**Home Energy Management Systems (HEMS): Optimized Implementation Using Spiking Neural Networks**

**1. Problem Formulation and Mathematical Model**

**1.1. Multi-Objective Optimization**

The HEMS optimization aims to minimize the total cost, maximize user comfort, and reduce environmental impact over a time horizon :

**Where:**

* : Control strategy vector (appliance schedules, battery, HVAC setpoints, etc.)
* : Electricity price at time (), possibly time-varying (e.g., TOU pricing)
* : Total household power consumption at time under control (kW)
* : Comfort utility, e.g., a sigmoid function of temperature deviation: where is the measured indoor temperature, is the user setpoint, and is a sensitivity parameter.
* : Environmental impact, e.g., total carbon emissions -

where is the grid emission factor for source .

* : User-defined weights for comfort and environmental impact.



**1.2. Constraints**

**a) Power Balance**

* : Set of all appliances.
* : Power drawn by appliance at time .
* : Power exchanged with the grid (import, export).
* : PV generation at .
* : Battery net output (discharging, charging).

**b) Battery Dynamics**

* : Battery state-of-charge at (kWh)
* , : Battery charging/discharging power at (kW)
* : Charging/discharging efficiency ()
* : Time step duration (h)
* : Battery capacity bounds (kWh)
* : Max charge/discharge rate (kW)

**c) Thermal Comfort and Dynamics**

* : Indoor temperature at
* : User-defined comfort bounds
* : Max allowed temperature change per time step

**Building Thermal Model:**

*Discretized:*

* : Thermal capacitance (kWh/°C)
* : Thermal resistance (°C/kW)
* : Outdoor temperature at
* : HVAC efficiency
* : HVAC power input (kW)
* : Internal heat gains (kW)

**d) Appliance Operational Constraints**

* : Maximum power for appliance

**e) Grid Exchange Limits**

* : Maximum grid import/export



**2. Data Acquisition and Preprocessing**

**2.1. Dataset Selection**

* **Pecan Street Dataport:** Appliance-level, high-resolution (1s) data (HVAC, PV, EV, etc.)
* **NREL ResStock:** End-use and weather-aligned U.S. household data
* **Others:** EAD, Plegma, Spanish Smart Meter, etc.

**2.2. Data Preparation Steps**

* **Integration:** Harmonize units, align timestamps, merge weather, price, and load data.
* **Feature Engineering:** Calculate derived features (e.g., cooling degree hours, occupancy proxies).
* **Normalization:** Scale variables for SNN input (e.g.,[[2]](#bppjg8qzjtf6) or z-score).
* **Spike Encoding:** Use rate or temporal coding to convert features to spike trains for SNN input.



**3. SNN-Based HEMS Architecture**

**3.1. Leaky Integrate-and-Fire (LIF) Neuron Model**

* : Membrane potential of neuron at
* : Membrane decay, = time constant
* : Synaptic current input
* : Synaptic current decay
* : Synaptic weight from neuron to
* : Output spike from neuron at

**3.2. Network Topology**

* **Input Layer:** Encodes all relevant features (power, temp, price, etc.) as spike trains.
* **Hidden Layers:** Two layers (e.g., 400 and 300 LIF neurons) extract temporal and nonlinear patterns.
* **Output Layer:** Generates real-valued control actions (e.g., appliance setpoints, battery dispatch).

**3.3. Encoding/Decoding**

* **Input Encoding:** Direct current injection or population rate coding.
* **Output Decoding:** Use membrane potential or spike count as continuous/discrete control signals.



**4. Reinforcement Learning Integration (TD3 Framework)**

**4.1. State Space**

**4.2. Action Space**

**4.3. Reward Function**

* Penalizes cost, comfort violations, and peak demand, rewards PV self-consumption.

**4.4. Policy Update (TD3)**

* : Policy parameters
* : Learning rate



**5. SNN-RL Co-Training and Temporal Credit Assignment**

* **STDP (Spike-Timing-Dependent Plasticity):**
  + : Spike times for pre/post-synaptic neurons
  + : Time constants for potentiation/depression



**6. Safety Layer (Constraint Enforcement)**

Quadratic programming ensures RL actions are feasible:

* : RL-suggested action
* : Constraint matrices (from above constraints)



**7. Implementation Workflow**

**Phase 1: Data Preparation**

* Integrate, clean, and encode data for SNN input.

**Phase 2: SNN-RL Co-Training**

* Train SNN encoder and RL actor-critic jointly with surrogate gradients and TD3.

**Phase 3: Real-Time Control**

* At each step, encode state, compute action, enforce constraints, and apply to HEMS.



**8. Conclusion**

This framework provides a mathematically rigorous, stepwise approach for deploying SNN-based HEMS with explicit constraint handling and deep RL integration. Every variable, equation, and step is defined for clarity and reproducibility. Future work will focus on neuromorphic hardware deployment and real-world pilot validation.