Evolving Robust and Reconfigurable Multi-Objective Controllers for Advanced Power Systems

Shauharda Khadka (Shaw)

PI: Kagan Tumer
Oregon State University
kagan.tumer@oregonstate.edu

NETL Project Manager: Sydni Credle
DE-FE0012302

March 22, 2017
Motivation: Energy Systems

• Where are we?
  - Advanced energy systems becoming more interconnected
  - Advanced Power Plants
  - Computation pushed further down the pipe
  - More powerful, cheaper, smaller devices
Motivation: Energy Systems

• Where are we?
  - Advanced energy systems becoming more interconnected
    - Advanced Power Plants
  - Computation pushed further down the pipe
  - More powerful, cheaper, smaller devices

• Where are we going?
  - Hybrid systems (eg. Hyper)
  - Competing objectives
  - Smart sensors, actuators
Motivation: Energy Systems

• Where are we?
  - Difficult to model
  - Distributed decision making
  - Scaling
Motivation: Energy Systems

• Where are we?
  - Difficult to model
  - Distributed decision making
  - Scaling

• Where are we going?
  - Even more difficult to model
  - Even more distributed decision making
  - Even more scaling
Motivation: Energy Systems

- We need to account for?
  - Model inaccuracies (or lack of models)
  - Thousands of actors (sensors, controllers, users)
  - Failing components
  - Competing objectives
  - Dynamic and stochastic environments

- And still control systems to result in safe, efficient operation
Outline

• Motivation: multiagent, multi-objective control in complex systems
• Roadmap & objectives

• Key Milestones for last year
  • M 5: Develop robust controller
  • M 6: Develop reconfigurable controller

• Summary & Project Status
Roadmap and Objectives

- Learning-Based Control: **multiagent, multi-objective control in complex systems**

- Multiagent
  - Biomimetic distributed subsystem-level control
  - System-level results

- Multi-objective Optimization
  - Simultaneously optimize multiple competing objective functions

- Reconfigurable
  - Adapt to changing power system needs
  - Develop new policies with previously unconsidered objective functions
Roadmap and Goals

- Learning-Based Control: multiagent, multi-objective control in complex systems

- Multiagent
  - Biomimetic distributed subsystem-level control
  - System-level results

- Multi-objective Optimization
  - Simultaneously optimize multiple competing objective functions

- Reconfigurable
  - Adapt to changing power system needs
  - Develop new policies with previously unconsidered objective functions
## Project Milestones

<table>
<thead>
<tr>
<th>Milestone Number</th>
<th>Milestone Title</th>
<th>Planned Completion Date</th>
<th>Actual Completion Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Develop an abstract simulator for advanced power systems</td>
<td>June 2014</td>
<td>June 2014 ✓</td>
</tr>
<tr>
<td>2</td>
<td>Develop bio-mimetic control algorithm for advanced power systems</td>
<td>Sept. 2014</td>
<td>Sept. 2014 ✓</td>
</tr>
<tr>
<td>3</td>
<td>Develop system metrics to measure tradeoffs of plant objectives</td>
<td>March 2015</td>
<td>March 2015 ✓</td>
</tr>
<tr>
<td>4</td>
<td>Develop multi-objective control algorithm for advanced power systems</td>
<td>Sept. 2015</td>
<td>Sept. 2015 ✓</td>
</tr>
<tr>
<td>5</td>
<td>Develop robust controller for advanced power system</td>
<td>June 2016</td>
<td>June 2017 Ongoing</td>
</tr>
<tr>
<td>6</td>
<td>Develop reconfigurable, multi-objective controller for advanced power system</td>
<td>Sept. 2016</td>
<td>September 2017 Ongoing</td>
</tr>
</tbody>
</table>
## Project Milestones

<table>
<thead>
<tr>
<th>Milestone Number</th>
<th>Milestone Title</th>
<th>Planned Completion Date</th>
<th>Actual Completion Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Develop an abstract simulator for advanced power systems</td>
<td>June 2014</td>
<td>June 2014 ✓</td>
</tr>
<tr>
<td>2</td>
<td>Develop bio-mimetic control algorithm for advanced power systems</td>
<td>Sept. 2014</td>
<td>Sept. 2014 ✓</td>
</tr>
<tr>
<td>3</td>
<td>Develop system metrics to measure tradeoffs of plant objectives</td>
<td>March 2015</td>
<td>March 2015 ✓</td>
</tr>
<tr>
<td>4</td>
<td>Develop multi-objective control algorithm for advanced power systems</td>
<td>Sept. 2015</td>
<td>Sept. 2015 ✓</td>
</tr>
<tr>
<td>5</td>
<td>Develop robust controller for advanced power system</td>
<td>June 2016</td>
<td>June 2017 Ongoing</td>
</tr>
<tr>
<td>6</td>
<td>Develop reconfigurable, multi-objective controller for advanced power system</td>
<td>Sept. 2016</td>
<td>September 2017 Ongoing</td>
</tr>
</tbody>
</table>
Project Overview

HyPer Facility
Run Data

State Modeling
Neural Network
Project Overview

HyPer Facility
Run Data

State Modeling
Neural Network

Successor State

Time Domain Simulator (milestone 1)
Project Overview

HyPer Facility Run Data

State Modeling Neural Network

Time Domain Simulator (milestone 1)

Candidate Controller

Bio-Mimetic Controller (milestone 2)

Neuroevolutionary Algorithm

Fitness

Successor State

Control, Action, Current State
### Project Overview

- **HyPer Facility Run Data**
- **Multi-Objective Fitness Assignment (milestones 3)**
- **State Modeling Neural Network**
- **Candidate Controller**
  - **Neuroevolutionary Algorithm**
  - **Bio-Mimetic Controller (milestone 2, 4)**
- **Evaluation of Robustness and Reconfigurability (milestones 5, 6)**
- **Time Domain Simulator (milestone 1)**

**Flowchart Diagram:**
- **Control**
- **Action**
- **Current**
- **State**
- **Successor State**
Project Overview

Multi-Objective Fitness Assignment (milestones 3)

HyPer Facility Run Data

State Modeling Neural Network

Successor State

Time Domain Simulator (milestone 1)

Control Action Current State

Candidate Controller

Neuroevolutionary Algorithm

Bio-Mimetic Controller (milestone 2, 4)

Evaluation of Robustness and Reconfigurability (milestones 5, 6)
Outline

- Motivation: multiagent, multi-objective control in complex systems
- Roadmap & objectives

  - Key Milestones for last year
    - M 5: Develop robust controller
    - M 6: Develop reconfigurable controller

- Summary & Project Status
Milestone 5: Robust Controller

- Train the controller for robustness to noise
  - Neural networks are known to be more robust to noisy inputs
  - Translate this robustness onto the controller

Deliverable: Train controller that are robust to actuator and sensor noise
Input and desired State trajectory

Start State

Desired State
Add controller

Start State

Controller

Control Action

Desired State
Add Simulator

Start State → Control Action

Controller

Desired State

Neural Network Simulator

New State
Compute fitness

- Start State
- Desired State
- Control Action
- New State
- Fitness

- Controller
- Neural Network Simulator
Close the Loop

Start State → Controller → Neural Network Simulator

Desired State → Fitness

New State → Control Action

Start State
Desired Turbine profile

![Graph showing Desired Turbine profile with time on the x-axis and normalized ST-502 (Turbine Speed) on the y-axis, with peaks at 0.6 and troughs at 0.4 over a 200-minute period.]

Kagan Tumer, Oregon State University
Controller trained with Perfect Information

![Graphs showing normalized ST-502 (Turbine Speed) over time.](image)
Controller trained with Perfect Information
Controller trained with Perfect Information

![Graph showing normalized ST-502 turbine speed over time with different noise conditions.](Image)
Training with Perfect Information Takeaways

• Not robust to noise
• Detrimentally rapid fluctuations in Turbine speed
Training with Perfect Information Takeaways

- Not robust to noise
- Detrimentally rapid fluctuations in Turbine speed

Solution:

- Integrate noise in controller training scheme
- Gaussian noise
Controller Learning Setup with Perfect information

Start State -> Controller -> Neural Network Simulator -> Desired State -> Fitness

Fitness -> New State -> Controller

Control Action
Ading Actuator Noise
Adding Sensor Noise

Start State → Controller → Neural Network Simulator → Fitness

Start State → Desired State → Fitness

New State → Noise → Control Action

Noise
Controller trained with 5% noise
Controller trained with 5% noise

[Graphs showing normalized ST-502 turbine speed over time with and without noise conditions.]
Controller trained with 5% noise
Controller trained with 5% noise
Controller trained with 10% noise
Controller trained with 10% noise
Controller trained with 10% noise
Controller trained with 10% noise
Training Neuro-Controllers With Noise

• Integrate Gaussian noise to controller and simulator output and train the controller in a loop

• Optimize for overall performance using an evolutionary algorithm
Training Neuro-Controllers With Noise

- Integrate Gaussian noise to controller and simulator output and train the controller in a loop
- Optimize for overall performance using an evolutionary algorithm

Robust controller capable of handling noise for both sensors and actuators
Outline

- Motivation: multiagent, multi-objective control in complex systems
- Roadmap & objectives

  - Key Milestones for last year
    - M 5: Develop robust controller
    - M 6: Develop reconfigurable controller

- Summary & Project Status
Milestone 6: Reconfigurable Controller

• Reconfigurable controller that can adapt to fluctuating demands

  - Demand profile normally periodic and stable
  - Not always!
  - Special circumstances can briskly alter demand profile

Need: Need controller than can reconfigure to demand, on demand!
Let’s look at reconfigurability

• Adapt behaviors to different performance profiles
• Rudimentary solution
  - Enumerate different performance profiles
  - Learn specific controller for them
  - Pick a controller, based on performance profile required
Let’s look at reconfigurability

• Adapt behaviors to different performance profiles
• Naive solution
  - Enumerate different performance profiles
  - Learn specific controller for them
  - Pick a controller, based on performance profile required

PROBLEM:
• Enumeration intractable
• Ignores dynamics
How may that look like?
Problematic transitions in dynamic space

- **Heat Wave**
- **Normal Demand**

![Diagram showing turbine speed over time with heat waves and normal demand](image-url)
A better transition

- Heat Wave
- Normal Demand
Need to account for dynamics

- Non-Markovian state
- Need to account for path taken to get there and where it’s headed next

One possible solution is:
Need to account for dynamics

- Non-Markovian state
- Need to account for path taken to get there and where it’s headed next

One possible solution is:

- **MEMORY**
  - Consider path taken to get there and direction headed
  - Controller utilizes this information to reconfigure efficiently
Memory-Augmented Controller

- Use **Memory-Augmented Neural Networks (MANNs)**
  - Neural Networks augmented with memory
  - Deep Neural Network (perhaps the deepest kind)
  - “External” Memory
  - Capture long-term dependencies in the data
  - Capture variable term dependencies in the data
Two Major Types of MANNs

1. Small Memory tied with computation
   a. Long Short term Memory (LSTM)
   b. Gated Recurrent Unit (GRU)

2. Big external Memory Bank that is interacted with
   a. Neural Turing Machine (NTM)
   b. Differential Neural Network (DNC)
Two Major Types of MANNs

1. Small Memory tied with computation
   a. Long Short term Memory (LSTM)
   b. Gated Recurrent Unit (GRU)

2. Big external Memory Bank that is interacted with
   a. Neural Turing Machine (NTM)
   b. Differential Neural Network (DNC)

Solution:

• Combine the best of both worlds!
Gated Recurrent Unit with Memory Block (GRU-MB)

- Detached memory from computation
- Retained adjustable size tractable to train
Feedforward Net
Read from external memory
Write to memory
Gate Input
Gate what’s read from memory
Gate what’s written to memory
Gated Recurrent Unit with Memory Block (GRU-MB)


Kagan Tumer, Oregon State University
GRU-MB Results

• Sequence Classification Task

<table>
<thead>
<tr>
<th>Input Sequence</th>
<th>Target Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1 0....0 1 0....0 -1</td>
<td>-1 .... 1 .... -1</td>
</tr>
<tr>
<td>-1 0....0 -1 0....0 1</td>
<td>-1 .... -1 .... -1</td>
</tr>
<tr>
<td>-1 0....0 1 0....0 1</td>
<td>-1 .... 1 .... 1</td>
</tr>
<tr>
<td>1 0....0 1 0....0 -1</td>
<td>1 .... 1 .... 1</td>
</tr>
</tbody>
</table>
Classification Accuracy

![Bar chart showing classification accuracy for different task depths and models.]

- NEAT-RNN (4-6 deep)
- NEAT-LSTM (4-6 deep)
- GRU-MB

Success Percentage

<table>
<thead>
<tr>
<th>Task Depth</th>
<th>4-deep</th>
<th>5-deep</th>
<th>6-deep</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEAT-RNN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEAT-LSTM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRU-MB</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Classification Accuracy

The diagram illustrates the classification accuracy for different task depths using various models:

- **NEAT-RNN (4-6 deep)**
- **NEAT-LSTM (4-6 deep)**
- **GRU-MB**

The success percentage is shown for task depths ranging from 4-deep to 21-deep.
Depth generalization

![Graph showing depth generalization with success percentage on the y-axis and depth on the x-axis. The line graph indicates a decrease in success percentage as depth increases.](Image)
Depth generalization

![Graph showing depth generalization](image)

- 4-deep
- 5-deep
- 6-deep
- 15-deep
- 21-deep

Success Percentage vs. Depth
Next Steps

• GRU-MB tested and verified on benchmark sequence classification tasks
• Translate this onto an advanced power plant application
• Customize GRU-MB
• Train GRU-MB as reconfigurable power plant controllers
## Project Milestones

<table>
<thead>
<tr>
<th>Milestone Number</th>
<th>Milestone Title</th>
<th>Planned Completion Date</th>
<th>Actual Completion Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Develop an abstract simulator for advanced power systems</td>
<td>June 2014</td>
<td>June 2014 ✔ Ongoing</td>
</tr>
<tr>
<td>2</td>
<td>Develop bio-mimetic control algorithm for advanced power systems</td>
<td>Sept. 2014</td>
<td>Sept. 2014 ✔</td>
</tr>
<tr>
<td>3</td>
<td>Develop system metrics to measure tradeoffs of plant objectives</td>
<td>March 2015</td>
<td>March 2015 ✔</td>
</tr>
<tr>
<td>4</td>
<td>Develop multi-objective control algorithm for advanced power systems</td>
<td>Sept. 2015</td>
<td>Sept. 2015 ✔</td>
</tr>
<tr>
<td>5</td>
<td>Develop robust controller for advanced power system</td>
<td>June 2016</td>
<td>June 2017</td>
</tr>
<tr>
<td>6</td>
<td>Develop reconfigurable, multi-objective controller for advanced power system</td>
<td>Sept. 2016</td>
<td>September 2017</td>
</tr>
</tbody>
</table>
Publications


Publications


Acknowledgements:

• Department of Energy, NETL
• Sydni Credle, Project Manager

• Students:
  - Shauharda (Shaw) Khadka, Drew Wilson, Logan Yliniemi, Drew Gabler
• Postdocs: Mitchell Colby, Jen Jen Chung

• Dave Tucker, NETL
• Paolo Pezzini, Kenneth Mark Bryden, Ames laboratory
Questions?

Contact Info:
Shauharda (Shaw) Khadka, Kagan Tumer
Oregon State University
khadkas@oregonstate.edu
kagan.tumer@oregonstate.edu
engr.oregonstate.edu/~ktumer/
Distributed multi-objective Control?

Multiagent control     multi-objective control

Many agents, one objective   One agent, many objectives
- Who does what?            - trade-off objectives

Many agents, many objectives
Project Overview

- **HyPer Facility Run Data**
- **State Modeling Neural Network**
- **Successor State**
- **Time Domain Simulator (milestone 1)**
- **Multi-Objective Fitness Assignment (milestones 3)**
- **Candidate Controller**
- **Neuroevolutionary Algorithm**
- **Bio-Mimetic Controller (milestones 2, 4)**
- **Evaluation of Robustness and Reconfigurability (milestones 5, 6)**
Feedforward Net
Read from an external memory

![Diagrams showing the process of reading from an external memory](image-url)
Write to memory
Gate input
Gate what’s read from memory
Gate what’s written to memory