Experiments 3 - Market 2

In this problem, four energy producers participate in a day-ahead merit-order electricity market, including three conventional ones and a renewable one. These producers submit hourly bids that specify a delivery quantity of electricity and an offer price for each hour of the next day. The market operator uses these bids to determine the market-clearing price and dispatch schedule. If a bid is accepted, the producer is obligated to deliver the specified quantity of electricity at the clearing price. Otherwise, the producer earns zero profit for that hour.

However, renewable generation is deeply uncertain, as it depends on weather conditions that cannot be predicted. If the producer under-delivers relative to its bid (due to overestimation of production), or over-delivers (due to underestimation), penalties or imbalance costs would apply. These costs are imposed by the system operator to maintain grid stability and to ensure supply-demand balance in real time. Meanwhile, the clearing price is unknown to the producer in advance, so they need to compete in this market.

The renewable energy producer’s goal is to maximize its expected profit by strategically selecting the quantity of electricity to bid for each hour of the next day, accounting for the uncertainty in both renewable generation and clearing prices.

Market structure and timing:
1. The day-ahead market is cleared once per day (e.g., at noon) for 24 hourly intervals of the following day.
2. Each producer submits a bid (b_t, p_t) for each hour $t=1, ..., 24$, where:
2.1. b_t: the quantity of energy (in MWh) the producer is willing to supply.
2.2. p_t: the minimum acceptable price per MWh.
3. The market operator computes the market-clearing price c_t for each hour.
4. If $p_t \leq c_t$, the bid is accepted, and the producer is committed to delivering b_t MWh in hour t.
5. The total energy demand in the market is D_t, which follows a normal distribution with constant mean μ_D and variance ${\sigma_D}^2$.

Penalty structure:
1. If actual generation $P_t < b_t$, the shortfall $b_t - P_t$ incurs a penalty of q_u per MWh.
2. If actual generation $P_t \geq b_t$, the surplus $P_t - b_t$ incurs a curtailment penalty of q_o per MWh.

Conventional producers:
1. For each conventional producer i, its bid (b_{it}, p_{it}) for each hour t follows normal distributions with constant means and variances.

Please formalize this problem description as a model. Please also specify key components, such as the state variables, initial state, uncertainties and transition functions.

Excellent — this is a rich stochastic decision problem in the context of a **day-ahead electricity market with uncertain renewable generation and prices**. Let’s formalize it as a **stochastic optimization /**

decision-theoretic model that captures the renewable producer's bidding problem.

---

1. Decision Framework

We model the problem from the perspective of **one renewable energy producer** participating in a **day-ahead electricity market** with uncertainty in both:

- its own future renewable generation, and
- the market-clearing price (which depends on bids and uncertain demand).

We'll focus on a single day with hourly time steps $t = 1, 2, \dots, 24$.

---

2. State, Decisions, and Randomness

State variables

At each hour t:

- S_t: information state available to the renewable producer before submitting bid (b_t, p_t). This can include:
  - Forecast of renewable generation $\hat{P}_t$
  - Forecast of market price $\hat{c}_t$
  - Historical errors or weather conditions
  - Time index t

Formally, we can define:

$$S_t = (\hat{P}_t, \hat{c}_t, t)$$

representing the producer's belief state (forecasts).

Decision variables

At each stage t:

$$a_t = (b_t, p_t)$$

- b_t: bid quantity (MWh)
- p_t: bid price (minimum acceptable price per MWh)

Uncertain variables

After the decision a_t is made:

- P_t: realized renewable generation (MWh)
- c_t: realized market clearing price (USD/MWh)

- D_t: realized market demand (which affects c_t)

Uncertainty enters through the joint distribution of (P_t, c_t).

We assume:

$$P_t \sim f_P(\cdot \mid \hat{P}_t), \quad D_t \sim \mathcal{N}(\mu_D, \sigma_D^2), \quad c_t = g(D_t, \{(b_{it}, p_{it})\}_i, (b_t, p_t))$$

where $g(\cdot)$ is the market-clearing function determined by the supply-demand intersection.

3. Profit Function

For each hour t:

1. **If the bid is not accepted** ($p_t > c_t$):

$$\pi_t = 0$$

2. **If the bid is accepted** ($p_t \le c_t$):
   The producer is committed to deliver b_t MWh at the clearing price c_t.
   - If actual generation $P_t < b_t$: shortfall penalty $q_u(b_t - P_t)$
   - If actual generation $P_t > b_t$: curtailment penalty $q_o(P_t - b_t)$

   Expected profit:

$$\pi_t = c_t b_t - q_u(b_t - P_t)^+ - q_o(P_t - b_t)^+$$

   where $(x)^+ = \max(x, 0)$.

4. Objective Function

The renewable producer aims to **maximize expected total profit** over the next day:

$$\max_{\{b_t, p_t\}_{t=1}^{24}} \mathbb{E}\left[\sum_{t=1}^{24} \pi_t\right]$$

subject to:

- bid and price limits: $0 \le b_t \le b_{\max}, \quad p_{\min} \le p_t \le p_{\max}$
- possibly intertemporal constraints (e.g., ramp limits, if relevant).

5. System Dynamics and Transitions

The problem can be viewed as a **finite-horizon stochastic dynamic program**:

$$V_t(S_t) = \max_{a_t} \mathbb{E}\left[\pi_t(S_t, a_t, P_t, c_t) + V_{t+1}(S_{t+1}) \mid S_t, a_t\right]$$

with terminal condition $V_{25}(\cdot) = 0$.

The **state transition** can be represented as:

$$S_{t+1} = f(S_t, a_t, P_t, c_t)$$

Typically, forecasts for $t+1$ depend on new weather information or updated price expectations.

6. Initial State

At the day-ahead market clearing time (e.g., noon of the previous day):

$$S_1 = (\hat{P}_1, \hat{c}_1, 1)$$

where forecasts for all 24 hours are available.

7. Uncertainty Modeling

1. **Renewable generation uncertainty:**

   $$P_t = \hat{P}_t + \varepsilon_t^P, \quad \varepsilon_t^P \sim \mathcal{N}(0, \sigma_P^2)$$

   possibly correlated across hours.
2. **Demand and price uncertainty:**

   $$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2), \quad c_t = g(D_t, \text{all bids})$$

   so c_t is a random variable indirectly driven by D_t and competitors' bids.
3. **Competitors' bids:**
   Conventional producers' bids:

   $$(b_{it}, p_{it}) \sim \mathcal{N}((\mu_{b_i}, \mu_{p_i}), \Sigma_i)$$

8. Compact Mathematical Model

$$\begin{aligned}
\max_{\{b_t, p_t\}} \quad & \mathbb{E}\left[\sum_{t=1}^{24} \pi_t\right] \\
\text{s.t.} \quad & \pi_t = \begin{cases} 0, & p_t > c_t, \\ c_t b_t - q_u (b_t - P_t)^+ - q_o (P_t - b_t)^+, & p_t \le c_t, \end{cases} \\
& P_t \sim f_P(\cdot \mid \hat{P}_t), \\
& D_t \sim \mathcal{N}(\mu_D, \sigma_D^2), \\
& c_t = g(D_t, (b_t, p_t), \{(b_{it}, p_{it})\}_i), \\
& 0 \le b_t \le b_{\max}, \quad p_{\min} \le p_t \le p_{\max}.
\end{aligned}$$

9. Interpretation of Key Components

Component	Meaning
S_t	Information available to the renewable producer at hour t (forecasts, time index).
$a_t = (b_t, p_t)$	Decision: quantity and price bid into day-ahead market.
P_t, c_t	Random realizations: renewable generation and market clearing price.
Transition $S_{t+1} = f(S_t, a_t, P_t, c_t)$	Forecasts updated as time advances.
Reward π_t	Profit in hour t (depends on penalties).
Objective	Maximize expected total profit over 24 hours.

---

10. Extensions (Optional for completeness)

- **Risk aversion:** include a CVaR or variance penalty term:

$$\max \mathbb{E}\left[\sum_t \pi_t\right] - \lambda \operatorname{Var}\left(\sum_t \pi_t\right)$$

- **Dynamic bidding:** if bids can be updated based on new forecasts, the problem becomes a multistage stochastic control process.

✅ Summary

The renewable producer faces a **finite-horizon stochastic optimization problem** with:

Type	Symbol	Description
State	$S_t = (\hat{P}_t, \hat{c}_t, t)$	Forecasts & time
Decision	$a_t = (b_t, p_t)$	Bid quantity and price
Random variables	(P_t, c_t, D_t)	Realized generation, price, and demand
Reward	$\pi_t = c_t b_t - q_u (b_t - P_t)^+ - q_o (P_t - b_t)^+$	
Transition	$S_{t+1} = f(S_t, a_t, P_t, c_t)$	Forecast update
Objective	$\max \mathbb{E}[\sum_t \pi_t]$	

Would you like me to express this model as a **stochastic programming formulation** (e.g., scenario-based with probability trees) or as a **dynamic programming (Bellman) recursion** explicitly next?

There are two areas where you can improve the specification: (1) the renewable energy generation is deeply uncertain, so its probability distribution cannot be determined; (2) explicitly specify that for each conventional producer $i$, its bid $(b_{it}, p_{it})$ for each hour $t$ follows normal distributions with constant means and variances. Please refine your specification.

Excellent — you're absolutely right.
Let's refine the model accordingly to capture **deep uncertainty in renewable generation** and to **explicitly formalize the stochastic structure of the conventional producers' bids**.

The refined version below keeps the same overall structure (a stochastic decision model for the renewable producer), but adjusts these two aspects with care and precision.

Refined Formal Model

1. Problem Overview

We consider a **day-ahead merit-order electricity market** with four producers:

- One **renewable** producer (decision-maker),
- Three **conventional** producers (indexed by $i = 1, 2, 3$).

The renewable producer must choose hourly bids (b_t, p_t) for $t = 1, \ldots, 24$, aiming to maximize expected profit while facing **deep uncertainty** in renewable generation and **stochastic competition** from conventional producers.

---

2. Time Structure

- The market is cleared **once per day** for 24 hourly intervals.
- For each hour $t$, all producers submit their bids $(b_{it}, p_{it})$ (for $i = 1, 2, 3$) and the renewable producer submits (b_t, p_t).

The market operator then determines:

- Market clearing price c_t,
- Dispatch quantities based on merit order.

---

3. Model Components

(a) Decision Variables

For the renewable producer:

$$a_t = (b_t, p_t), \quad t = 1, \ldots, 24,$$

where:

- $b_t \geq 0$: bid quantity (MWh),

- $p_t \geq 0$: bid price (USD/MWh).

(b) State Variables

At each decision stage t:

$$S_t = (\hat{P}_t, \hat{c}_t, t),$$

where:

- $\hat{P}_t$: forecast of available renewable generation (MWh),
- $\hat{c}_t$: forecast of market-clearing price,
- t: current hour index.

This information is available when submitting bids.

(c) Uncertainties

There are **three major sources of uncertainty**:

(1) Renewable Generation (Deep Uncertainty)

The actual renewable generation P_t is **deeply uncertain** — meaning its probabilistic distribution is **unknown or ambiguous**.

We only know that:

$$P_t \in \mathcal{U}_t,$$

where $\mathcal{U}_t$ is a **bounded uncertainty set**, derived from forecast ranges or physical limits (e.g., based on wind speed, solar irradiance).

Example:

$$\mathcal{U}_t = [\underline{P}_t, \overline{P}_t],$$

representing the plausible range of realizations.

This introduces a **robust** or **distributionally ambiguous** component to the problem.

(2) Market Demand

The total system demand in hour t:

$$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2),$$

independent across hours (or weakly correlated).

(3) Conventional Producers' Bids

For each conventional producer $i = 1, 2, 3$, and each hour t:

$$b_{it} \sim \mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2), \quad p_{it} \sim \mathcal{N}(\mu_{p_i}, \sigma_{p_i}^2),$$

independent across hours and across producers.

Their bids define random supply offers in the market.

(d) Market Clearing Mechanism

The market-clearing price c_t is determined as the intersection of total supply and demand:

$$c_t = g(D_t, (b_t, p_t), \{(b_{it}, p_{it})\}_{i=1}^3),$$

where $g(\cdot)$ is the deterministic clearing rule (usually defined by sorting bids by price and matching to total demand).

Thus, c_t is a random variable influenced by:

- stochastic demand D_t,
- random competitor bids (b_{it}, p_{it}),
- and the renewable producer's bid (b_t, p_t).

---

4. Penalty and Profit Structure

(a) Delivery and Penalties

If the renewable producer's bid is **accepted** (i.e., $p_t \leq c_t$):

- It is committed to deliver b_t MWh.

If actual generation P_t differs from b_t:

- **Shortfall penalty:** if $P_t < b_t$, a penalty $q_u(b_t - P_t)$,
- **Surplus (curtailment) penalty:** if $P_t > b_t$, a penalty $q_o(P_t - b_t)$.

Thus, realized profit:

$$\pi_t = \begin{cases} c_t b_t - q_u(b_t - P_t)^+ - q_o(P_t - b_t)^+, & \text{if } p_t \leq c_t, \\ 0, & \text{if } p_t > c_t. \end{cases}$$

(b) Total Profit Objective

The renewable producer aims to **maximize the worst-case expected profit** over the 24-hour horizon, accounting for both stochastic and deeply uncertain variables:

$$\max_{\{b_t, p_t\}_{t=1}^{24}} \min_{P_t \in \mathcal{U}_t} \mathbb{E}_{D_t, (b_{it}, p_{it})} \left[\sum_{t=1}^{24} \pi_t \right]$$

This is a **distributionally robust stochastic optimization** problem:

- Expectation over known stochastic factors (demand, competitor bids),
- Inner minimization over the set of plausible renewable realizations $\mathcal{U}_t$.

---

5. System Dynamics and Transitions

The system evolves hourly. After realization of outcomes at time t:

$$S_{t+1} = f(S_t, a_t, P_t, c_t),$$

where forecasts for hour $t+1$ are updated based on realized weather, generation, and market information.

The dynamic programming recursion (conceptually):

$$V_t(S_t) = \max_{a_t} \min_{P_t \in \mathcal{U}_t} \mathbb{E}_{D_t,(b_{it},p_{it})} \left[\pi_t(S_t, a_t, P_t, c_t) + V_{t+1}(S_{t+1})\right]$$

with terminal condition $V_{25}(\cdot) = 0$.

---

6. Initial State

At the beginning of day-ahead scheduling (e.g., noon of the previous day):

$$S_1 = (\hat{P}_1, \hat{c}_1, 1),$$

where forecasts $\hat{P}_t$ and $\hat{c}_t$ for all t are available.

---

7. Compact Formulation Summary

$$\begin{aligned}
\max_{\{b_t,p_t\}} \quad & \min_{P_t \in \mathcal{U}_t} \mathbb{E}_{D_t,(b_{it},p_{it})} \left[\sum_{t=1}^{24} \pi_t\right] \\
\text{s.t.} \quad & \pi_t = \begin{cases} c_t b_t - q_u (b_t - P_t)^+ - q_o (P_t - b_t)^+, & p_t \le c_t, \\ 0, & p_t > c_t, \end{cases} \\
& c_t = g(D_t, (b_t, p_t), \{(b_{it}, p_{it})\}_{i=1}^{3}), \\
& D_t \sim \mathcal{N}(\mu_D, \sigma_D^2), \\
& b_{it} \sim \mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2),\ p_{it} \sim \mathcal{N}(\mu_{p_i}, \sigma_{p_i}^2), \quad i = 1, 2, 3, \\
& P_t \in \mathcal{U}_t, \\
& 0 \le b_t \le b_{\max}, \quad p_{\min} \le p_t \le p_{\max}.
\end{aligned}$$

---

8. Interpretation of Key Modeling Elements

Component	Meaning
S_t	Information state (forecast of renewable output and price).
$a_t = (b_t, p_t)$	Renewable producer's hourly bid decision.
P_t	Actual renewable generation (deeply uncertain, only known to lie in $\mathcal{U}_t$).
D_t	Market demand (normally distributed).
(b_{it}, p_{it})	Competitors' stochastic bids (normal distributions with constant means and variances).
c_t	Market clearing price (function of demand and all bids).
q_u, q_o	Penalty rates for under- and over-delivery.
Objective	Maximize expected total profit under worst-case renewable realizations.

---

9. Conceptual Type of Model

This model is a **hybrid robust–stochastic optimization** problem:

- **Stochastic:** due to random demand and conventional producers' bids;
- **Robust (deep uncertainty):** due to lack of known distribution for renewable output;
- **Dynamic:** decisions made hourly with updated information.

✅ In summary

The renewable producer's problem is a **distributionally robust, finite-horizon stochastic optimization model** with:

$$\boxed{\max_{b_t, p_t} \ \min_{P_t \in \mathcal{U}_t} \ \mathbb{E}_{D_t, (b_{it}, p_{it})} \left[\sum_t \pi_t \right]}$$

where:

- Conventional producers' bids follow fixed normal distributions,
- Renewable generation lies within an uncertainty set $\mathcal{U}_t$,
- Market price c_t is determined by the merit-order clearing rule.

Would you like me to make this **explicitly scenario-based** next (e.g., define discrete scenarios for demand and competitor bids, and uncertainty sets for generation)? That would make it ready for numerical implementation.

Various stakeholders approach this problem from different perspectives. Please refine the model specification from the perspectives of: (1) the renewable producer; and (2) the system regulator. These two perspectives should share a common environment, with each stakeholder independently controlling its own control variables and indirectly influencing each other through the shared environment. In the specification, specify the common environment, including global parameters and variables. For each perspective, also specify the corresponding model and its key components, such as control variables, transition functions, and objective functions. If there are multiple objectives, provide multiple objective functions instead of a weighted objective function.

Excellent — that's a sophisticated and important refinement.
We will now **reformulate the model as a multi-stakeholder dynamic system**, in which:

- There is a **shared environment** representing the day-ahead electricity market and the physical power system.
- Two key **stakeholders** act within this environment:
  1. The **renewable energy producer** (a strategic market participant making bids under uncertainty), and
  2. The **system regulator or operator** (who enforces reliability, balance, and efficiency rules, possibly regulating penalties, pricing mechanisms, or capacity limits).

They interact **indirectly** through the shared environment — each controlling its own decisions, but influencing market outcomes and each other's objectives.

Multi-Stakeholder Model Specification

---

1. Common Environment (Shared System)

This is the **market–physical environment** that both the renewable producer and the system regulator operate within.

Global parameters

Symbol	Description
$T = \{1, \ldots, 24\}$	Set of hourly time steps (decision epochs).
$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$	System demand in hour t.
$\{i = 1, 2, 3\}$	Conventional producers.
$b_{it} \sim \mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2)$	Conventional producer i's bid quantity.
$p_{it} \sim \mathcal{N}(\mu_{p_i}, \sigma_{p_i}^2)$	Conventional producer i's bid price.
$\mathcal{U}_t = [\underline{P}_t, \overline{P}_t]$	Uncertainty set for renewable output (deep uncertainty).
$g(\cdot)$	Market-clearing function (price–quantity matching rule).
q_u, q_o	Penalty parameters for under-delivery and curtailment.
$b_{\max}, p_{\min}, p_{\max}$	Feasible bounds on bids.

Global variables (state of the environment)

Variable	Description
c_t	Market-clearing price (USD/MWh).
$P_t \in \mathcal{U}_t$	Actual renewable generation (MWh).
S_t	Market information state at hour t: $S_t = (\hat{P}_t, \hat{c}_t, D_t, t)$.
π_t	Realized renewable producer profit at time t.
R_t	System reliability measure (e.g., supply-demand imbalance).

The environment evolves according to:

$$S_{t+1} = f_{\text{env}}(S_t, a_t^R, a_t^O, \xi_t)$$

where:

- a_t^R: renewable producer's control action,
- a_t^O: regulator's control action,
- ξ_t: exogenous random factors (e.g., demand and competitor bids).

---

2. Perspective 1: Renewable Energy Producer

Purpose

Maximize economic performance (profit) under market uncertainty and penalties for imbalances, given the regulatory structure imposed by the system operator.

Control variables

$$a_t^R = (b_t, p_t)$$

- b_t: bid quantity,
- p_t: bid price.

State variables

$$S_t^R = (\hat{P}_t, \hat{c}_t, t)$$

representing the renewable producer's information set (forecasts and time index).

Transition function

$$S_{t+1}^R = f^R(S_t^R, a_t^R, P_t, c_t)$$

The state is updated as new forecasts are issued and the previous hour's outcomes are observed.

Objective functions

The renewable producer typically pursues two related objectives:

1. **Profit maximization**

$$J_1^R = \mathbb{E}_{D_t,(b_{it},p_{it})}\left[\sum_{t=1}^{24} \pi_t\right]$$

where

$$\pi_t = \begin{cases} c_t b_t - q_u(b_t - P_t)^+ - q_o(P_t - b_t)^+, & p_t \le c_t, \\ 0, & p_t > c_t. \end{cases}$$

2. **Risk / exposure minimization (optional)**
   Because renewable generation is deeply uncertain, the producer might also aim to minimize worst-case or variability measures, e.g.

$$J_2^R = -\min_{P_t \in \mathcal{U}_t} \sum_{t=1}^{24} \pi_t, \quad \text{or} \quad J_2^R = \mathrm{Var}\left(\sum_t \pi_t\right)$$

   representing robustness or profit-variance control.

The renewable producer's **multi-objective problem** is then:

$$\begin{aligned} \max_{\{b_t,p_t\}} \quad & (J_1^R, -J_2^R) \\ \text{s.t.} \quad & P_t \in \mathcal{U}_t, \quad c_t = g(D_t, (b_t, p_t), \{(b_{it}, p_{it})\}), \\ & 0 \le b_t \le b_{\max}, \quad p_{\min} \le p_t \le p_{\max}. \end{aligned}$$

This captures the **renewable producer's internal problem** given the market and regulatory settings.

---

3. Perspective 2: System Regulator / Market Operator

Purpose

Ensure **system reliability, efficiency, and fairness** in the market clearing process, while maintaining economic signals that incentivize appropriate bidding behavior.

Control variables

$$a_t^O = (q_u, q_o, r_t, \theta_t)$$

where:

- q_u, q_o: imbalance penalty parameters (set before or updated periodically),
- r_t: reserve requirement or flexibility margin,
- θ_t: possibly adjustment parameters in the price-clearing rule $g(\cdot)$ (e.g., caps, floors, or market constraints).

State variables

$$S_t^O = (D_t, \{(b_{it}, p_{it})\}_{i=1}^3, (b_t, p_t), t)$$

representing full market information and forecasts available to the regulator.

Transition function

The system evolves based on market equilibrium and realized uncertainty:

$$S_{t+1}^O = f^O(S_t^O, a_t^O, a_t^R, \xi_t)$$

where ξ_t captures stochastic demand and generation changes.

Objective functions

The regulator has multiple policy objectives (explicitly separate, **not combined**):

1. **Reliability objective**
   Maintain supply-demand balance and adequate reserves:

   $$J_1^O = -\mathbb{E}\left[\sum_{t=1}^{24} R_t^2\right], \quad R_t = D_t - \left(P_t + \sum_i b_{it}\mathbf{1}_{\{p_{it} \le c_t\}}\right)$$

   (negative sign indicates minimization of imbalance).
2. **Economic efficiency objective**
   Maximize social welfare (consumer surplus – producer cost):

   $$J_2^O = \mathbb{E}\left[\sum_{t=1}^{24}\left(\int_0^{D_t} p(q)\,dq - c_t D_t\right)\right]$$

   or equivalently, minimize total system cost given bids and penalties.
3. **Sustainability / renewable integration objective**
   Encourage participation of renewable generation while maintaining grid stability:

   $$J_3^O = \mathbb{E}\left[\sum_{t=1}^{24} P_t - \lambda_{\text{imb}}|R_t|\right]$$

   balancing renewable utilization with imbalance tolerance.

Thus, the regulator's problem is:

$$\max_{a_t^O} \quad (J_1^O, J_2^O, J_3^O)$$

subject to system and market equilibrium constraints.

---

4. Coupling Through the Shared Environment

The two stakeholders **interact indirectly** through the shared market environment:

Coupling Mechanism	Description
Market clearing	$c_t = g(D_t, (b_t, p_t), \{(b_{it}, p_{it})\})$. The renewable's bid affects the clearing price and thus total supply, influencing regulator metrics R_t and J_2^O.

Coupling Mechanism	Description
Penalty parameters	The regulator's choice of q_u, q_o influences the renewable's bidding strategy and expected profit.
Physical generation	Realized P_t affects both renewable profit and system imbalance R_t.
Information flow	The regulator observes aggregate market outcomes, while the renewable observes forecasts and past results.

This shared environment can be viewed as a **two-agent stochastic game** (or **bi-level control system**):

- The renewable producer's optimization defines the **lower level** (strategic bidding under uncertainty),
- The regulator's optimization defines the **upper level** (system policy design).

---

5. Compact Representation

Layer	Agent	Control Variables	State Variables	Transition Function	Objectives
Environment	Shared	–	$S_t = (\hat{P}_t, \hat{c}_t, D_t, t)$	$S_{t+1} = f_{\text{env}}(S_t, a_t^R, a_t^O, \xi_t)$	–
Renewable Producer	$a_t^R = (b_t, p_t)$	$S_t^R = (\hat{P}_t, \hat{c}_t, t)$	$S_{t+1}^R = f^R(S_t^R, a_t^R, P_t, c_t)$	J_1^R, J_2^R	
System Regulator	$a_t^O = (q_u, q_o, r_t, \theta_t)$	$S_t^O = (D_t, \text{all bids}, t)$	$S_{t+1}^O = f^O(S_t^O, a_t^O, a_t^R, \xi_t)$	J_1^O, J_2^O, J_3^O	

---

6. Conceptual View

This structure defines a **shared-environment, multi-agent system**:

- The **environment** governs stochastic demand, price clearing, and renewable uncertainty.
- The **renewable producer** optimizes profit and robustness under deep uncertainty.
- The **regulator** optimizes reliability, efficiency, and sustainability.
- Both agents act simultaneously but influence each other only through the **shared market clearing and physical balance dynamics**.

✅ In summary

- **Common environment:** stochastic electricity market and physical system, with random demand, conventional bids, and deeply uncertain renewable generation.
- **Renewable producer's model:** dynamic, robust–stochastic optimization over bids (b_t, p_t), maximizing profit and minimizing risk.
- **Regulator's model:** multi-objective system control over penalty parameters and reserve policies $(q_u, q_o, r_t, \theta_t)$, ensuring reliability, efficiency, and sustainability.

- **Coupling:** both interact indirectly through prices c_t, physical imbalances R_t, and penalties, forming a two-level stochastic control/game system.

Would you like me to express this formally as a **bi-level optimization (Stackelberg) model**, where the regulator leads and the renewable producer responds optimally? That would make the interdependence mathematically explicit.

> Your specification should only include implementable and quantifiable components. For those conceptual components, please refine them or remove them.

Excellent — this is a crucial refinement.
We now move from a **conceptual, qualitative** system to a **quantifiable, implementable** model that is mathematically well-defined, suitable for computation, estimation, or simulation.

We will therefore keep only **operationally measurable** quantities and **implementable decision variables** for both the **renewable producer** and the **system regulator**, while maintaining their coupling through a **shared market environment**.

Refined Implementable Model

We describe the model in three layers:

1. **Common Environment** (shared parameters, variables, and stochastic processes)
2. **Renewable Producer's Model**
3. **System Regulator's Model**

Each component contains only **quantifiable** entities: random variables, parameters, decision variables, and functions that can be explicitly computed or simulated.

---

1. Common Environment

1.1 Time and Index Sets

$$t \in \{1, 2, \ldots, 24\}, \quad i \in \{1, 2, 3\}$$

1.2 Global Parameters

Symbol	Unit	Description
μ_D, σ_D	MWh, MWh	Mean and standard deviation of hourly demand.
μ_{b_i}, σ_{b_i}	MWh, MWh	Mean and std. dev. of bid quantity for conventional producer i.
μ_{p_i}, σ_{p_i}	USD/MWh	Mean and std. dev. of bid price for conventional producer i.

Symbol	Unit	Description
q_u, q_o	USD/MWh	Penalty rates for under-delivery and over-delivery (set by regulator).
$b_{\max}$	MWh	Upper bound on bid quantity for renewable producer.
$p_{\min}, p_{\max}$	USD/MWh	Lower and upper bounds on bid price.
$\underline{P}_t, \overline{P}_t$	MWh	Minimum and maximum feasible renewable generation (derived from forecasts or physical limits).

1.3 Random Variables

All random variables are observable or measurable ex-post.

Symbol	Distribution / Domain	Description
$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$	$\mathbb{R}^+$	System demand.
$b_{it} \sim \mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2)$	$\mathbb{R}^+$	Bid quantity of conventional producer i.
$p_{it} \sim \mathcal{N}(\mu_{p_i}, \sigma_{p_i}^2)$	$\mathbb{R}^+$	Bid price of conventional producer i.
$P_t \in [\underline{P}_t, \overline{P}_t]$	bounded	Realized renewable generation (deep uncertainty modeled by range, not distribution).

1.4 Market Clearing Function

The market-clearing price c_t is computed deterministically given demand and supply bids.
Define:

$$\mathcal{S}_t = \{(p_t, b_t)\} \cup \{(p_{it}, b_{it}) : i = 1, 2, 3\}$$

Sort all bids by price ascending and find the smallest c_t such that total accepted quantity meets demand:

$$\sum_{(p,b)\in\mathcal{S}_t : p \le c_t} b = D_t$$

This yields a unique **market clearing price** c_t and identifies which bids are accepted.

This function $c_t = g(D_t, \mathcal{S}_t)$ is **computable** from observed data.

---

2. Renewable Producer’s Model

2.1 Control Variables

$$a_t^R = (b_t, p_t)$$

- $b_t \in [0, b_{\max}]$ (MWh): bid quantity.
- $p_t \in [p_{\min}, p_{\max}]$ (USD/MWh): bid price.

2.2 Observed State Variables

$$S_t^R = (\hat{P}_t, \hat{c}_t)$$

where:

- $\hat{P}_t$: numerical generation forecast (MWh);
- $\hat{c}_t$: numerical price forecast (USD/MWh).

These are measurable and updated from available data at time of bidding.

2.3 Realized Quantities

After bids are submitted and the market clears:

Variable	Definition	Unit
c_t	clearing price from $g(\cdot)$	USD/MWh
P_t	realized renewable output	MWh
π_t	realized profit	USD

2.4 Profit Function

$$\pi_t = \begin{cases} c_t b_t - q_u (b_t - P_t)^+ - q_o (P_t - b_t)^+, & p_t \le c_t, \\ 0, & p_t > c_t. \end{cases}$$

where $(x)^+ = \max(x, 0)$.

This is directly computable from measurable quantities.

2.5 Objective Functions

Two quantifiable objectives:

1. **Expected Profit Maximization**

$$J_1^R = \mathbb{E}_{D_t,(b_{it},p_{it})}\left[\sum_{t=1}^{24} \pi_t\right]$$

   subject to $P_t \in [\underline{P}_t, \overline{P}_t]$.

2. **Worst-Case Profit Robustness**

$$J_2^R = \min_{P_t \in [\underline{P}_t, \overline{P}_t]} \sum_{t=1}^{24} \pi_t$$

   (quantifiable using deterministic bounds).

The renewable producer solves:

$$\max_{(b_t, p_t)_{t=1}^{24}} (J_1^R, J_2^R)$$

2.6 Transition Function

Hourly updates use observable forecast updates:

$$\hat{P}_{t+1} = \hat{P}_t + \Delta\hat{P}_t, \quad \hat{c}_{t+1} = \hat{c}_t + \Delta\hat{c}_t,$$

where $\Delta\hat{P}_t, \Delta\hat{c}_t$ are derived from available forecasting models.
These are measurable quantities, not conceptual placeholders.

---

3. System Regulator's Model

3.1 Control Variables

$$a_t^O = (q_u, q_o)$$

Penalty rates are the regulator's instruments. They are **numerical parameters** directly implementable in the market settlement rules.

(If desired, these can be set once per day or adjusted hourly within allowed ranges.)

3.2 Observed State Variables

$$S_t^O = (D_t, c_t, b_t, P_t)$$

These are measurable from market clearing and physical metering.

3.3 Quantifiable System Performance Metrics

1. **System imbalance (reliability measure):**

$$R_t = D_t - \left(P_t + \sum_{i=1}^{3} b_{it}\mathbf{1}_{\{p_{it} \le c_t\}} + b_t\mathbf{1}_{\{p_t \le c_t\}}\right)$$

   (MWh difference between demand and actual delivered supply).

2. **Penalty payments collected:**

$$\Psi_t = q_u(b_t - P_t)^+ + q_o(P_t - b_t)^+$$

   These are observable from settlements.

3.4 Objective Functions

The regulator pursues multiple quantifiable objectives (measurable from data):

1. **Reliability Objective (minimize imbalance):**

$$J_1^O = -\mathbb{E}\left[\sum_{t=1}^{24} R_t^2\right]$$

2. **Economic Efficiency Objective (minimize total penalties paid):**

$$J_2^O = -\mathbb{E}\left[\sum_{t=1}^{24} \Psi_t\right]$$

3. **Renewable Integration Objective (maximize renewable energy share actually delivered):**

$$J_3^O = \mathbb{E}\left[\sum_{t=1}^{24} P_t \mathbf{1}_{\{p_t \le c_t\}}\right]$$

All these metrics can be measured directly from market outcomes and metering data.

The regulator's implementable problem:

$$\max_{q_u, q_o} (J_1^O, J_2^O, J_3^O)$$

subject to the same market-clearing relationships.

3.5 Transition Function

The system state evolves deterministically given observed random draws:

$$S_{t+1}^O = (D_{t+1}, c_{t+1}, b_{t+1}, P_{t+1})$$

All components are observable and updated from the environment.

---

4. Coupling and Interaction

Both stakeholders interact through **explicit, quantifiable shared variables**:

Shared Variable	Controlled By	Appears In	Description
c_t	Environment (depends on all bids)	Renewable and Regulator	Market-clearing price.
q_u, q_o	Regulator	Renewable profit and penalties	Penalty rates.
P_t	Exogenous physical realization	Both	Realized renewable generation.
R_t	Derived	Regulator objective	System imbalance measure.

Thus, both influence and respond to one another **through numeric variables** in the same stochastic environment.

---

5. Compact Implementable Structure

Layer	Stakeholder	Control Variables	State Variables	Observable Quantities	Objective Functions
Environment	Shared	–	$D_t, (b_{it}, p_{it}), P_t$	$c_t = g(\cdot)$	–
Renewable Producer	$a_t^R = (b_t, p_t)$	$(\hat{P}_t, \hat{c}_t)$	π_t, P_t, c_t	J_1^R, J_2^R	
System Regulator	$a_t^O = (q_u, q_o)$	(D_t, c_t, b_t, P_t)	R_t, Ψ_t	J_1^O, J_2^O, J_3^O	

All quantities in this table are **numerically defined and measurable** in a real or simulated system.

✅ In summary

- The model now contains only **quantifiable, implementable** components:
  - Random variables with explicit distributions or bounds.
  - Decision variables with finite, interpretable ranges.
  - Observable performance metrics (profit, imbalance, penalties).
- The renewable producer optimizes **bidding decisions** b_t, p_t.
- The regulator optimizes **penalty parameters** q_u, q_o.
- Their interaction is mediated through the **market-clearing price**, **penalty terms**, and **physical balance** equations — all computable.

Would you like me next to formalize this as a **numerical simulation framework or optimization structure** (e.g., a bilevel or game-theoretic algorithm ready for implementation in code)?

> To ensure internal consistency across these models, please compose them into a single, unified model that shares a common environment and states, while embedding perspective-specific controls and objectives. The composed model should: 1. embed each perspective's control variables and objectives; 2. enable modular evaluation of each perspective; 3. for each perspective, treat other perspectives' controls as exogenous inputs; and 4. incorporate all other components from every perspective.

Excellent — this is the right next step.
We'll now build a **unified, internally consistent model** that:

1. **Integrates** both the *renewable producer* and *system regulator* perspectives;
2. **Shares** a common environment (state, parameters, and random variables);
3. **Embeds** each stakeholder's control variables and objectives;
4. **Allows modular evaluation**, i.e. each stakeholder can be simulated or optimized independently by treating the other's decisions as fixed inputs; and

5. **Includes all implementable components only** (everything measurable, computable, or simulatable).

Unified Market–Regulator–Renewable Model

We define a single system $\mathcal{M} = (\mathcal{E}, \mathcal{R}, \mathcal{O})$
where:

- $\mathcal{E}$: shared **environment** (state variables, parameters, random processes, transition rules);
- $\mathcal{R}$: **renewable producer** (controls a_t^R, objectives J^R);
- $\mathcal{O}$: **system regulator** (controls a_t^O, objectives J^O).

All parts are numerically defined.

---

1. Common Environment $\mathcal{E}$

(a) Time and indices

$$t = 1, 2, \dots, 24, \quad i = 1, 2, 3$$

(b) Global parameters (fixed, known to all)

Symbol	Units	Description
μ_D, σ_D	MWh	Mean and standard deviation of system demand.
μ_{b_i}, σ_{b_i}	MWh	Mean and std. dev. of bid quantity for conventional producer i.
μ_{p_i}, σ_{p_i}	USD/MWh	Mean and std. dev. of bid price for conventional producer i.
$b_{\max}$	MWh	Max bid quantity for renewable producer.
$p_{\min}, p_{\max}$	USD/MWh	Price bounds for bids.
$\underline{P}_t, \overline{P}_t$	MWh	Lower/upper bounds on renewable generation.

(c) Stochastic and uncertain inputs

Variable	Type	Description
$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$	stochastic	Hourly system demand.
$b_{it} \sim \mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2)$	stochastic	Quantity bid by conventional producer i.
$p_{it} \sim \mathcal{N}(\mu_{p_i}, \sigma_{p_i}^2)$	stochastic	Price bid by conventional producer i.
$P_t \in [\underline{P}_t, \overline{P}_t]$	bounded uncertain	Realized renewable generation (deep uncertainty).

(d) Shared state variables

At each hour t:

$$S_t = (D_t, \{(b_{it}, p_{it})\}_{i=1}^{3}, b_t, p_t, q_u, q_o, P_t, c_t)$$

where:

- b_t, p_t: renewable's bid,
- q_u, q_o: regulator's penalty rates,
- c_t: market-clearing price (computed below).

All elements of S_t are **observable or computable** at or after time t.

(e) Market clearing rule (shared function)

$$c_t = g(D_t, (b_t, p_t), \{(b_{it}, p_{it})\}_{i=1}^{3})$$

implemented as the lowest price c_t satisfying:

$$\sum_{(p,b)\in\mathcal{S}_t : p \le c_t} b = D_t, \quad \mathcal{S}_t = \{(p_t, b_t)\} \cup \{(p_{it}, b_{it})\}_{i=1}^{3}$$

This determines accepted bids for all producers.

(f) Energy delivery and imbalance measures

After market clearing and physical realization:

1. **Accepted quantities:**

$$b_t^{\text{acc}} = b_t \mathbf{1}_{\{p_t \le c_t\}}, \quad b_{it}^{\text{acc}} = b_{it} \mathbf{1}_{\{p_{it} \le c_t\}}$$

2. **System imbalance:**

$$R_t = D_t - \left(P_t + \sum_{i=1}^{3} b_{it}^{\text{acc}} + b_t^{\text{acc}} \right)$$

3. **Penalty cost to renewable:**

$$\Psi_t = q_u (b_t - P_t)^+ + q_o (P_t - b_t)^+$$

All are **numerically computable** from the environment and controls.

(g) Transition of the environment

At each step:

$$S_{t+1} = f_{\text{env}}(S_t, a_t^R, a_t^O, \xi_t)$$

where:

- $a_t^R = (b_t, p_t)$,

- $a_t^O = (q_u, q_o)$,
- $\xi_t = (D_{t+1}, \{(b_{i,t+1}, p_{i,t+1})\}, P_{t+1})$.

The transition is **deterministic given exogenous draws ξ_t** and decisions.

---

2. Renewable Producer Submodel $\mathcal{R}$

(a) Control variables

$$a_t^R = (b_t, p_t), \quad b_t \in [0, b_{\max}],\ p_t \in [p_{\min}, p_{\max}]$$

(b) Local observable state

$$S_t^R = (\hat{P}_t, \hat{c}_t)$$

— the producer's forecasts for generation and price (numeric predictions).

(c) Payoff function (per hour)

$$\pi_t = \begin{cases} c_t b_t - q_u (b_t - P_t)^+ - q_o (P_t - b_t)^+, & p_t \le c_t, \\ 0, & p_t > c_t. \end{cases}$$

(d) Objective functions

1. **Expected Profit:**

$$J_1^R = \mathbb{E}\left[\sum_{t=1}^{24} \pi_t\right]$$

2. **Worst-case (robust) Profit:**

$$J_2^R = \min_{P_t \in [\underline{P}_t, \overline{P}_t]} \sum_{t=1}^{24} \pi_t$$

The renewable's decision problem:

$$\max_{(b_t, p_t)_{t=1}^{24}} (J_1^R, J_2^R)$$

given q_u, q_o as **exogenous inputs** from regulator.

---

3. System Regulator Submodel $\mathcal{O}$

(a) Control variables

$$a_t^O = (q_u, q_o), \quad q_u, q_o \ge 0$$

(these may be static per day or updated hourly).

(b) Local observable state

$$S_t^O = (D_t, c_t, b_t, P_t)$$

(c) Performance metrics (computed from shared state)

1. **System imbalance (reliability):**

$$R_t = D_t - \left(P_t + \sum_{i=1}^{3} b_{it}^{\text{acc}} + b_t^{\text{acc}}\right)$$

2. **Penalty cost to renewable:**

$$\Psi_t = q_u(b_t - P_t)^+ + q_o(P_t - b_t)^+$$

3. **Delivered renewable energy:**

$$E_t^{\text{ren}} = P_t \mathbf{1}_{\{p_t \le c_t\}}$$

(d) Objective functions

All are quantifiable, using measurable quantities:

1. **Reliability objective (minimize imbalance):**

$$J_1^O = -\mathbb{E}\left[\sum_{t=1}^{24} R_t^2\right]$$

2. **Economic efficiency objective (minimize total penalties):**

$$J_2^O = -\mathbb{E}\left[\sum_{t=1}^{24} \Psi_t\right]$$

3. **Renewable integration objective (maximize accepted renewable energy):**

$$J_3^O = \mathbb{E}\left[\sum_{t=1}^{24} E_t^{\text{ren}}\right]$$

Regulator's problem:

$$\max_{q_u, q_o} (J_1^O, J_2^O, J_3^O)$$

treating b_t, p_t as **exogenous inputs** from the renewable producer.

---

4. Unified State–Control–Outcome Structure

Category	Symbol	Owner	Description	Domain / Range
State (shared)	S_t	Environment	$(D_t, b_{it}, p_{it}, b_t, p_t, q_u, q_o, P_t, c_t)$	numeric, measurable
Renewable control	$a_t^R = (b_t, p_t)$	Renewable	Bid quantity and price	bounded numeric
Regulator control	$a_t^O = (q_u, q_o)$	Regulator	Penalty parameters	nonnegative numeric
Outcome variables	$c_t, R_t, \pi_t, \Psi_t, E_t^{\text{ren}}$	Shared / derived	Market price, imbalance, profit, penalty, renewable share	measurable numeric

---

5. Unified Transition and Evaluation

System dynamics

For each hour t:

1. Environment draws stochastic inputs $(D_t, b_{it}, p_{it})_{i=1}^{3}$ and uncertain P_t;
2. Renewable chooses $a_t^R = (b_t, p_t)$;
3. Regulator parameters $a_t^O = (q_u, q_o)$ are applied;
4. Market clears: $c_t = g(\cdot)$;
5. Outcomes $R_t, \pi_t, \Psi_t, E_t^{\text{ren}}$ computed;
6. Environment transitions to S_{t+1}.

State transition equation

$$S_{t+1} = f_{\text{env}}(S_t, a_t^R, a_t^O, \xi_t)$$

where $\xi_t = (D_{t+1}, \{(b_{i,t+1}, p_{i,t+1})\}_{i=1}^{3}, P_{t+1})$.

This ensures both perspectives evolve within the **same quantitative environment**.

---

6. Modular Evaluation

- **Renewable producer module:**
  Evaluate (J_1^R, J_2^R) given exogenous (q_u, q_o).
- **Regulator module:**
  Evaluate (J_1^O, J_2^O, J_3^O) given exogenous (b_t, p_t).
- Both share the same environment $\mathcal{E}$, ensuring **consistency** and **comparability**.

---

7. Unified Model Summary

$$\boxed{\begin{array}{rl} & \text{Environment: } S_{t+1} = f_{\text{env}}(S_t, a_t^R, a_t^O, \xi_t) \\ & \text{Renewable: } a_t^R = (b_t, p_t),\ \max(J_1^R, J_2^R) \\ & \text{Regulator: } a_t^O = (q_u, q_o),\ \max(J_1^O, J_2^O, J_3^O) \\ \text{Shared Outcomes:} & \begin{cases} c_t = g(D_t, (b_t, p_t), \{(b_{it}, p_{it})\}), \\ R_t = D_t - (P_t + \sum_i b_{it}^{\text{acc}} + b_t^{\text{acc}}), \\ \pi_t, \Psi_t, E_t^{\text{ren}} \text{ as defined above.} \end{cases} \end{array}}$$

All components are **measurable, implementable, and internally consistent**.

✅ In summary

- The unified model defines a single, shared stochastic environment with explicit states and transitions.
- Each stakeholder (renewable producer and regulator) operates within this environment with its **own control variables** and **quantifiable objectives**.
- Each perspective treats the other's controls as exogenous, allowing **modular evaluation**.
- Every equation and variable is **computable or observable** — no conceptual placeholders remain.

Would you like me to now formalize this unified model in **bilevel or stochastic game form** (for example, as a Stackelberg or Nash structure) so you can analyze equilibrium behaviors quantitatively?
Here is one internally consistent, **quantitative composite model** that keeps a single environment while leaving each stakeholder's levers and goals explicit and separable.

---

1. Common environment $\mathcal{E}$

Time index: $t = 1, \ldots, 24.$
Conventional producers: $i = 1, 2, 3.$

Symbol	Units	Meaning
μ_D, σ_D	MWh	mean / st.dev. of hourly demand
μ_{b_i}, σ_{b_i}	MWh	mean / st.dev. of bid quantity of producer i
μ_{p_i}, σ_{p_i}	USD/MWh	mean / st.dev. of bid price of producer i
$b_{\max}$	MWh	upper bound on renewable bid quantity
$p_{\min}, p_{\max}$	USD/MWh	bounds on bid price
$\underline{P}_t, \overline{P}_t$	MWh	physical min / max renewable generation

Random inputs each hour

$$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2), \qquad b_{it} \sim \mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2), \qquad p_{it} \sim \mathcal{N}(\mu_{p_i}, \sigma_{p_i}^2),$$

and uncertain renewable output $P_t \in [\underline{P}_t, \overline{P}_t]$.

State vector

$$S_t = (D_t, \{(b_{it}, p_{it})\}_{i=1}^{3}, b_t, p_t, q_u, q_o, P_t, c_t).$$

Market-clearing rule

Given bids $\mathcal{S}_t = \{(b_t, p_t)\} \cup \{(b_{it}, p_{it})\}$,
$c_t = g(D_t, \mathcal{S}_t)$ is the smallest price satisfying
$\sum_{(p,b)\in\mathcal{S}_t : p \le c_t} b = D_t$.

Accepted quantities
$b_t^{\text{acc}} = b_t \mathbf{1}_{\{p_t \le c_t\}}, \; b_{it}^{\text{acc}} = b_{it} \mathbf{1}_{\{p_{it} \le c_t\}}$.

System imbalance

$$R_t = D_t - \Big(P_t + \sum_i b_{it}^{\text{acc}} + b_t^{\text{acc}}\Big),$$

and renewable penalty
$\Psi_t = q_u(b_t - P_t)^+ + q_o(P_t - b_t)^+$.

State transition
$S_{t+1} = f_{\text{env}}(S_t, a_t^R, a_t^O, \xi_t)$
with new random draws $\xi_t = (D_{t+1}, \{(b_{i,t+1}, p_{i,t+1})\}, P_{t+1})$.

---

2. Renewable-producer module $\mathcal{R}$

Control: $a_t^R = (b_t, p_t)$,
$b_t \in [0, b_{\max}], \; p_t \in [p_{\min}, p_{\max}]$.

Hourly profit

$$\pi_t = \begin{cases} c_t b_t - q_u(b_t - P_t)^+ - q_o(P_t - b_t)^+, & p_t \le c_t, \\ 0, & p_t > c_t. \end{cases}$$

Objectives

1. Expected profit
   $J_1^R = \mathbb{E}\Big[\sum_{t=1}^{24} \pi_t\Big]$.
2. Worst-case profit
   $J_2^R = \min_{P_t \in [\underline{P}_t, \overline{P}_t]} \sum_{t=1}^{24} \pi_t$.

Renewable problem (given q_u, q_o):

$$\max_{(b_t, p_t)_{t=1}^{24}} \; (J_1^R, J_2^R).$$

---

3. Regulator module $\mathcal{O}$

Control: $a_t^O = (q_u, q_o), \; q_u, q_o \ge 0$.

Derived measurable quantities

$$\Psi_t = q_u(b_t - P_t)^+ + q_o(P_t - b_t)^+, \qquad E_t^{\mathrm{ren}} = P_t \mathbf{1}_{\{p_t \le c_t\}}.$$

Objectives

1. Reliability $J_1^O = -\mathbb{E}[\sum_t R_t^2]$ (minimize imbalance).
2. Efficiency $J_2^O = -\mathbb{E}[\sum_t \Psi_t]$ (minimize penalties).
3. Renewable share $J_3^O = \mathbb{E}[\sum_t E_t^{\mathrm{ren}}]$ (maximize accepted renewable energy).

Regulator problem (given b_t, p_t):

$$\max_{q_u, q_o} (J_1^O, J_2^O, J_3^O).$$

---

4. Unified system summary

Category	Symbol / Function	Owner	Domain	Description
State	S_t	Shared	numeric	Market-physical state
Renewable controls	$a_t^R = (b_t, p_t)$	Producer	bounded	Bid quantity & price
Regulator controls	$a_t^O = (q_u, q_o)$	Regulator	≥0	Penalty rates
Market rule	$c_t = g(D_t, \mathcal{S}_t)$	Env.	deterministic	Clearing price
Imbalance	R_t	Shared	$\mathbb{R}$	Supply–demand gap
Profit / penalty	π_t, Ψ_t	Shared	USD	Producer profit & penalties

Transition: $S_{t+1} = f_{\mathrm{env}}(S_t, a_t^R, a_t^O, \xi_t)$.

---

5. Modular evaluation

- **Renewable module:** evaluate (J_1^R, J_2^R) with fixed (q_u, q_o).
- **Regulator module:** evaluate (J_1^O, J_2^O, J_3^O) with fixed (b_t, p_t).
  Both draw from the same stochastic environment $\mathcal{E}$; all quantities are numeric and measurable.

Result:

$$\boxed{\begin{aligned}&\text{Environment: } S_{t+1} = f_{\mathrm{env}}(S_t, a_t^R, a_t^O, \xi_t),\\ &\text{Renewable: } \max_{a^R}(J_1^R, J_2^R),\\ &\text{Regulator: } \max_{a^O}(J_1^O, J_2^O, J_3^O),\\ &\text{Shared outputs: } (c_t, R_t, \pi_t, \Psi_t, E_t^{\mathrm{ren}}).\end{aligned}}$$

This single structure maintains a **shared, fully quantifiable state and transition system**, embeds both stakeholders' controls and objectives, and permits **modular or joint analysis** (e.g., simulation, optimization, or equilibrium search).

Now, provide a modular Python implementation of the unified model, using classes to represent different perspectives. At the end of the program, define a function as the interface. This function should accept a dictionary of control variables and a dictionary of uncertain parameter values as input, simulate the model dynamics, and return the value of each objective function as output. This function does not need to be executed.

Here's a **modular Python implementation** of the unified renewable–regulator market model.

It uses object-oriented design with three main classes:

- **`MarketEnvironment`**: shared market and stochastic environment
- **`RenewableProducer`**: renewable participant's behavior and objectives
- **`SystemRegulator`**: regulator's behavior and objectives

At the end, a single **interface function `simulate_market()`** allows the user to pass in:

- a dictionary of **control variables** (for both perspectives), and
- a dictionary of **uncertain/stochastic parameters**,

and receive back **objective values** for each stakeholder.

All equations correspond exactly to the implementable unified model specified earlier.
This code is structured for analysis or simulation but **is not executed here**.

```python
import numpy as np
from dataclasses import dataclass, field
from typing import Dict, Tuple, List

# ============================================================
# 1. Shared Environment
# ============================================================

@dataclass
class MarketEnvironment:
    """Shared stochastic market environment."""
    T: int = 24
    mu_D: float = 1000.0
    sigma_D: float = 50.0
    mu_bi: List[float] = field(default_factory=lambda: [300.0, 300.0, 300.0])
    sigma_bi: List[float] = field(default_factory=lambda: [30.0, 30.0, 30.0])
    mu_pi: List[float] = field(default_factory=lambda: [50.0, 55.0, 60.0])
    sigma_pi: List[float] = field(default_factory=lambda: [5.0, 5.0, 5.0])
    P_bounds: List[Tuple[float, float]] = field(default_factory=lambda: [(0.0, 500.0)] * 24)

    def sample_demand(self) -> float:
        return np.random.normal(self.mu_D, self.sigma_D)

    def sample_conventional_bids(self) -> List[Tuple[float, float]]:
        bids = []
        for i in range(3):
            b_i = np.random.normal(self.mu_bi[i], self.sigma_bi[i])
            p_i = np.random.normal(self.mu_pi[i], self.sigma_pi[i])
            bids.append((b_i, p_i))
        return bids

    def sample_renewable_output(self, t: int) -> float:
        low, high = self.P_bounds[t - 1]
        return np.random.uniform(low, high)
    @staticmethod
    def market_clearing_price(D_t: float, bids: List[Tuple[float, float]]) -> float:
        """
        Simple deterministic market clearing:
        Sort bids by price ascending, accumulate quantities until demand is met.
        """
        sorted_bids = sorted(bids, key=lambda x: x[1])
        cum_quantity = 0.0
        for b, p in sorted_bids:
            cum_quantity += b
            if cum_quantity >= D_t:
                return p
        return sorted_bids[-1][1]


# ============================================================
# 2. Renewable Producer
# ============================================================

@dataclass
class RenewableProducer:
    """Renewable producer perspective."""
    b_max: float = 500.0
    p_min: float = 0.0
    p_max: float = 200.0

    def profit(self, b_t: float, p_t: float, c_t: float, P_t: float, q_u: float, q_o: float) ->
float:
        if p_t > c_t:
            return 0.0
        shortfall = max(b_t - P_t, 0.0)
        surplus = max(P_t - b_t, 0.0)
        return c_t * b_t - q_u * shortfall - q_o * surplus

    def evaluate_objectives(
        self,
        env: MarketEnvironment,
        b_seq: List[float],
        p_seq: List[float],
        q_u: float,
        q_o: float
    ) -> Dict[str, float]:
        """Compute expected and worst-case profit objectives."""
        T = env.T
        profits = []
        for t in range(1, T + 1):
            D_t = env.sample_demand()
            conv_bids = env.sample_conventional_bids()
            P_t = env.sample_renewable_output(t)
            bids = conv_bids + [(b_seq[t - 1], p_seq[t - 1])]
            c_t = env.market_clearing_price(D_t, bids)
            pi_t = self.profit(b_seq[t - 1], p_seq[t - 1], c_t, P_t, q_u, q_o)
            profits.append(pi_t)

        expected_profit = np.mean(profits)
        worst_case_profit = np.min(profits)
        return {
            "ExpectedProfit": expected_profit,
            "WorstCaseProfit": worst_case_profit
        }


# ============================================================
# 3. System Regulator
# ============================================================

@dataclass
class SystemRegulator:
    """System regulator perspective."""
    def compute_metrics(
        self,
        env: MarketEnvironment,
        b_seq: List[float],
        p_seq: List[float],
        q_u: float,
        q_o: float
    ) -> Dict[str, float]:
        T = env.T
        imbalance_sq = []
        penalty_sum = []
        accepted_renewable = []

        for t in range(1, T + 1):
            D_t = env.sample_demand()
            conv_bids = env.sample_conventional_bids()
            P_t = env.sample_renewable_output(t)
            bids = conv_bids + [(b_seq[t - 1], p_seq[t - 1])]
            c_t = env.market_clearing_price(D_t, bids)

            # accepted quantities
            b_acc = b_seq[t - 1] if p_seq[t - 1] <= c_t else 0.0
            b_conv_acc = sum(b for b, p in conv_bids if p <= c_t)
            R_t = D_t - (P_t + b_acc + b_conv_acc)
            Psi_t = q_u * max(b_seq[t - 1] - P_t, 0.0) + q_o * max(P_t - b_seq[t - 1], 0.0)
            E_t_ren = P_t if p_seq[t - 1] <= c_t else 0.0

            imbalance_sq.append(R_t ** 2)
            penalty_sum.append(Psi_t)
            accepted_renewable.append(E_t_ren)

        # Quantifiable objectives
        J1 = -np.mean(imbalance_sq)       # reliability (minimize imbalance)
        J2 = -np.mean(penalty_sum)        # efficiency (minimize penalties)
        J3 = np.mean(accepted_renewable)  # renewable integration
        return {"Reliability": J1, "Efficiency": J2, "RenewableShare": J3}


# ============================================================
# 4. Unified Interface Function
# ============================================================

def simulate_market(
    controls: Dict[str, any],
    uncertainties: Dict[str, any]
) -> Dict[str, Dict[str, float]]:
    """
    Unified simulation interface.
    Inputs:
        controls: dictionary with control sequences/parameters
            {
              "renewable_b": [...],   # list of 24 bid quantities
              "renewable_p": [...],   # list of 24 bid prices
              "q_u": float,           # under-delivery penalty
              "q_o": float            # over-delivery penalty
            }
        uncertainties: dictionary with environment parameter values
            {
              "mu_D", "sigma_D", "mu_bi", "sigma_bi",
              "mu_pi", "sigma_pi", "P_bounds"
            }
    Output:
        Dictionary of objective values:
            {
              "Renewable": {"ExpectedProfit": ..., "WorstCaseProfit": ...},
              "Regulator": {"Reliability": ..., "Efficiency": ..., "RenewableShare": ...}
            }
    """

    # --- Initialize environment and agents ---
    env = MarketEnvironment(
        mu_D=uncertainties.get("mu_D", 1000.0),
        sigma_D=uncertainties.get("sigma_D", 50.0),
        mu_bi=uncertainties.get("mu_bi", [300.0, 300.0, 300.0]),
        sigma_bi=uncertainties.get("sigma_bi", [30.0, 30.0, 30.0]),
        mu_pi=uncertainties.get("mu_pi", [50.0, 55.0, 60.0]),
        sigma_pi=uncertainties.get("sigma_pi", [5.0, 5.0, 5.0]),
        P_bounds=uncertainties.get("P_bounds", [(0.0, 500.0)] * 24)
    )
    renewable = RenewableProducer()
    regulator = SystemRegulator()

    b_seq = controls["renewable_b"]
    p_seq = controls["renewable_p"]
    q_u = controls["q_u"]
    q_o = controls["q_o"]

    # --- Evaluate objectives ---
    renewable_results = renewable.evaluate_objectives(env, b_seq, p_seq, q_u, q_o)
    regulator_results = regulator.compute_metrics(env, b_seq, p_seq, q_u, q_o)

    return {
        "Renewable": renewable_results,
        "Regulator": regulator_results
    }

# ============================================================
# End of unified model module
# ============================================================
```

✅ How it works

- **`MarketEnvironment`**
  Generates random demand and conventional bids, samples renewable output, and computes the market-clearing price.
- **`RenewableProducer`**
  Computes profits and expected / worst-case objectives given market outcomes and regulator's penalties.
- **`SystemRegulator`**
  Calculates measurable system metrics (imbalance, penalties, renewable share) and corresponding objectives.
- **`simulate_market()`**
  Serves as the **unified interface**:
  - Inputs: dictionaries of control variables and uncertain parameters,
  - Runs stochastic sampling for 24 hours,
  - Returns all objective values in structured form.

This modular structure enables:

- Independent optimization or evaluation of each perspective,
- Shared and consistent environment simulation,
- Easy extension to bilevel or equilibrium computation.

mu_D, sigma_D, mu_bi, sigma_bi, mu_pi, sigma_pi are constants instead of uncertain parameters, so the implementation should not take them as inputs. Please refine the implementation.

Got it ✅ — below is a **refined, modular Python implementation** of the unified model where:

- The structural parameters **`mu_D`**, **`sigma_D`**, **`mu_bi`**, **`sigma_bi`**, **`mu_pi`**, **`sigma_pi`** are **constants inside the model** (not inputs).
- The interface only accepts:
  - a dictionary of **control variables** (bids and penalty rates), and
  - a dictionary of **uncertain parameters** (such as renewable output bounds, if desired).

The model remains fully modular and consistent with the unified formal specification.

```python
import numpy as np
from dataclasses import dataclass, field
from typing import Dict, Tuple, List

# ============================================================
# 1. Shared Environment
# ============================================================

@dataclass
class MarketEnvironment:
    """Shared stochastic market environment."""
    T: int = 24

    # === Fixed structural constants ===
    mu_D: float = 1000.0       # mean demand
    sigma_D: float = 50.0      # std dev of demand
    mu_bi: List[float] = field(default_factory=lambda: [300.0, 300.0, 300.0])
    sigma_bi: List[float] = field(default_factory=lambda: [30.0, 30.0, 30.0])
    mu_pi: List[float] = field(default_factory=lambda: [50.0, 55.0, 60.0])
    sigma_pi: List[float] = field(default_factory=lambda: [5.0, 5.0, 5.0])

    # === Uncertain renewable generation bounds ===
    P_bounds: List[Tuple[float, float]] = field(default_factory=lambda: [(0.0, 500.0)] * 24)

    # ----------------------------------------------------
    # Stochastic sampling methods
    # ----------------------------------------------------
    def sample_demand(self) -> float:
        """Sample hourly electricity demand."""
        return np.random.normal(self.mu_D, self.sigma_D)

    def sample_conventional_bids(self) -> List[Tuple[float, float]]:
        """Sample bids (quantity, price) for conventional producers."""
        bids = []
        for i in range(3):
            b_i = np.random.normal(self.mu_bi[i], self.sigma_bi[i])
            p_i = np.random.normal(self.mu_pi[i], self.sigma_pi[i])
            bids.append((b_i, p_i))
        return bids

    def sample_renewable_output(self, t: int) -> float:
        """Sample renewable generation within bounds for hour t."""
        low, high = self.P_bounds[t - 1]
        return np.random.uniform(low, high)

    @staticmethod
    def market_clearing_price(D_t: float, bids: List[Tuple[float, float]]) -> float:
        """
        Determine clearing price: sort bids by price ascending,
        find smallest price such that cumulative quantity >= demand.
        """
        sorted_bids = sorted(bids, key=lambda x: x[1])
        cum_quantity = 0.0
        for b, p in sorted_bids:
            cum_quantity += b
            if cum_quantity >= D_t:
                return p
        return sorted_bids[-1][1]  # if not enough supply, last price


# ============================================================
# 2. Renewable Producer
# ============================================================

@dataclass
class RenewableProducer:
    """Renewable producer perspective."""
    b_max: float = 500.0
    p_min: float = 0.0
    p_max: float = 200.0

    def profit(self, b_t: float, p_t: float, c_t: float, P_t: float,
               q_u: float, q_o: float) -> float:
        """Compute realized profit for one hour."""
        if p_t > c_t:
            return 0.0
        shortfall = max(b_t - P_t, 0.0)
        surplus = max(P_t - b_t, 0.0)
        return c_t * b_t - q_u * shortfall - q_o * surplus

    def evaluate_objectives(
        self,
        env: MarketEnvironment,
        b_seq: List[float],
        p_seq: List[float],
        q_u: float,
        q_o: float
    ) -> Dict[str, float]:
        """Compute expected and worst-case profit objectives."""
        T = env.T
        profits = []
        for t in range(1, T + 1):
            D_t = env.sample_demand()
            conv_bids = env.sample_conventional_bids()
            P_t = env.sample_renewable_output(t)
            bids = conv_bids + [(b_seq[t - 1], p_seq[t - 1])]
            c_t = env.market_clearing_price(D_t, bids)
            pi_t = self.profit(b_seq[t - 1], p_seq[t - 1], c_t, P_t, q_u, q_o)
            profits.append(pi_t)

        expected_profit = float(np.mean(profits))
        worst_case_profit = float(np.min(profits))
        return {
            "ExpectedProfit": expected_profit,
            "WorstCaseProfit": worst_case_profit
        }


# ============================================================
# 3. System Regulator
# ============================================================

@dataclass
class SystemRegulator:
    """System regulator perspective."""

    def compute_metrics(
        self,
        env: MarketEnvironment,
        b_seq: List[float],
        p_seq: List[float],
        q_u: float,
        q_o: float
    ) -> Dict[str, float]:
        """Compute quantifiable system-level objectives."""
        T = env.T
        imbalance_sq = []
        penalty_sum = []
        accepted_renewable = []
        for t in range(1, T + 1):
            D_t = env.sample_demand()
            conv_bids = env.sample_conventional_bids()
            P_t = env.sample_renewable_output(t)
            bids = conv_bids + [(b_seq[t - 1], p_seq[t - 1])]
            c_t = env.market_clearing_price(D_t, bids)

            # accepted quantities
            b_acc = b_seq[t - 1] if p_seq[t - 1] <= c_t else 0.0
            b_conv_acc = sum(b for b, p in conv_bids if p <= c_t)
            R_t = D_t - (P_t + b_acc + b_conv_acc)
            Psi_t = q_u * max(b_seq[t - 1] - P_t, 0.0) + q_o * max(P_t - b_seq[t - 1], 0.0)
            E_t_ren = P_t if p_seq[t - 1] <= c_t else 0.0

            imbalance_sq.append(R_t ** 2)
            penalty_sum.append(Psi_t)
            accepted_renewable.append(E_t_ren)

        # Quantifiable objectives
        J1 = -float(np.mean(imbalance_sq))       # reliability (minimize imbalance)
        J2 = -float(np.mean(penalty_sum))        # efficiency (minimize penalties)
        J3 = float(np.mean(accepted_renewable))  # renewable integration
        return {"Reliability": J1, "Efficiency": J2, "RenewableShare": J3}


# ============================================================
# 4. Unified Interface Function
# ============================================================

def simulate_market(
    controls: Dict[str, any],
    uncertainties: Dict[str, any]
) -> Dict[str, Dict[str, float]]:
    """
    Unified simulation interface.

    Parameters
    ----------
    controls : dict
        {
          "renewable_b": [b1,...,b24],  # renewable bid quantities
          "renewable_p": [p1,...,p24],  # renewable bid prices
          "q_u": float,                 # under-delivery penalty
          "q_o": float                  # over-delivery penalty
        }
    uncertainties : dict
        {
          "P_bounds": [(low_t, high_t), ...]   # optional renewable output bounds
        }

    Returns
    -------
    dict
        {
          "Renewable": {"ExpectedProfit": ..., "WorstCaseProfit": ...},
          "Regulator": {"Reliability": ..., "Efficiency": ..., "RenewableShare": ...}
        }
    """

    # --- Initialize environment and agents ---
    P_bounds = uncertainties.get("P_bounds", [(0.0, 500.0)] * 24)
    env = MarketEnvironment(P_bounds=P_bounds)
    renewable = RenewableProducer()
    regulator = SystemRegulator()

    b_seq = controls["renewable_b"]
    p_seq = controls["renewable_p"]
    q_u = controls["q_u"]
    q_o = controls["q_o"]

    # --- Evaluate objectives ---
    renewable_results = renewable.evaluate_objectives(env, b_seq, p_seq, q_u, q_o)
    regulator_results = regulator.compute_metrics(env, b_seq, p_seq, q_u, q_o)
    return {
        "Renewable": renewable_results,
        "Regulator": regulator_results
    }

# ==============================================================
# End of unified model module
# ==============================================================
```

✅ Key refinements

- **`mu_D`**, **`sigma_D`**, **`mu_bi`**, **`sigma_bi`**, **`mu_pi`**, and **`sigma_pi`** are now **hard-coded constants** inside **`MarketEnvironment`**.
- The only uncertain parameter accepted by **`simulate_market()`** is now **`P_bounds`**, representing renewable generation ranges.
- The **`simulate_market()`** interface remains the main entry point for simulation and evaluation.

Example (not executed):

```
python

results = simulate_market(
    controls={
        "renewable_b": [200.0]*24,
        "renewable_p": [50.0]*24,
        "q_u": 40.0,
        "q_o": 10.0
    },
    uncertainties={
        "P_bounds": [(100.0, 400.0)]*24
    }
)
```

This returns a dictionary with all objective values for both the renewable producer and the regulator, consistent with the unified quantitative model.