Thesis Model 1

Lander Maes

October 2025

1 Problem Formulation

Businesses often purchase their expected electricity needs on the day-ahead market. However, their actual consumption on the day of delivery can differ from what was planned. This difference, or *imbalance*, must be settled on the imbalance market, where prices are volatile and can be significantly higher than on the day-ahead market.

To mitigate the financial risks associated with this price volatility, businesses can use a Battery Energy Storage System (BESS). A battery allows a business to store energy when prices are low (or even negative) and use that stored energy when prices are high. This avoids the need to buy expensive electricity from the grid to cover an imbalance.

The challenge is to develop an intelligent **control policy** for such a battery. This policy must decide at any given moment whether to:

- **Charge** the battery from the grid,
- **Discharge** the battery to meet the business's needs or sell back to the grid,
- **Do nothing** and wait for a better opportunity.

The goal of this policy is to operate the battery in a way that minimizes the total electricity cost for the business.

2 Model Formulation

This documents describes the models we will be using, based on the framework by Powell. Their are two basic models that will be described.

- **Model 1:** A model without an explicit, external forecaster. The future price is treated directly as the uncertain exogenous information.
- **Model 2:** A model with an external forecaster. The error between the forecast and the real value will be treated as uncertain exogenous information.

2.1 Variable Definitions

The variables used are defined as follows:

R_t The amount of energy (in MWh) stored in the battery at time t.

D_t The electricity demand of the user at a given time t.

p_t The imbalance price (€/MWh) on the electricity grid at time t.

x_t The amount of power (in MW) bought from the grid ($x_t > 0$) or sold to the grid ($x_t < 0$) during the interval at time t.

$\mathbb{I}_t$ An indicator variable which is 1 with probability p^{jump} (the probability of a jump occurring).

ε_{t+1} The normal random noise term, $\varepsilon_{t+1} \sim N(0, \sigma^2)$.

ε^J_{t+1} The jump noise term, $\varepsilon^J_{t+1} \sim N(\mu^{jump}, (\sigma^{jump})^2)$, representing the magnitude of the jump.

3 Model 1

3.1 Model Components

1. **State Variable (S_t)**
   The state describes the information available to the agent at time t.

   $$S_t = (R_t, D_t, p_t)$$

2. **Decision Variable (x_t)**
   This is the action taken by the agent at time t. We define four distinct flows of power:

   x^{gb}_t Amount of power delivered from the grid to the battery (charging).

   x^{gd}_t Amount of power delivered from the grid to meet the business's demand.

   x^{bg}_t Amount of power delivered from the battery to the grid (selling).

   x^{bd}_t Amount of power delivered from the battery to meet the business's demand.

   These variables are subject to the following constraints at every time step t:

$$
\begin{aligned}
x_t^{gd} + x_t^{bd} &= D_t \\
x_t^{bg} + x_t^{bd} &\leq R_t \\
x_t^{gb} &\leq (R^{max} - R_t) \\
x_t^{gb} &\leq r^{charge} \\
x_t^{bg} + x_t^{bd} &\leq r^{discharge} \\
x_t^{gb}, x_t^{gd}, x_t^{bg}, x_{[}t]^{bd} &\geq 0
\end{aligned}
$$

3. **Exogenous Information (W_{t+1})**
   This is the new, unknown information that arrives after decision x_t is made.
   $$W_{t+1} = (p_{t+1}, D_{t+1})$$

4. **Transition Function**
   Since the future price p_{t+1} is directly observed as the new exogenous information, no explicit transition function is needed to model the price evolution. The transition of the battery's state of charge, R_t, is deterministically determined by the decision x_t.

5. **Objective Function ($C(S_t, x_t)$)**
   This represents the immediate cost or reward of taking decision x_t in state S_t.
   $$C(S_t, x_t) = -p_t x_t$$
   The overall goal is to maximize the sum of these contributions over time.

3.2 Modeling Uncertainty

To make decisions, the agent needs a model for the exogenous information, $W_{t+1} = p_{t+1}$. We start with a simple price model where the future price is the current price plus some random noise.

We model the price evolution according to the equation:

$$p_{t+1} = p_t + \varepsilon_{t+1}$$

Here, ε_{t+1} represents the uncertainty about the price change in the next time step. The challenge lies in modeling this error term.

3.2.1 Jump diffusion

We can model the evolution of the price with the Jump Diffusion approach, which explicitly accounts for the extreme, non-normal price spikes (jumps) common in electricity markets. The price is modeled as:

$$p_{t+1} = \bar{\theta}_t^\top \phi_t + \varepsilon_{t+1} + \mathbb{I}_t \varepsilon_{t+1}^J$$

Here, $\bar{\theta}_t^\top \phi_t$ can be represented by a time series model, ε_{t+1} is the normal random noise, and $\mathbb{I}_t \varepsilon_{t+1}^J$ is the jump component.

3.2.2 Hybrid Forecasting with the Empirical Distribution

To generate a forecast p_{t+1}, we use a hybrid method that transforms the data to a domain where it is easier to model. The calculation follows three distinct steps:

1. **Step 1: Data Transformation (Price to Normal Score)**
   First, we transform the current price p_t into a standardized, normally distributed value z_t.

   - *Action 1a: Find the quantile.* We build a cumulative distribution function, $F_P(p)$, from our entire historical price dataset. We use this function to find the quantile of our current price: $u_t = F_P(p_t)$. For example, if p_t is higher than 85% of historical prices, $u_t = 0.85$.
   - *Action 1b: Convert to a z-score.* We convert this quantile u_t into a standard normal value (a z-score) using the inverse of the standard normal CDF: $z_t = \Phi^{-1}(u_t)$.

   This transforms a volatile price series into a well-behaved, normally distributed series.

2. **Step 2: Time-Series Forecast (Predicting the Next Score)**
   Now, we forecast the next z-score, z_{t+1}, using the history of past z-scores. For this, Here we can use the equation given in the jump diffusion model.

3. **Step 3: Inverse Transformation (Score back to Price)**
   Finally, we transform the predicted z-score, z_{t+1}, back into a price forecast, $\hat{p}_{t+1}$, by reversing the process.

   - *Action 3a: Convert z-score back to quantile.* We use the standard normal CDF to find the predicted quantile: $u_{t+1} = \Phi(z_{t+1})$.
   - *Action 3b: Find the price.* We use the inverse of our historical empirical distribution to find the price that corresponds to this predicted quantile: $p_{t+1} = F_P^{-1}(u_{t+1})$.

3.3 Policies

There are different policies to try. As a baseline, a simple heurisitic is chosen.

3.3.1 A Heuristic Policy with Trend Confirmation

This policy extends the simple "buy-low, sell-high" strategy by adding a condition that requires a sustained price trend. When using this policy, our state model changes to:

$$S'_t = (R_t, p_t, p_{t-1}, p_{t-2}, p_{t-3}, p_{t-4}, p_{t-5})$$

The policy $X^{trend}(S_t|\theta)$ is defined as follows:

$$x_t = \begin{cases} x_{charge} & \text{if } p_t \leq \theta^{buy} \text{ and } p_{t-k} < 0 \text{ for all } k \in \{1, ..., 5\} \\ x_{discharge} & \text{if } p_t \geq \theta^{sell} \text{ and } p_{t-k} > 0 \text{ for all } k \in \{1, ..., 5\} \\ 0 & \text{otherwise} \end{cases}$$

where the parameters to be tuned are $\theta = (\theta^{buy}, \theta^{sell})$.

3.3.2 Parameter Tuning

The performance of the policy for a given parameter set θ is evaluated by simulating its behavior over one or more price sequences, known as sample paths (ω^n). The average performance over N sample paths is given by:

$$\bar{F}^{trend}(\theta) = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=0}^{T} C(S_t(\omega^n), X^{trend}(S'_t(\omega^n)|\theta))$$

We can solve this problem using a **grid search**, a straightforward method for parameter optimization. The process consists of the following steps:

1. **Define a search space:** We define a plausible range of values for both θ^{buy} and θ^{sell}. For example, we might search for the optimal buying threshold between -50€ and 50€, and the optimal selling threshold between 80€ and 200€.

2. **Discretize the space:** We divide each range into a discrete number of steps. For example, we could test every 5 increment within the defined ranges. This creates a finite "grid" of all possible $(\theta^{buy}, \theta^{sell})$ combinations.

3. **Simulate and evaluate:** For *each* combination of parameters on the grid, we run a full simulation of our policy against the historical price data. We calculate the total financial performance for that specific parameter set.

4. **Select the best parameters:** After evaluating all combinations, we simply choose the $(\theta^{buy}, \theta^{sell})$ pair that resulted in the highest overall performance. This pair represents our tuned, optimal heuristic policy.