The City College of New York

CSC 59866-E: Senior Project I

AI Agents for Decision Making in the Real World

By Saptarashmi Bandyopadhyay
Email: sbandyopadhyay@ccny.cuny.edu, sbandyopadhyay@gc.cuny.edu
Assistant Professor of Computer Science
City College of New York and Graduate Center at the City University of New York

April 20, 2026 CSC 59866

Advanced Topics: AI Agents for Smart Grid Power Orchestration

Logistics and Motivation

Recall Lecture 20: We looked at Agentic AI for scientific discovery, using diffusion models and robotic labs to autonomously synthesize new materials.

What powers those robotic labs? The modern electric grid is under unprecedented stress from extreme weather, aging infrastructure, and the massive energy demands of AI data centers themselves.

Today's Agenda

- From Predictive AI to Agentic Smart Grids.
- Multi-Agent Architectures: Decentralized & Hierarchical Frameworks.
- Interactive Problem: Multi-Agent Economic Dispatch (Calculus).
- The PowerAgent Framework: Foundation Models & Tool Use.
- Safe & Explainable Control via Digital Twins.

From Predictive AI to Agentic Smart Grids

The Smart Grid Crisis

The Traditional Grid: Based on deterministic models and human-in-the-loop decision making. Generation follows load.

The Modern Challenge: Renewable energy (solar/wind) is volatile and unpredictable. Distributed Energy Resources (DERs) like home solar panels and EVs turn consumers into "prosumers."

The Failure of Legacy Control: Traditional SCADA systems and human operators cannot react fast enough to the sub-second volatility of millions of decentralized energy endpoints.

Predictive AI vs. Agentic AI in Smart Grids

Predictive AI (Past 10 Years): Machine Learning in power systems was strictly used as a static oracle. "Predict the wind power output for tomorrow." A human operator takes that prediction and makes the decision.

Agentic AI (The Future): Next-generation Smart Grids use *autonomous decision-making*.

The Agentic Loop: An agent perceives grid states (voltage, frequency), reasons about goals (cost reduction, preventing blackouts), plans multi-step actions, and *executes* them autonomously.

Why Should we want to decentralize?

We can’t afford to have a single point of failure in a national energy grid!

Instead, we can leverage **Decentralized Multi-Agent Systems** that only get deployed at individual microgrids or substations, and then communicate only with their neighbors.

This approach results in less latency, more privacy for consumer data, and much more scalability compared to a centralized system relying on one node.

A Hierarchical Multi-Agent Framework

In recent research papers, there is a trend towards *Decentralized Hierarchies* that can be composed of several layers. As an example:

Local Agents can be in charge of the charging / discharging of an individual device (e.g. an electric car charger).

Microgrid Agents can manage the power of a small neighborhood area with potentially many of these individual devices.

Global Agents would coordinate the flow of power between these disparate microgrid systems.

The Power Balance Constraint

At every level of the hierarchy, the agents are bound by the fundamental law of physics (Kirchhoff's Laws).

The Power Balance Equation: Total generated power must exactly equal total load demand plus grid losses at every millisecond t:

$$\sum_{i=1}^{N_G} P_{G,i}(t) + \sum_{j=1}^{N_{RES}} P_{RES,j}(t) \pm \sum_{k=1}^{N_{ESS}} P_{ESS,k}(t) = \sum_{m=1}^{N_L} P_{L,m}(t) + P_{loss}(t)$$

P_G (Fossil Generators), P_{RES} (Renewables), P_{ESS}(Battery Storage), P_L(Load).

If this equation is violated, the grid frequency drops from 60Hz, causing catastrophic cascading blackouts.

Example: Economic Dispatch

Setup for Problem

Consider a Microgrid Agent that controls two Generators G1 and G2 with a combined 100MW of power that it needs to supply to a neighnorhood:

Each generator has a quadratic cost function (in dollars per hour) based on how much power P they produce:

- Cost of G1: $C_1(P_1) = 0.1P_1^2 + 20P_1$
- Cost of G2: $C_2(P_2) = 0.15P_2^2 + 15P_2$

The Constraint: The agent must balance the grid: P1 + P2 = 100
Your Task: Using calculus (the Method of Lagrange Multipliers), find the optimal power output for P1 and P2 that minimizes total cost.

Solution (Part 1)

Step 1: Set up the Lagrangian.

We want to minimize C1 + C2 subject to 100 - P1 - P2 = 0.

$$\mathcal{L} = (0.1P_1^2 + 20P_1) + (0.15P_2^2 + 15P_2) + \lambda(100 - P_1 - P_2)$$

Step 2: Take partial derivatives and set to zero.

$$\frac{\partial \mathcal{L}}{\partial P_1} = 0.2P_1 + 20 - \lambda = 0 \implies \lambda = 0.2P_1 + 20$$

$$\frac{\partial \mathcal{L}}{\partial P_2} = 0.3P_2 + 15 - \lambda = 0 \implies \lambda = 0.3P_2 + 15$$

Solution (Part 2)

Step 3: Equate the lambdas.

$$0.2P_1 + 20 = 0.3P_2 + 15$$
$$0.2P_1 - 0.3P_2 = -5$$

Step 4: Substitute the constraint (P2 = 100 - P1).

$$0.2P_1 - 0.3(100 - P_1) = -5$$
$$0.2P_1 - 30 + 0.3P_1 = -5$$
$$0.5P_1 = 25 \implies \mathbf{P_1 = 50}\ \text{MW}$$

Step 5: Solve for P2.

$$\mathbf{P_2 = 50}\ \text{MW}$$

The Agentic Takeaway: Instead of doing this math, a multi-agent Reinforcement Learning setup naturally converges to this exact cost-optimal state through decentralized reward maximization!

Foundation Models for Power Systems

LLMs in Power Systems

Can Large Language Models (LLMs) operate the power grid?

The Gap: LLMs hallucinate and lack real-time context. They cannot directly change the voltage of a transformer (the electric kind, not the machine learning kind).

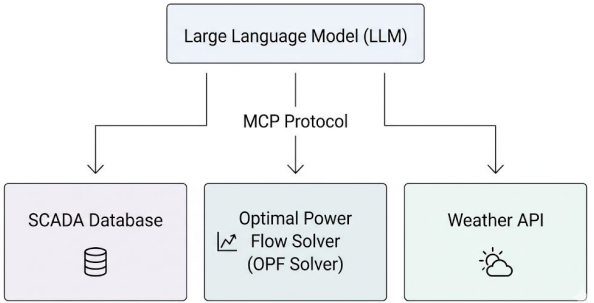
The PowerAgent Roadmap: Recent research proposes wrapping Foundation Models with strict, standardized tool interfaces to augment human operators.

The Model Context Protocol (MCP)

How does an agent actually interact with the physical grid? It uses the **Model Context Protocol (MCP)**.

Instead of generating text, the Agent uses Physical tools.

1. *Perception:* Agent uses MCP to query grid sensors: "Node 4 voltage is dropping."
2. *Reasoning:* LLM infers a storm has disrupted a solar farm.
3. *Action:* Agent uses MCP to call an external optimization script to re-route power, then presents the final, verified plan to the human operator for 1-click execution.



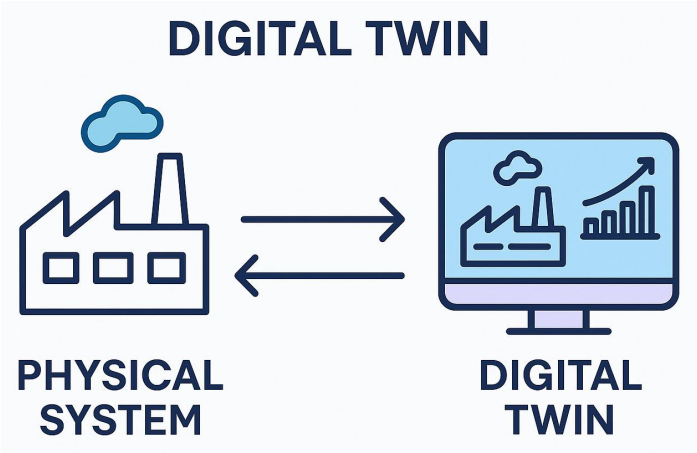
Safety, Explainable AI, and Digital Twins

Digital Twins for Safe RL Training

The RL Safety Problem: You cannot train an RL agent on the real power grid. Random exploration = rolling blackouts.

Digital Twins: High-fidelity, real-time virtual replicas of the physical power system.

Agents are trained inside the Digital Twin, where catastrophic failures carry no real-world cost. Once the policy stabilizes, it is transferred to the physical grid (Sim-to-Real transfer).



Physics-Informed and Explainable Control

Black-box neural networks are not trusted by grid operators.

Physics-Informed Neural Networks (PINNs): We hardcode Kirchhoff's laws into the loss function of the agent. If the neural network suggests an action that violates physical conservation of energy, the loss approaches infinity.

Explainability (XAI): As discussed in Lecture 18, we must use methods like SHAP or LIME to allow the human operator to ask the Agent: *"Why did you decide to turn on the backup diesel generator?"* -

Summary and Project Discussino

The Grid Evolution: Transitioning from centralized, deterministic models to decentralized, autonomous multi-agent hierarchies.

Mathematical Constraints: Agents are strictly bound by the physics of power balance and cost optimization (Economic Dispatch).

PowerAgent & Tools: LLMs cannot control the grid alone; they must use structured protocols (MCP) to access simulators and external solvers.

For your Senior Projects: If you are building multi-agent systems, how are you enforcing real-world constraints? Consider using a "Digital Twin" approach for your methodology, separating your agent's training environment from its final execution environment.

Questions?