Distributed Dynamic Programming (DDP): Forward Pass Decomposition for Copper Plate MPOPF

November 12, 2025

1 Problem Overview

Distributed Dynamic Programming (DDP) decomposes a multi-period optimal power flow (MPOPF) problem into T sequential subproblems using a forward pass approach. Each time step t solves a local optimization problem that couples with adjacent time steps through:

- **Backward coupling**: State trajectory from previous time step $t-1$ (current iteration)
- **Forward coupling**: Shadow prices (dual variables) from next time step $t+1$ (previous iteration)

The copper plate formulation ignores network constraints and focuses on energy arbitrage with battery storage.

2 Sets, Indices, and Parameters

2.1 Sets and Indices

- $\mathcal{T} = \{1, 2, \ldots, T\}$: Set of time periods
- $t \in \mathcal{T}$: Time period index
- k: DDP iteration index

2.2 System Parameters

$$P_{\text{BASE}} = 1000 \text{ kW (power base)} \tag{1}$$
$$E_{\text{BASE}} = 1000 \text{ kWh (energy base)} \tag{2}$$
$$\Delta t = 1.0 \text{ hour (time step duration)} \tag{3}$$

2.3 Load and Price Profiles

- $P_L[t]$: Load demand at time t [pu on P_{BASE}]
- $C[t]$: Energy price at time t [\$/kWh]
- C_B: Battery quadratic cost coefficient [\$/kW^2/h], typically $10^{-6} \times \min(C[t])$

2.4 Battery Parameters

$$
\begin{aligned}
E_{\text{Rated}} &: \text{Battery energy capacity [pu on } E_{\text{BASE}}] && (4)\\
P_{B,R} &: \text{Battery power rating [pu on } P_{\text{BASE}}] && (5)\\
\text{SOC}_{\min}, \text{SOC}_{\max} &: \text{SOC limits (fraction of } E_{\text{Rated}}) && (6)\\
B_0 &: \text{Initial SOC [pu on } E_{\text{BASE}}] && (7)\\
B_{T,\text{target}} &: \text{Terminal SOC constraint (optional) [pu on } E_{\text{BASE}}] && (8)
\end{aligned}
$$

SOC bounds:

$$\underline{B} = \text{SOC}_{\min} \cdot E_{\text{Rated}} \tag{9}$$

$$\overline{B} = \text{SOC}_{\max} \cdot E_{\text{Rated}} \tag{10}$$

3 Variable Color Coding

Following DDP forward pass decomposition:

- $\mathbf{B}[\mathbf{t}]$: Battery SOC at time t in **current iteration** k (blue)
- $\mathbf{B}[\mathbf{t}]$: Battery SOC at time t from **previous iteration** $k-1$ (red, used for coupling)
- $\mu[\mathbf{t}]$: Shadow price (dual variable) for battery dynamics at time t (green)
- $P_{\text{subs}}[t]$: Substation power at time t [pu]
- $P_B[t]$: Battery power at time t [pu] (positive = discharge, negative = charge)

4 DDP Forward Pass Subproblem

4.1 Subproblem for Time Step t (Iteration k)

Each time step $t \in \{1, \dots, T\}$ solves the following optimization problem:

$$
\begin{aligned}
\min_{\substack{P_{\text{subs}}[t], P_B[t], \mathbf{B}[\mathbf{t}],\\ P_B[t+1] \text{ (if } t<T)}} \quad & \underbrace{C[t] \cdot P_{\text{subs}}[t] \cdot \Delta t}_{\text{Energy cost at } t} + \underbrace{C_B \cdot (P_B[t])^2 \cdot \Delta t}_{\text{Battery cost at } t} \\
& + \underbrace{\mu^{\mathbf{k-1}}[\mathbf{t+1}] \cdot \left(\mathbf{B}^{\mathbf{k-1}}[\mathbf{t+1}] - \mathbf{B}[\mathbf{t}] + \Delta t \cdot P_B[t+1] \right)}_{\text{Soft coupling with next time step (if } t<T)}
\end{aligned} \tag{11}
$$

Subject to:
Power Balance:

$$P_{\text{subs}}[t] + P_B[t] = P_L[t] \tag{12}$$

Battery Dynamics:

$$\mathbf{B}[\mathbf{t}] - B_{\text{prev}}[t] + \Delta t \cdot P_B[t] = 0 \tag{13}$$

where:

$$B_{\text{prev}}[t] = \begin{cases} B_0 & \text{if } t = 1 \\ \mathbf{B^k[t-1]} & \text{if } t > 1 \quad \text{(from current forward pass)} \end{cases} \tag{14}$$

SOC Box Constraints (Negative Inequality Form):

$$\underline{B} - \mathbf{B[t]} \leq 0 \quad \text{(i.e., } \mathbf{B[t]} \geq \underline{B}\text{)} \tag{15}$$

$$\mathbf{B[t]} - \overline{B} \leq 0 \quad \text{(i.e., } \mathbf{B[t]} \leq \overline{B}\text{)} \tag{16}$$

Battery Power Limits:

$$-P_{B,R} \leq P_B[t] \leq P_{B,R} \tag{17}$$

$$-P_{B,R} \leq P_B[t+1] \leq P_{B,R} \quad \text{(if } t < T\text{)} \tag{18}$$

Terminal SOC Constraint (if $t = T$ and terminal constraint exists):

$$\mathbf{B[T]} = B_{T,\text{target}} \tag{19}$$

4.2 Key Formulation Insights

4.2.1 Coupling Structure

- **Backward (state) coupling**: Each subproblem uses $\mathbf{B[t-1]}$ from the *current* forward pass (iteration k), ensuring consistency within the same pass.
- **Forward (cost-to-go) coupling**: The coupling term with time $t+1$ uses:
  - $\mathbf{B^{k-1}[t+1]}$: State from *previous* iteration (parameter)
  - $\mu^{\mathbf{k-1}}[\mathbf{t+1}]$: Shadow price from *previous* iteration (parameter)
  - $P_B[t+1]$: Battery power at $t+1$ as a *decision variable*
- The coupling term $\mu[\mathbf{t+1}] \cdot (\mathbf{B[t+1]} - \mathbf{B[t]} + \Delta t \cdot P_B[t+1])$ penalizes violations of the battery dynamics constraint between t and $t+1$.
- By optimizing $P_B[t+1]$ as a decision variable, each subproblem can "soft-enforce" consistency with the next time step, even though $\mathbf{B[t+1]}$ is fixed from the previous iteration.

4.2.2 Shadow Price Interpretation

The dual variable $\mu[\mathbf{t}]$ from the battery dynamics constraint represents:

- Marginal value of increasing $\mathbf{B[t]}$ by one unit
- Shadow price / cost-to-go from state $\mathbf{B[t]}$ to terminal time T
- How much the objective would improve if we had one more unit of energy stored at time t

For terminal time step $t = T$ with terminal constraint $\mathbf{B[T]} = B_{T,\text{target}}$:

$$\mu_{\text{total}}[\mathbf{T}] = \mu_{\text{dynamics}}[\mathbf{T}] + \mu_{\text{terminal}}[\mathbf{T}] \tag{20}$$

where $\mu_{\text{terminal}}[\mathbf{T}]$ is the dual of the terminal constraint.

4.2.3 No Explicit Dual Update

Unlike ADMM-based methods, DDP does *not* require a separate dual update step. The shadow prices $\mu[\mathbf{t}]$ are automatically extracted from each subproblem's battery dynamics constraint dual:

$$\mu[\mathbf{t}] = -\lambda_{\text{battery\_dynamics}}[t] \tag{21}$$

where $\lambda_{\text{battery\_dynamics}}[t]$ is the JuMP dual variable (negated for proper economic sign convention).

5 DDP Algorithm

Algorithm 1 DDP Forward Pass for Copper Plate MPOPF

Input: Problem data $\mathcal{D} = \{T, C[t], P_L[t], B_0, E_{\text{Rated}}, P_{B,R}, \text{SOC}_{\min}, \text{SOC}_{\max}, \Delta t\}$
Parameters: max_iter, ϵ_{tol}, optional $\boldsymbol{\mu}^{\mathbf{0}}$ (initial dual variables)

Initialize $(k = 0)$:
  $\mathbf{B^0[t]} \leftarrow B_0$ for all $t \in \mathcal{T}$
  $\boldsymbol{\mu}^{\mathbf{0}}\mathbf{[t]} \leftarrow 0$ (or provided warm start) for all $t \in \mathcal{T}$
  $P_B^0[t] \leftarrow 0$, $P_{\text{subs}}^0[t] \leftarrow 0$ for all $t \in \mathcal{T}$

for $k = 1, 2, \ldots, \text{max\_iter}$ **do**
  // Store previous iteration values for coupling
  $\mathbf{B^{k-1}} \leftarrow \mathbf{B^{k-1}}$, $\boldsymbol{\mu}^{\mathbf{k-1}} \leftarrow \boldsymbol{\mu}^{\mathbf{k-1}}$, $P_B^{k-1} \leftarrow P_B^{k-1}$

  // Forward Pass: Solve each time step sequentially
  for $t = 1, 2, \ldots, T$ **do**
    Solve subproblem for time t using:
      $\mathbf{B^k[t-1]}$ from current forward pass (if $t > 1$)
      $\mathbf{B^{k-1}[t+1]}$, $\boldsymbol{\mu}^{\mathbf{k-1}}\mathbf{[t+1]}$ from previous iteration (if $t < T$)

    Extract solution:
      $\mathbf{B^k[t]} \leftarrow$ optimal SOC
      $P_B^k[t] \leftarrow$ optimal battery power
      $P_{\text{subs}}^k[t] \leftarrow$ optimal substation power
      $\boldsymbol{\mu}^{\mathbf{k}}\mathbf{[t]} \leftarrow -\text{dual}(\text{battery\_dynamics}[t])$
      If $t = T$ with terminal constraint:
        $\boldsymbol{\mu}^{\mathbf{k}}\mathbf{[T]} \leftarrow \boldsymbol{\mu}^{\mathbf{k}}\mathbf{[T]} + (-\text{dual}(\text{terminal\_soc}))$

    if $t < T$ **then**
      $P_B^k[t+1] \leftarrow$ optimal value of $P_B[t+1]$ variable
    end if
  end for

  // Compute convergence metrics
  $\text{err}_{\text{state}} \leftarrow \|\mathbf{B^k} - \mathbf{B^{k-1}}\|_2$
  $\text{err}_{\text{dual}} \leftarrow \|\boldsymbol{\mu}^{\mathbf{k}} - \boldsymbol{\mu}^{\mathbf{k-1}}\|_2$
  $\text{err}_{\text{KKT}} \leftarrow \max(\text{err}_{\text{state}}, \text{err}_{\text{dual}})$

  if $\text{err}_{\text{KKT}} < \epsilon_{\text{tol}}$ **then**
    break ▷ Converged
  end if
end for

Return: $\{\mathbf{B^k[t]}, P_B^k[t], P_{\text{subs}}^k[t], \boldsymbol{\mu}^{\mathbf{k}}\mathbf{[t]}\}_{t=1}^{T}$, convergence status

6 Convergence Criteria

6.1 State Trajectory Error

$$\mathrm{err}_{\mathrm{state}}^k = \left\| \mathbf{B^k} - \mathbf{B^{k-1}} \right\|_2 = \sqrt{\sum_{t=1}^{T} (\mathbf{B^k}[\mathbf{t}] - \mathbf{B^{k-1}}[\mathbf{t}])^2} \tag{22}$$

6.2 Dual Variable Error

$$\mathrm{err}_{\mathrm{dual}}^k = \left\| \mu^{\mathbf{k}} - \mu^{\mathbf{k-1}} \right\|_2 = \sqrt{\sum_{t=1}^{T} (\mu^{\mathbf{k}}[\mathbf{t}] - \mu^{\mathbf{k-1}}[\mathbf{t}])^2} \tag{23}$$

6.3 KKT Error (Combined)

$$\mathrm{err}_{\mathrm{KKT}}^k = \max\left(\mathrm{err}_{\mathrm{state}}^k, \mathrm{err}_{\mathrm{dual}}^k\right) \tag{24}$$

The algorithm converges when:

$$\mathrm{err}_{\mathrm{KKT}}^k < \epsilon_{\mathrm{tol}} \quad (\text{typically } \epsilon_{\mathrm{tol}} = 10^{-5}) \tag{25}$$

7 Objective Function

7.1 Total Objective (Physical Units)

$$J = \sum_{t=1}^{T} \left[C[t] \cdot (P_{\mathrm{subs}}[t] \cdot P_{\mathrm{BASE}}) \cdot \Delta t + C_B \cdot (P_B[t] \cdot P_{\mathrm{BASE}})^2 \cdot \Delta t \right] \tag{26}$$

7.2 Objective Components

$$J_{\mathrm{energy}} = \sum_{t=1}^{T} C[t] \cdot (P_{\mathrm{subs}}[t] \cdot P_{\mathrm{BASE}}) \cdot \Delta t \quad [\$] \tag{27}$$

$$J_{\mathrm{battery}} = \sum_{t=1}^{T} C_B \cdot (P_B[t] \cdot P_{\mathrm{BASE}})^2 \cdot \Delta t \quad [\$] \tag{28}$$

$$J = J_{\mathrm{energy}} + J_{\mathrm{battery}} \quad [\$] \tag{29}$$

7.3 Battery Quadratic Cost Rationale

The battery cost term $C_B \cdot (P_B[t])^2$ serves multiple purposes:

1. **Regularization**: Prevents numerical issues and excessive battery cycling
2. **Smoothing**: Encourages gradual power transitions rather than bang-bang control
3. **Conditioning**: Improves numerical stability of the optimization problem
4. **Physical realism**: Approximates wear-and-tear and efficiency losses

Typically set as: $C_B = 10^{-6} \times \min(C[t])$ to ensure it's a small regularization term.

8 Dual Variable Extraction and Sign Conventions

8.1 Battery Dynamics Dual

From the constraint:

$$\mathbf{B[t]} - B_{\text{prev}}[t] + \Delta t \cdot P_B[t] = 0 \tag{30}$$

JuMP returns the dual $\lambda_{\text{dynamics}}[t]$. The shadow price is:

$$\mu_{\text{dynamics}}[\mathbf{t}] = -\lambda_{\text{dynamics}}[t] \tag{31}$$

Economic interpretation: $\mu[\mathbf{t}] > 0$ means increasing $\mathbf{B[t]}$ would *decrease* the objective (energy stored is valuable). $\mu[\mathbf{t}] < 0$ means increasing $\mathbf{B[t]}$ would *increase* the objective (energy stored is costly).

8.2 SOC Box Constraint Duals

From constraints (negative inequality form):

$$g_{\text{Bmin}}[t] = \underline{B} - \mathbf{B[t]} \leq 0 \quad \text{(i.e., } \mathbf{B[t]} \geq \underline{B}) \tag{32}$$

$$g_{\text{Bmax}}[t] = \mathbf{B[t]} - \overline{B} \leq 0 \quad \text{(i.e., } \mathbf{B[t]} \leq \overline{B}) \tag{33}$$

JuMP dual variables:

$$\lambda_{\text{Bmin}}[t] = -\text{dual}(g_{\text{Bmin}}[t]) \quad \text{(negate to get positive when active)} \tag{34}$$

$$\lambda_{\text{Bmax}}[t] = \text{dual}(g_{\text{Bmax}}[t]) \quad \text{(already correct sign)} \tag{35}$$

Complementary slackness:

$$\lambda_{\text{Bmin}}[t] \cdot (\mathbf{B[t]} - \underline{B}) = 0, \quad \lambda_{\text{Bmin}}[t] \geq 0 \tag{36}$$

$$\lambda_{\text{Bmax}}[t] \cdot (\overline{B} - \mathbf{B[t]}) = 0, \quad \lambda_{\text{Bmax}}[t] \geq 0 \tag{37}$$

8.3 Terminal SOC Constraint Dual

If terminal constraint exists: $\mathbf{B[T]} = B_{T,\text{target}}$

The total shadow price at $t = T$ is:

$$\mu_{\text{total}}[\mathbf{T}] = \mu_{\text{dynamics}}[\mathbf{T}] + \mu_{\text{terminal}}[\mathbf{T}] \tag{38}$$

where:

$$\mu_{\text{terminal}}[\mathbf{T}] = -\text{dual}(\text{terminal\_soc}) \tag{39}$$

9 KKT Optimality Conditions

For each time step t, the KKT conditions are:

9.1 Stationarity

$$\frac{\partial \mathcal{L}}{\partial P_{\text{subs}}[t]} = C[t] \cdot \Delta t + \lambda_{\text{power\_balance}}[t] = 0 \tag{40}$$

$$\frac{\partial \mathcal{L}}{\partial P_B[t]} = 2C_B \cdot P_B[t] \cdot \Delta t + \lambda_{\text{power\_balance}}[t] + \Delta t \cdot \mu[\mathbf{t}] = 0 \tag{41}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{B}[\mathbf{t}]} = \mu[\mathbf{t}] - \mu[\mathbf{t+1}] + \lambda_{\text{Bmin}}[t] - \lambda_{\text{Bmax}}[t] = 0 \tag{42}$$

For terminal step $t = T$ with terminal constraint:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{B}[\mathbf{T}]} = \mu_{\text{dynamics}}[\mathbf{T}] + \mu_{\text{terminal}}[\mathbf{T}] + \lambda_{\text{Bmin}}[T] - \lambda_{\text{Bmax}}[T] = 0 \tag{43}$$

9.2 Primal Feasibility

All constraints satisfied:

$$P_{\text{subs}}[t] + P_B[t] - P_L[t] = 0 \tag{44}$$

$$\mathbf{B}[\mathbf{t}] - B_{\text{prev}}[t] + \Delta t \cdot P_B[t] = 0 \tag{45}$$

$$\underline{B} \leq \mathbf{B}[\mathbf{t}] \leq \overline{B} \tag{46}$$

$$-P_{B,R} \leq P_B[t] \leq P_{B,R} \tag{47}$$

9.3 Dual Feasibility

$$\lambda_{\text{Bmin}}[t] \geq 0 \tag{48}$$

$$\lambda_{\text{Bmax}}[t] \geq 0 \tag{49}$$

9.4 Complementary Slackness

$$\lambda_{\text{Bmin}}[t] \cdot (\mathbf{B}[\mathbf{t}] - \underline{B}) = 0 \tag{50}$$

$$\lambda_{\text{Bmax}}[t] \cdot (\overline{B} - \mathbf{B}[\mathbf{t}]) = 0 \tag{51}$$

10 Comparison with Brute Force Solution

10.1 Centralized (Brute Force) Formulation

The brute force approach solves the full multi-period problem in one shot:

$$\min_{\{P_{\text{subs}}[t], P_B[t], B[t]\}_{t=1}^{T}} \quad \sum_{t=1}^{T} \left[C[t] \cdot P_{\text{subs}}[t] \cdot \Delta t + C_B \cdot (P_B[t])^2 \cdot \Delta t \right] \tag{52}$$

Subject to:

$$P_{\text{subs}}[t] + P_B[t] = P_L[t], \quad \forall t \in \mathcal{T} \tag{53}$$

$$B[1] - B_0 + \Delta t \cdot P_B[1] = 0 \tag{54}$$

$$B[t] - B[t-1] + \Delta t \cdot P_B[t] = 0, \quad \forall t \in \{2, \ldots, T\} \tag{55}$$

$$\underline{B} \leq B[t] \leq \overline{B}, \quad \forall t \in \mathcal{T} \tag{56}$$

$$-P_{B,R} \leq P_B[t] \leq P_{B,R}, \quad \forall t \in \mathcal{T} \tag{57}$$

$$B[T] = B_{T,\text{target}} \quad \text{(if terminal constraint exists)} \tag{58}$$

10.2 Key Differences

- **Brute Force**: Solves all time steps simultaneously, guarantees global optimum
- **DDP**: Solves time steps sequentially with iterative coupling, may require multiple passes to converge
- **Scalability**: DDP can parallelize across time steps (not implemented in current version)
- **Memory**: DDP has smaller subproblems, lower memory footprint per solve
- **Warm starting**: DDP can initialize with duals from brute force for rapid convergence

11 Convergence Properties

11.1 Optimality Guarantee

Under convexity assumptions (which hold for the copper plate problem):

- If DDP converges (i.e., $\text{err}_{\text{KKT}} < \epsilon_{\text{tol}}$), the solution is **globally optimal**
- Convergence is guaranteed if the coupling penalty (implicitly through dual variables) is strong enough
- With warm start from brute force duals, DDP often converges in 1-2 iterations

11.2 Convergence Rate

- **Linear convergence**: $\text{err}^{k+1} \leq \alpha \cdot \text{err}^k$ for some $\alpha < 1$
- Convergence rate depends on problem conditioning and coupling strength
- Tighter SOC bounds $\Rightarrow$ slower convergence (more binding constraints)
- More time periods $T \Rightarrow$ longer horizon coupling, potentially slower convergence

11.3 Practical Observations

From numerical experiments with $T = 2, 4$ time periods:

- **Cold start** ($\mu = 0$): Converges in 5-15 iterations to $\epsilon_{\text{tol}} = 10^{-5}$
- **Warm start** (μ from BF): Converges in 1-2 iterations
- **Objective accuracy**: Matches brute force to within 10^{-6} relative error
- **Solution accuracy**: State and control trajectories match BF to numerical precision

12 Implementation Notes

12.1 Solver Choices

- **Gurobi**: Fast for quadratic programs (QP), commercial license required
- **Ipopt**: Open-source nonlinear solver, slightly slower but freely available
- Both solvers produce identical solutions for this convex problem

12.2 Numerical Considerations

- Use per-unit system to avoid ill-conditioning from large physical values
- Set C_B small enough to be a regularization, not a dominant cost
- Terminal SOC constraint should not be at SOC bounds to avoid numerical issues
- Monitor dual variable magnitudes to detect potential infeasibility

12.3 Parallelization Potential

Current implementation is *sequential* forward pass. Future extensions:

- Parallelize subproblems within each iteration (requires asynchronous updates)
- Use GPU acceleration for large-scale systems
- Implement backward pass for improved convergence

13 Example Problem Instance

13.1 System Configuration

$$
\begin{aligned}
T &= 4 \text{ time periods} && (59)\\
\Delta t &= 1.0 \text{ hour} && (60)\\
P_{\text{BASE}} &= 1000 \text{ kW} && (61)\\
E_{\text{BASE}} &= 1000 \text{ kWh} && (62)
\end{aligned}
$$

13.2 Battery Specifications

$$
\begin{aligned}
E_{\text{Rated}} &= 4000 \text{ kWh} = 4.0 \text{ pu} && (63)\\
P_{B,R} &= 800 \text{ kW} = 0.8 \text{ pu} && (64)\\
\text{SOC}_{\text{min}} &= 0.30 \quad (30\%) && (65)\\
\text{SOC}_{\text{max}} &= 0.95 \quad (95\%) && (66)\\
B_0 &= 0.30 \times 4.0 = 1.2 \text{ pu} \quad (30\% \text{ initial SOC}) && (67)\\
B_{T,\text{target}} &= 0.32 \times 4.0 = 1.28 \text{ pu} \quad (32\% \text{ terminal SOC}) && (68)
\end{aligned}
$$

13.3 Load Profile (Rated = 1250 kW)

$$P_L[t] = 1250 \times \text{LoadShape}[t]/1000 \quad [\text{pu}] \tag{69}$$

$$\text{LoadShape} = 0.8 + 0.2 \times \frac{\sin(\omega t - 0.8) + 1}{2}, \quad \omega = \frac{2\pi}{T} \tag{70}$$

Example for $T = 4$: $P_L \approx [1.00, 1.10, 1.15, 1.05]$ pu (approximate sinusoidal)

13.4 Price Profile (Charge-Discharge Scenario)

Design to test look-ahead capability:

$$C[t] = \begin{cases} 0.05 \text{ \$/kWh} & t \in \{1, 2\} \quad \text{(cheap: charge)} \\ 0.20 \text{ \$/kWh} & t \in \{3, 4\} \quad \text{(expensive: discharge)} \end{cases} \tag{71}$$

This creates a 4× price difference to incentivize:

- **Periods 1-2**: Charge battery (low prices)
- **Periods 3-4**: Discharge battery (high prices)

13.5 Battery Cost Coefficient

$$C_B = 10^{-6} \times \min(C[t]) = 10^{-6} \times 0.05 = 5 \times 10^{-8} \text{ \$/kW}^2\text{/h} \tag{72}$$

14 Appendix: Physical Interpretation

14.1 Battery Power Sign Convention

- $P_B[t] > 0$: Battery **discharging** (injecting power into grid)
- $P_B[t] < 0$: Battery **charging** (consuming power from grid)
- $P_B[t] = 0$: Battery idle

14.2 Economic Signals

- High $C[t] \Rightarrow$ Expensive energy $\Rightarrow$ Discharge battery ($P_B[t] > 0$) $\Rightarrow$ Reduce $P_{\text{subs}}[t]$
- Low $C[t] \Rightarrow$ Cheap energy $\Rightarrow$ Charge battery ($P_B[t] < 0$) $\Rightarrow$ Increase $P_{\text{subs}}[t]$
- $\mu[\mathbf{t}] > 0 \Rightarrow$ Energy stored at t is valuable (future high prices)
- $\mu[\mathbf{t}] < 0 \Rightarrow$ Energy stored at t is costly (future low prices)

14.3 SOC Trajectory Interpretation

- $\mathbf{B}[\mathbf{t}]$ decreases when $P_B[t] > 0$ (discharging)
- $\mathbf{B}[\mathbf{t}]$ increases when $P_B[t] < 0$ (charging)
- Optimal trajectory balances energy arbitrage profit against SOC constraints
- Terminal constraint $B[T] = B_{T,\text{target}}$ ensures battery returns to desired state