Padhi
Dept. of Aerospace Engineering
Indian Institute of Science - Bangalore
Adaptive Critic vs. Single Network
Adaptive Critic: A Comparison Study

Comparison Studies:
Three Motivating Examples

Scalar Problem: \[ \dot{x} = x - x^3 + u \]

Van-der Pol’s Oscillator: \[ \ddot{x} + \alpha (x^2 - 1) \dot{x} + x - (1 + x^2 + \dot{x}^2)u = 0 \]

Electrostatic Actuator: \[ \dot{Q} - \frac{1}{R} \left( V_{in} - \frac{Qg}{\varepsilon A} \right) = 0 \]
\[ m\ddot{g} + b\dot{g} + k(g - g_0) + \frac{Q^2}{2\varepsilon A} = 0 \]
Numerical Studies: Computer Platform

- Processor: Pentium III, 930 MHz speed
- RAM: 320 MB
- Software used:
  - MATLAB V. 5.2, Release 12
  - Neural Network Toolbox V.3.0
- Training algorithm used: Levenberg-Marquardt back-propagation scheme
- Number of runs for averaging: 10

Scalar Example

- System Model \( \dot{x} = x - x^3 + u \)
- Cost Function \( J = \frac{1}{2} \int_0^\infty (q x^3 + ru^3) \, dt \)
- Closed Form Solution (used for comparison)
  \[ u = - \left( x - x^3 \right) - x\sqrt{x^4 - 2x^2 + 2} \]
Necessary Conditions of Optimality

- State Equation: \( x_{k+1} = x_k + \Delta t \left( x_k - x_k^3 + u_k \right) \)
- Cost Function: \( J = \sum_{k=1}^{N} \left( \frac{1}{2} (q x_k^2 + r u_k^2) \Delta t \right) \)
- Costate Equation: \( \lambda_i = \lambda_{k+1} + \Delta t \left[ q x_k + \lambda_{k+1} (1-3x_k^2) \right] \)
- Optimal Control Equation: \( u_k = -\left( \frac{\lambda_{k+1}}{r} \right) \)

Implementation

<table>
<thead>
<tr>
<th>AC</th>
<th>SNAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critic Network: 1-4-1</td>
<td>Critic Network: 1-4-1</td>
</tr>
<tr>
<td>Action Network: 1-4-1</td>
<td>Convergence Criteria:</td>
</tr>
</tbody>
</table>

Critic:
\[ \| \lambda_p^k - \lambda_p^{k+1} \|_1 / \| \lambda_p^k \| < 0.01 \]

Action:
\[ \| u_p^k - u_p^{k+1} \|_1 / \| u_p^k \| < 0.01 \]

Cycle:
\[ | err_{c_k} - err_{c_{k+1}} | < 0.0002 \]
\[ | err_{a_k} - err_{a_{k+1}} | < 0.0002 \]

Weights: \( q = r = 1 \)
Numerical Results

<table>
<thead>
<tr>
<th>Time</th>
<th>State</th>
<th>Neural Control</th>
<th>Closed Form Control</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>X(0) = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>X(0) = 0.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>X(0) = -1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>X(0) = -0.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Numerical Results
Time Comparison: \( T_{SNAC} = 0.43 \, T_{AC} \)

| # OF iterations | CRITIC ACTION CONVERGENCE TOTAL |
|-----------------|-------------------------------|-------------------------------|
|                 | TIME (sec) | TIME (sec) | CHECK TIME (sec) | TOTAL TIME (sec) |
| 1               | 5.713      | 4.48      | 3.724            | 13.884          |
| 2               | 4.955      | 4.137     | 3.689            | 12.782          |
| 3               | 4.776      | 4.183     | 3.658            | 12.627          |

SNAC

TOTAL TIME (sec) = 53.907

Cost Function Comparison \((t_f = 6)\)
For different initial conditions

<table>
<thead>
<tr>
<th>INITIAL STATE X(0)</th>
<th>COST (SNAC)</th>
<th>COST (AC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>1.4499</td>
<td>1.4591</td>
</tr>
<tr>
<td>-0.8</td>
<td>1.0521</td>
<td>1.0586</td>
</tr>
<tr>
<td>-0.6</td>
<td>0.6461</td>
<td>0.6476</td>
</tr>
<tr>
<td>-0.4</td>
<td>0.3032</td>
<td>0.302</td>
</tr>
<tr>
<td>-0.2</td>
<td>0.0779</td>
<td>0.0773</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0779</td>
<td>0.0772</td>
</tr>
<tr>
<td>0.4</td>
<td>0.3031</td>
<td>0.3021</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6461</td>
<td>0.6478</td>
</tr>
<tr>
<td>0.8</td>
<td>1.052</td>
<td>1.0588</td>
</tr>
<tr>
<td>1</td>
<td>1.4497</td>
<td>1.4592</td>
</tr>
</tbody>
</table>

Conclusion: \( J_{AC} \approx J_{SNAC} \)
Van-der Pol's Oscillator

- System Model:
  \[ \ddot{x} + \alpha(x^2 - 1)\dot{x} + x - (1 + x^2 + \dot{x}^2)u = 0 \]

- Cost Function:
  \[ J = \frac{1}{2} \left( X'^T Q X + R u^2 \right) dt \]

- Cost Function:
  \[ J = \frac{N}{2} \sum_{k=1}^{N-1} \left( X'^T Q X_k + R u_k^2 \right) \Delta t \] (discrete)

Necessary Conditions of Optimality

- State Equation:
  \[
  \begin{bmatrix}
  x_{k+1} \\
  x_{2,k+1}
  \end{bmatrix} = \begin{bmatrix}
  x_k \\
  x_{2,k}
  \end{bmatrix} + \Delta t \begin{bmatrix}
  x_k \\
  x_{2,k}
  \end{bmatrix} + \alpha(1 - x_k^2)x_k - x_k + (1 + x_k^2 + x_{2,k}^2)u_k
  \]

- Costate Equation:
  \[
  \begin{bmatrix}
  \dot{\lambda}_1 \\
  \dot{\lambda}_{2,k}
  \end{bmatrix} = \Delta t \begin{bmatrix}
  q_1 x_k \\
  q_{2,k}
  \end{bmatrix} + \frac{1}{\Delta t} \begin{bmatrix}
  \Delta t[(2x_k u_k) - 1 - (2\alpha x_k x_{2,k})] \\
  (1 + \Delta t)(2x_{2,k} u_k) + (\alpha(1 - x_k^2))
  \end{bmatrix} \begin{bmatrix}
  \lambda_{1,k+1} \\
  \lambda_{2,k+1}
  \end{bmatrix}
  \]

- Optimal Control Equation:
  \[ u_k = -r^{-1}(1 + x_k^2 + x_{2,k}^2) \dot{\lambda}_{2,k} \]
Implementation

<table>
<thead>
<tr>
<th>AC</th>
<th>SNAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Critic Network: 2-6-2</td>
<td>- Critic Network: 2-6-2</td>
</tr>
<tr>
<td>- Action Network: 2-6-1</td>
<td>- Convergence Criteria:</td>
</tr>
<tr>
<td>Critic: ( | \hat{\lambda}_p^\mu - \hat{\lambda}_p^\nu | / | \hat{\lambda}_p^\nu | &lt; 0.05 )</td>
<td>( | \hat{\lambda}_p^\mu - \hat{\lambda}_p^\nu | / | \hat{\lambda}_p^\nu | &lt; 0.05 )</td>
</tr>
<tr>
<td>Action: ( | u_p^\mu - u_p^\nu | / | u_p^\nu | &lt; 0.05 )</td>
<td>( p = 1, 2, \ldots n )</td>
</tr>
<tr>
<td>Cycle: (</td>
<td>err_{c_i} - err_{c_{i-1}}</td>
</tr>
<tr>
<td>(</td>
<td>err_{c_i} - err_{c_{i-1}}</td>
</tr>
</tbody>
</table>

Weighing matrices: \( Q = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad R = 1 \)

Telescopic Method: State Generation

Define: \( S_i = \{ X_k : \| X_k \| \leq c_i \} \)
where \( c_i \) is a positive constant. \( (i = 1, 2, \ldots) \)
\[ c_1 \leq c_2 \leq c_3 \ldots. \]

Increase \( c_i \) until \( S_i \) includes initial conditions.

One way: \( c_1 = 0.05, c_i = c_{i-1} + 0.05(i-1) \)
Numerical Results
Different Initial Conditions

Telescopic Cycle Training Time
**Total Training Time** \( T_{SNAC} \approx 0.43 T_{AC} \)

![Histogram showing Total Training Time comparison between AC and SNAC](image)

**Cost Function Comparison For different initial conditions** \( (t_f = 10) \)

![Column chart showing Cost Function comparison between AC and SNAC](image)

**Conclusion**: \( J_{SNAC} \approx J_{AC} \)

---

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION
Prof. Radhakant Padhi, AE Dept., IISc-Bangalore
**Electrostatic Actuator**

- System Model: \[ \dot{Q} = \frac{1}{R} \left( V_{in} - \frac{Qg}{\varepsilon A} \right) = 0 \]
  \[ m\ddot{z} + b\dot{z} + k(z - g_0) + \frac{Q^2}{2\varepsilon A} = 0 \]

- Cost Function: \[ J = \frac{1}{2} \int_0^T (X^T Q_x X + R_u u^2) \, dt \]

- Cost Function: \[ J = \sum_{k=1}^{N-1} \frac{1}{2} \left( X_k^T Q_w X_k + R_w u_k^2 \right) \Delta t \]
  (Discrete)

---

**Electrostatic Actuator: A Real-life Problem**

\[
Z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} Q \\ g \\ \dot{g} \end{bmatrix} \quad \dot{z}_1 = \frac{1}{R} \left( V_{in} - \frac{z_1 z_2}{\varepsilon A} \right) \\
\dot{z}_2 = z_3 \\
\dot{z}_3 = -\frac{1}{m} \left( \frac{z_1^2}{2\varepsilon A} + bz_3 + k(z_2 - g_0) \right)
\]

\[ v = V_{in} \]
Plant Parameters

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>SYMBOL</th>
<th>VALUE</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Area</td>
<td>A</td>
<td>100</td>
<td>μm²</td>
</tr>
<tr>
<td>Permittivity</td>
<td>ε</td>
<td>1</td>
<td>C/N μm²</td>
</tr>
<tr>
<td>Initial gap</td>
<td>ε₀</td>
<td>1</td>
<td>μm</td>
</tr>
<tr>
<td>Mass</td>
<td>m</td>
<td>1</td>
<td>mg</td>
</tr>
<tr>
<td>Damping constant</td>
<td>b</td>
<td>0.5</td>
<td>mg/s</td>
</tr>
<tr>
<td>Spring constant</td>
<td>k</td>
<td>1</td>
<td>mg/s²</td>
</tr>
<tr>
<td>Resistance</td>
<td>R</td>
<td>0.001</td>
<td>Ω</td>
</tr>
</tbody>
</table>

Electrostatic Actuator Dynamics

- Equilibrium point \( Z_0 \) : \( z_2 = 0.5, \dot{Z} = 0 \)

\[
\begin{bmatrix}
0 \\
0 \\
0 \\
\end{bmatrix}
= 
\begin{bmatrix}
v \\
R \frac{z_0}{2ReA} \\
\frac{z_0^2}{2eA} - \frac{k}{m} (0.5 - g_0) - \frac{b}{m} z_0 \\
\end{bmatrix}
\]

- At equilibrium (operating) point

\[
Z_0 = \begin{bmatrix}
\sqrt{eA} \\
0.5 \\
0 \\
\end{bmatrix}, \quad V_0 = \frac{1}{2\sqrt{eA}}\]
Electrostatic Actuator: Error Dynamics

Error Dynamics:

\[
\begin{align*}
X &\triangleq Z - Z_0 \\
u &\triangleq v - v_0 \\
\dot{x}_1 &= \frac{1}{R} \left( u - \frac{x_1}{2\epsilon A} - \frac{x_2}{\sqrt{\epsilon A}} - \frac{x_1^2}{\epsilon A} \right) \\
\dot{x}_2 &= x_3 \\
\dot{x}_3 &= -\frac{1}{m} \left( \frac{x_1^2}{2\epsilon A} + \frac{x_1}{\sqrt{\epsilon A}} + kx_2 + bx_3 + \frac{1}{2} + \frac{k}{2} - \frac{g_0}{k} \right)
\end{align*}
\]

Necessary Conditions of Optimality

- State Equation:

\[
\begin{bmatrix}
\dot{x}_{1i} \\
\dot{x}_{2i} \\
\dot{x}_{3i}
\end{bmatrix} =
\begin{bmatrix}
x_{1i} \\
x_{2i} \\
x_{3i}
\end{bmatrix} + \Delta \\
\begin{bmatrix}
\frac{u}{R} \\
\frac{2\epsilon AR}{R} \\
-\frac{R\sqrt{\epsilon A}}{m\sqrt{\epsilon A}} \\
\frac{x_1}{2\epsilon A} \\
\frac{x_2}{\sqrt{\epsilon A}} \\
-kx_3 \\
\frac{1}{2} + \frac{k}{2} - \frac{g_0}{k}
\end{bmatrix}
\]

- Costate Equation:

\[
\begin{bmatrix}
\lambda_{x_1} \\
\lambda_{x_2} \\
\lambda_{x_3}
\end{bmatrix} =
\begin{bmatrix}
Q_{x_1} \\
Q_{x_2} \\
Q_{x_3}
\end{bmatrix} =
\begin{bmatrix}
\left(1 - \frac{\Delta t}{2m\sqrt{\epsilon A}} \right) \\
-\frac{\Delta t}{R\sqrt{\epsilon A}} \\
0
\end{bmatrix}
\begin{bmatrix}
\Delta x_1 \\
\Delta x_2 \\
\Delta x_3
\end{bmatrix} + 
\begin{bmatrix}
0 \\
\frac{m\epsilon A}{m\epsilon A} \\
\frac{1}{m}
\end{bmatrix}
\begin{bmatrix}
\lambda_{\dot{x}_1} \\
\lambda_{\dot{x}_2} \\
\lambda_{\dot{x}_3}
\end{bmatrix} + 
\begin{bmatrix}
\left(-\frac{\Delta t}{m\epsilon A} \right) \\
\left(-\frac{\Delta t}{m\epsilon A} \right) \\
0
\end{bmatrix}
\begin{bmatrix}
\lambda_{x_1} \\
\lambda_{x_2} \\
\lambda_{x_3}
\end{bmatrix}
\]

- Optimal Control Equation:

\[
u = -R^{-1} \lambda_{x_1i}
\]
### Implementation

<table>
<thead>
<tr>
<th><strong>AC</strong></th>
<th><strong>SNAC</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Critic Network: 3-6-3</td>
<td>- Critic Network: 3-6-3</td>
</tr>
<tr>
<td>- Action Network: 3-6-1</td>
<td>- Convergence Criteria:</td>
</tr>
<tr>
<td>- Convergence Criteria:</td>
<td>[ | \lambda_p^j - \lambda_p^\infty | / | \lambda_p^\infty | &lt; 0.05 ]</td>
</tr>
<tr>
<td>Critic:</td>
<td>Action:</td>
</tr>
<tr>
<td>[ | u_p^j - u_p^{\infty} | / | u_p^{\infty} | &lt; 0.05 ]</td>
<td>[ | u_p^j - u_p^{\infty} | / | u_p^{\infty} | &lt; 0.05 ]</td>
</tr>
<tr>
<td>Cycle:</td>
<td>Cycle:</td>
</tr>
<tr>
<td>[ \text{err}<em>{r_i} - \text{err}</em>{r_{i+1}} &lt; 0.1 ]</td>
<td>[ \text{err}<em>{r_i} - \text{err}</em>{r_{i+1}} &lt; 0.1 ]</td>
</tr>
<tr>
<td>Weighing matrices:</td>
<td>Weighing matrices:</td>
</tr>
<tr>
<td>[ Q_W = I, \quad R_W = 1 ]</td>
<td>[ Q_W = I, \quad R_W = 1 ]</td>
</tr>
</tbody>
</table>

### Numerical Results

![Graphs showing numerical results for Critic Network (AC) and SNAC Network](image)

OPTIMAL CONTROL, GUIDANCE AND ESTIMATION
Prof. Radhakant Padhi, AE Dept., IISc-Bangalore
Numerical Results

Telescopic Cycle Training Time
Total Training Time $T_{\text{SNAC}} \approx 0.44 T_{\text{AC}}$

Cost Function Comparison For different initial conditions ($t_f = 25$)

**Conclusion**: $J_{\text{SNAC}} \approx J_{\text{AC}}$
Conclusions: SNAC

- SNAC retains all good features of AC
- Elimination of Action Networks:
  - Eliminates the approximation errors due to action network
  - No necessity of iterative training loops between Action and Critic networks
- Computationally simpler
- Leads to a significant amount of computational savings

Comment:
SNAC is applicable whenever the optimal control equation is explicitly (symbolically) solvable for control in terms of state and costate variables.

References

Thanks for the Attention....!!

Questions ... ??