A Comparison of DHP Based Antecedent Parameter Tuning Strategies for Fuzzy Control

Michael S. Carroll, Alec M. Rogers, Thaddeus T. Shannon and George G. Lendaris

Northwest Computational Intelligence Laboratory
Portland, Oregon USA

Supported by: National Science Foundation ECS-9904378
Overview

- Problem context
- Plant description
- Experimental methods
- Results
- Conclusion
Context

- Fuzzy TSK models

- Training of antecedents may improve input space partitioning

- DHP

- Modules using the same inputs
  - Controller
  - Critic
  - (Model)

- Research Questions
  - Does antecedent training improve performance?
  - Should the modules be tuned independently?
  - Which error signals should be used to do the tuning?
The Plant
(Narendra Benchmark)

- **System Equations**
  \[
  x_1(t + 1) = 0.9x_1(t)\sin(x_2(t)) + \left[2 + 1.5\frac{x_1(t)u_1(t)}{1 + x_1^2(t)u_1^2(t)}\right]u_1(t) + \left[x_1(t) + \frac{2x_1(t)}{1 + x_1^2(t)}\right]u_2(t)
  \]
  \[
  x_2(t + 1) = x_3[1 + \sin(4x_3(t))] + \frac{x_3(t)}{1 + x_3^2(t)}
  \]
  \[
  x_3(t + 1) = [3 + \sin(2x_1(t))]u_2(t)
  \]

- **Reference Signals**
  \[
  \tilde{x}_1(t) = 0.75\sin\left(\frac{2\pi t}{50}\right) + 0.75\sin\left(\frac{2\pi t}{10}\right)
  \]
  \[
  \tilde{x}_2(t) = 0.75\sin\left(\frac{2\pi t}{30}\right) + 0.75\sin\left(\frac{2\pi t}{20}\right)
  \]

- **Utility Function**
  \[
  U(t) = (\tilde{x}_1 - x_1)^2 + (\tilde{x}_2 - x_2)^2
  \]
Experiments

• Fuzzy TSK critic and controller modules

• Two sets of experiments
  - Shared antecedent parameters
  - Unshared antecedent parameters

• Four cases in each set
  - No antecedent training
  - Training using the critic
  - Training using the controller
  - Training using both
Methods

• Initial membership functions

• Training procedure
  - Random reference signal [-1.5, 1.5] each 40 time steps
  - 18 runs of 240,000 trials for each experimental case
  - Performance assessment every 4,000 trials
  - Simultaneous controller and critic training
  - Training rates:

<table>
<thead>
<tr>
<th></th>
<th>Controller</th>
<th>Critic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antecedents</td>
<td>0.0002</td>
<td>0.000005</td>
</tr>
<tr>
<td>Consequents</td>
<td>0.002</td>
<td>0.005</td>
</tr>
</tbody>
</table>
# Experimental Cases

- **Shared antecedent parameters**

<table>
<thead>
<tr>
<th></th>
<th>Critic not used in training</th>
<th>Critic used in training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller not used in training</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Controller used in training</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

- **Independent antecedent parameters**

<table>
<thead>
<tr>
<th></th>
<th>Critic not used in training</th>
<th>Critic used in training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller not used in training</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Controller used in training</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
Results:
Reference Tracking
Results:
Should modules be tuned independently?

Case 4: Shared
Case 8: Independent
Results:
Does antecedent training improve performance?
Which error signals should be used?

Shared

Independent

Case 1: No antecedent training
Case 2: Critic used
Case 3: Controller used
Case 4: Critic and Controller used

Case 5: No antecedent training
Case 6: Critic used
Case 7: Controller used
Case 8: Critic and Controller used
Results:
Boxplot

Case 1: No antecedent training
Case 2: Critic used
Case 3: Controller used
Case 4: Critic and Controller used
Case 5: No antecedent training
Case 6: Critic used
Case 7: Controller used
Case 8: Critic and Controller used
Conclusions

- Some improvement in training with antecedent tuning

- Antecedents tuned with controller or combined error signals performed slightly better in early training

- Shared antecedent parameters work as well as, or slightly better than, independently trained parameters

Therefore, the additional computational cost of separately tuned parameters is not justified—shared tuning is recommended