Convergence of AI, Physics, Computing, and Control for Intelligent Power System Control

Qiuhua Huang
Associate Professor
Colorado School of Mines
(qiuhuahuang@mines.edu)

PSERC Webinar
February 15, 2023
Acknowledgement

Collaborators

• PNNL (including former colleagues): Yousu Chen, Renke Huang, Tianzhixi (Tim) Yin, Sohom Datta, Xueqing Sun, Bruce Palmer, Yuan Liu, Z. Jason Hou, Long Vu, Sayak Mukherjee, Kaveri Mahapatra, Xinda Ke, Shaobu Wang, Xinya Li, Yan Du, Yujiao Chen, Ramij Raja Hossain

• V&R Energy: Mariana Vaiman, Michael Vaiman

• Google: Jie Tan, Wenhao Yu

• EPRI: Adrian Kelly

• RTE-France: Antoine Marot, Benjamin Donnot, Karim Chao

• Song Wang (now with PGE, previously with PacifiCorp), Rui Fan (DU), Wei Qiao (UNL), Jochen Cremer (DTU),

**Funding support:** ARPA-E OPEN 2018 Program, DOE OE AGM, PNNL DeepScience LDRD
Presentation Outline

• Motivation and background

• Introduction to deep reinforcement learning

• Convergence of AI, physics, computing and control for intelligent power system control
  o Integrated framework
  o Open-source environments for DRL-based grid control
  o Advanced DRL algorithms for grid control
  o Test results, demo and applications

• Future Work and Potential Applications
Increasing Renewables and Rapidly Changing Operation Conditions

EIA projects renewables share of U.S. electricity generation mix will double by 2050

California ISO net load “duck curve”

Sources: EIA
Power System Stability Risks Are Becoming A Global Issue

Power System Stability Control and Operation for Keeping the Lights On

Local control

Wide-area control[1]

System control and operation (primarily by human operators today)

Big Challenges in Grid Operation and Control

September 8, 2011 Pacific Southwest Blackout in U.S. [1]

Texas was “seconds and minutes” away from catastrophic months long blackouts[2]

[2] https://www.texastribune.org/2021/02/18/texas-power-outages-ercot/
Two Types of System-level Control for Dealing with Contingencies

• Preventive control (prior to contingency)
  • Minimize the chance of emergencies
  • Often used conservatively to ensure sufficient security margin against creditable contingency events (e.g., N-1) all the time
  • Typical methods: generation redispatch, var compensation, demand response/load curtailment
  • Emerging methods: transmission switching

• Emergency control (post-contingency/disturbance)
  • Minimize impacts of large disturbances; serve as a safety net for the system
  • Usually used against severe or low probability-high impact contingencies
  • Typical control methods: load shedding, generator tripping, controlled islanding
  • Emerging methods: emergent demand response, HVDC redispatch

Focus of the talk today
The Grand Challenge of Achieving Intelligent Emergency Control

- Power system post-event emergency control has **strong requirements:**
  - **Scalability:** >20,000 buses (with 1000s of control devices now → millions in the near future)
  - **Solution time:** < 5 seconds
  - **Security and adaptability** (to fast-changing conditions)

- Existing control methods and their issues:
  - Rule-based control (**not adaptive, time-consuming to develop and update them**)
  - Model-predictive control (**scalability and solution time issues**)
  - Learning-based (or data-driven) control (**scalability, security and adaptability issues**)
Can We Transfer AI Successes in Games and Robotics to Complex Grid Control?

Credit: Nature

Credit: OpenAI

https://www.wesc.org/epubs/StateOfTheInterconnection/Pages/Western-Interconnection.aspx
Introduction to Deep Reinforcement Learning
# Reinforcement Learning vs Other ML Methods

<table>
<thead>
<tr>
<th>Machine learning categories</th>
<th>Reinforcement learning</th>
<th>Supervised learning</th>
<th>Unsupervised learning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of data</strong></td>
<td>Data for training include observations and rewards from the environment. They are usually not predefined. Data is generated on-the-fly by interacting with the environment.</td>
<td>Data for training include pairs of input and the desired output <em>(Needs a large amount of data and labels)</em></td>
<td>Data for training do not include desired outputs</td>
</tr>
</tbody>
</table>
| **Example methods** | • Q-Learning  
• Deep reinforcement learning | • Neural networks  
• Decision trees | • K-mean clustering  
• Autoencoders |
| **Specific tasks** | • Control  
• Scheduling  
• Sequential decision-making | • Classification  
• Regression (prediction) | • Discover clusters  
• Identify factors/structures |
Reinforcement Learning in a Nutshell

• **(Why)** Reinforcement learning (RL) is designed for solving *sequential decision-making problems* in a stochastic environment.

• **(What)** The goal is to learn a control policy to *maximizes expected accumulative reward* over time.

• **(How)** The agent learns a control policy *iteratively* through interacting with the environment via *trial-and-errors guided by the reward signal*.

• **Deep** reinforcement learning = *deep learning* + reinforcement learning
  - Not just deep neural network, but more about deep learning techniques and ecosystem.

Agent (policy) is represented by deep neural network

- Observations
- Control Actions

Power Grid Control
- Actions
- Observations

\[ a_t \rightarrow s_t \rightarrow s_{t+1} \rightarrow r_t \]

- Power Grid
## Key Challenges in DRL for Grid Control

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>large system (observation space), large number of control points (action space) and scenarios (exploration space)</td>
</tr>
<tr>
<td>Adaptability</td>
<td>Adapt to changing operation conditions and network topology</td>
</tr>
<tr>
<td>Security</td>
<td>High security requirement (Insecure or outage is almost prohibitive)</td>
</tr>
<tr>
<td>Trustworthiness</td>
<td>Control actions or decisions should be trusted by operators and engineers</td>
</tr>
<tr>
<td>Training and test environments</td>
<td>• Grid environments are essential for DRL training and test; should be OpenAI gym-compatible</td>
</tr>
<tr>
<td></td>
<td>• Grid simulator ≠ environment for DRL training and test</td>
</tr>
<tr>
<td>Reproducibility</td>
<td>• The complexity of DRL for grid control makes them hard to reproduce</td>
</tr>
<tr>
<td></td>
<td>• Open-source is a proven, good practice for reproducibility</td>
</tr>
</tbody>
</table>

A single domain breakthrough is unlikely to address these challenges! → Need holistic solutions
Convergence of AI, physics, computing and control for intelligent power system control
A Holistic Approach to Intelligent Grid Control

An Integrated Framework for achieving convergence of key techniques

Advanced DRL Algorithms for large-scale power grid control

Open Source Environments for developing, testing and benchmarking solutions
Four Technology Pillars

Intelligent Grid Control for enabling grid transformation

Physics  Control  AI  Computing

How to achieve technology convergence to maximize impacts?
An Integrated Framework for Technology Convergence

Grid Control (e.g., load shedding)

Physics
(Power system models, performance requirements)

AI
(e.g. DMRL, surrogate model)

Data-driven control

Models & Data
Design & Validation
Actions

Models & Data
Analyze

Models & Data

HPC: High performance computing

DMRL: Deep Meta-Reinforcement Learning

HPC & Grid Simulator

Computing
RLGC: A Lightweight, Open-source Environment for RL for Grid Dynamic Control

- The first open-source grid environment for developing, testing and benchmarking RL algorithms for grid dynamic control[1]
- Applications: load shedding for voltage control, generator tripping, dynamic breaking, controlled islanding (to be released), T&D coordination (to be released)

Open-sourced on GitHub: https://github.com/RLGC-Project/RLGC

An HPC-based Grid Environment

- Scalable from laptop to HPC clusters/clouds with the RAY platform [1].
- GridPACK: HPC-based advanced grid dynamic simulator [2]
- The environment and DRL algorithms are open-sourced: https://github.com/pnnl/HADREC/

[1] https://www.ray.io/
An Example: Load Shedding for Emergency Voltage Control

**Dynamic simulation engine**

- **IEEE 300-bus system model**

**A Neural Network for representing agent’s policy**

- **$O_t$: Observations**
  - 154 bus voltage magnitudes and 46 bus load levels

- **$r_t$ (for training)**

- **$a_t$: Actions**
  - 46 load substations could shed load.
  - Each area, for each training time step, the load could be shed between 0% and 20%.
  - The action space is 46.

---

We Need Scalable RL Methods

• Training a RL agent for IEEE 39-bus system using the DQN algorithm on a laptop took about 2 days [1]
  • Existing methods are very difficult, if not impossible, to scale up for controlling large power systems (e.g., >2000 buses)

• Rich Sutton: “The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin” [2]

• We target at real-world large systems→ Scalable RL methods

Three Types of RL Methods

- **Model-based RL**
  - PILCO
  - Dyna

- **Model-free RL**
  - PPO
  - DDPG
  - DQN

- **Derivative-free RL**
  - ARS
  - ES

---

Parallel Augmented Random Search Algorithm

• Based on evolutionary strategy
• Unique features/advantages
  • Easy to scale and parallel (via workers)
  • Easy to tune (only 5 main hyper-parameters vs 20+ in existing DRL algorithms)
  • More robust for training (much less sensitive to random seeds and hyper-parameters)
  • Support larger learning rates to achieve faster training

Parallel ARS Algorithm Test Results

- Emergency voltage control on the IEEE 300-bus system

High scalability of Parallel ARS

Much faster and more robust training with larger average rewards using Parallel ARS

Learning a Context to Enhance Control Adaptiveness and Effectiveness

- We can enhance the control adaptiveness and effectiveness if we could efficiently construct a context for the control agent.
- **Challenge:** What should be the proper information for representing the context for different grid controls?
- Learn the context automatically from data and learning experiences via **meta-learning**
- Mathematical formulation:

\[
\theta^*, c(\varepsilon)^* = \arg \max_{\theta, c(\varepsilon)} \mathbb{E}_{\varepsilon \sim P(E_i)} \left[ J(c(\varepsilon), \theta) \right]
\]

- **\(\theta\)** – neural network weights for control policy
- **\(\varepsilon\)** – one training environment
- **\(c(\varepsilon)\)** – context representation for environment \(\varepsilon\)
- **\(P(E_i)\)** – a distribution of training environments

Source: CAISO
Realization of Deep Meta-Reinforcement Learning

- Reformulate (1) as (2) and (3) and solve them in an iterative manner (similar to the coordinate descent method):
  - **RL**: Learn the control policy $\theta$ while leveraging the latest context $c(\varepsilon)$ by solving equation (2)
  - **Meta-RL**: Learn the low dimensional latent context $c$ using Bayesian Optimization to solve (3)
- Fast adaptation by only solving (3)

---

Our DMRL Algorithm Fits Well into Power System Operation Time Frames and Procedures
IEEE 300-Bus Power Flow Conditions for Training and Test

We use this small system to compare our algorithms with existing solutions that are difficult to scale.

<table>
<thead>
<tr>
<th>Training power flow case</th>
<th>Generation</th>
<th>Load</th>
<th>Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Total 22929.5 MW (100%)</td>
<td>Total 22570.2 MW (100%)</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>120% for all generators</td>
<td>120% for all loads</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>135% for all generators</td>
<td>135% for all loads</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>115% for all generators</td>
<td>150% for loads in Zone 1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Adaptation/testing power flow case</th>
<th>Generation</th>
<th>Load</th>
<th>Generation</th>
<th>Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>90% for all generators</td>
<td>90% for all loads</td>
<td>92.4% for all generators</td>
<td>92.4% for all loads</td>
</tr>
<tr>
<td>2</td>
<td>110% for all generators</td>
<td>110% for all loads</td>
<td>107.7% for all generators</td>
<td>107.7% for all loads</td>
</tr>
<tr>
<td>3</td>
<td>115% for all generators</td>
<td>115% for all loads</td>
<td>117.2% for all generators</td>
<td>117.2% for all loads</td>
</tr>
<tr>
<td>4</td>
<td>125% for all generators</td>
<td>125% for all loads</td>
<td>122.5% for all generators</td>
<td>122.5% for all loads</td>
</tr>
<tr>
<td>5</td>
<td>140% for all generators</td>
<td>140% for all loads</td>
<td>142.1% for all generators</td>
<td>142.1% for all loads</td>
</tr>
<tr>
<td>6</td>
<td>95% for all generators</td>
<td>82.8% for loads in Zone 1</td>
<td>97.1% for all generators</td>
<td>85.32% for loads in Zone 1</td>
</tr>
<tr>
<td>7</td>
<td>107% for all generators</td>
<td>124.4% for loads in Zone 1</td>
<td>104.6% for all generators</td>
<td>121.5% for loads in Zone 1</td>
</tr>
<tr>
<td>8</td>
<td>110% for all generators</td>
<td>134.3% for loads in Zone 1</td>
<td>112.3% for all generators</td>
<td>137.2% for loads in Zone 1</td>
</tr>
<tr>
<td>9</td>
<td>119% for all generators</td>
<td>159.7% for loads in Zone 1</td>
<td>121.1% for all generators</td>
<td>162.6% for loads in Zone 1</td>
</tr>
</tbody>
</table>

Deep Meta-Reinforcement Learning Test Results

<table>
<thead>
<tr>
<th></th>
<th>DMRL</th>
<th>PARS</th>
<th>MPC*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>11.6 hours</td>
<td>9.5 hours</td>
<td>N/A</td>
</tr>
<tr>
<td>Adaptation</td>
<td>5.3 mins</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Solution time</td>
<td>0.7 sec</td>
<td>0.7 sec</td>
<td>63.3 sec</td>
</tr>
</tbody>
</table>

*MPC: Model-predictive control

All test scenarios are unseen during training

Physics-informed DMRL Enhances Training Efficiency and Control Robustness

• Power system community have developed vast amount of domain knowledge in forms of physics laws, standards, rules, and performance requirements.

• Physics-informed DMRL: we incorporated system performance requirements as a trainable action mask (TAM) into the agent and significantly improved its sampling efficiency and robustness[1].

Incorporate prior knowledge into the agent with a fixed action mask [1]

31

A Summary of Our Algorithms

- **Parallel Augmented Random Search (PARS) algorithm** [1],
- **High performance power system simulation platform GridPACK**
- **Physics-informed PARS** [2]: incorporate physics knowledge through a trainable action mask
- **Safe PARS** [3]: control barrier function + PARS
- **Deep meta-reinforcement learning (meta-learning + PARS)** [4]: realize fast adaptation (~5 mins) of control policies to changing grid conditions

Test Results and Demo on Large Systems
A Synthetic 3000-bus Texas System

Single-line diagram

Generation mix

Hourly renewable outputs and net load demands

Source: https://electricgrids.engr.tamu.edu/electric-grid-test-cases/activsg2000/
Texas 3000-bus System Dataset for Training and Testing

1440 power flow cases (2 months)

100 power flow cases

140 fault buses, 1 fault duration (0.1 s)

14,000 scenarios

280 fault buses, 4 fault durations ∈ [0.08, 0.4]

200 power flow cases

2000 buses

Smart sampling [1]

Training data set

Test data set

Texas 3000-bus System Off-line Training Results

- **Action**: load shedding fractions at 258 load buses, continuous, range [-0.2, 0]
- **Observation**: 726 bus voltage magnitudes + 258 substation load demand

- 100 power flow cases
- 140 fault buses, 1 fault duration (0.1 s)
- 14,000 scenarios
- Objective value (total reward) differences (positive is better)

- Training curve (running avg. rewards vs training time)
  - Training time: ~2 days
  - Objective value: 99.74%
Off-line Testing Results

200 power flow cases

56,000 scenarios

280 fault buses, 4 fault durations ∈ [0.08, 0.4]

Histogram of % reduced load shedding compared with ruled-based UVLS (positive is better)

26% reduction on average

Better in 99.7% scenarios

Objective value (total reward) differences (positive is better)
Demo on a Real Large System

• In the ARPA-E HADREC project, the platform and DMRL algorithm are integrated with V&R’s real-time situational awareness tool (ROSE)

• Trained and tested on a large system based on real-world EMS snapshots provided by PacifiCorp

• We considered three emergency controls:
  • Load shedding for voltage control
  • Generator tripping for transient stability (next slide)
  • Controlled islanding for avoiding system-wide instability
Case study of a PacifiCorp system

- HADREC: 1 generator tripped, 300MW total
- Out of Step (OOS): 8 generators tripped, 1 GW total
- ~20% improvement in responding time and 70% reduction in tripped generator output
Other Applications
Improve DFIG Performance in Grid Fault Scenarios

• Reduce the DFIG rotor over-current and DC-link over-voltage under different fault conditions

Improve Inter-area Damping Control through HVDC and FACTS

Test results on the MinniWECC system from [1]


Test results on the MinniWECC system from [2]

Key takeaways

• The latest AI show “generalist” capabilities – the same (very similar) NN model and DRL algorithm can be applied to different domains and applications

• This helps reduce extensive work in special designs based on strong domain expertise.
Future Work and Perspectives
Trustworthy DRL for Grid Control

• Human-AI collaboration for intelligent grid control
• How to make DRL agent more trustworthy?
  • Good performance
  • Know its know limitations and alert human operators in advance
• “Learning to run power network with trust” formulation and competition [1]
• Research directions
  • Reinforcement learning from human feedback (RLHF)
  • Off-line reinforcement learning from historical human control actions
  • Explainable reinforcement learning (XRL)

Distributed Control with Edge AI and Coordination with Centralized Control

- Distributed control with edge computing and AI helps manage millions of DERs.
- Coordinating centralized and distributed control as well as computing is critical for large-scale clean energy integration and FERC 2222 compliance.
Other Potential Applications

• AI-Copilot for system operators
• Next-generation decision support tools for preventive control and network optimization
  • Optimal transmission switching [1]
• Mitigation plan for N-1-1 and N-2 contingencies during planning
• AI-bot for dispatcher(Operator) training system (DTS/OTS)
  • Generate training scenarios and optimal control actions

A Summary

- Intelligent power system control is key to safeguard the grid while enabling grid transformation and modernization.
- Solving such a grand challenge requires the convergence of Physics, AI, Control and Computing.
- We developed an integrated framework, open-sourced environments and advanced DRL algorithms.
- We showed promising results on large systems.
- Trustworthy AI, human-AI collaboration and AI-enhanced T&D coordination are important future research directions.
Publications


8. R. Hossain, Q. Huang, R. Huang. “Graph Convolutional Network-Based Topology Embedded Deep Reinforcement Learning for Voltage Stability Control”, in IEEE Transactions on Power Systems,
Questions?

Qiuhua Huang
qiuhuahuang@mines.edu
BACKUP SLIDES
**Reward Function Design: An Example**

- RL agents try to maximize the accumulative rewards
  - Opposite to the objective function of optimal control
  - In general, no convexity requirement \( \rightarrow \) more flexibility in design
- The reward function is usually application-specific

\[
\text{Reward} = \begin{cases} 
  c_i \sum_j \Delta V_i - c_j \sum_j \Delta P_j (p.u.) - c_i u_{invalid} & \\
  -10000, \text{ if } V_i(t) < 0.95, \ t > T_{pf} + 4 \\
  \min[V_i(t) - 0.7, 0], \text{ if } T_{post\_fault} < t < T_{post\_fault} + 0.33 \\
  \min[V_i(t) - 0.8, 0], \text{ if } T_{post\_fault} + 0.33 < t < T_{post\_fault} + 0.5 \\
  \min[V_i(t) - 0.9, 0], \text{ if } T_{post\_fault} + 0.5 < t < T_{post\_fault} + 1.5 \\
  \min[V_i(t) - 0.95, 0], \text{ if } T_{post\_fault} + 1.5 < t 
\end{cases}
\]

- Voltage criteria
- Load shedding
- Invalid actions

Large-penalty for non-acceptable performance
Meet the minimum performance requirement

Two-level Parallelism of PARS

Update Policy

ARS Learner

Direction Worker 0

Direction Worker i

Direction Worker a

Environment Rollout p

Environment Rollout p

Environment Rollout p

RLGC/HADREC

RLGC/HADREC

RLGC/HADREC
Parallel Augmented Random Search (ARS) Algorithm

- Highly scalable when combined with high performance computing and grid simulator [1] (see results in the next slide)
- Guided search by exploiting estimated gradients [2]

\[ \varepsilon_i = \alpha \varepsilon + (1 - \alpha) \mathcal{U} \varepsilon', \quad \varepsilon \sim N(0, \sigma I_n) \text{ and } \varepsilon' \sim N(0, \sigma I_k) \]

Basic idea: Estimate the gradient using random search

Inherent parallelism in the exploration

Parallel ARS Algorithm

\[ \nabla F(\theta) \approx \frac{1}{n \sigma} \sum_{i=1}^{n} \{ F(\theta + \sigma \varepsilon) \} \quad \text{where} \quad \varepsilon \sim N(0, I) \]

References:
Physics-inspired training

Physics: the voltage stability problem in power systems are mostly local issues

1. Areas are loosely coupled for voltage problems

2. Yet, actions in two or three of the regions are required for faults near or at the boundary of the regions.

3. Solutions: divided training and then coordinative training