

Market_narrative 4



In this problem, a renewable energy producer participates in a day-ahead merit-order electricity market competing against three other conventional producers. The producer submits hourly bids that specify a delivery quantity of electricity and an offer price for each hour of the next day. 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. Meanwhile, the hourly clearing price depends on the bids of all producers and is unknown in advance. The renewable energy producer's goal is to maximize its expected profit by strategically selecting the quantity of electricity b_t and offer price p_t to bid for each hour of the next day, accounting for the uncertainty in both renewable generation and clearing prices.

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

Excellent — you've provided a clear and realistic market setup. Let's formalize this as a **stochastic optimization model** for the renewable producer's *day-ahead bidding problem*.

1. Model Overview

A renewable producer participates in a **day-ahead electricity market** over a set of hours $t \in \mathcal{T} = \{1, 2, \dots, T\}$.

For each hour t , the producer decides:

- a **bid quantity** $b_t \geq 0$,
- an **offer price** $p_t \geq 0$.

These bids determine whether and how much energy will be sold depending on market clearing outcomes and realized renewable generation.

2. Uncertainties

Two sources of uncertainty exist:

1. Market clearing price uncertainty

- P_t^ω : random variable for the *market clearing price* in hour t , scenario $\omega \in \Omega$.
- Depends on bids of all producers (unknown at the time of bidding).

2. Renewable generation uncertainty

- G_t^ω : random variable for the *available renewable generation* (in MWh) in hour t , scenario ω .

- Determined by weather (e.g., wind, solar irradiance).

3. Decision Variables

Symbol	Description	Decision Stage
b_t	Quantity of energy bid (MWh) for hour t	Day-ahead (first-stage)
p_t	Offer price (\$/MWh) for hour t	Day-ahead (first-stage)
q_t^ω	Actual delivered energy (MWh) in scenario ω	Real-time (second-stage)
s_t^ω	Shortfall or deviation variable (if generation < commitment)	Real-time (second-stage)

4. Market Clearing Mechanism

- The producer's bid (b_t, p_t) is **accepted** if $p_t \leq P_t^\omega$.
- If accepted, the producer is **committed** to deliver b_t at the clearing price P_t^ω .
- If rejected ($p_t > P_t^\omega$), the producer earns zero profit in that hour.

To capture this in expectation, define the **acceptance indicator**:

$$\delta_t^\omega = \begin{cases} 1, & \text{if } p_t \leq P_t^\omega, \\ 0, & \text{otherwise.} \end{cases}$$

5. Real-Time Delivery and Penalties

If the bid is accepted ($\delta_t^\omega = 1$), the producer must deliver b_t using its available generation G_t^ω .

- If $G_t^\omega \geq b_t$: full delivery is possible (no penalty).
- If $G_t^\omega < b_t$: there is a shortfall $s_t^\omega = b_t - G_t^\omega$, and a **penalty** (or imbalance cost) applies, denoted by penalty rate π_t (\$/MWh shortfall).

Thus,

$$s_t^\omega = \max(0, b_t - G_t^\omega).$$

6. Profit Function

Expected profit in scenario ω :

$$\Pi^\omega(b, p) = \sum_{t \in \mathcal{T}} \delta_t^\omega (P_t^\omega b_t - \pi_t s_t^\omega).$$

Expected profit over all scenarios:

$$\mathbb{E}[\Pi(b, p)] = \sum_{\omega \in \Omega} \pi_{\omega} \sum_{t \in \mathcal{T}} \delta_t^{\omega} (P_t^{\omega} b_t - \pi_t s_t^{\omega}),$$

where π_{ω} is the probability of scenario ω .

7. Optimization Problem

$$\begin{aligned} \max_{b_t, p_t} \quad & \mathbb{E}_{\omega} \left[\sum_{t \in \mathcal{T}} \delta_t^{\omega} (P_t^{\omega} b_t - \pi_t \max(0, b_t - G_t^{\omega})) \right] \\ \text{s.t.} \quad & b_t \geq 0, \quad p_t \geq 0, \quad \forall t \in \mathcal{T}. \end{aligned}$$

8. Model Components Summary

Category	Symbol	Meaning
Decision variables	b_t, p_t	Bid quantity and price
Stochastic variables	$P_t^{\omega}, G_t^{\omega}$	Clearing price and generation
State variables	(Optional, if sequential formulation) $x_t = (G_t, P_t)$	Market and generation state
Parameters	π_t	Penalty rate for shortfall
	π_{ω}	Scenario probability
Transition function	$x_{t+1} = f(x_t, \xi_t)$	Evolution of uncertain states (e.g. Markov weather process)
Objective	Maximize expected profit	
Constraints	Bid non-negativity, delivery balance (through s_t^{ω})	

9. Extensions (Optional Enhancements)

- **Risk aversion:** Replace expectation with CVaR or utility function.
- **Dynamic coupling:** Introduce storage or inter-hour constraints (then x_t includes stored energy).
- **Strategic interaction:** Model clearing price P_t as a function of all bids (game-theoretic extension).

Compact Stochastic Programming Form

$$\begin{aligned} \max_{b, p} \quad & \mathbb{E}_{P, G} \left[\sum_t \mathbf{1}_{\{p_t \leq P_t\}} (P_t b_t - \pi_t \max(0, b_t - G_t)) \right] \\ \text{s.t.} \quad & b_t, p_t \geq 0. \end{aligned}$$

Would you like me to extend this into a **mathematical program with scenario discretization** (i.e., explicit constraints per scenario) or keep it in **expectation form** for theoretical formulation?



First, the day-ahead market is cleared once per day for 24 hourly intervals of the following day, so you can assume the problem only considers a 24-hour period. Second, penalties would apply if the renewable producer under-delivers or over-delivers relative to its bid. Please refine the specification.

Excellent clarification — that narrows the model to a single **24-hour day-ahead bidding problem** and adds symmetric imbalance penalties (for both under- and over-delivery).

Below is a refined, formalized **stochastic optimization model** incorporating these details.

1. Model Overview

A renewable energy producer participates in a **day-ahead electricity market** for the next day, consisting of

$$\mathcal{T} = \{1, 2, \dots, 24\}$$

hourly trading intervals.

For each hour t , the producer submits:

- a **bid quantity** b_t (MWh),
- an **offer price** p_t (\$/MWh).

The bid is accepted if it is *at or below* the market clearing price.

Once accepted, the producer must deliver the bid quantity b_t during real-time operation.

Actual renewable generation, however, is uncertain, potentially leading to **under-** or **over-delivery penalties**.

2. Model Components

Decision Variables (First-stage decisions)

Symbol	Description	Domain
b_t	Bid quantity for hour t	$b_t \geq 0$
p_t	Offer price for hour t	$p_t \geq 0$

These are chosen *before* uncertainty (in price and generation) is realized.

Stochastic Variables (Uncertainties)

Symbol	Description	Domain
P_t^ω	Market clearing price for hour t under scenario ω	\mathbb{R}_+
G_t^ω	Available renewable generation for hour t under scenario ω	\mathbb{R}_+

where $\omega \in \Omega$ indexes the set of possible scenarios with probability π_ω .

Market Clearing Indicator

$$\delta_t^\omega = \begin{cases} 1, & \text{if } p_t \leq P_t^\omega, \\ 0, & \text{otherwise.} \end{cases}$$

If $\delta_t^\omega = 1$, the bid is accepted and the producer must deliver b_t MWh.

Imbalance Quantities

After realization of generation G_t^ω :

$$\begin{aligned} u_t^\omega &= \max(0, b_t - G_t^\omega) && \text{(under-delivery),} \\ o_t^\omega &= \max(0, G_t^\omega - b_t) && \text{(over-delivery).} \end{aligned}$$

Penalty Parameters

Symbol	Description	Units
π_t^U	Penalty rate for under-delivery (shortfall)	\$/MWh
π_t^O	Penalty rate for over-delivery (excess generation)	\$/MWh

3. Profit in Each Scenario

For each hour t and scenario ω :

$$\Pi_t^\omega = \delta_t^\omega \left[P_t^\omega b_t - \pi_t^U u_t^\omega - \pi_t^O o_t^\omega \right].$$

If the bid is not accepted ($\delta_t^\omega = 0$), profit is zero.

4. Expected Profit Maximization

The producer chooses bids $\{b_t, p_t\}_{t=1}^{24}$ to maximize the expected profit:

$$\begin{aligned} \max_{b_t, p_t} \quad & \mathbb{E}_\omega \left[\sum_{t=1}^{24} \delta_t^\omega (P_t^\omega b_t - \pi_t^U \max(0, b_t - G_t^\omega) - \pi_t^O \max(0, G_t^\omega - b_t)) \right] \\ \text{s.t.} \quad & b_t \geq 0, \quad p_t \geq 0, \quad t = 1, \dots, 24. \end{aligned}$$

In discrete scenario form:

$$\max_{b_t, p_t} \sum_{\omega \in \Omega} \pi_{\omega} \sum_{t=1}^{24} \delta_t^{\omega} \left(P_t^{\omega} b_t - \pi_t^U \max(0, b_t - G_t^{\omega}) - \pi_t^O \max(0, G_t^{\omega} - b_t) \right).$$

5. State Representation (if sequential hourly coupling considered)

For this one-day problem (24 hours, independent markets), there is no inter-hour dependency unless additional dynamics (e.g., storage, ramping) are introduced.

However, a *state variable* formulation can be introduced if modeling day-to-day or inter-hour correlations:

$$x_t = (G_t, P_t),$$

with transitions governed by a stochastic process

$$x_{t+1} = f_t(x_t, \xi_t),$$

where ξ_t represents random weather and market factors.

6. Summary Table of Components

Category	Symbol(s)	Description
Decision Variables	b_t, p_t	Bid quantity and offer price
Stochastic Variables	$P_t^{\omega}, G_t^{\omega}$	Clearing price and available generation
State Variables	$x_t = (G_t, P_t)$	(optional) market-weather state
Parameters	π_t^U, π_t^O	Penalty rates for imbalance
Derived Variables	$\delta_t^{\omega}, u_t^{\omega}, o_t^{\omega}$	Acceptance and imbalances
Objective	Maximize expected profit	
Constraints	Nonnegativity; market clearing acceptance via δ_t^{ω}	

7. Interpretation

- The model captures *day-ahead strategic bidding* under **dual uncertainty** (prices and generation).
- The expected profit accounts for the possibility of:
 - **bid rejection** (no profit),
 - **under-delivery penalties** when generation falls short,
 - **over-delivery penalties** when generation exceeds commitment.
- The 24-hour structure implies **independent optimization per hour**, unless explicitly linked (e.g., correlated scenarios or storage constraints).

Would you like me to extend this formulation into a **mathematical program with explicit scenario-wise constraints and binary acceptance variables** (i.e., a mixed-integer stochastic program suitable for numerical solution)?



There are still several areas where the model specification can be improved: (1) the renewable energy generation is deeply uncertain, so its probability distribution cannot be determined; and (2) the market-clearing price is not random but endogenous, determined by merit-order of bids. Please refine your specification.

Excellent — those two refinements fundamentally change the structure of the problem. You are now describing a **strategic, robust (distributionally uncertain) stochastic optimization** with **endogenous market clearing**.

Let's carefully formalize this refined version.

Refined Model Specification

Setting

- A **day-ahead market** is cleared **once per day** for **24 hourly intervals**, $\mathcal{T} = \{1, \dots, 24\}$.
- There are N producers (indexed $i = 1, \dots, N$), among which producer $i = 0$ is **the renewable producer of interest**.
- Each producer i submits:
 - bid quantity $b_{i,t}$,
 - offer price $p_{i,t}$.

The **market clearing mechanism** sets a single clearing price P_t and dispatch levels $q_{i,t}$ by intersecting total supply with demand.

Unlike before, P_t is **not random** — it is **endogenously determined** from all submitted bids.

Meanwhile, the renewable producer's **available generation** is deeply uncertain — its probability law is not fully known.

Therefore, the producer adopts a **robust (or distributionally robust)** decision model.

1. Sets and Indices

Symbol	Description
$\mathcal{T} = \{1, \dots, 24\}$	Hours in the next day
$\mathcal{I} = \{0, 1, \dots, N\}$	Producers (0 = renewable)

2. Decision Variables

For the renewable producer $i = 0$:

Variable	Description
$b_{0,t}$	Bid quantity (MWh) for hour t
$p_{0,t}$	Offer price (\$/MWh) for hour t
$q_{0,t}$	Accepted quantity (MWh) determined endogenously by market clearing
$s_{0,t}^+, s_{0,t}^-$	Over- and under-delivery quantities realized ex post

3. Parameters

Symbol	Description
$b_{i,t}, p_{i,t}$ (for $i \neq 0$)	Competitors' submitted bids (assumed known or estimated by the renewable producer)
D_t	Total demand in hour t
π_t^U, π_t^O	Penalty rates for under- and over-delivery (\$/MWh)
\mathcal{U}_t	Uncertainty set for renewable generation $G_{0,t}$, e.g., an interval or convex set
$G_{0,t}$	Actual available renewable generation (unknown at bidding time)

4. Market Clearing (Endogenous Prices)

The **day-ahead market** clears bids by **merit order** (lowest offer prices first).

For each hour t , the clearing price P_t and dispatched quantities $q_{i,t}$ satisfy:

$$\begin{aligned}
 \sum_{i \in \mathcal{I}} q_{i,t} &= D_t, \quad (\text{supply-demand balance}) \\
 0 &\leq q_{i,t} \leq b_{i,t}, \quad \forall i \in \mathcal{I}, \\
 q_{i,t} &= \begin{cases} b_{i,t}, & p_{i,t} < P_t, \\ \text{partial}, & p_{i,t} = P_t, \\ 0, & p_{i,t} > P_t, \end{cases} \\
 P_t &= p_{i^*,t} \text{ of the marginal unit such that total supply meets } D_t.
 \end{aligned}$$

Hence, P_t depends *endogenously* on all bids, including $(b_{0,t}, p_{0,t})$.

The renewable producer must anticipate how its bid affects P_t and whether its offer is accepted.

5. Real-Time Delivery and Penalties

After market clearing, actual renewable generation $G_{0,t} \in \mathcal{U}_t$ is realized.

Delivery deviation:

$$\begin{aligned}
 s_{0,t}^+ &= \max(0, G_{0,t} - q_{0,t}) \quad (\text{over-delivery}), \\
 s_{0,t}^- &= \max(0, q_{0,t} - G_{0,t}) \quad (\text{under-delivery}).
 \end{aligned}$$

The renewable producer's realized profit in hour t is:

$$\Pi_{0,t}(b_{0,t}, p_{0,t}; G_{0,t}, P_t, q_{0,t}) = P_t q_{0,t} - \pi_t^U s_{0,t}^- - \pi_t^O s_{0,t}^+.$$

6. Deep Uncertainty Formulation

Because the probability distribution of renewable generation is unknown, the producer seeks a **robust** (or **distributionally robust**) bid that performs well across all plausible realizations $G_{0,t} \in \mathcal{U}_t$.

Robust Optimization Objective

$$\max_{b_{0,t}, p_{0,t}} \min_{G_{0,t} \in \mathcal{U}_t} \sum_{t=1}^{24} \Pi_{0,t}(b_{0,t}, p_{0,t}; G_{0,t}, P_t(b, p), q_{0,t}(b, p)).$$

That is, the renewable producer maximizes its **worst-case profit** over the uncertainty set of feasible generation levels.

If the producer has partial statistical information (e.g., mean and variance), one could replace the inner minimization with a **distributionally robust** expectation:

$$\max_{b_{0,t}, p_{0,t}} \min_{\mathbb{P} \in \mathcal{D}} \mathbb{E}_{\mathbb{P}} \left[\sum_t \Pi_{0,t}(\cdot) \right],$$

where \mathcal{D} is a family of plausible probability distributions.

7. State and Transition Representation

Although this is a single-day problem, we can express a dynamic representation if multi-day or sequential decision processes are modeled:

- **State variable:** $x_t = G_{0,t}$ (available generation or forecast state)
- **Decision variables:** $b_{0,t}, p_{0,t}$
- **Transition:** $x_{t+1} = f_t(x_t, \xi_t)$, where ξ_t are weather shocks.

For this single-day formulation, the uncertainty is static and scenario-independent per hour.

8. Model Summary Table

Category	Symbol(s)	Description
Decision Variables	$b_{0,t}, p_{0,t}$	Bid quantity and price (renewable)
Endogenous Variables	$q_{i,t}, P_t$	Dispatch and clearing price (market outcome)
Uncertain Variables	$G_{0,t} \in \mathcal{U}_t$	Renewable generation (deep uncertainty)
Parameters	$D_t, \pi_t^U, \pi_t^O, (b_{i,t}, p_{i,t})_{i \neq 0}$	Demand, penalties, competitors' bids

Category	Symbol(s)	Description
State Variables	(optional) $x_t = G_{0,t}$	Renewable forecast state
Transition Function	$x_{t+1} = f_t(x_t, \xi_t)$	Weather evolution
Objective	$\max_{b_0, p_0} \min_{G_0 \in \mathcal{U}} \sum_t \Pi_{0,t}$	Robust expected profit
Constraints	Market clearing, bid bounds, imbalance definitions	

9. Interpretation

- The renewable producer recognizes that:
 - Market prices P_t are **endogenous** and depend on its own bid.
 - Renewable generation $G_{0,t}$ is **deeply uncertain**, so a probabilistic expectation is unreliable.
- Thus, it optimizes against **worst-case generation outcomes** within credible bounds (robust optimization).
- The resulting equilibrium (if all producers behave strategically) would correspond to a **Nash equilibrium** in a **robust bidding game**.

Compact Formulation

$$\begin{aligned} \max_{b_{0,t}, p_{0,t}} \quad & \min_{G_{0,t} \in \mathcal{U}_t} \sum_{t=1}^{24} \left[P_t(b, p) q_{0,t}(b, p) - \pi_t^U \max(0, q_{0,t}(b, p) - G_{0,t}) - \pi_t^O \max(0, G_{0,t} - q_{0,t}(b, p)) \right] \\ \text{s.t.} \quad & (P_t, q_t) \text{ satisfy market clearing conditions for all } t. \end{aligned}$$

Would you like me to take this one step further — for example, by **formulating the market-clearing process as a complementarity (KKT) system** so that the whole model can be expressed as a **Mathematical Program with Equilibrium Constraints (MPEC)**, suitable for simulation or numerical analysis?



You can assume the total energy demand D_t follows a normal distribution with constant mean and variance. You can also assume that for each conventional producer i , its bid (b_i, p_i) for each hour t follows normal distributions with constant means and variances. Please refine the model specification.

Perfect — this refinement now gives us the **final, fully specified structural model**:

- We have (1) a **renewable producer** who strategically bids in a **day-ahead merit-order market**,
 (2) **endogenous clearing prices** determined by the intersection of random bids and random demand, and
 (3) **deep uncertainty in renewable generation**, modeled via robust optimization.

Below is the rigorous, formal specification consistent with those assumptions.

1. Model Context

A renewable producer (indexed $i = 0$) participates in a **day-ahead market** cleared once per day for **24 hourly intervals**,

$$\mathcal{T} = \{1, 2, \dots, 24\}.$$

Each producer $i \in \mathcal{I} = \{0, 1, \dots, N\}$ submits for each hour t :

- a **bid quantity** b_{it} (MWh),
- an **offer price** p_{it} (\$/MWh).

The market operator ranks bids in increasing order of p_{it} and determines the **market clearing price** P_t such that total accepted supply equals total demand D_t .

2. Stochastic Environment

2.1 Uncertain Demand

Total energy demand D_t is random but well-characterized:

$$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2), \quad \text{independent across } t.$$

Thus, demand uncertainty is represented probabilistically with **known distribution** parameters.

2.2 Competitors' Bids

For each **conventional producer** $i = 1, \dots, N$:

$$\begin{aligned} p_{it} &\sim \mathcal{N}(\mu_{p_i}, \sigma_{p_i}^2), \\ b_{it} &\sim \mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2), \end{aligned}$$

independent across t , and with correlations between p_{it} and b_{it} if desired (often independent for simplicity).

The renewable producer knows the distributions $(\mu_{p_i}, \sigma_{p_i}^2), (\mu_{b_i}, \sigma_{b_i}^2)$ but not the realized values when bidding.

2.3 Renewable Generation (Deeply Uncertain)

The renewable producer's available generation G_{0t} is **not probabilistically known**.

Instead, it belongs to an **uncertainty set**:

$$G_{0t} \in \mathcal{U}_t = [\underline{G}_t, \overline{G}_t],$$

where \underline{G}_t and \overline{G}_t are derived from forecasts or historical bounds.

This is a **robust (distribution-free)** representation of renewable uncertainty.

3. Decision Variables (Renewable Producer)

Symbol	Description	Timing
b_{0t}	Bid quantity (MWh)	Day-ahead (before uncertainty)
p_{0t}	Offer price (\$/MWh)	Day-ahead (before uncertainty)
q_{0t}	Accepted quantity after clearing	Determined by market clearing
s_{0t}^-, s_{0t}^+	Under- and over-delivery deviations	Realized ex post

4. Market Clearing Mechanism (Endogenous Price Formation)

For each hour t , the market clearing price P_t and accepted quantities q_{it} are determined as follows.

4.1 Merit-Order Clearing Conditions

1. Supply-Demand Balance

$$\sum_{i \in \mathcal{I}} q_{it} = D_t.$$

2. Individual Feasibility

$$0 \leq q_{it} \leq b_{it}, \quad \forall i \in \mathcal{I}.$$

3. Dispatch Order

$$q_{it} = \begin{cases} b_{it}, & \text{if } p_{it} < P_t, \\ \text{partial}, & \text{if } p_{it} = P_t, \\ 0, & \text{if } p_{it} > P_t. \end{cases}$$

4. Clearing Price

$$P_t = p_{i^*, t}, \quad \text{where } i^* \text{ is the marginal unit satisfying } \sum_{i \in \mathcal{I}: p_{it} \leq P_t} b_{it} = D_t.$$

Thus, P_t is a **function of all bids** and random variables $D_t, \{b_{it}, p_{it}\}_{i=1}^N$.

5. Real-Time Delivery and Penalties

After market clearing, the renewable producer's actual generation G_{0t} is realized.

Define deviations:

$$\begin{aligned} s_{0t}^- &= \max(0, q_{0t} - G_{0t}), & (\text{under-delivery}) \\ s_{0t}^+ &= \max(0, G_{0t} - q_{0t}), & (\text{over-delivery}). \end{aligned}$$

Penalties:

Under-delivery penalty: $\pi_t^U s_{0t}^-$, Over-delivery penalty: $\pi_t^O s_{0t}^+$.

6. Profit Function

Given the realized clearing price P_t , accepted quantity q_{0t} , and generation G_{0t} :

$$\Pi_{0t} = P_t q_{0t} - \pi_t^U s_{0t}^- - \pi_t^O s_{0t}^+.$$

Total daily profit:

$$\Pi_0 = \sum_{t=1}^{24} \Pi_{0t}.$$

7. Renewable Producer's Optimization Problem

The renewable producer anticipates the distribution of competitors' bids and demand, but not generation.

It thus solves a **two-level robust-stochastic program**:

$$\begin{aligned} \max_{b_{0t}, p_{0t}} \quad & \min_{G_{0t} \in \mathcal{U}_t} \mathbb{E}_{D_t, \{b_{it}, p_{it}\}_{i \neq 0}} \left[\sum_{t=1}^{24} (P_t(b, p, D, b_{-0}, p_{-0}) q_{0t}(b, p, D, b_{-0}, p_{-0}) - \pi_t^U s_{0t}^- - \pi_t^O s_{0t}^+) \right] \\ \text{s.t.} \quad & \text{Market clearing conditions (4.1) hold for all } t, \\ & b_{0t} \geq 0, \quad p_{0t} \geq 0, \quad \forall t. \end{aligned}$$

where:

- the **expectation** accounts for the known stochastic distributions of D_t and competitors' bids,
- the **inner minimization** enforces robustness to renewable generation uncertainty.

8. Model Components Summary

Category	Symbol(s)	Description
Decision variables	b_{0t}, p_{0t}	Renewable producer's bid quantity & price
Endogenous variables	P_t, q_{it}	Clearing price and accepted quantities (merit-order outcome)
Stochastic variables (known distributions)	$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2); b_{it}, p_{it} \sim \mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2), (\mu_{p_i}, \sigma_{p_i}^2)$	Demand and competitors' bids
Deep uncertainty	$G_{0t} \in [\underline{G}_t, \overline{G}_t]$	Renewable availability (unknown distribution)
Parameters	π_t^U, π_t^O	Imbalance penalties
Objective	$\max_{b_{0t}, p_{0t}} \min_{G_{0t} \in \mathcal{U}_t} \mathbb{E}[\Pi_0]$	Maximize robust expected profit

Category	Symbol(s)	Description
Constraints	Market clearing (merit-order), nonnegativity	Feasibility & system balance

9. Interpretation

- **Endogenous clearing:** P_t emerges from all producers' bids and random market conditions.
- **Stochastic competitors & demand:** Represented by normal distributions — the renewable producer forms expectations over them.
- **Deep uncertainty in generation:** No reliable distribution; handled via a robust "worst-case within bounds" formulation.
- **Objective:** A *hybrid robust-stochastic optimization* balancing expected market outcomes with guaranteed resilience to poor renewable realizations.

Compact Representation

$$\begin{aligned}
 & \max_{b_{0t}, p_{0t}} \min_{G_{0t} \in [\underline{G}_t, \overline{G}_t]} \mathbb{E}_{D_t, b_{-0,t}, p_{-0,t}} \left[\sum_{t=1}^{24} \left(P_t(b, p, D, b_{-0}, p_{-0}) q_{0t}(b, p, D, b_{-0}, p_{-0}) - \pi_t^U [q_{0t} - G_{0t}]^+ - \pi_t^O \right) \right] \\
 & \text{s.t. market clearing conditions (merit order) for each } t.
 \end{aligned}$$

Would you like me to **write this explicitly as an equilibrium (MPEC)** formulation next, showing how P_t and q_{it} are solved via complementarity (Karush–Kuhn–Tucker) conditions for the market clearing problem? That would make the model ready for numerical solution or simulation.



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 — this is now a **multi-agent decision environment** (or *multi-level stochastic-robust game*) with a shared market system.

We'll formalize it in a **structured systems modeling language**:

- A **common environment** defines shared stochastic and endogenous variables (the "market system").

- Each **stakeholder** (renewable producer and regulator) controls distinct decision variables and pursues independent objectives.
- They interact *indirectly* through shared variables and constraints (e.g., market clearing, penalties, emissions targets, etc.).

Below is a **rigorous system-level model specification**, structured by these components.

COMMON ENVIRONMENT (Shared Market System)

The common environment represents the *physical and economic system* where both stakeholders act. It includes **shared variables**, **global parameters**, and **stochastic processes** describing the 24-hour day-ahead electricity market.

A. Time and Agents

- Time horizon: $\mathcal{T} = \{1, \dots, 24\}$ (hours of next day)
- Producers: $\mathcal{I} = \{0, 1, \dots, N\}$, where
 $i = 0$: renewable producer (strategic decision-maker),
 $i = 1, \dots, N$: conventional producers.

B. Global Parameters

Symbol	Meaning	Type
$D_t \sim \mathcal{N}(\mu_D, \sigma_D^2)$	Stochastic hourly demand	Random variable
π_t^U, π_t^O	Penalty rates for under/over delivery	Constant
$\underline{G}_t, \overline{G}_t$	Bounds on renewable availability	Constants
$\mu_{b_i}, \sigma_{b_i}, \mu_{p_i}, \sigma_{p_i}$	Mean and variance of conventional bids	Constants
$C_i(q)$	Cost function for producer i	Function
$E_i(q)$	Emissions function for producer i	Function
$\mathcal{P}_t = \{(b_{it}, p_{it})\}_{i=1}^N$	Random conventional bids	Random variable set
M	Market operator enforcing merit-order dispatch	Mechanism (not agent)

C. Shared Variables (Endogenous Market Variables)

These variables are determined jointly by all stakeholders' decisions and system conditions.

Variable	Meaning	Determination
P_t	Market clearing price	Determined by merit-order mechanism
q_{it}	Accepted quantity for producer i	Result of market clearing
G_{0t}	Renewable generation realization	Exogenous, deeply uncertain
$S_t = \sum_i E_i(q_{it})$	System emissions	Function of dispatch

Variable	Meaning	Determination
D_t	System demand	Exogenous random variable

Key relationship (market clearing constraint):

$$\sum_{i \in \mathcal{I}} q_{it} = D_t, \quad \forall t.$$

The regulator and producer both indirectly influence P_t , q_{it} , and S_t through their own controls.

D. Transition Functions

- The **system state** at hour t is:

$$x_t = (D_t, G_{0t}, P_t, q_t, S_t).$$

- State transitions:

$$x_{t+1} = f_t(x_t, \xi_t),$$

where ξ_t captures new random draws for D_{t+1} and uncertain bids \mathcal{P}_{t+1} .

Renewable generation G_{0t} evolves in an uncertain but bounded manner:

$$G_{0,t+1} \in [\underline{G}_{t+1}, \overline{G}_{t+1}].$$

STAKEHOLDER (1): RENEWABLE PRODUCER

A. Role and Control

The renewable producer is a **strategic agent** submitting bids (b_{0t}, p_{0t}) into the day-ahead market. Its actions influence:

- its own dispatch q_{0t} ,
- market clearing price P_t ,
- system-wide balance (through the merit-order mechanism).

B. Control Variables

Variable	Description
b_{0t}	Bid quantity for hour t (MWh)
p_{0t}	Offer price for hour t (\$/MWh)

Control vector:

$$u_t^R = (b_{0t}, p_{0t}).$$

C. Local Uncertainty and State

Variable	Description
$G_{0t} \in [\underline{G}_t, \overline{G}_t]$	Renewable availability (deep uncertainty)
P_t, q_{0t}	Realized from market clearing given all bids
D_t, \mathcal{P}_t	Stochastic environment inputs

Local state (observable to producer):

$$x_t^R = (G_{0t}^{\text{forecast}}, D_t^{\text{forecast}}, \text{market history}).$$

D. Objective Functions

The renewable producer typically has **two competing objectives**:

1. Profit Maximization (Expected-robust hybrid objective)

$$J_1^R = \max_{u^R} \min_{G_{0t} \in [\underline{G}_t, \overline{G}_t]} \mathbb{E}_{D_t, \mathcal{P}_t} \left[\sum_{t=1}^{24} (P_t q_{0t} - \pi_t^U [q_{0t} - G_{0t}]^+ - \pi_t^O [G_{0t} - q_{0t}]^+) \right].$$

2. Reliability / Commitment Consistency Objective

(optional if the producer values accuracy in commitments)

$$J_2^R = - \sum_{t=1}^{24} \mathbb{E}[|q_{0t} - G_{0t}|],$$

minimizing deviations between committed and available generation.

E. Transition Function (Producer's Local Dynamics)

The producer's forecast state evolves via an uncertain (possibly Markovian) weather model:

$$G_{0,t+1}^{\text{forecast}} = f^R(G_{0t}^{\text{forecast}}, \eta_t),$$

where η_t represents stochastic meteorological inputs.

F. Constraints

$$\begin{aligned} b_{0t} &\geq 0, \quad p_{0t} \geq 0, \\ \sum_i q_{it} &= D_t, \quad (\text{global coupling}), \\ q_{0t} &= \text{dispatch result of } M(b, p, D). \end{aligned}$$

STAKEHOLDER (2): SYSTEM REGULATOR

The **system regulator** acts as a policy-maker or market operator aiming to ensure **system reliability**, **price stability**, and **environmental sustainability**, while not directly maximizing profit.

A. Control Variables

Variable	Description
π_t^U, π_t^O	Penalty rates for under-/over-delivery (imbalance pricing policy)
\bar{E}	Maximum allowed total emissions (system cap)
τ_t	Carbon tax or renewable subsidy (optional policy variable)

Control vector:

$$u_t^G = (\pi_t^U, \pi_t^O, \tau_t, \bar{E}).$$

B. Observed State

The regulator observes system-wide conditions:

$$x_t^G = (D_t, P_t, S_t, q_t, E_t),$$

where S_t = system emissions, P_t = clearing price, q_t = dispatch profile.

C. Objective Functions

The regulator typically has **multiple independent objectives** — these are not combined via weights (multi-objective policy design).

1. Market Efficiency Objective

Minimize total expected system cost:

$$J_1^G = \min_{u^G} \mathbb{E} \left[\sum_{t=1}^{24} \sum_{i \in \mathcal{I}} C_i(q_{it}) \right].$$

2. Emissions Objective

Minimize total expected system emissions subject to cap:

$$J_2^G = \min_{u^G} \mathbb{E} \left[\sum_{t=1}^{24} S_t \right], \quad \text{s.t.} \quad \sum_t S_t \leq \bar{E}.$$

3. Price Stability Objective

Minimize volatility of clearing prices:

$$J_3^G = \min_{u^G} \mathbb{E} \left[\sum_{t=1}^{24} (P_t - \mu_P)^2 \right],$$

where μ_P is the desired average price level.

4. Renewable Integration Objective

Encourage higher renewable participation:

$$J_4^G = \max_{u^G} \mathbb{E} \left[\sum_{t=1}^{24} q_{0t} \right].$$

D. Transition Function

The regulator's policy influences future states via economic and environmental feedbacks:

$$x_{t+1}^G = f^G(x_t^G, u_t^G, \zeta_t),$$

where ζ_t represents new demand and bid shocks.

E. Constraints

$$\sum_i q_{it} = D_t \quad (\text{market balance}),$$

$$S_t = \sum_i E_i(q_{it}),$$

P_t, q_t determined by merit-order dispatch mechanism M .

◆ SYSTEM COUPLING (Shared Environment Interaction)

Interaction Pathway	Description
Price coupling	P_t links the renewable producer's bidding strategy and the regulator's penalty/tax policy.
Quantity coupling	q_{0t} and S_t depend jointly on all producers' bids and demand realizations.
Penalty policy	Regulator's (π_t^U, π_t^O) directly affect the renewable producer's expected profit and bidding risk.
Emission policy	Regulator's \bar{E} and τ_t alter the relative competitiveness of renewable vs. conventional bids.

Thus, both agents share the **same stochastic and equilibrium market environment**, but each optimizes distinct objectives with separate control levers.

🧩 Summary Table

Aspect	Renewable Producer	System Regulator	Shared Environment
Decision variables	b_{0t}, p_{0t}	$\pi_t^U, \pi_t^O, \tau_t, \bar{E}$	—
Objective functions	J_1^R (profit), J_2^R (reliability)	J_1^G (efficiency), J_2^G (emissions), J_3^G (price stability), J_4^G (renewable integration)	—
State variables	$x_t^R = (G_{0t}, D_t^{\text{forecast}}, P_t, q_{0t})$	$x_t^G = (D_t, P_t, S_t)$	$x_t = (D_t, G_{0t}, P_t, q_t, S_t)$
Uncertainty	G_{0t} deep uncertainty	Demand and bids	Demand and bids

Aspect	Renewable Producer	System Regulator	Shared Environment
Transition functions	$f^R(G_{0t})$ (weather model)	$f^G(S_t, P_t)$ (system evolution)	$f(x_t, \xi_t)$
Coupling mechanisms	Affects market price P_t , dispatch q_{0t}	Sets penalties and policy affecting P_t, q_t	Merit-order clearing mechanism

✓ Interpretation:

- The **renewable producer** is a *strategic, profit-maximizing agent* operating under price uncertainty and deep renewable uncertainty.
- The **system regulator** is a *policy-optimizing agent* balancing efficiency, emissions, and market stability.
- Both interact through the **shared market environment** — a stochastic equilibrium system where demand, bids, prices, and emissions evolve jointly.

Would you like me to extend this into a **bi-level (Stackelberg) equilibrium model** — where the regulator acts as leader (setting policies first) and the renewable producer (and others) respond strategically? That would turn this into a **mathematical program with equilibrium constraints (MPEC)** suitable for simulation or policy analysis.



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

Excellent — we'll now **refine the model specification** to make it *implementable and quantitatively well-defined*, using only variables and parameters that can be measured, simulated, or directly estimated from data.

We'll retain the **two stakeholder perspectives** —

(1) the **renewable producer**, and

(2) the **system regulator** —

linked through a **common, quantifiable market environment**.

All conceptual placeholders (e.g. "market efficiency", "policy incentives", "history") will be replaced with measurable terms like costs, emissions, penalties, and expected values.

✓ COMMON ENVIRONMENT

1. Time and Agents

- Hours: $\mathcal{T} = \{1, 2, \dots, 24\}$
- Producers: $\mathcal{I} = \{0, 1, \dots, N\}$
 - $i = 0$: renewable producer

- $i = 1, \dots, N$: conventional producers

2. Global Parameters

Symbol	Description	Units / Domain
μ_D, σ_D	Mean and standard deviation of hourly demand	MW, MW
μ_{b_i}, σ_{b_i}	Mean, std. dev. of bid quantity for conventional producer i	MW, MW
μ_{p_i}, σ_{p_i}	Mean, std. dev. of offer price for conventional producer i	\$/MWh
$\underline{G}_t, \overline{G}_t$	Lower and upper bounds of renewable generation potential	MW
π_t^U, π_t^O	Penalty rates for under- and over-delivery	\$/MWh
α_i, β_i	Marginal cost parameters for conventional producer i : $C_i(q) = \alpha_i q + \frac{1}{2} \beta_i q^2$	\$/MWh, \$/MWh ²
γ_i	Emission factor for producer i	ton CO ₂ /MWh

All these quantities can be **observed**, **estimated**, or **statistically modeled** from historical data.

3. Stochastic Inputs

Random Variable	Distribution	Meaning
D_t	$\mathcal{N}(\mu_D, \sigma_D^2)$	Hourly system demand
b_{it}	$\mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2)$	Bid quantity of producer $i > 0$
p_{it}	$\mathcal{N}(\mu_{p_i}, \sigma_{p_i}^2)$	Offer price of producer $i > 0$

These distributions are empirically estimable.

4. Shared Variables (Endogenous)

Variable	Description	Determined by
P_t	Market clearing price	Merit-order clearing
q_{it}	Accepted dispatch for producer i	Market clearing
S_t	System emissions = $\sum_i \gamma_i q_{it}$	Output of market clearing
G_{0t}	Renewable generation realization	Scenario within $[\underline{G}_t, \overline{G}_t]$

5. Market Clearing (Quantifiable Rule)

For each hour t :

1. Order bids by price:

$(p_{1t}, b_{1t}), \dots, (p_{Nt}, b_{Nt}), (p_{0t}, b_{0t})$ sorted ascending in p .

2. Determine clearing price P_t :

P_t is the lowest price such that cumulative accepted bids equal demand:

$$\sum_{i:p_{it} \leq P_t} b_{it} \geq D_t.$$

3. Dispatch accepted quantities q_{it} :

- If $p_{it} < P_t$: $q_{it} = b_{it}$
- If $p_{it} = P_t$: q_{it} adjusted to meet demand exactly
- If $p_{it} > P_t$: $q_{it} = 0$

This defines P_t and q_{it} deterministically given all bids and demand.

(1) RENEWABLE PRODUCER MODEL

A. Decision Variables

Variable	Description
b_{0t}	Bid quantity submitted for hour t (MWh)
p_{0t}	Offer price submitted for hour t (\$/MWh)

Decision vector:

$$u_t^R = (b_{0t}, p_{0t})$$

B. Local Uncertainty and Inputs

Variable	Description
$G_{0t} \in [\underline{G}_t, \overline{G}_t]$	Realized renewable generation (bounded)
P_t, q_{0t}	Determined by market clearing given all bids and demand
D_t, b_{it}, p_{it}	Stochastic environmental inputs (with known distributions)

C. Realized Profit

$$\Pi_{0t}(G_{0t}, D_t, b, p) = P_t(b, p, D_t)q_{0t}(b, p, D_t) - \pi_t^U[q_{0t} - G_{0t}]^+ - \pi_t^O[G_{0t} - q_{0t}]^+.$$

D. Objectives (Quantifiable)

1. Expected Profit (robustified for generation bounds):

$$J_1^R = \max_{b_{0t}, p_{0t}} \min_{G_{0t} \in [\underline{G}_t, \overline{G}_t]} \mathbb{E}_{D_t, b_{it}, p_{it}} \left[\sum_{t=1}^{24} \Pi_{0t}(G_{0t}, D_t, b, p) \right].$$

2. Commitment Reliability:

$$J_2^R = \min_{b_{0t}, p_{0t}} \mathbb{E}_{D_t, b_{it}, p_{it}, G_{0t}} \left[\sum_{t=1}^{24} |q_{0t} - G_{0t}| \right].$$

Both J_1^R and J_2^R are fully measurable once data and simulation distributions are specified.

E. Constraints

$$b_{0t} \geq 0, \quad p_{0t} \geq 0, \quad \forall t.$$

Market clearing relations define q_{0t}, P_t .

(2) SYSTEM REGULATOR MODEL

The **regulator** sets operational policy parameters that shape the market environment but does not control bids directly.

A. Control Variables (Policy Levers)

Variable	Description
π_t^U	Penalty rate for under-delivery (\$/MWh)
π_t^O	Penalty rate for over-delivery (\$/MWh)
\bar{E}	Daily emissions cap (ton CO ₂)

These are observable, enforceable policy parameters.

Decision vector:

$$u_t^G = (\pi_t^U, \pi_t^O, \bar{E})$$

B. Observed Inputs and Shared Variables

Variable	Description
P_t, q_{it}	Market price and dispatch (from clearing)
$S_t = \sum_i \gamma_i q_{it}$	Emissions (computed)
D_t, b_{it}, p_{it}	Stochastic environment inputs

C. Objectives (All Quantifiable)

1. System Cost Minimization:

$$J_1^G = \min_{\pi_t^U, \pi_t^O, \bar{E}} \mathbb{E} \left[\sum_{t=1}^{24} \sum_{i=1}^N C_i(q_{it}) \right].$$

2. Emission Minimization:

$$J_2^G = \min_{\pi_t^U, \pi_t^O, \bar{E}} \mathbb{E} \left[\sum_{t=1}^{24} S_t \right], \quad \text{s.t.} \quad \sum_t S_t \leq \bar{E}.$$

3. Price Volatility Minimization:

$$J_3^G = \min_{\pi_t^U, \pi_t^O} \mathbb{E} \left[\sum_{t=1}^{24} (P_t - \mathbb{E}[P_t])^2 \right].$$

All objectives are data-based: costs, prices, and emissions are observable outcomes of the market simulation or historical records.

D. Constraints

$$\sum_i q_{it} = D_t, \quad S_t = \sum_i \gamma_i q_{it}, \quad \sum_t S_t \leq \bar{E}.$$

◆ SYSTEM INTERACTION SUMMARY

Component	Renewable Producer	Regulator	Shared Through
Controls	b_{0t}, p_{0t}	$\pi_t^U, \pi_t^O, \bar{E}$	Market clearing mechanism
Random Inputs	$D_t, b_{it}, p_{it}, G_{0t}$	D_t, b_{it}, p_{it}	Shared distributions
Outputs	q_{0t}, P_t, Π_{0t}	$S_t, P_t, C_i(q_{it})$	Market results
Feedback loop	Bids affect P_t, S_t	Penalty rates affect renewable profit	Shared clearing mechanism

⚙ IMPLEMENTATION STRUCTURE

Each stakeholder model can be **implemented numerically** as:

- A **simulation-optimization** model:
 - Sample D_t, b_{it}, p_{it} ,
 - Solve market clearing,
 - Evaluate profits, costs, and emissions,
 - Optimize respective controls.
- Or a **bi-level optimization** (regulator–producer) model where:
 - Upper level: Regulator chooses $(\pi_t^U, \pi_t^O, \bar{E})$,
 - Lower level: Producers (including the renewable) optimize their bids (b_{0t}, p_{0t}) .

All parameters, variables, and objectives are **numerically evaluable** given data on:

- demand and bid distributions,
- cost and emissions coefficients,
- penalty and cap levels.

✓ In summary:

- The **common environment** defines quantifiable market inputs (demand, costs, emissions, bids) and clearing logic.
- The **renewable producer** controls b_{0t}, p_{0t} to maximize profit and reliability.
- The **regulator** controls $\pi_t^U, \pi_t^O, \bar{E}$ to minimize cost, emissions, and price volatility.
- Both influence each other only via **observable market quantities**: price P_t , dispatch q_{it} , and emissions S_t .

This specification is now **fully implementable, data-driven**, and ready for quantitative simulation or optimization analysis.



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. enables 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 now the **final integration step**, producing a single **unified, internally consistent model** of the day-ahead electricity market system that embeds both perspectives — the **renewable producer** and the **system regulator** — within one **shared environment** and **state structure**.

We will construct the composed model so that:

1. Both agents' controls and objectives are included;
2. Each agent's optimization can be evaluated modularly;
3. When evaluating one agent, the other's controls are treated as fixed exogenous inputs;
4. All environment dynamics, stochastic processes, and measurable variables are common and shared.

Unified System Model

1. Global Setting and Notation

- **Time horizon:**
 $\mathcal{T} = \{1, 2, \dots, 24\}$ (24 hourly intervals)
- **Producers:**
 $\mathcal{I} = \{0, 1, \dots, N\}$, with $i = 0$ the renewable producer and $i > 0$ the conventional producers.
- **System regulator:**
Governs market-wide penalty rates and emission caps.

2. Common Environment

All agents share the same environment, consisting of global parameters, random variables, endogenous states, and transition dynamics.

(a) Global Parameters (known, measurable)

Symbol	Description	Units
μ_D, σ_D	Mean and standard deviation of demand	MW
μ_{b_i}, σ_{b_i}	Mean and std. dev. of bid quantity for conventional producer $i > 0$	MW
μ_{p_i}, σ_{p_i}	Mean and std. dev. of offer price for conventional producer $i > 0$	\$/MWh
$\underline{G}_t, \overline{G}_t$	Bounds on renewable generation	MW
α_i, β_i	Cost coefficients for $C_i(q) = \alpha_i q + \frac{1}{2} \beta_i q^2$	\$/MWh, \$/MWh ²
γ_i	Emission factor for producer i	ton CO ₂ /MWh

(b) Random Variables (stochastic environment)

Symbol	Distribution	Description
D_t	$\mathcal{N}(\mu_D, \sigma_D^2)$	System demand
b_{it}	$\mathcal{N}(\mu_{b_i}, \sigma_{b_i}^2), i > 0$	Conventional bid quantity
p_{it}	$\mathcal{N}(\mu_{p_i}, \sigma_{p_i}^2), i > 0$	Conventional offer price

(c) Endogenous Market Variables (shared state)

Variable	Description	Determined by
P_t	Market clearing price	Merit-order clearing
q_{it}	Accepted dispatch	Merit-order clearing
S_t	Total emissions $S_t = \sum_i \gamma_i q_{it}$	System outcome
G_{0t}	Renewable generation realization $\in [\underline{G}_t, \overline{G}_t]$	Exogenous bounded uncertainty

(d) State Vector

For each hour t ,

$$x_t = (D_t, G_{0t}, P_t, q_t, S_t)$$

encapsulates all shared environmental and market information.

(e) State Transition Function

Between hours:

$$x_{t+1} = f(x_t, \xi_t),$$

where ξ_t represents independent realizations of $(D_{t+1}, b_{i,t+1}, p_{i,t+1})$.

This enables hour-to-hour stochastic simulation.

3. Control Structure

(a) Renewable Producer Controls

Control	Description
b_{0t}	Bid quantity (MWh)
p_{0t}	Offer price (\$/MWh)

Control vector:

$$u_t^R = (b_{0t}, p_{0t})$$

(b) System Regulator Controls

Control	Description
π_t^U	Under-delivery penalty rate (\$/MWh)
π_t^O	Over-delivery penalty rate (\$/MWh)
\bar{E}	Daily emissions cap (ton CO ₂)

Control vector:

$$u_t^G = (\pi_t^U, \pi_t^O, \bar{E})$$

(c) Control Relationships

- u_t^R and u_t^G jointly affect market outcomes P_t, q_{it}, S_t .
- Each perspective treats the other's control vector as fixed when optimizing.

4. Market Clearing Mechanism

Given all bids and demand D_t :

1. **Merit-order sorting:** bids (p_{it}, b_{it}) are ranked ascending by price.
2. **Clearing price:** P_t is set where cumulative accepted supply equals demand:

$$\sum_{i:p_{it} \leq P_t} b_{it} \geq D_t.$$

3. Dispatch:

$$q_{it} = \begin{cases} b_{it}, & p_{it} < P_t, \\ D_t - \sum_{j:p_{jt} < P_t} b_{jt}, & p_{it} = P_t, \\ 0, & p_{it} > P_t. \end{cases}$$

This procedure produces deterministic outputs P_t and q_{it} given all bids, including the renewable's u_t^R .

5. Renewable Producer Model (Perspective 1)

(a) Inputs (from environment and regulator)

- $D_t, \{b_{it}, p_{it}\}_{i>0}$: random conventional bids and demand
- $u_t^G = (\pi_t^U, \pi_t^O, \bar{E})$: regulator's fixed policy parameters

(b) Control Variables

$$u_t^R = (b_{0t}, p_{0t})$$

(c) Realized Profit per Hour

$$\Pi_{0t} = P_t q_{0t} - \pi_t^U [q_{0t} - G_{0t}]^+ - \pi_t^O [G_{0t} - q_{0t}]^+.$$

(d) Objectives

1. Expected Robust Profit

$$J_1^R(u^R; u^G) = \max_{u^R} \min_{G_{0t} \in [\underline{G}_t, \bar{G}_t]} \mathbb{E}_{D_t, b_{it}, p_{it}} \left[\sum_{t=1}^{24} \Pi_{0t} \right].$$

2. Commitment Reliability

$$J_2^R(u^R; u^G) = \min_{u^R} \mathbb{E}_{D_t, b_{it}, p_{it}, G_{0t}} \left[\sum_{t=1}^{24} |q_{0t} - G_{0t}| \right].$$

(e) Constraints

$$b_{0t} \geq 0, p_{0t} \geq 0, \forall t, \quad \sum_i q_{it} = D_t.$$

6. System Regulator Model (Perspective 2)

(a) Inputs (from environment and renewable producer)

- D_t, b_{it}, p_{it} : stochastic environment
- $u_t^R = (b_{0t}, p_{0t})$: renewable producer's fixed strategy

(b) Control Variables

$$u_t^G = (\pi_t^U, \pi_t^O, \bar{E})$$

(c) Measurable System Outcomes

- P_t, q_{it} : market results
- $S_t = \sum_i \gamma_i q_{it}$: emissions

- $C_i(q_{it}) = \alpha_i q_{it} + \frac{1}{2} \beta_i q_{it}^2$: generation cost

(d) Objectives

All are measurable expectations over the stochastic environment.

1. System Cost Minimization

$$J_1^G(u^G; u^R) = \min_{u^G} \mathbb{E} \left[\sum_{t=1}^{24} \sum_{i=1}^N C_i(q_{it}) \right].$$

2. Emission Minimization

$$J_2^G(u^G; u^R) = \min_{u^G} \mathbb{E} \left[\sum_{t=1}^{24} S_t \right], \quad \text{s.t.} \quad \sum_t S_t \leq \bar{E}.$$

3. Price Volatility Minimization

$$J_3^G(u^G; u^R) = \min_{u^G} \mathbb{E} \left[\sum_{t=1}^{24} (P_t - \mathbb{E}[P_t])^2 \right].$$

(e) Constraints

$$\sum_i q_{it} = D_t, \quad S_t = \sum_i \gamma_i q_{it}, \quad \sum_t S_t \leq \bar{E}.$$

7. Unified Composition

All components now combine into one structured system:

(a) Shared State Evolution

$$x_{t+1} = f(x_t, u_t^R, u_t^G, \xi_t),$$

where $f(\cdot)$ embeds demand and bid sampling and merit-order clearing.

(b) Stakeholder Objectives

$$\begin{aligned} \text{Renewable Producer:} & \quad \{J_1^R, J_2^R\}(u^R; u^G) \\ \text{System Regulator:} & \quad \{J_1^G, J_2^G, J_3^G\}(u^G; u^R) \end{aligned}$$

Each can be optimized modularly by holding the other's control fixed.

(c) Shared Constraints (common to both perspectives)

$$\sum_i q_{it} = D_t, \quad q_{it} \in [0, b_{it}], \quad P_t \text{ satisfies merit order,}$$

$$S_t = \sum_i \gamma_i q_{it}, \quad \sum_t S_t \leq \bar{E}.$$

8. Modular Evaluation Procedure

This structure allows **modular evaluation** of each stakeholder:

Step	Description	Control Treated as Exogenous
1	Fix regulator policy u^G . Optimize renewable producer bids via J_1^R, J_2^R .	u^G
2	Fix renewable bids u^R . Optimize regulator policy via J_1^G, J_2^G, J_3^G .	u^R
3	Optionally iterate toward equilibrium (if analyzing joint behavior).	—

This ensures each submodel is internally consistent and operational within the unified environment.

9. Summary of Unified Components

Category	Symbol / Definition	Shared?	Quantifiable?
State variables	$x_t = (D_t, G_{0t}, P_t, q_t, S_t)$	Shared	Yes
Renewable controls	$u_t^R = (b_{0t}, p_{0t})$	No (producer)	Yes
Regulator controls	$u_t^G = (\pi_t^U, \pi_t^O, \bar{E})$	No (regulator)	Yes
Transition	$x_{t+1} = f(x_t, u_t^R, u_t^G, \xi_t)$	Shared	Yes
Producer objectives	J_1^R, J_2^R	Modular	Yes
Regulator objectives	J_1^G, J_2^G, J_3^G	Modular	Yes
Constraints	Market clearing, emissions cap	Shared	Yes

✓ In summary:

The unified model consists of:

1. A **shared, quantifiable environment** x_t governed by measurable stochastic and economic rules (demand, bids, generation, prices, emissions);
2. **Two agent modules**, each with its own control variables and quantifiable objectives;
3. Each module treats the other's decisions as *given* when optimized;
4. All system relations (market clearing, penalties, emissions) are consistent and measurable.

This model is now **internally consistent, implementable, and modular**, suitable for:

- simulation (Monte Carlo over stochastic inputs),
- optimization (bi-level or iterative fixed-point approaches), or
- policy evaluation (testing penalty/emission schemes vs. producer bidding strategies).



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.

Below is a **clean, modular, and implementable Python design** for the unified model.

It uses **object-oriented structure** to represent the **common environment**, **renewable producer**, and **system regulator**, with full internal consistency and modularity.

At the end, a single function `simulate_market()` provides the **interface**:

- Input:
 - **controls**: a dictionary of both agents' control variables;
 - **uncertainty**: a dictionary specifying stochastic inputs (e.g. demand distribution, competitor bid distributions, etc.).
- Output:
 - A dictionary with each **objective value** (for renewable producer and regulator).

python

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

# =====
# ===== 1. COMMON ENVIRONMENT =====
# =====

@dataclass
class MarketEnvironment:
    """Shared market environment for all stakeholders."""
    T: int = 24 # hours in the day

    # --- Global parameters ---
    mu_D: float = 1000.0 # mean demand (MW)
    sigma_D: float = 100.0 # std dev of demand
    mu_b: List[float] = field(default_factory=lambda: [200.0, 250.0, 300.0])
    sigma_b: List[float] = field(default_factory=lambda: [20.0, 25.0, 30.0])
    mu_p: List[float] = field(default_factory=lambda: [40.0, 50.0, 60.0])
    sigma_p: List[float] = field(default_factory=lambda: [5.0, 5.0, 5.0])

    alpha: List[float] = field(default_factory=lambda: [10.0, 12.0, 15.0]) # cost linear coeff
    beta: List[float] = field(default_factory=lambda: [0.05, 0.04, 0.03]) # cost quadratic
    gamma: List[float] = field(default_factory=lambda: [0.5, 0.4, 0.3]) # emissions ton
    CO2/MWh

    G_bounds: List[Tuple[float, float]] = field(
        default_factory=lambda: [(100.0, 300.0)] * 24
    )

    # --- Random sampling of stochastic variables ---
    def sample_demand(self) -> np.ndarray:
        return np.random.normal(self.mu_D, self.sigma_D, self.T)
```

```

def sample_conventional_bids(self) -> Tuple[np.ndarray, np.ndarray]:
    """Return random bid quantities and prices for conventional producers."""
    N = len(self.mu_b)
    b = np.zeros((N, self.T))
    p = np.zeros((N, self.T))
    for i in range(N):
        b[i, :] = np.random.normal(self.mu_b[i], self.sigma_b[i], self.T)
        p[i, :] = np.random.normal(self.mu_p[i], self.sigma_p[i], self.T)
    return b, p

# =====
# ===== 2. RENEWABLE PRODUCER =====
# =====

class RenewableProducer:
    """Renewable producer perspective with bidding and profit evaluation."""

    def __init__(self, env: MarketEnvironment):
        self.env = env

    def profit(self, b0, p0, piU, pi0, D, b_conv, p_conv, G_real):
        """Compute total profit across all hours given fixed bids and outcomes."""
        T = self.env.T
        profits = np.zeros(T)

        for t in range(T):
            # 1. Collect all bids (including renewable)
            b_all = np.append(b_conv[:, t], b0[t])
            p_all = np.append(p_conv[:, t], p0[t])

            # 2. Sort by offer price (merit order)
            order = np.argsort(p_all)
            b_sorted, p_sorted = b_all[order], p_all[order]

            # 3. Determine clearing price and dispatch
            cum_supply = np.cumsum(b_sorted)
            idx = np.searchsorted(cum_supply, D[t])
            idx = np.clip(idx, 0, len(b_sorted) - 1)
            P_t = p_sorted[idx]
            q_sorted = np.zeros_like(b_sorted)
            q_sorted[:idx] = b_sorted[:idx]
            q_sorted[idx] = D[t] - np.sum(b_sorted[:idx])

            # 4. Find renewable dispatch
            # last element in sorted corresponds to renewable producer (added last)
            renewable_index = np.where(order == len(order) - 1)[0][0]
            q0_t = q_sorted[renewable_index]

            # 5. Compute deviation penalties
            G_t = np.clip(G_real[t], *self.env.G_bounds[t])
            s_minus = max(0, q0_t - G_t)
            s_plus = max(0, G_t - q0_t)
            profits[t] = P_t * q0_t - piU * s_minus - pi0 * s_plus

        return profits.sum()

    def evaluate_objectives(self, controls: Dict, uncertainty: Dict) -> Dict[str, float]:
        """Evaluate renewable producer objectives."""
        b0 = np.array(controls.get("b0", [200.0] * self.env.T))
        p0 = np.array(controls.get("p0", [50.0] * self.env.T))
        piU = uncertainty["piU"]
        pi0 = uncertainty["pi0"]

        D = uncertainty["D"]
        b_conv = uncertainty["b_conv"]
        p_conv = uncertainty["p_conv"]
        G_real = uncertainty["G_real"]

        # Objective 1: robust expected profit (simulate min over G bounds)
        profit_mean = self.profit(b0, p0, piU, pi0, D, b_conv, p_conv, G_real)

        # Objective 2: reliability (expected absolute deviation)
        T = self.env.T
        reliability = np.mean([abs(G_real[t] - b0[t]) for t in range(T)])

```

```

        return {"J1_profit": profit_mean, "J2_reliability": reliability}

# =====
# ===== 3. SYSTEM REGULATOR =====
# =====

class SystemRegulator:
    """Regulator perspective: evaluate costs, emissions, and price volatility."""

    def __init__(self, env: MarketEnvironment):
        self.env = env

    def market_clearing(self, b0, p0, D, b_conv, p_conv):
        """Perform market clearing and return prices, dispatch, and emissions."""
        T = self.env.T
        N = len(self.env.mu_b)
        P = np.zeros(T)
        q_conv = np.zeros((N, T))
        q0 = np.zeros(T)

        for t in range(T):
            b_all = np.append(b_conv[:, t], b0[t])
            p_all = np.append(p_conv[:, t], p0[t])
            order = np.argsort(p_all)
            b_sorted, p_sorted = b_all[order], p_all[order]
            cum_supply = np.cumsum(b_sorted)
            idx = np.searchsorted(cum_supply, D[t])
            idx = np.clip(idx, 0, len(b_sorted) - 1)
            P[t] = p_sorted[idx]
            q_sorted = np.zeros_like(b_sorted)
            q_sorted[:idx] = b_sorted[:idx]
            q_sorted[idx] = D[t] - np.sum(b_sorted[:idx])

            # revert to original order
            q_all = np.zeros_like(q_sorted)
            q_all[order] = q_sorted
            q_conv[:, t] = q_all[:N]
            q0[t] = q_all[N]

        # Emissions
        gamma = np.array(self.env.gamma)
        S = np.dot(gamma, q_conv)
        return P, q_conv, q0, S

    def evaluate_objectives(self, controls: Dict, uncertainty: Dict) -> Dict[str, float]:
        """Evaluate regulator objectives."""
        piU = controls["piU"]
        pi0 = controls["pi0"]
        E_cap = controls["E_cap"]

        b0 = uncertainty["b0"]
        p0 = uncertainty["p0"]
        D = uncertainty["D"]
        b_conv = uncertainty["b_conv"]
        p_conv = uncertainty["p_conv"]

        P, q_conv, q0, S = self.market_clearing(b0, p0, D, b_conv, p_conv)

        # Cost
        alpha, beta = np.array(self.env.alpha), np.array(self.env.beta)
        cost = np.sum(alpha[:, None] * q_conv + 0.5 * beta[:, None] * q_conv ** 2)

        # Emissions
        emissions = np.sum(S)
        penalty_emission = max(0, emissions - E_cap)

        # Price volatility
        price_vol = np.var(P)

        return {
            "J1_system_cost": cost,
            "J2_emissions": emissions,
            "J3_price_volatility": price_vol,

```

```

        "penalty_emission_violation": penalty_emission,
    }

# =====
# ===== 4. UNIFIED SIMULATION =====
# =====

def simulate_market(controls: Dict, uncertainty: Dict) -> Dict[str, float]:
    """
    Interface function for unified model simulation.

    Inputs:
        controls: dict containing all control variables
        {
            "b0": [...], "p0": [...],          # renewable bids
            "piU": float, "pi0": float, "E_cap": float # regulator policy
        }

        uncertainty: dict containing sampled or fixed uncertainty variables
        {
            "D": np.array,          # hourly demand
            "b_conv": np.ndarray,    # competitor bid quantities
            "p_conv": np.ndarray,    # competitor bid prices
            "G_real": np.array      # realized renewable generation
        }

    Output:
        dict of all objective values from both perspectives
    """

    # --- Initialize environment ---
    env = MarketEnvironment()

    # --- Instantiate modules ---
    renewable = RenewableProducer(env)
    regulator = SystemRegulator(env)

    # --- Evaluate renewable producer objectives ---
    renewable_objs = renewable.evaluate_objectives(controls, uncertainty)

    # --- Evaluate regulator objectives ---
    regulator_objs = regulator.evaluate_objectives(controls, uncertainty)

    # --- Combine results ---
    results = {**renewable_objs, **regulator_objs}
    return results

# =====
# ===== EXAMPLE USAGE (not executed) =====
# =====
"""
Example (not to execute):

controls = {
    "b0": [220.0]*24,
    "p0": [45.0]*24,
    "piU": 30.0,
    "pi0": 20.0,
    "E_cap": 10000.0,
}

# Generate uncertainty samples
env = MarketEnvironment()
D = env.sample_demand()
b_conv, p_conv = env.sample_conventional_bids()
G_real = np.random.uniform(100, 300, 24)

uncertainty = {
    "D": D,
    "b_conv": b_conv,
    "p_conv": p_conv,
    "G_real": G_real,
}

```

```
results = simulate_market(controls, uncertainty)
print(results)
"""
```

✓ Model Structure Summary

Component	Python Class	Key Methods
Common environment	MarketEnvironment	sample_demand(), sample_conventional_bids()
Renewable producer	RenewableProducer	profit(), evaluate_objectives()
System regulator	SystemRegulator	market_clearing(), evaluate_objectives()
Unified interface	Function simulate_market()	Integrates both perspectives and returns all objectives

This program:

- Is **modular**, allowing independent evaluation of each stakeholder's objectives;
- Treats the other agent's controls as **exogenous inputs**;
- Shares a consistent **environment and market mechanism**;
- Returns **quantifiable objective values** for both perspectives in one unified call.



piU and piO are control variable inputs, instead of uncertainty inputs. Please refine the implementation.

Perfect — you're absolutely right:

The penalty rates **piU** and **piO** are **control variables** of the **system regulator**, not random uncertainties.

Below is a **refined, internally consistent** Python implementation of the unified modular model.

All definitions, roles, and data flows have been corrected accordingly.

python

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

# =====
# ===== 1. COMMON ENVIRONMENT =====
# =====

@dataclass
class MarketEnvironment:
    """Shared market environment for all stakeholders."""
    T: int = 24 # number of hourly intervals

    # --- Global parameters ---
    mu_D: float = 1000.0
    sigma_D: float = 100.0
```

```
mu_b: List[float] = field(default_factory=lambda: [200.0, 250.0, 300.0])
sigma_b: List[float] = field(default_factory=lambda: [20.0, 25.0, 30.0])
mu_p: List[float] = field(default_factory=lambda: [40.0, 50.0, 60.0])
sigma_p: List[float] = field(default_factory=lambda: [5.0, 5.0, 5.0])

alpha: List[float] = field(default_factory=lambda: [10.0, 12.0, 15.0])
beta: List[float] = field(default_factory=lambda: [0.05, 0.04, 0.03])
gamma: List[float] = field(default_factory=lambda: [0.5, 0.4, 0.3])
```

```
G_bounds: List[Tuple[float, float]] = field(
    default_factory=lambda: [(100.0, 300.0)] * 24
)
```

```
# --- Random sampling ---
```

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

```
def sample_conventional_bids(self) -> Tuple[np.ndarray, np.ndarray]:
    """Random bid quantities and prices for conventional producers."""
    N = len(self.mu_b)
    b = np.zeros((N, self.T))
    p = np.zeros((N, self.T))
    for i in range(N):
        b[i, :] = np.random.normal(self.mu_b[i], self.sigma_b[i], self.T)
        p[i, :] = np.random.normal(self.mu_p[i], self.sigma_p[i], self.T)
    return b, p
```

```
# =====
# ===== 2. RENEWABLE PRODUCER =====
# =====
```

```
class RenewableProducer:
    """Renewable producer model with bid-based profit evaluation."""

    def __init__(self, env: MarketEnvironment):
        self.env = env

    def profit(self, b0, p0, piU, pi0, D, b_conv, p_conv, G_real):
        """Compute total profit across all hours given bids and penalties."""
        T = self.env.T
        profits = np.zeros(T)

        for t in range(T):
            # Combine all bids
            b_all = np.append(b_conv[:, t], b0[t])
            p_all = np.append(p_conv[:, t], p0[t])

            # Merit-order sort
            order = np.argsort(p_all)
            b_sorted, p_sorted = b_all[order], p_all[order]

            # Determine clearing price and dispatch
            cum_supply = np.cumsum(b_sorted)
            idx = np.searchsorted(cum_supply, D[t])
            idx = np.clip(idx, 0, len(b_sorted) - 1)
            P_t = p_sorted[idx]
            q_sorted = np.zeros_like(b_sorted)
            q_sorted[:idx] = b_sorted[:idx]
            q_sorted[idx] = D[t] - np.sum(b_sorted[:idx])

            # Retrieve renewable dispatch
            renewable_index = np.where(order == len(order) - 1)[0][0]
            q0_t = q_sorted[renewable_index]

            # Compute penalties and profit
            G_t = np.clip(G_real[t], *self.env.G_bounds[t])
            s_minus = max(0, q0_t - G_t)
            s_plus = max(0, G_t - q0_t)
            profits[t] = P_t * q0_t - piU * s_minus - pi0 * s_plus

        return profits.sum()

    def evaluate_objectives(self, controls: Dict, uncertainty: Dict) -> Dict[str, float]:
        """Evaluate renewable producer objectives."""
```

```

b0 = np.array(controls.get("b0", [200.0] * self.env.T))
p0 = np.array(controls.get("p0", [50.0] * self.env.T))
piU = controls["piU"]
pi0 = controls["pi0"]

D = uncertainty["D"]
b_conv = uncertainty["b_conv"]
p_conv = uncertainty["p_conv"]
G_real = uncertainty["G_real"]

# Objective 1: profit
profit_val = self.profit(b0, p0, piU, pi0, D, b_conv, p_conv, G_real)

# Objective 2: reliability (expected absolute deviation)
reliability = float(np.mean(np.abs(G_real - b0)))

return {"J1_profit": profit_val, "J2_reliability": reliability}

# =====
# ===== 3. SYSTEM REGULATOR =====
# =====

class SystemRegulator:
    """Regulator model for evaluating cost, emissions, and price volatility."""

    def __init__(self, env: MarketEnvironment):
        self.env = env

    def market_clearing(self, b0, p0, D, b_conv, p_conv):
        """Return prices, dispatch quantities, and system emissions."""
        T = self.env.T
        N = len(self.env.mu_b)
        P = np.zeros(T)
        q_conv = np.zeros((N, T))
        q0 = np.zeros(T)

        for t in range(T):
            b_all = np.append(b_conv[:, t], b0[t])
            p_all = np.append(p_conv[:, t], p0[t])
            order = np.argsort(p_all)
            b_sorted, p_sorted = b_all[order], p_all[order]
            cum_supply = np.cumsum(b_sorted)
            idx = np.searchsorted(cum_supply, D[t])
            idx = np.clip(idx, 0, len(b_sorted) - 1)
            P[t] = p_sorted[idx]
            q_sorted = np.zeros_like(b_sorted)
            q_sorted[:idx] = b_sorted[:idx]
            q_sorted[idx] = D[t] - np.sum(b_sorted[:idx])

            q_all = np.zeros_like(q_sorted)
            q_all[order] = q_sorted
            q_conv[:, t] = q_all[:N]
            q0[t] = q_all[N]

        gamma = np.array(self.env.gamma)
        S = np.dot(gamma, q_conv)
        return P, q_conv, q0, S

    def evaluate_objectives(self, controls: Dict, uncertainty: Dict) -> Dict[str, float]:
        """Evaluate regulator objectives based on market outcomes."""
        piU = controls["piU"]
        pi0 = controls["pi0"]
        E_cap = controls["E_cap"]

        b0 = controls["b0"]
        p0 = controls["p0"]
        D = uncertainty["D"]
        b_conv = uncertainty["b_conv"]
        p_conv = uncertainty["p_conv"]

        P, q_conv, q0, S = self.market_clearing(b0, p0, D, b_conv, p_conv)

        # System cost
        alpha, beta = np.array(self.env.alpha), np.array(self.env.beta)

```

```

cost = np.sum(alpha[:, None] * q_conv + 0.5 * beta[:, None] * q_conv ** 2)

# Emissions
emissions = np.sum(S)
emission_penalty = max(0, emissions - E_cap)

# Price volatility
price_volatility = float(np.var(P))

return {
    "J1_system_cost": float(cost),
    "J2_emissions": float(emissions),
    "J3_price_volatility": price_volatility,
    "emission_cap_violation": emission_penalty,
}

# =====
# ===== 4. UNIFIED SIMULATION =====
# =====

def simulate_market(controls: Dict, uncertainty: Dict) -> Dict[str, float]:
    """
    Unified model interface.

    Parameters
    -----
    controls : dict
        Control variables for both stakeholders:
        {
            "b0": [...], "p0": [...], # renewable bids
            "piU": float, "pi0": float, # penalty rates (regulator)
            "E_cap": float # emissions cap
        }

    uncertainty : dict
        Stochastic / uncertain inputs:
        {
            "D": np.array, # hourly demand
            "b_conv": np.ndarray, # competitor bid quantities
            "p_conv": np.ndarray, # competitor offer prices
            "G_real": np.array # realized renewable generation
        }

    Returns
    -----
    dict
        Objective function values from both perspectives.
    """

    # Shared environment
    env = MarketEnvironment()

    # Instantiate agents
    renewable = RenewableProducer(env)
    regulator = SystemRegulator(env)

    # Evaluate renewable producer objectives
    renewable_objs = renewable.evaluate_objectives(controls, uncertainty)

    # Evaluate regulator objectives
    regulator_objs = regulator.evaluate_objectives(controls, uncertainty)

    # Combine and return results
    results = {**renewable_objs, **regulator_objs}
    return results

# =====
# ===== EXAMPLE USAGE (not run) =====
# =====
"""
# Example configuration (not executed):

controls = {

```

```

    "b0": [220.0]*24,
    "p0": [45.0]*24,
    "piU": 30.0,
    "pi0": 20.0,
    "E_cap": 10000.0,
}

# Generate uncertainty samples
env = MarketEnvironment()
D = env.sample_demand()
b_conv, p_conv = env.sample_conventional_bids()
G_real = np.random.uniform(100, 300, 24)

uncertainty = {
    "D": D,
    "b_conv": b_conv,
    "p_conv": p_conv,
    "G_real": G_real,
}

results = simulate_market(controls, uncertainty)
print(results)
"""

```

✓ Key Refinements

Aspect	Change Made	Rationale
Control vs. Uncertainty	piU, pi0 moved to controls	They are regulator-controlled policy levers
Renewable producer	Reads piU, pi0 from controls	Consistent with policy control
Regulator model	Reads all controls (b0, p0, piU, pi0, E_cap)	Single source for control state
Unified interface	simulate_market() now passes controls consistently	Enables modular, reproducible simulation

🧩 Structure Overview

Component	Python Class	Key Purpose
MarketEnvironment	Defines global parameters, stochastic sampling	Shared across modules
RenewableProducer	Evaluates renewable's objectives (profit, reliability)	Uses controls + environment
SystemRegulator	Evaluates regulator's objectives (cost, emissions, price volatility)	Uses controls + environment
simulate_market()	Unified interface combining both	Returns all objective values

✓ Result:

A **self-contained, modular, and internally consistent** Python framework where each stakeholder's control variables are explicit and separable, the shared environment is consistent, and all objectives are quantitatively computable.