**CMPE412- Computer Simulation**

**Project 2- Manufacturing**

**Shams Nassrullah**

**20201701075**

**Introduction**

In Project 2, I created a discrete-event simulation to simulate a high-volume manufacturing line for automotive parts. Initially, the focus was on a single product line to simplify the development process, with the option to extend the simulation to accommodate multiple product types with unique manufacturing requirements.

**Description of the System**

The production line consists of several stages: raw material handling, machining, assembly, quality control, and packaging. Each stage requires specific machinery and labor across multiple shifts.

**Goals**

The goal was to optimize throughput for a single product line, identify and address production bottlenecks, and analyze the effects of operational variables through scenario analysis. An additional challenge was to extend the simulation to handle multiple product types and analyze the added complexity and resource allocation challenges.

**Tools and Programming Requirements**

The project was carried out using Python and the SimPy library for discrete-event simulation. The fundamental functionalities included event scheduling, time progression, and state updates.

**System Modeling**

Raw Materials, Intermediate Parts, and Finished Products

The system consists of the following components:

- Raw Materials: The initial input for the manufacturing process.

- Intermediate Parts: Parts that are processed but not yet complete.

- Finished Products: The final output of the production line.

**Production Stages**

The production line is divided into several stages:

- Raw Material Handling: Loading raw materials into the system.

- Machining: Processing raw materials into intermediate parts.

- Assembly: Combining intermediate parts into finished products.

- Quality Control: Inspecting finished products for defects.

- Packaging: Packing the finished products for shipment.

Resources (Machines and Operators)

**Each stage requires specific resources:**

- Machines: Specialized equipment for each production stage.

- Operators: Workers assigned to operate machines and perform tasks.

Data Requirements

Processing Times

Establishing hypothetical but realistic processing times for each stage. For instance, raw material handling may take 5 minutes per unit, machining 10 minutes, assembly 15 minutes, quality control 8 minutes, and packaging 7 minutes.

**Failure Rates**

Determining typical failure rates and maintenance times for machines. For example, machines may fail every 500 hours of operation on average, requiring 2 hours of maintenance.

**Shift Patterns**

Configuring shift patterns and worker allocations. For example, a typical shift pattern might include three 8-hour shifts with a specific number of operators assigned to each stage.

Simulation Implementation

**Event Management**

Developing a system to manage events such as the start and end of processes, and resource breakdowns. This involves creating events for each significant occurrence in the simulation, such as the beginning of machining a part or the failure of a machine.

**Simulation Clock**

Implementing a simulation clock to move to the next event. This clock will track the passage of time within the simulation, ensuring events occur in the correct order.

**Modifiable Input Variables**

They were allowing users to modify input variables like machine count and shift timing. This feature provides flexibility to experiment with different configurations and their impacts on production efficiency.

**Experimentation and Analysis**

Basic scenarios establish baseline performance metrics for throughput and identify initial bottlenecks. Adjusting key variables, such as machine counts and processing times, reveals their impact on production efficiency. Bottlenecks are identified and addressed through data collection, analysis of machine utilization, downtime, and throughput rates.

**Conclusion**

This project created a discrete-event simulation for a high-volume automotive part manufacturing line, initially focusing on a single product type and potentially expanding to handle multiple types. Using Python and SimPy, the simulation modeled stages like raw material handling, machining, assembly, quality control, and packaging, to optimize throughput and identify bottlenecks. Through scenario analysis, it was found that adjusting machine configurations and optimizing maintenance schedules could significantly improve efficiency. Future enhancements include integrating real-time data for adaptive production management, aiming to continually enhance manufacturing processes.

**References**

* SimPy Documentation. Retrieved from <https://simpy.readthedocs.io/en/latest/>
* Microsoft. (2024). Visual Studio Code. Retrieved from <https://code.visualstudio.com/>
* Python Software Foundation. (2024). Python Programming Language. Retrieved from <https://www.python.org/>