Northeastern University
Fall 2025
IE 6300 Manufacturing Systems Design

Prof. Xiaoning Jin

Assembly Line Optimization for Battery Pack Production

Project Report

Group 7: Advait Jigarbhai Naik,
Ashutosh Kulkarni

1. Introduction

1.1. Background & Motivation

The global transition towards electric vehicles (EVs) and renewable energy systems has created unprecedented demand for battery packs. As the automotive sector moves towards electrification, battery manufacturers face mounting pressure to increase production capacity while maintaining stringent quality standards and cost competitiveness. Battery pack production represents a critical stage in this supply chain, where individual cells are transformed into complete, functional modules through a complex series of operations.

This project focuses on a 15-station assembly line dedicated to battery module production. The manufacturing system combines automated and manual operations, processing battery cells from initial quality testing through final assembly. The line begins with pallets of four individual cells (Stations 0-4) and transitions to stacked cell (module) configurations (Station 5 onward), culminating in fully assembled and tested battery modules ready for integration into larger battery systems.

The motivation for this optimization study stems from observed inefficiencies in the production line, particularly excessive queue times and unbalanced workload distribution across stations directly impacting the line's throughput. In an industry where production scalability and time-to-market are competitive differentiators, even modest improvements in line efficiency can translate to significant business value.

1.2. Problem Definition and Scope

The primary problem addressed in this project is the identification and mitigation of bottleneck stations that constrain throughput and create excessive queue times in the line. Initial observations revealed that certain stations experienced prolonged job queuing, while others operated with significant idle time. This imbalance indicates that variability in process times has led to imbalance in the line.

<u>Specific Problem Characteristics:</u>

- Variable Process Times: Manual stations exhibit comparatively more process variability, leading to unpredictable cycle times and queue formation.
- Sequential Bottlenecks: As one bottleneck is resolved, constraints shift downstream, requiring iterative analysis and targeted improvements.
- Mixed Automation Levels: The line combines fully automated stations with manual operations, creating complexity in balancing cycle times.

<u>Project Scope:</u>

This study encompasses the following elements:

- System Modelling: Development of a simulation model using Python's ‘SimPy’ library to accurately represent the 15-station assembly line, including:
  - Process times and variability for each station
  - Queue dynamics
  - Station capacities
  - CONWIP system using tokens
- Bottleneck Analysis: Systematic identification of constraint stations through:
  - Queue time analysis
  - Cycle time breakdown by station
- Optimization Strategies: Evaluation and implementation of targeted improvement initiatives, including:
  - Equipment upgrades
  - Capacity expansion
  - Process automation
- Performance Validation: Quantitative assessment of improvements through simulation-based metrics such as throughput, total cycle time, and average queue times.

1.3. Importance of the Problem in Industry and Research

Cost Competitiveness:
Battery pack cost represents a sizeable portion of total EV manufacturing costs. Assembly efficiency directly impacts per-unit production costs.

Quality and Traceability:
Battery packs are safety-critical components requiring complete traceability from individual cells through final assembly. Queue times and process variability can negatively impact quality control effectiveness, as longer residence times increase contamination risks and potential handling damage. Optimized flow reduces these quality risks while maintaining rigorous testing requirements.

Production Flexibility:
As battery technologies evolve (e.g., different cell chemistries, form factors, and configurations), assembly lines must adapt quickly. Understanding and optimizing current processes provides the foundation for efficient reconfiguration to accommodate future product variants.

Bottleneck Dynamics:
The project demonstrates the dynamic nature of manufacturing constraints, where resolving one bottleneck shifts limitations downstream.

Human-Machine Integration:
The assembly line's mix of fully automated stations and manual operations highlights ongoing challenges in manufacturing automation. Understanding how to balance automation investment with operator flexibility remains a critical research question as Industry 4.0 technologies mature.

Digital Twin Development:
The simulation model serves as a digital twin of the physical assembly line, enabling what-if analysis and predictive optimization without disrupting actual production. This approach represents best practices in modern manufacturing engineering and provides a framework for continuous improvement initiatives.

2. Methodology

2.1. Dataset

The dataset used for the production line is a synthetic dataset.

Table 1: Process Time Dataset

Station #	Sub Process #	Process Name	Sub Process Time (s)	Mean Process time (s)
1	1	OCV test	10	10
2	1	No replacement	5	
	1	1 replaced	25	
	1	2 replaced	50	
	1	3 replaced	75	
	1	4 replaced	100	conditional
3	1	Orientation Correction	25	25
4	1	Tape application	60	60
5	1	Cell stacking	60	60
6	1	Stack placement	30	
	2	Press	25	
	3	Post press placement	30	85
7	1	Scan	30	
	2	Place Sticker	30	60
8	1	Insulation Detection	45	45
9	1	Pole imaging	60	60
10	1	Initial scan	30	
	2	Laser cleaning	40	70
11	1	CCS installation	60	60
12	1	Initial scan	30	30
	2	Welding	40	70
13	1	Post weld cleaning	60	60
14	1	EOL testing	60	60
15	1	Off loading	45	45

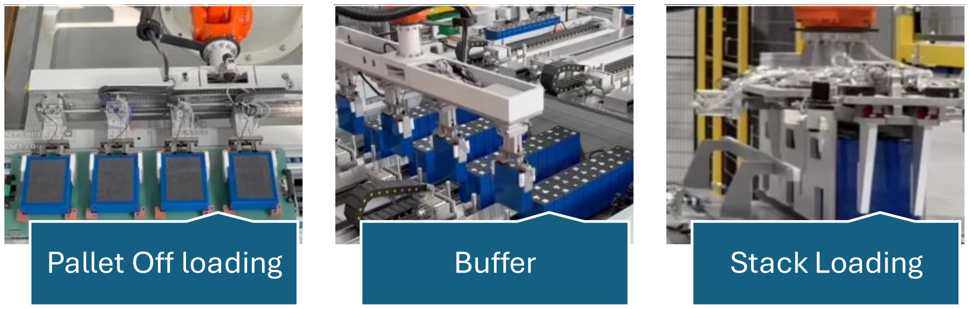
2.2. Experimental Setup

We have used the 'SimPy' library in Python to simulate the Production system.

The simulation is divided in 2 parts:
Part 1: Station 1 to 4 [processes on individual cells]
Part 2: Station 6 to 14 [Processes on 'module' configuration]

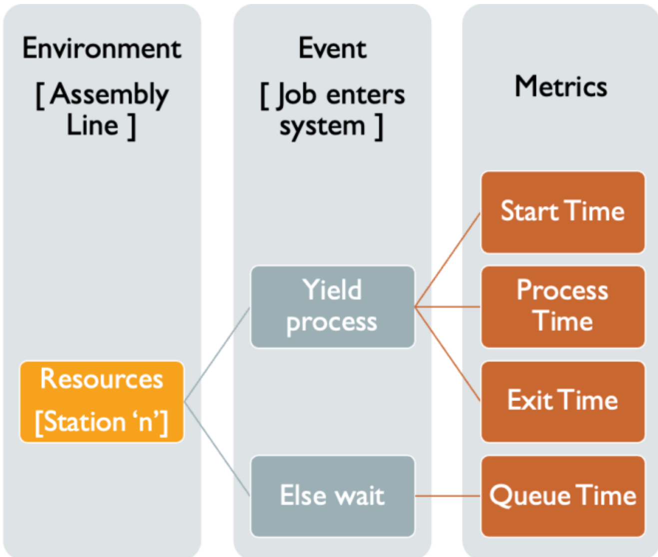
This is due to the nature of station 5 where individual cells [pallet with 4 cells] are off loaded and stacked in a module configuration [stack of 12 cells] in a buffer. Once the buffer reaches a specified capacity, the module is loaded at station 6. The module configuration is now one job unit that will be processed from station 6 onwards.

Figure 1: Station 5 process



<u>Simulation Program:</u>

Figure 2: Code Structure



<u>Environment:</u> Defined as a 'class' in python. This environment is our operating space i.e. our Assembly Line. Within the environment we create 'Resources' that are defined as a 'function' in python. These resources are our stations. Within these resources we define process times, and process variability.

<u>Event:</u> This is defined as a 'function' in python and utilized to send jobs to the environment. Within this event each job will enter the environment and request the process at each station. If the station is busy the job will be held in a queue. We also implement the WIP level control here. The job requires a free token to enter the environment.

<u>Metrics:</u> Metrics are included in the Environment and the Event to record when the job enters the environment, queue time if station is busy, process times at each station and when the job exits.

2.3. Parameter Choices and Justification

Process Variability:

We have chosen gamma distribution for Coefficient of Variability for processes times with values between 0.1 and 0.5. Automated stations have a lower CV, and manual stations have a relatively higher CV.

Number of Jobs:

For both the simulations, we have selected around 50 jobs to be sent in the system to get enough data for analysis of queue times and bottleneck progression after upgrades.

2.4. Assumptions, Constraints & Limitations

WIP level:

Since the production line is a rigid conveyor belt system, we have assumed that it can hold up to 3-4 jobs between 2 stations. Based on this value we have selected a WIP level of 5 for part 1 of the simulation and WIP level of 8 for part 2.

Arrival rate:

For Part 1 of the simulation, the loading time of the pallet at station 0 is taken as the interarrival time for station 1 and for part 2, unloading + loading time at station 5 is taken as the interarrival time for station 6.

Other Factors:

- For the simulation, we are utilizing 'random.seed(43)' function to replicate the same pattern of randomness to effectively compare the process improvement results.
- Factors like machine downtime, failure, Mean Time to Repair (MTTR) and Mean Time to Failure (MTTF) are not considered in the simulation. It is assumed that all stations are fully operational.
- The travel time between two stations which is decided by the speed of the conveyor belt is not considered in the simulation.

3. Problem & Results

3.1. Simulation Part 1

Station 2: Analysis:

Problem Description:

Station 2 replaces defective cells with qualified ones. This station is equipped with a gantry robot that has single cell gripper, i.e. it can pick up one cell at a time. The process time depends on the number of cells to be replaced.

Table 2: Station 2 Process Times

No replacement	5 s
1 replaced	25 s
2 replaced	50 s
3 replaced	75 s
4 replaced	100 s

This creates variability in the system leading to queue buildup. We observe following queue characteristics at station 2:

- No of jobs that experience queue: 30. Four jobs experience queue times greater than 100s.
- Maximum queue time recorded:167.53 s (job 54).

Figure 3: Station 2 queue time before upgrade

```
Job 54:
  CONWIP Wait: 1720.07 seconds
Station       Arrival    Exit        Process     Queue
-----------------------------------------------------
Station 1     3045.07    3053.86     8.79        0.00
Station 2     3053.86    3226.39     5.00        167.53
Station 3     3226.39    3273.65     32.35       14.91
Station 4     3273.65    3328.44     54.78       0.00
```

<u>Reference files:</u>
Code file: 'Code1.py' | Output File: 'Code1_output'.

<u>Upgrade:</u>
To reduce the variability, we suggest the following.

Replace the 'single cell gripper' of the gantry robot with a '4 cell gripper' that can utilize data from Station 1 (OCV test) to replace the required defective cells. This will reduce variability in process times to [5 or 25] s. Another option is to replace Station 2 with a Kuka robot (similar to Station 0) with a '4 cell gripper'. The former upgrade option is more viable as integration of the Kuka robot will be expensive and space requirements constraints will have to be considered.

<u>Upgrade Result:</u>

After reducing variability, we observe significant reduction in queue times. Please note that even if we see queue times as zero, we do not consider the queue as eliminated as the results are affected by factors like no of jobs, selected WIP level, interarrival times and CV for process times and the random.seed function.

But due to this upgrade, we see significant queue times at station 4. Multiple jobs experience a queue of more than 200 seconds. Jobs 14 and 15 experience maximum queue times.

Figure 4: Station 2 queue times after upgrade-1

```
Job 14:
  CONWIP Wait: 246.47 seconds
Station       Arrival    Exit        Process     Queue
-----------------------------------------------------
Station 1     571.47     582.80      11.33       0.00
Station 2     582.80     587.80      5.00        0.00
Station 3     587.80     608.08      20.28       0.00
Station 4     608.08     914.48      34.41       271.98
```

Figure 5: Station 2 queue times after upgrade-2

```
Job 15:
  CONWIP Wait: 255.10 seconds
Station      Arrival    Exit       Process    Queue
----------------------------------------------------
Station 1    605.10     613.48     8.38       0.00
Station 2    613.48     618.48     5.00       0.00
Station 3    618.48     650.92     32.44      0.00
Station 4    650.92     933.19     18.72      263.56
```

<u>Reference Files:</u>
Code file: 'Code2.py' | Output File: 'Code2_output'.

<u>Station 4 Analysis:</u>

<u>Problem Description:</u>

This is a manual station with a single operator and mean process time of 60 seconds. As the pallet arrives, the operator places a double-sided tape on the 4 cells with the help of an alignment tool. We have considered comparatively high process CV for this station and as the alignment is important there a chance of rework too.

We consider 2 cases where we set a target process time to reduce the queue times.

<u>Case 1:</u>

Reduce mean process time to 30 seconds. This can be achieved by increasing the number of operators on the station, specific training or removing the need of the alignment tool.

<u>Upgrade Result [Case 1]:</u>

We observe that 45 jobs face a queue at station 4 and the maximum queue time is 119.11 seconds (job 14). Approximately 50% reduction has been achieved and except job 14, other jobs experienced queue times of less than 100 seconds.

Figure 6: Station 4 upgrade result, Case 1

```
Job 14:
  CONWIP Wait: 29.15 seconds
Station      Arrival    Exit       Process    Queue
----------------------------------------------------
Station 1    354.15     366.35     12.20      0.00
Station 2    366.35     371.35     5.00       0.00
Station 3    371.35     400.20     20.24      8.61
Station 4    400.20     532.84     13.53      119.11
```

<u>Reference Files:</u>
Code File: 'Code3.py' | Output File: 'Code3_output'

Case 2:

We install a parallel station which increases the capacity to 2. The job waiting in queue will be sent to the next available station. Compared to case 1, this is relatively expensive and difficult to integrate due to the conveyor belt upgrade and consideration of space constraints. This case has an advantage if one station is down and further chance of improvement if two or more operators are assigned to each station.

Upgrade Result [Case 2]:

We observe that 42 jobs experience queue times at station 4 and the maximum queue time is 71.18 seconds (job 37). Approximately 70% reduction is achieved and except job 37, other jobs experienced queue times of less than 70 seconds. Compared to Case 1, approximately 40% reduction has been achieved in the maximum queue time.
After the station 4 upgrade cases, we can observe queue times at station 2 again. No significant difference is observed between Case 1 & 2 at station 2 queue times due to the upgrade at station 4.

Figure 7: Station 4 upgrade result, Case 2

```
Job 37:
  CONWIP Wait: 72.54 seconds
Station      Arrival    Exit       Process    Queue
--------------------------------------------------
Station 1    972.54     987.60     8.29       6.78
Station 2    987.60     993.41     5.00       0.81
Station 3    993.41     1047.39    20.25      33.73
Station 4    1047.39    1224.76    106.19     71.18
```

Reference Files:

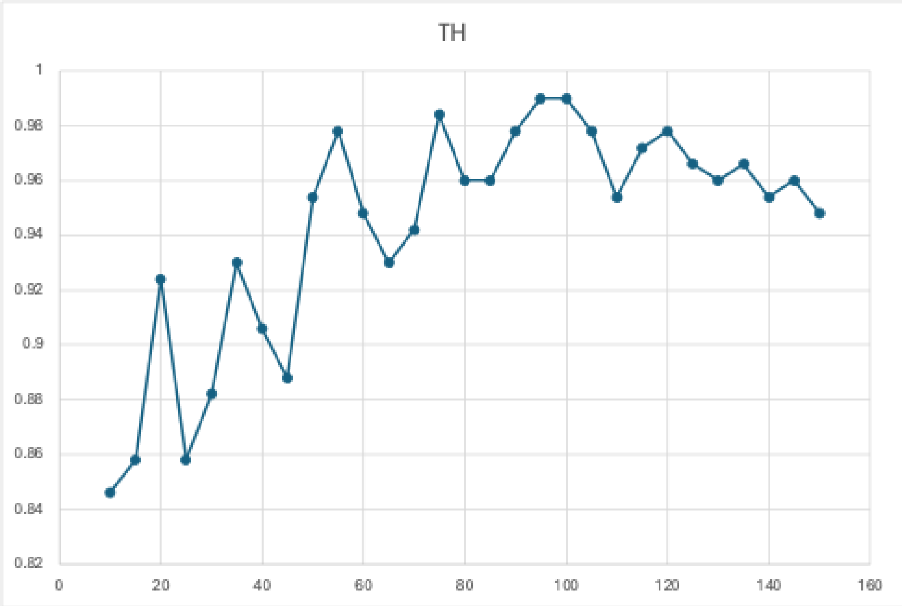
Code File: ‘Code4.py’ | Output File: ‘Code4_output’

Overall Case 2 is better as we have parallel stations, lesser queue times and similar queue times at station 2.

Finally, we compare the throughput achieved after upgrades. Below is the plot of Throughput vs number of jobs processed by the system.

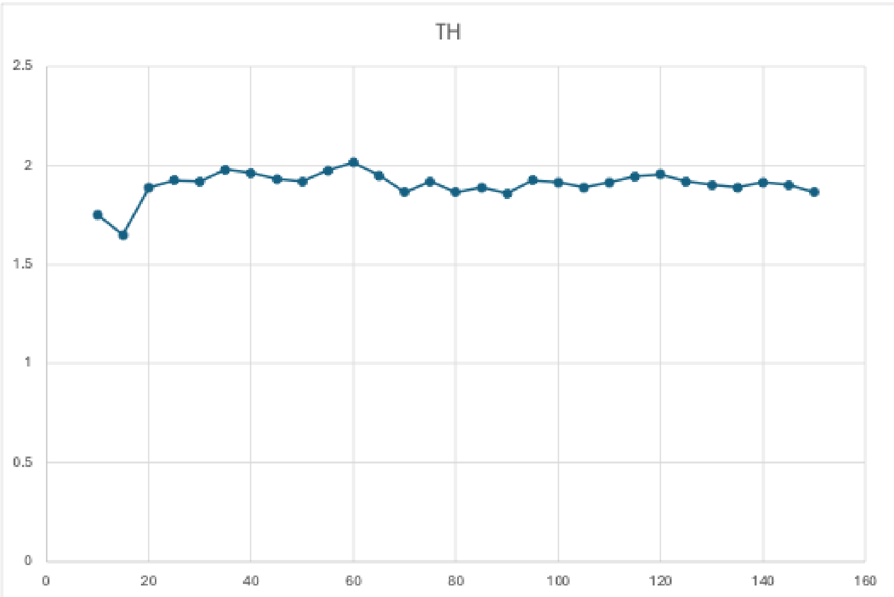
Before Upgrade:

Figure 8: Sim 1, Throughput before upgrade



After Upgrade:

Figure 9: Sim 1, Throughput after upgrade



Observation:

The throughput before upgrade is highly variable and dependent on the number of jobs sent to the system. After upgrade we can observe increased throughput and significantly more streamlined performance. The upgrades make the system more stable irrespective of the number of jobs to be processed.

3.2. Simulation Part 2

Station 6: Analysis:

Problem Description:

During the initial system warm-up and job arrivals, a significant queue developed upstream of Station 6, indicating its role as a primary bottleneck in the process due to high process time. The maximum observed queue time at this station reached 121 seconds for Job 35, while the average queue time at Station 6 was approximately 54 seconds.

Reference Files:

Code File: 'Code_5.py' | Output File: 'Result_5'

Figure 10: Station 6 queue times before upgrade

Job 35:

Station	Arrival	Exit	Process	Queue	Cycle
Station 6	2862.33	3032.32	48.78	121.21	169.99
Station 7	3032.32	3091.26	44.33	14.61	58.94
Station 8	3091.26	3156.39	65.13	0.00	65.13
Station 9	3156.39	3216.84	60.45	0.00	60.45
Station 10	3216.84	3278.76	55.28	6.64	61.92
Station 11	3278.76	3340.64	61.88	0.00	61.88
Station 12	3340.64	3439.72	63.71	35.37	99.08
Station 13	3439.72	3472.90	33.18	0.00	33.18
Station 14	3472.90	3543.03	49.39	20.75	70.13
Station 15	3543.03	3582.24	39.21	0.00	39.21

The simulation results identified Station 6 as the primary bottleneck, with a large and persistent queue and delays affecting approximately 30 jobs. Because Station 6 operates as a semi-manual station, its capacity could be increased at relatively low cost by adding a parallel setup or an additional operator.

Station 6 Upgrade:

Given the high queue times and minimal investment required, doubling the station's capacity was selected as the improvement strategy. Following this enhancement, queueing at Station 6 decreased sharply: the average queue time dropped from roughly 54 seconds to 0.9 seconds, and the maximum queue time reduced from 121 seconds to 29 seconds. This improvement led to a significant enhancement in overall line flow, minimizing waiting times and alleviating upstream congestion.

Figure 11: Station 6 queue time after upgrade

Job 37:

Station	Arrival	Exit	Process	Queue	Cycle
Station 6	2872.37	2939.73	67.36	0.00	67.36
Station 7	2939.73	3062.08	57.42	64.92	122.35
Station 8	3062.08	3108.98	45.92	0.98	46.90
Station 9	3108.98	3219.56	54.73	55.84	110.58
Station 10	3219.56	3325.98	88.65	17.77	106.42
Station 11	3325.98	3387.00	61.02	0.00	61.02
Station 12	3387.00	3447.62	60.62	0.00	60.62
Station 13	3447.62	3503.48	37.11	18.75	55.86
Station 14	3503.48	3589.72	57.23	29.02	86.24
Station 15	3589.72	3620.18	26.25	4.21	30.46

Reference Files:

Code File: 'Code_6.py' | Output File: 'Result_6'

Station 7: Analysis:

Bottleneck Shift and Improvement Proposal for Station 7:

After increasing the capacity of Station 6, the bottleneck shifted to Station 7, where a notable queue developed, affecting a total of 14 jobs. Station 7 operates as a fully manual station with a process time of 64 seconds (Fig. 11), making it the new constraint in the production flow. The delays primarily stem from manual QR code scanning and the placement of the module identification sticker.

To address this issue, Station 7 can be converted into a semi-automated station, in which an automated scanning system handles the QR code reading while the operator focuses solely on sticker placement. This change is expected to reduce the station's process time from 60 seconds to approximately 45 seconds, significantly improving throughput and reducing queue buildup.

Figure 12: Station 7 queue time after upgrade

Job 39:

Station	Arrival	Exit	Process	Queue	Cycle
Station 6	2902.08	2970.62	68.55	0.00	68.55
Station 7	2970.62	3004.26	33.64	0.00	33.64
Station 8	3004.26	3037.63	33.36	0.00	33.36
Station 9	3037.63	3151.03	67.12	46.29	113.41
Station 10	3151.03	3292.06	59.86	81.17	141.03
Station 11	3292.06	3388.31	69.56	26.69	96.25
Station 12	3388.31	3475.08	70.87	15.89	86.76
Station 13	3475.08	3553.10	78.03	0.00	78.03
Station 14	3553.10	3622.48	69.38	0.00	69.38
Station 15	3622.48	3655.74	33.26	0.00	33.26

Figure 13: Station 7 queue time after upgrade

Job 48:

Station	Arrival	Exit	Process	Queue	Cycle
Station 6	3726.52	3823.79	97.28	0.00	97.28
Station 7	3823.79	3859.92	36.12	0.00	36.12
Station 8	3859.92	3899.99	40.07	0.00	40.07
Station 9	3899.99	3956.54	56.55	0.00	56.55
Station 10	3956.54	4006.37	49.83	0.00	49.83
Station 11	4006.37	4041.67	33.45	1.85	35.30
Station 12	4041.67	4207.13	75.25	90.21	165.46
Station 13	4207.13	4267.29	60.15	0.00	60.15
Station 14	4267.29	4313.56	46.27	0.00	46.27
Station 15	4313.56	4350.04	36.49	0.00	36.49

Reference Files:

Code File: 'Code_7.py' | Output File: 'Result_7'

Station 10 & 12 Analysis:

Following the Station 7 capacity enhancement, the simulation reveals load shifting to Stations 10 (Laser Surface Cleaning) and 12 (Laser Welding), with queue times increasing significantly. These fully automated stations present unique optimization challenges that preclude conventional solutions. Station 10's laser ablation process requires precise power control and systematic cleaning patterns to ensure optimal surface preparation, while Station 12's coordinate-based welding system must maintain specific penetration depths and heat-affected zones to guarantee joint integrity both operating at optimized cycles that cannot be reduced without compromising quality.

Furthermore, capacity expansion through equipment duplication would require substantial capital investment, complex system integration including synchronized control protocols and shared coordinate data management, and extensive facility infrastructure modifications. Given these technical and economic constraints, the optimal solution involves implementing strategic buffer zones of 3-4 pallets before each station, which will effectively decouple these bottlenecks from upstream variability while requiring minimal investment and no compromise to critical process parameters. This buffer strategy is expected to provide the flexibility to accommodate production variations without the complexity and cost of automated system duplication.

Results after double the capacity of station 6 and optimizing process time of station 7:

Table 3: Simulation part 2 results

Metric	Before (initial sim)	After (ST6 cap ×2 + ST7 optimized)	Change
Throughput (jobs/s)	0.0103	0.0111	+7.8%
Total cycle time (CT, s)	703.22	642.30	-8.7%
Simulation time (s)	4843.34	4512.69	-6.8%
Sum of avg queue (all stations)	97.18	54.83	-43.6%

4. Discussions

4.1. Bottleneck Evolution and System Behaviour

In Part 1 of the simulation, Station 2 emerged as the first major bottleneck due to high process time variability arising from conditional cell replacement logic. Reducing variability through a four-cell gripper significantly decreased queue times, validating that variability not only mean process time plays a critical role in flow performance. However, this improvement immediately caused the bottleneck to shift to Station 4, a manual task with high cycle time variability.

Parallelization at Station 4 was found to reduce queue times by approximately 70%, outperforming simple cycle-time reduction. Yet this again caused upstream queue rebalancing at Station 2, illustrating that the line is highly sensitive to WIP levels, arrival rates, and randomness in cycle times.

4.2. Impact of Capacity Changes in Part 2

In Part 2, doubling the capacity at Station 6 resulted in dramatic queue reduction and improved overall line flow. This station, being semi-manual, offered a cost-effective improvement path. Once the bottleneck at Station 6 was alleviated, the constraint migrated downstream to Station 7. Optimizing the manual activities at Station 7 improved throughput and reduced delays, further increasing system stability.

However, once these human-dependent stations were optimized, the constraint migrated to Stations 10 and 12 fully automated, precision laser operations. Unlike earlier bottlenecks, these stations are **not candidates for simple process-time reductions** due to strict quality requirements and high capital requirement.

4.3. Influence of Variability, WIP, and Queue Dynamics

Across both simulations, queue formation was heavily influenced by:

- Process variability (CV values)
- WIP constraints imposed by conveyor capacity
- Interarrival variability driven by upstream processes
- Sequential dependence of station outputs

The results validate classical flow principles: high variability and high utilization produce long queues. By reducing variability (Station 2), adding capacity (Stations 4 and 6), and decoupling rigid automated processes with buffers (Stations 10 and 12), the system began to operate nearer to a balanced flow condition.

4.4. System-Level Performance Improvements

These outcomes demonstrate that simulation-driven optimization provides quantifiable improvements while avoiding excessive capital expenditure.

Overall, the study shows that a structured simulation approach allows decision-makers to pinpoint high-impact interventions and avoid unnecessary investment, particularly in technologically advanced operations like battery pack manufacturing.

5. Conclusion

This project used simulation to better understand how the battery pack assembly line behaves under real operating conditions. By identifying where delays were forming and why we were able to test targeted improvements such as reducing process variability, adding capacity, and streamlining manual tasks. These changes led to meaningful gains in throughput, shorter cycle times, and smoother flow across the line.

A key insight from the study is that fixing one bottleneck often reveals the next, especially in a system with both manual and automated processes. For the more rigid automated stations, thoughtful buffer placement proved more practical than equipment changes. Overall, the work shows how a digital twin can support smarter, lower-risk decisions and help the production line run more efficiently and consistently.

6. References

- Factory Physics, Third Edition
- Production Line Video
- https://about.bnef.com/insights/clean-transport/electric-vehicle-outlook/#overview
- https://realpython.com/simpy-simulating-with-python/
- https://www.porffor.com/newsinfo/1062410.html
- https://flex-lineautomation.com/industry-insights/buffer-conveyor-101/

7. Appendix

Please refer to ‘Appendix’ folder in supplemental files.

- Code Files: ‘sim_part_1’ and ‘sim_part_2’
- Assembly Line Detailed Description: ‘Assembly_Line’