© August 2025| IJIRT | Volume 12 Issue 3 | ISSN: 2349-6002

Dynamic Risk Balancing in Manufacturing using Agentic Reinforcement Learning

Dr. Gopichand Agnihotram[1], Joydeep Sarkar[2]
[1,2]Wipro Ltd.

Abstract— In modern manufacturing, achieving a dynamic equilibrium between producer and consumer risks is critical for sustainable profitability and market competitiveness. This paper introduces a novel framework using Agentic Reinforcement Learning (RL) to address this complex optimization challenge. We formulate the problem by defining an RL agent designed to navigate the manufacturing environment by making sequential decisions on production, pricing, and inventory. The core of our approach is a comprehensive reward function that maximizes profitability while explicitly penalizing risks for both producers and consumers. Producer risks, such as inventory holding costs, stockouts, and production inefficiency, are balanced against consumer risks like price volatility, product unavailability, and quality dissatisfaction. By training the agent to optimize a policy that maximizes the cumulative discounted reward, the framework enables autonomous, data-driven decisions that adapt to real-time market dynamics. This methodology provides a robust solution for enhancing operational efficiency and resilience in complex manufacturing systems.

Index Terms— Agentic Framework for Operations, AI in Production Scheduling, Autonomous Decision-Making, Dynamic Risk Balancing, Producer and Consumer Risk, Reinforcement Learning in Manufacturing, Smart Manufacturing, Supply Chain Optimization.

I. INTRODUCTION

In real-world decision-making contexts such as manufacturing, quality control, and service delivery, organizations must navigate a trade-off between risks experienced by both consumers and producers. These are commonly categorized as consumer risk and producer risk, each significantly influencing strategies related to quality assurance, cost efficiency, and customer satisfaction. With recent advances in machine learning, particularly in Reinforcement Learning (RL), there is now a powerful means to dynamically model, analyse, and balance these risks in pursuit of optimal operational outcomes. The application of such intelligent systems is becoming a cornerstone of modern risk management in manufacturing, allowing for the prediction and mitigation of operational disruptions[1][2] . This document introduces the foundational concepts of consumer and producer risks, formulates the optimization problem of cost minimization through risk balancing, and provides formal definitions and illustrative examples of each risk category.

II. PROBLEM STATEMENT

Manufacturing operations must continuously balance cost efficiency, product quality, and customer satisfaction under uncertain and dynamic conditions. Key challenges include managing inventory levels, production schedules, pricing strategies, and supply chain disruptions—all while minimizing risks that affect both producers and consumers.

The problem is to optimize operational decisions in a way that:

- Minimizes producer risks such as excess inventory, stockouts, and production inefficiencies.
- Minimizes consumer risks such as price volatility, product unavailability, and dissatisfaction.
- Maximizes overall profitability and responsiveness to market conditions.

This requires a dynamic decision-making framework that can adapt to real-time data and evolving constraints, ensuring robust performance across diverse manufacturing scenarios[3] .

III. RL FRAMEWORK COMPONENTS

A. Applying Reinforcement Learning to Minimize Costs by Balancing Risks

Balancing the risks faced by consumers and producers is a classic business dilemma. Producers want to maximize profit by minimizing costs (inventory, production) and maximizing revenue, which can lead to higher prices or lower stock levels. Consumers want fair prices, high-quality products, and consistent availability. These goals are often in conflict. This complex, dynamic optimization problem is an ideal use case for Reinforcement Learning (RL), which excels at making sequential decisions in an uncertain environment to maximize a long-term goal [4].

B. Framing the Problem in RL Terms

To apply RL, we must first define the core components of the model: the Agent, the Environment, the State, the Actions, and the Reward.

1. The Agent:

The agent is the decision-making entity that we are training. It could be an algorithm controlling a specific business function, such as:

- A Pricing Engine
- An Inventory Management System
- A Production Scheduler
- A Marketing Campaign Manager

2. The Environment:

The environment is the market in which the agent operates. It is everything the agent interacts with but doesn't directly control. This includes:

- Consumer demand and behaviour patterns.
- Competitor pricing and actions.
- Raw material costs and availability.
- Supply chain latency
- Market trends, seasonality, and economic factors.

3. State Space:

The state space encapsulates all the critical information required for effective decision-making at any given time. The manufacturing environment at any time step is described by a multidimensional state vector. This state captures all relevant information necessary for decision-making, including inventory levels for each product, forecasted demand, current market prices etc. These elements provide a comprehensive snapshot of the manufacturing system's operational context, enabling the agent to make informed decisions that account for both supply-side and demand-side uncertainties.

Formally, the state at time t is represented as: $S_t = \{I_t, D_t, P_t, L_t, Q_t, R_t\}$

Where,

I_t denotes inventory levels,
D_t is forecasted demand,
P_t represents current prices,
L_t is supplier lead time,
Q_t is the production queue status,
R_t captures recent return rates or customer satisfaction signals.

4. Action Space:

The action space defines the set of decisions available to the agent at each decision point. Typical actions include determining the production quantity, allocating stock, and setting product prices. The latter, known as dynamic pricing, is itself a well-studied application of RL for revenue management[5][6]. By selecting appropriate actions, the agent directly influences inventory levels, market responsiveness, and overall system efficiency. These actions are designed to be flexible, allowing the agent to adapt to changing market conditions and operational constraints.

At each decision epoch, the reinforcement learning (RL) agent selects an action vector that determines operational choices.

At each time step t, the agent selects an action At defined as: $A_t = \{X_{i,t}, P_{i,t}, S_{i,t}\}$

Where:

$X_{i,t}$: Units to produce of product i at time t

$P_{i,t}$: Price to set for product i at time t

$S_{i,t}$: Stock to allocate or ship for product i at time t

5. The Reward Function

The Core of Risk Balancing: This is the most critical component. To minimize cost and balance risks, the reward function must be carefully designed to reflect this objective. Cost minimization is equivalent to maximizing a negative cost (i.e., maximizing profit and minimizing penalties).

The reward function is designed to balance profitability with risk management. At each time step, the agent receives a reward that reflects the net profit, calculated as total revenue minus the cost of goods sold and operational expenses. Importantly, the reward is adjusted by subtracting both producer and consumer risks, thereby incentivizing the agent to consider the broader impact of its decisions beyond immediate financial gain.

At each time step t, the reward is defined as:

$$R_t = Revenue_t - COGS_t - OpCost_t - Producer\ Risk_t - Consumer\ Risk_t$$

Where:

i. Revenue

$$Revenue_t = \sum_i p_{i,t}\, y_{i,t}$$

Where:

$p_{i,t}$: Price of product i at time t

$y_{i,t}$: Units of product i sold at time t

ii. COGS (Cost of Goods Sold)

$$COGS_t = \sum_i c_{i,t}\, x_{i,t}$$

Where:

$c_{i,t}$: Unit production cost of product i

$x_{i,t}$: Units produced of product i

iii. Operational Cost

$$OpCost_t = Fixed_t + \sum_i Variable_{i,t}$$

IV. RISK FUNCTIONS

A. Producer Risk Function

Producer risk represents the potential costs and inefficiencies experienced by the manufacturer. It is modelled as a weighted sum of inventory holding costs, stockout costs, and penalties for production schedule inefficiency:

$$Producer\ Risk_t = \propto_1 H_t + \propto_2 S_t^{out} + \propto_3 D_t^{ineff}$$

Where,

H_t: is the total inventory holding cost, calculated as the sum over all products of the squared deviation from ideal inventory levels, weighted by storage cost.

S_t^{out}: is the stockout cost, representing the cost of unmet demand

D_t^{ineff}: Downtime or production schedule inefficiency penalty

$\propto_1, \propto_2, \propto_3$: Weighting hyperparameters for the producer's risk components.

B. Consumer Risk Function

Consumer risk captures the negative outcomes experienced by customers because of the manufacturer's decisions. Like producer risk, consumer risk is aggregated using weighted terms, with each component representing a distinct aspect of the customer experience.

The Consumer Risk at time t is defined as:

$$Consumer\ Risk_t = \beta_1 V_t + \beta_2 U_t + \beta_3 Q_t$$

Where:

- V_t: Price Volatility
  This is typically measured as the sum of squared price changes across all products:
  $$V_t = \sum_i (p_{i,t} - p_{i,t-1})^2$$
- U_t: Stockout Impact
  The total number of consumers who experienced stockouts (i.e., demand not met due to insufficient inventory) at time t.
- Q_t: Quality Dissatisfaction
  This could be measured by the rate of negative reviews or product return rate at time t, indicating dissatisfaction with the product.
- $\beta_1, \beta_2, \beta_3$: Weighting Coefficients
  These are modifiable parameters that reflect the relative importance of each risk factor from the consumer's perspective.

V. REINFORCEMENT LEARNING OBJECTIVE

The overarching objective for the RL agent is to maximize the expected cumulative reward over a specified time horizon. This is achieved by learning a policy—a mapping from states to actions—that optimizes long-term outcomes.

The agent aims to maximize the expected cumulative discounted reward over a finite time horizon T:

$$max_\pi\ E_\pi \left[\sum_{t=0}^{T} \gamma^t R_t \right]$$

Where:

- π: Policy
  A mapping from states to actions. It defines the agent's behaviour, how it chooses actions based on the current state.
- E_π: Expectation under policy π The expected value of the cumulative reward when the agent follows policy π.
- $\gamma \in [0,1]$: Discount Factor
  Determines the importance of future rewards. A lower γ prioritizes short-term gains, while a higher γ emphasizes long-term outcomes.
- R_t: Reward at time t

VI. FINAL OPTIMIZATION FUNCTION

The RL agent's goal is to choose actions over time that maximize the overall expected reward. The reward at each step is calculated as the profit (revenue minus costs), but with extra penalties for different types of risks faced by both producers and consumers.

- The agent tries to maximize profit (revenue minus costs).
- It is also penalized for producer-side risks, including:
  - Holding excess inventory (H_t)
  - Stockouts leading to lost sales (S_t^{out})
  - Production inefficiencies (D_t^{ineff})
- It is also penalized for consumer-side risks, including:
  - Price volatility (V_t)
  - Product unavailability (U_t)
  - Poor product quality (Q_t)

The agent seeks to maximize the expected cumulative discounted reward over a finite time horizon T, defined as:

$$max_\pi\, E_\pi \left[\sum_{t=0}^{T} \gamma^t (Revenue_t - COGS_t - OpCost_t - \propto_1 H_t - \propto_2 S_t^{out} - \propto_3 D_t^{ineff} - \beta_1 V_t - \beta_2 U_t - \beta_3 Q_t) \right]$$

This comprehensive objective ensures that the agent not only maximizes profit but also systematically balances and minimizes both producer and consumer risks.

VII. RL AGENT TRAINING

To solve the optimization function the following RL setup can be used:

- Algorithm: Use PPO or SAC for continuous action spaces. These algorithms are well-suited for complex control tasks due to their stability and sample efficiency[7][8].
- State Encoder: Normalize and encode state vector $S_t = \{I_t, D_t, P_t, L_t, Q_t, R_t\}$.
- Policy Network: Maps states to actions $A_t = \{X_{i,t}, P_{i,t}, S_{i,t}\}$
- Critic Network: Estimates the value function $V^\pi(S_t)$ to guide policy updates.
- Replay Buffer: Store transitions (S_t, A_t, S_{t+1}, R_t) for off-policy learning (SAC).
- Exploration Strategy: Use entropy regularization or epsilon-greedy.

Simulation Environment:

- Model demand, supply delays, and customer behaviour stochastically.
- Initialize dynamics using ERP/MES historical data.

Hyperparameter Tuning:

- Apply grid search or Bayesian optimization for learning rate, discount factor γ, entropy coefficient, and batch size.
- Track cumulative reward and policy entropy.

Convergence Monitoring:

- Monitor reward trends and entropy for learning progress.
- Use early stopping if performance plateaus or declines.

VIII. APPLICATION OF THE RL FRAMEWORK IN GENERAL MANUFACTURING

In a typical manufacturing setup, whether producing electronics, automotive parts, consumer goods, or industrial equipment, this document outlines a framework that applies optimization using RL to improve operational efficiency and responsiveness[9]. We used the following framework,

A. Agent: The agent acts as a digital controller or decision-support system, that can lead to more sustainable and lean production outcomes[10], by managing:

- Production scheduling: Deciding when and how much to produce based on demand, resource availability, and constraints.
- Procurement: Ordering raw materials, components, or subassemblies from suppliers.
- Resource allocation: Assigning machines, labour, and tools to appropriate production tasks or lines.

B. Environment: The manufacturing environment is dynamic and influenced by:

- Demand variability: Fluctuations in customer orders due to seasonality, market trends, or custom requirements.
- Raw material and component price fluctuations: Cost changes in inputs like metals, plastics, electronics, or chemicals.
- Supplier lead time variability: Uncertainties in delivery schedules affecting production continuity.
- Operational disruptions: Machine breakdowns, labour shortages, or quality issues impacting throughput.

This aligns with modern supply chain strategies that use RL to optimize inventory and order management in the face of uncertainty[11][12].

C. State: At any given time, the state of the manufacturing system includes:

- Inventory levels: For both raw materials and finished goods.
- Pending orders: Demand pipeline, including firm orders and forecasts.
- Machine and labour status: Whether resources are operational, idle, or under maintenance.
- Forecasted demand: Derived from historical sales, seasonality, or predictive models.

Metric	Before RL (Rule-Based Scheduling)	After RL (RL Agent Scheduling)	Improvement
Order Fulfilment Rate	89.5%	97.2%	+8.6%
Inventory Holding Costs	$1.25M	$870K	+30.4%
Stockout Events	42	11	+73.8%
Estimated Net Profit	$5.1M	$6.4M	+25.4%

D. Actions: The agent can take real-time actions such as:

- Start/stop production runs: Based on inventory positions and incoming demand.
- Adjust batch sizes: To optimize for cost, changeover time, and order fulfilment.
- Reassign resources: Switch machines, lines, or workforce to high-priority or delayed tasks.
- Expedite procurement: Place urgent material/component orders to recover from potential delays.

E. Reward Function: The reward function balances efficiency and customer satisfaction by penalizing both producer-side and consumer-side risks,

- Producer-side penalties:
  - Overstock: Excess finished goods or raw materials increasing storage and capital costs.
  - Idle time: Machines or labour not being used effectively.
  - Underutilization: Suboptimal use of production assets and resources.
- Consumer-side penalties:
  - Missed deliveries: Failure to meet customer delivery timelines.
  - Lead time uncertainty: Inconsistencies affecting planning reliability.
  - Dissatisfaction: Due to poor quality, stockouts, or pricing fluctuations.

This RL framework enables manufacturing organizations to make real-time, data-driven decisions that adapt to fluctuating demand and operational disruptions. By synchronizing factory output with the broader supply chain, it enhances robustness, reducing upstream risks by optimizing procurement and downstream risks by ensuring timely fulfilment and minimizing stockouts. Ultimately, this creates a resilient link between production and the market, helping to reduce waste, optimize resource usage, and enhance customer experience, regardless of the specific industry or product line.

IX. PERFORMANCE METRICS

A. Performance Gains Before vs After RL Deployment

B. Key Performance Indicators (KPIs) and Plots

Figure 1: Cumulative Reward Over Episodes

Shows how the agent's total reward increases over time, indicating learning progress. Higher values indicate better learning.

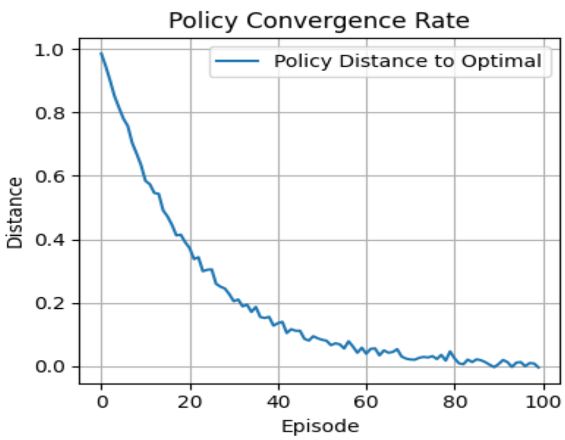

Figure 2: Policy Convergence Rate

Measures how quickly the agent's policy stabilizes, with lower values indicating convergence toward optimal behavior.

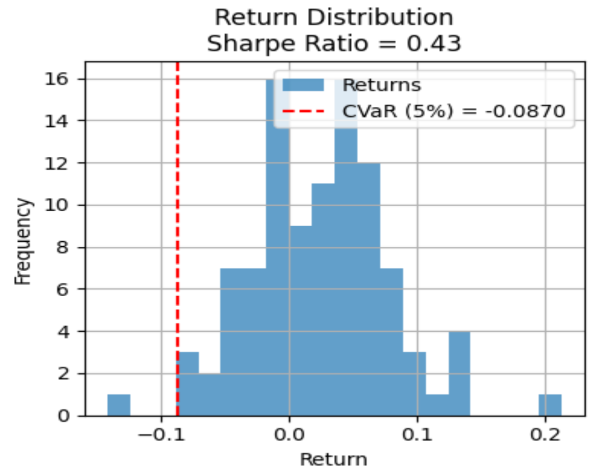

Figure 3: Risk-Adjusted Return

Sharpe Ratio: 0.43 — reflects moderate return relative to risk. A higher ratio reflects better risk-adjusted returns.

CVaR (5%): -0.0870 — captures expected loss in the worst 5% of scenarios. Lower CVaR indicates reduced exposure to extreme loss.

X. CONCLUSION

This paper successfully demonstrates that an Agentic Reinforcement Learning framework can effectively solve the complex challenge of balancing producer and consumer risks in manufacturing. By utilizing a comprehensive reward function that penalizes both operational inefficiencies and negative customer outcomes, the RL agent learns to make optimal, real-time decisions on production, pricing, and inventory. The resulting improvements in order fulfillment, cost reduction, and profitability highlight the framework's capacity to create more resilient, efficient, and adaptive manufacturing systems. This data-driven approach provides a powerful method for navigating the inherent trade-offs in a dynamic market, ultimately enhancing both business performance and customer satisfaction.

GLOSSARY

- LTPD (Lot Tolerance Percent Defective): Max defect rate a consumer tolerates in a lot.
- AQL (Acceptable Quality Level): Max defect rate considered acceptable as a process average.

- Consumer Risk (Type II Error, β-risk)

Consumer risk is the probability that a consumer will accept a product or service that does not meet the required quality standards. In statistical terms, it is the risk of a Type II error—failing to reject a bad product.

Hypothesis Testing:

- Null hypothesis H_0: The lot is good (defective rate ≤ AQL).
- Consumer risk $\beta = P(Accept\ Lot\ |Lot\ is\ Bad)$

- Producer Risk (Type I Error, $\propto$-risk)

Producer risk is the probability that a producer's good product or service is incorrectly rejected as not meeting quality standards. Statistically, it is the risk of a Type I error—rejecting a good product.

Hypothesis Testing:

- Alternative hypothesis H_1: The lot is bad (defective rate ≥ LTPD).

Producer risk $\propto = P(Reject\ Lot\ |Lot\ is\ Good)$

REFERENCE

[1] Meegle. (2024). Risk Management In Manufacturing.
[2] Pirani. (2025). How AI & Machine learning are shaping the future of Risk Management.
[3] Li, C., & Hong, Y. (2023). Deep Reinforcement Learning in Smart Manufacturing: A Review and Prospects.
[4] Wang, J., et al. (2020). A Survey of Reinforcement Learning for Optimization in Automation.
[5] Harry James, Kolawole Favour. (2024). Harnessing Reinforcement Learning for Optimized Dynamic Pricing in E-commerce.
[6] Shakya, S. (2023). Dynamic Pricing Using Reinforcement Learning.
[7] Schulman, J., et al. (2017). Proximal Policy Optimization Algorithms.
[8] Haarnoja, T., et al. (2018). Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor.
[9] Nievas, N., et al. (2023). Reinforcement Learning for Autonomous Process Control in Industry 4.0: Advantages and Challenges.
[10] Stavropoulos, P., et al. (2024). Reinforcement Learning-Based Optimization for Sustainable and Lean Production within the Context of Industry 4.0.
[11] Nair, A. (2024). Reinforcement Learning in Operations & Supply Chain Management. *Medium*.
[12] Oluwatumininu Anne Ajayi (2023). Optimizing inventory management through reinforcement learning.