Production, Robotics, and Integration Software for Manufacturing & Management Center

12-13-2020

KRS Flow Junction Case Study and Simulation

Byeongmok (Brad) Kim

Madhi Moghaddam

Maitreya Sreeram

Shimon Y. Nof

Comments are invited and should be directed to the author at the address listed below. Please do not reproduce in any way without the permission of the authors. A complete listing of reports may be obtained by contacting the PRISM Center, Grissom Hall, Purdue University, 315 N. Grant St., West Lafayette, IN, 47907-2023. Also refer to [PRISM Center (purdue.edu)]

KRS Flow Junction Case Study and Simulation

Kim, B., Moghaddam, M., Sreeram, M., and Nof, S.Y.

PRISM Center and School of IE, Purdue University

PRISM Research Memorandum No. 2020-P1

December 13, 2020

PRISM Lab/Purdue

PRISM Center
Production, Robotics, and Integration
Software for Mfg. & Management

“Knowledge through information; Wisdom through collaboration”

PRISM Center
Grissom Hall
315 N. Grant Street
Purdue University
West Lafayette, IN 47907-2023

KRS Flow Junction Case Study and Simulation

Kim, Byeongmok (Brad), Moghaddam, Mahdi, Sreeram, Maitreya, and Nof, Shimon Y.

December 13, 2020

1. Objective

The purpose of this report is to study fundamental workflow logic for cyber collaborative future work and factories (C2F) as part of our NSF Grant 1839971: Collaborative Research: Pre-Skilling Workers, Understanding Labor Force Implications and Designing Future Factory Human-Robot Workflows Using Physical Simulation Platform.

2. The Flow Junction as a fundamental manufacturing, production, and service workflow logic [This Section 2 is taken from [25] Sreeram and Nof, 2020, revised as PRISM Research Memo 2022-P2 in July 2022.]

With consumers demanding from suppliers a wider range of products, and quicker and more accurate delivery capabilities, many manufacturing, production, and service systems often employ as part of their material flow process- “flow junctions,” or “flow control stations.”

<u>Flow Junction definition</u>: In a Flow Junction, different parts or components arrive from multiple sources and are grouped and sorted based on common attributes: type of product, storage requirements, priority of order, destination in process, shipping or distribution plan, etc.

The goal of utilizing a Flow Junction is to improve flexibility and cost/time effectiveness. Examples of such junctions include sorting and merging stations in:

- Transportation (e.g., airports [1], shipping and distribution hubs, cross docking depots);
- Food and beverage industry [2];
- Manufacturing and logistics [3];
- Construction parts and materials;
- Automated storage and retrieval systems;
- Healthcare and medical supply chains;
- Test, maintenance and repair; and more.

The general flow logic of a Flow Junction is shown in Figure 1.

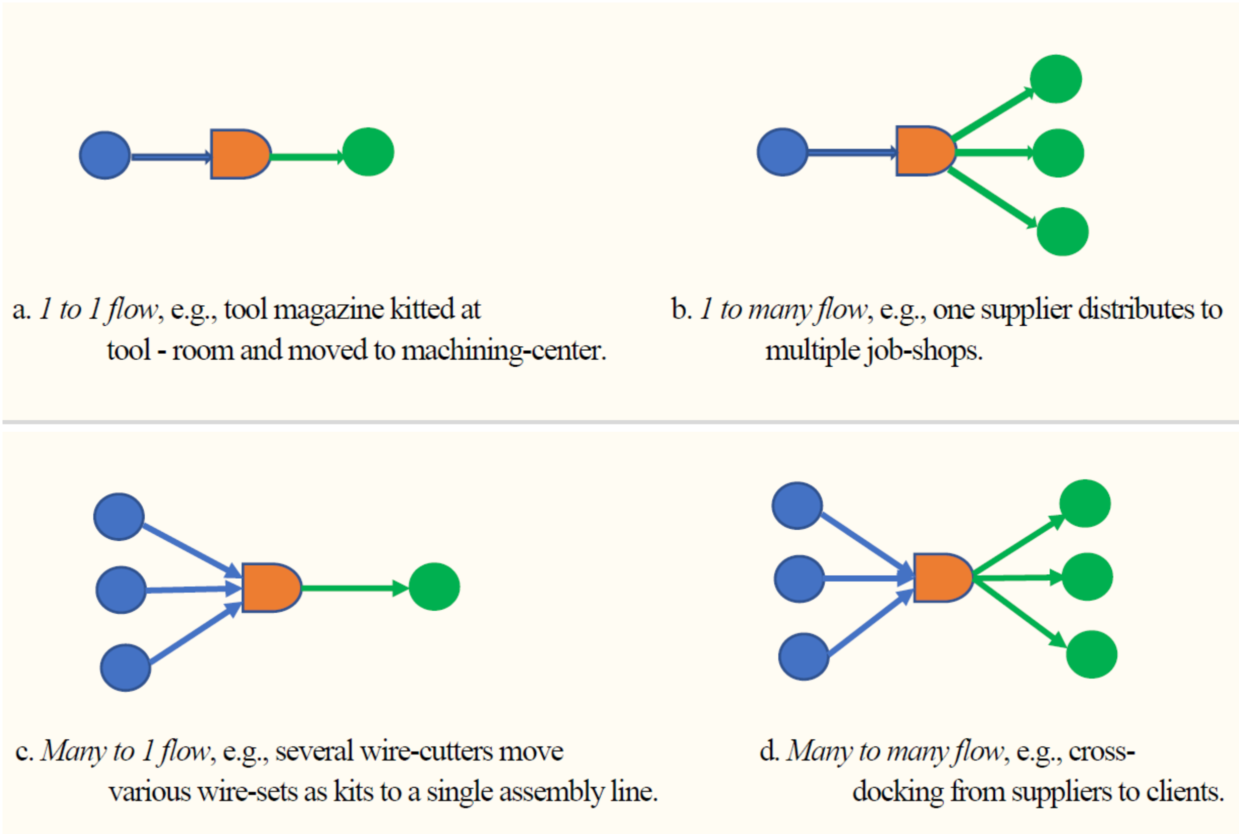

Figure 1. Four Typical flows. Flow junction -- Source-- Destination --
[From Sreeram and Nof, 2020]

The current scope of this research considers the modeling, improvement and optimization of kitting stations for the electrical harness industry (KRS Case Study, as an example) [4, 5, 6].

Parts kitting (Figure 2) is a frequently used method to deliver pre-organized and often pre-inspected parts to assembly lines/workstations. Kitting policies usually involve 1) grouping all parts required to assemble one complete unit of end product or sub-assembly; and 2) placing these grouped parts into one or more containers.

The main advantages of kitting: Material flow downstream is simplified, errors are prevented or eliminated early; inventories, space requirements and holding costs are reduced. These advantages, however, come at the additional expense of supporting the additional workforce and automation required to perform the kitting operations, and additional cost involved with errors induced from this additional workforce [6].

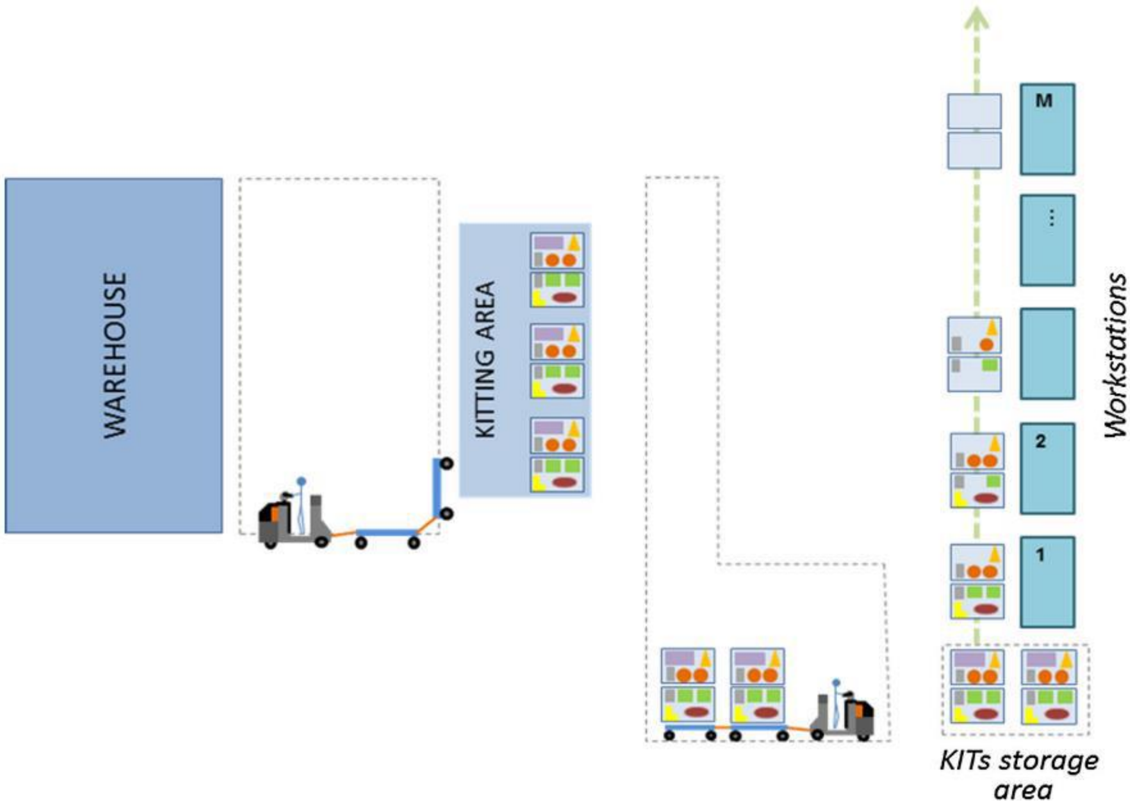

Figure 2: Kitting system scheme ([6])

Kitting tasks, taxonomy and future work advances

Tasks involved in the kitting stations need to be performed in a procedural, pre-determined and logical manner [7] , to ensure the process and involved tasks are simple and can be performed quickly. Small to medium enterprises depend on manual labor for kitting operations, and given the repetitiveness of these tasks, human error can be a prominent cause of errors and conflicts in material handling [8]. Any error arising from the kitting station can be a potential conflict for further steps downstream; additional correction costs, economic losses can be incurred if it is necessary to either correct the steps within the scope of the kitting policy or reduce the overall probability of these errors from arising [8]. Human operators perform a series of physical (picking, placing, traversing, storing, and scanning) and cognitive (decision-making, part-checking, and scan verification) – thus it becomes imperative to create a taxonomy that can address the following requirements:

1. Determine different types of errors that can arise and their classification (cognitive, physical, prior error, etc.);

2. Map these errors to different logical steps of the kitting workflow based activities;
3. Quantify the cost impact of these errors taking into consideration probability of these errors being detected, and cascading impact downstream.

The taxonomy should enable material planners to identify the current gaps (of skill, operation, workflow) and initiate corrective measures to reduce the occurrence of errors and conflicts [8, 9]. Some examples of corrective measures include:

1. Streamlined and simplified operations: By standardizing the involved process and ensuring minimal levels of specialization, human-induced errors can be minimized by reducing the cognitive load [10].
2. IoT/RFID based solutions: IoT/IoS based design has been shown to provide preventive maintenance of industrial systems in real-time [11]–[13]. While SMEs often rely on barcodes and hand-held scanners for material handling, the usage of RFID has gathered momentum within various supply chains [12], [14], [15]. We propose an IoT based kitting system that can reduce errors which originate due to cognitive actions (e.g., decisions such as acknowledging when the kit tray is complete and should be sent to temporary storage, when a part is placed in the wrong tray, or vice versa).
3. Advanced AR/MR dynamic and responsive training of human operators: Adaptive intelligent tutoring systems [16]–[18] can be used to improve cognitive knowledge and skill sharing, retention, error and conflict reduction, improve and minimize costs and delays for cognitive and physical task training. We can envision the use of (Adaptutor) to improve the task performance for physical tasks (and later, also cognitive tasks) within the kitting workflow, since they also involve local, body-coordinated and spatial tasks as well.

<u>HUB-CI for workflow optimization and harmonization:</u> To evaluate the usefulness of these improvements, we consider the development of a discrete-event simulator based on HUB-CI [19], [20] logic and services. This simulator takes a modular approach to integrate these improvements, and different levels of collaboration can be evaluated to determine the optimal operating parameters for such a system. HUB-CI is required to ensure that the proposed improvements are integrated into the workflow in a harmonized manner. We capture relevant simulated performance metrics such as operator error rate, average operation cost, time, and penalty costs to validate the different levels of HUB-CI. HUB-CI simulator is envisioned as part of our planned PRSP.

3. Learning Curve Models for task times and error rates

Evaluation of improved processing time

The learning curve model (JGLCM; Jaber et al., 2013) assumes that the operator learns with repetitive actions, which shortens the processing time. Therefore, we can assume that workers can learn from the avatar because it educates workers by demonstration and workers learn by following the avatar's demonstration.

Thus, we can use JGLCM (Jaber et al., 2013) to calculate workers' improved processing time by using their initial processing time and learning rates.

$$T_n = \mathrm{x}T_1 n^{-b_c} + (1-\mathrm{x})T_1 n^{-b_m} = T_1[x(n^{-b_c} - n^{-b_m}) + n^{-b_m}], \tag{1}$$

Where T_n is the time to produce *n* th unit, T_1 is the time to produce the first unit, *n* is the cumulative number of repetitions, and *b* is the learning exponent. The learning exponent is calculated as $b = -\log(\theta)/\log(2)$, where θ is the learning rate, *x* is a percentage of splitting T_1 into two components, cognitive and motor.

Evaluation of improved error rate

Humans are forced to generate errors, which can only be suppressed by learning (Duffey et al., 2008). Therefore, we can calculate the workers' error rates by using their learning rates and experiences (working hours).

$$\lambda(\varepsilon) = \lambda_m + (\lambda_0 - \lambda_m)exp - k(\varepsilon - \varepsilon_0), \tag{2}$$

Where $\lambda(\varepsilon)$ is error (or failure) rate, *k* is the learning rate constant of proportionality, positive for learning and negative for forgetting, ε is an experience, ε_0 is initial experience, for a novice or a new technology there is commonly no initial experience, so $\varepsilon_0 \to 0$, is initial error rate, and λ_m is the final or irreducible minimum error rate.

As a result, we can evaluate the improvement of workers' skill proficiency due to avatar and IoT learning using the learning curve model and human error model mentioned above. The evaluation metrics will be processing time and error rate, and the effectiveness of avatar learning can be verified through analysis of changes in evaluation metrics before and after avatar learning.

4. Simulation Description

The simulation of the Flow Juncture is developed using Python programming language and object-oriented programming. It uses a discrete-event based approach towards workflow simulation. Illustrated in Figure 3, the workflow of the simulation consists of four consecutive sections. It starts with parts of products arriving from wire-cutting in section “Scan and Verify”, where the worker scans the parts and verifies their health. A healthy part will continue to “Deposit” section in which it is deposited to its corresponding tray with other parts of the product. In section “Move to storage”, the worker moves full trays to storage. A tray with all of the parts required for a product is considered full. The worker selects a location for the full trays in the final section. What follows are the two elements of the workflow logic.

Worker

The worker at the Flow Junction performs five actions, which are represented by five functions, scan, verify, deposit in tray, move to storage and put in location. The actions of the worker are enumerated below as in Figure 3:

1. *Scan*: the worker scans the part that arrives at the Flow Junction.
2. *Verify*: the worker verifies the health of the product.
3. *Deposit in tray*: the worker selects a tray and deposits the part in that tray.
4. *Move to storage*: the selected tray is moved to the storage if it is full.
5. *Put in location*: the worker selects a location for the tray.

Simulation Logic

The yellow symbols in Figure 3 show those validity checks in the simulation. If a decision is correct the simulation moves on, otherwise the cost of the error is calculated and the loop starts over. These validity checks can be identified by their numbers in each logical section of Figure 3. They are defined as follows:

1. *[2.1/2.2] Verification check*: if the part is faulty and the worker correctly identifies it, the error is from the Wire-cutting. A rework will be requested to fix the faulty part. If the part is not faulty and the worker identifies it correctly, the simulation continues. If the worker's decision is the opposite of the part's health, the worker has made an E1 error.

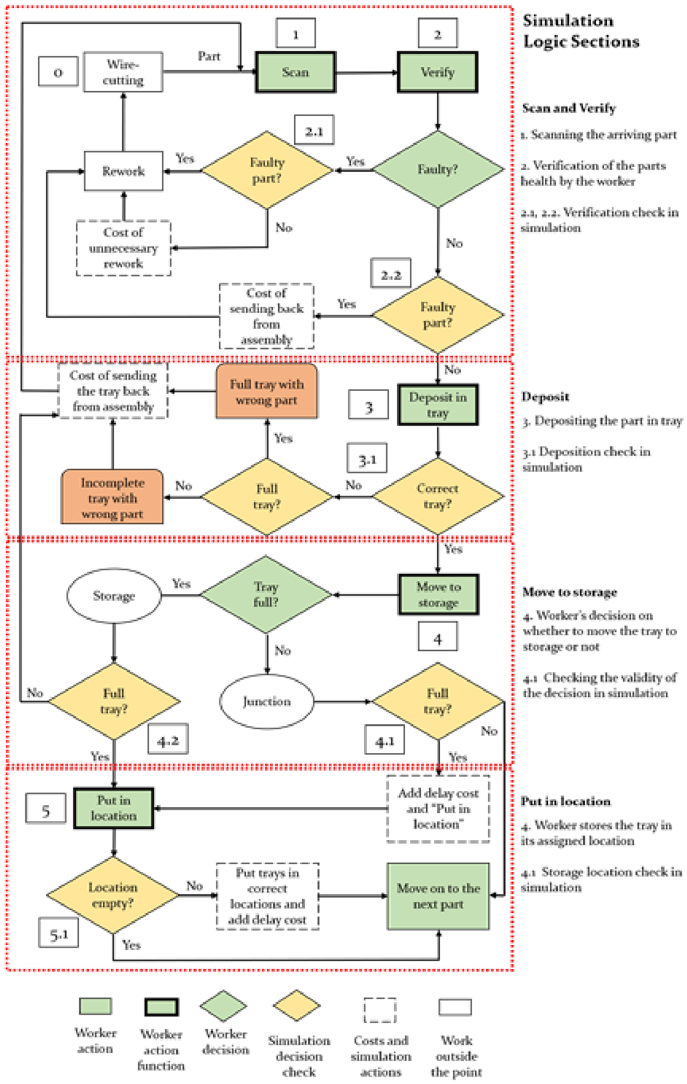

Figure 3: Workflow of the simulation

2. *[3.1] Deposition check*: if the tray that the worker selects is correct, the simulation continues. Otherwise, the worker has put the part in the wrong tray. In that case he/she has made an E2 error. The selected tray and part will eventually be sent to assembly or wire-set, where they will be sent back to the Flow Junction.
3. *[4.1/4.2] Moving to storage check*: the correct decision is to keep the incomplete tray at the Flow Junction, or take the complete tray to storage. If the worker

sends an incomplete tray to storage, error E4 has been made and the tray will be returned back to the Flow Junction. If he/she keeps a complete tray at the Flow Junction, an unnecessary delay has been added. This is error E5, which is caused by the worker.

4. *[5.1] Location selection check*: the location that the worker selects is either empty or occupied. In case of the former, the tray will be placed at the selected location, but the latter means E6 has occurred. When the location is not empty, it means either the worker wants to put the current tray in the wrong location or he/she has done it for a previous tray. In either case, a delay will be added to put both trays in their correct locations.

IoT and Avatar

IoT has been suggested as an improvement for the Flow Junction process. Sensors and miniature LED light bulbs can visually help workers in depositing parts, moving full trays to storage and selecting the correct storage location. This will significantly reduce the error rate and time of operation. The effects of IoT on workers has been simulated using Equation 2, through a function that can be applied to a worker object (referring to the object-oriented programming method). This will reduce the initial error rates of the worker in different tasks.

Avatar (or Adaptutor), is an AR based tutoring system for physical tasks in machine-based environments. As a proposed improvement for the Flow Junction process, we assume that workers are initiated to the AR tutoring for a fixed number of iterations, and then reintroduced into the workflow. The objective of avatar is to improve the task performance of operators in an adaptive manner, responding to individual learning styles and skill levels. Hence, the effect of avatar on task performance is simulated using Equation 1, which uses the worker skill level, their cognitive and physical task learning abilities and the number of iterations of avatar they require. Avatar is assumed to be applicable for each of the tasks (scanning, verification, deposit and location storage). We do not consider the cost of tutoring, worker time-outs due to the tutoring in this simulation.

Assumptions

In the simulation of the workflow, once an error occurs, its costs are calculated and added while the parts involved in that error will be sent back to the initial step of the simulation. The probability of two or more consecutive errors for the same part or the same tray will be so small that we can assume that it will not occur. Therefore, the tray containing an erroneous part (faulty or incorrect part) will be sent to the assembly, where it would get sent back to the Flow Juncture.

The other assumptions in the simulation are regarding the initial attributes of the worker, products and parts. However, these values can be easily modified based on the requirements. Also, it was assumed that there are a fixed number of parts and products in each run of the simulation, which will end once all the parts have been sent to the assembly in the correct trays.

Performance Metrics

The goal of the simulation is to quantify and measure the performance of workers in the Flow Junction and the effects of Avatar and IoT on their performance. To that end, the workers and Flow Junction have been simulated based on real-world attributes, some of which are designed as inputs while the rest are derived from them. The inputs include skill level of the worker, cognitive and motor learning ability coefficients, and initial experience for the worker, and irreducible error rate, base error rate, and base task time for the Flow Junction. Note that the worker inputs can change in multiple runs of the simulation as we create different workers, while the Flow Junction inputs will remain the same.

To achieve the goals of the simulation, i.e., four types of performance metrics have been designed as outputs. (i) The first performance metric is the cumulative time of operation (scan, verification, deposit, storage location) per run. (ii) The second is the number of errors per task and their cumulation. (iii) Using the second type and the total number of decisions the worker makes, the total error rate per run is calculated as the third performance metric. (iv) The final type of performance metric is the costs per run which is calculated according to the previous three.

The costs are threefold: cost of error, cost of conflict and cost of operation. Each error has its own cost, depending on the number of products it affects and the stage in which it occurs. Some of the errors will also cause conflicts in later stages. For instance, if the worker sends a faulty part to assembly, a conflict will occur, but if he sends a healthy part back to wire-cutting, only the cost of error, which is unnecessary rework will be added. The cost of operation is largely influenced by the time the worker takes to finish a task. It is important to note that every error that the worker makes will result in rework, which in turn will increase the total time spent on tasks and as a consequence, cost of operation.

Generated Data-frame

In the beginning of the simulation a data-frame is generated with the initial attributes of the worker and his performance on one simulation run as its columns. The attributes include learning abilities, error rates, required time and initial experience. The columns related performance include the time spent on each task and the number of errors. After each run the settings and results of the simulation will be recorded in the data-frame. Figure 4 shows a snippet of the generated data-frame.

	ID	Skill Level	Cognitive Learning Ability	Motor Learning Ability	Verification Error Rate	Deposit Error Rate	Storage Error Rate	Location Error Rate	Scan Time	Verification Time	Deposit Time	Storage Time	Location Time	Initial Experience	Total Scan Time
0	1	2	0.02	0.2	0.032423	0.038830	0.073584	0.037290	2.386294	2.386294	2.386294	2.386294	2.386294	0.5	187.892849
1	1	2	0.02	0.2	0.031325	0.037447	0.035914	0.032175	2.386294	2.386294	2.386294	2.386294	2.386294	0.5	166.672773
2	1	2	0.02	0.2	0.048244	0.036506	0.037792	0.046206	2.386294	2.386294	2.386294	2.386294	2.386294	0.5	181.682613
3	1	2	0.02	0.2	0.032591	0.032814	0.035781	0.122085	2.386294	2.386294	2.386294	2.386294	2.386294	0.5	196.857435

Figure 4: A snippet of the generated data-frame

5. Experiment Design

In this experiment, multiple simulation protocols are considered. Based on the traditional workflow, each proposed improvement (protocol) is introduced into the workflow using HUB-CI logic, to ensure that it is integrated and can communicate smoothly with the workflow. The protocols are described below:

Level 0: Traditional protocol (regular workflow)
HUB-CI Level 1A: Traditional Workflow + avatar training
HUB-CI Level 1B: Traditional workflow + IoT/MHS
HUB-CI Level 2: Traditional workflow + avatar + IoT/MHS

Based on the proposed simulation protocols, the following experiments are devised:

1. Experiment 1: Comparing effect of Avatar, IoT/MHS improvements on the same worker
2. Experiment 2: Comparing Avatar, IoT/MHS improvements across workers of different skill levels

In each of these experiments, we monitor the performance metrics across a fixed number of runs for each simulation protocol, which are then averaged out. Relevant tables and graphs are then populated.

6. Simulation Results

Experiment 1:

Experiment 1 monitors the overall costs (broken into error, operation and total) across multiple runs of the simulation. In the first case, we consider the working of a worker with lower skill ability (even though a range of values for low skill level can be considered, we assume singular values for this simulation).

a. Novice Worker

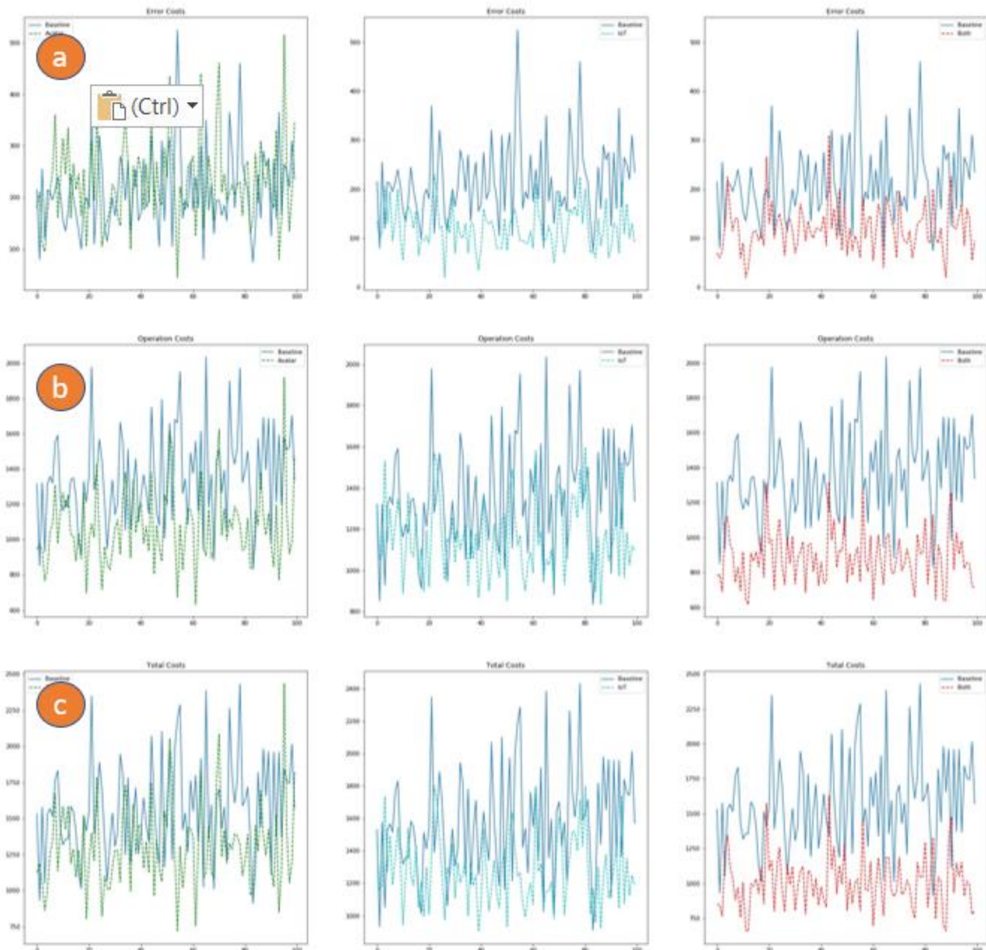

Figure 5: L-R - Cost Comparison across protocol levels. a) Error Costs, b) Operation Costs, c) Total Costs

Figure 1 plots each of the costs from top to bottom, while differentiating across HUB-CI protocol levels from left to right. It can be seen that HUB-CI Level 1A does not affect error costs significantly (actually showing a minimal increase in error costs, which is not significant enough to be deemed effective (Table 3), which is not the case for HUB-CI

Level 1B - which produces a significant effect on error costs. The reverse can be observed for Operation costs, where Level 1A seems to dominate while Level 1B has minimal effect. However, Level 2 shows the largest difference in costs when comparing to Level 0. Given our assumption that we do not consider avatar training costs or IoT operation costs, Level 2 provides the maximum benefits, as seen from Table 1 and 2 (36% reduction in total costs). Table 3 provides the statistical tests to compare the mean costs across different protocols. It can be seen that while error costs are not significantly different between baseline and level 1A, there is a significant difference in operation costs and total costs for level 1A.

Protocol	Errors		Operation		Total	
	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
Baseline	215.6	78.52	1355.55	261.74	1571.15	331.99
HUB-CI Level 1A (avatar only)	229.75	82.44	1072.32	213.85	1302.07	290.08
HUB-CI Level 1B (IoT only)	121.01	43.16	1156.38	179.57	1276.385	215.433
HUB-CI Level 2 (avatar + IOT)	119.7	48.93	887.15	151.89	1006.85	193.06

Table 1: Errors, Operation cost across Protocols

Protocol	Error Cost	Operation Cost	Total Cost
Level 1A	+6.5%	-21%	-18%
Level 1B	-44%	-15%	-19%
Level 2	-45%	-35%	-36%

Table 2: Percentage differences (compared to Level 0)

Hypothesis	Test result	Interpretation
Error Cost - Baseline = HUB-CI Level 1A	Welch Two sample t-test: t = -1.24, pval = 0.21	Not enough evidence to reject NH
Error Cost – Baseline = HUB-CI Level 1B	Welch Two sample t-test: t = 10.55, pval = 6e-20	Reject Null Hypothesis – HUB-CI Level 1B Error cost is statistically significant
Operation Cost – Baseline = HUB-CI Level 1A	Welch Two sample t-test: t = 8.37, pval = 1.15e-14	Reject Null Hypothesis – HUB-CI Level 1A Operation cost is statistically significant
Total Cost = Baseline = HUB-CI Level 2	Welch Two sample t-test: t = 14.69, pval = 1.98e-33	Reject Null Hypothesis – HUB-CI Level 2 Total cost is statistically significant

Table 3: Statistical tests for a novice worker

b. Experienced Worker

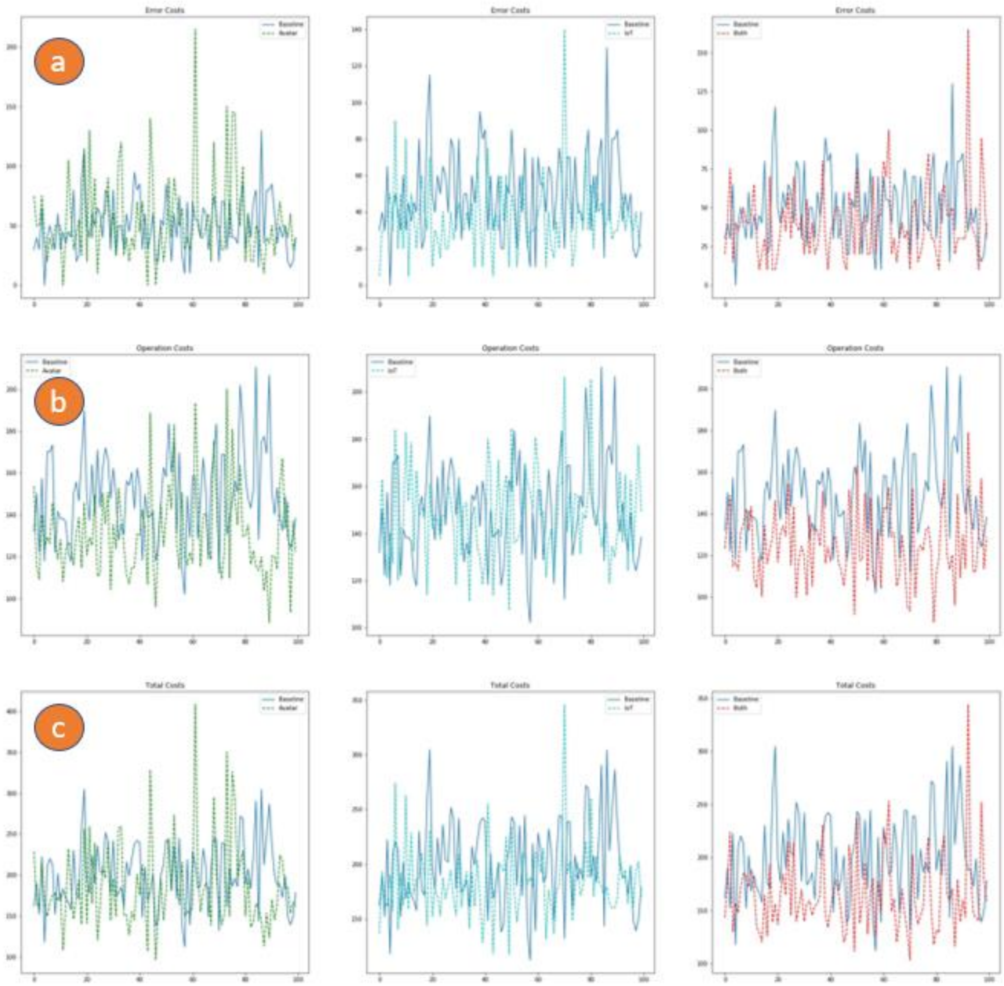

Figure 6: L-R - Cost Comparison across protocol levels. a) Error Costs, b) Operation Costs, c) Total Costs

For an experienced worker, the relevant cost graphs are shown in Figure 6. In this case, the benefits of each HUB-CI protocol are lower but still significant, owing to the skilled nature of the worker (Table 6) in relation to the learning curve model (Equation 1). While the magnitude of the costs for the skilled worker are lower compared to a novice worker, it can be seen that the HUB-CI Level 2 protocol provides statistically significant cost reductions (18% reduction in costs when compared to baseline). The statistical tests shown in table 6 allude to results similar to table 3, but in comparison the evidence against null hypothesis is of lower magnitude (as that of a novice worker).

Protocol	Errors		Operation		Total	
	Mean	Std.Dev	Mean	Std.Dev	Mean	Std.Dev
Baseline	50.6	23.6	149.72	20.91	200.32	40.39
Level 1A (avatar only)	54.25	35.98	130.89	21.00	185.14	52.94
Level 1B (IoT only)	38.6	20.05	147.69	19.08	184.49	33.46
Level 2 (avatar + IOT)	39.6	24.01	125.07	17.40	164.67	36.44

Table 4: Errors, Operation cost across Protocols

Protocol	Error Cost	Operation Cost	Total Cost
Level 1A	+7.2%	-13%	-7.5%
Level 1B	-24%	-1.4%	-8%
Level 2	-22%	-17%	-18%

Table 5: Percentage differences (compared to Level 0)

Hypothesis	Test result	Interpretation
Error Cost - Baseline = HUB-CI Level 1A	Welch Two sample t-test: t = -0.85, pval = 0.39	Not enough evidence to reject NH
Error Cost – Baseline = HUB-CI Level 1B	Welch Two sample t-test: t = 4.45, pval = 1.4e-5	Reject Null Hypothesis – HUB-CI Level 1B Error cost is statistically significant
Operation Cost – Baseline = HUB-CI Level 1A	Welch Two sample t-test: t = 6.35, pval = 1.4e-9	Reject Null Hypothesis – HUB-CI Level 1A Operation cost is statistically significant
Operation Cost – Baseline = HUB-CI Level 1B	Welch Two sample t-test: t = 0.71, pval = 0.47	Not enough evidence to reject NH
Total Cost = Baseline = HUB-CI Level 2	Welch Two sample t-test: t = 6.55, pval = 4.8e-10	Reject Null Hypothesis – HUB-CI Level 2 Total cost is statistically significant

Table 6: Statistical tests for Experienced worker

Experiment 2: Novice vs Experienced Worker

Worker	Error Cost	Operation Cost	Total Cost
Novice before improvements	105.25	451.34	556.59
Novice after improvements	70.15	335.21	405.36
Percentage change	**34%**	**26%**	**28%**
Experienced before improvements	52.05	148.96	210.01
Experienced after improvements	37.6	123.03	160.63
Percentage change	**28%**	**18%**	**24%**

Table 7: Cost comparisons across workers

In this experiment, for each type of worker, the costs are compared before and after the improvements are introduced. Figure 7 shows the relevant performance metrics, and it can be seen that while both types of workers show positive improvements, the magnitude of these improvements are larger in the case of novice workers (Reduction in total cost: 28% for novice worker, 24% for experienced worker). This relates also to the fact that with increasing skill level, it is more difficult to reduce error rate and task performance. This suggests maturity of the avatar tutoring, and any improvements beyond the current level require exponentially more investment/time.

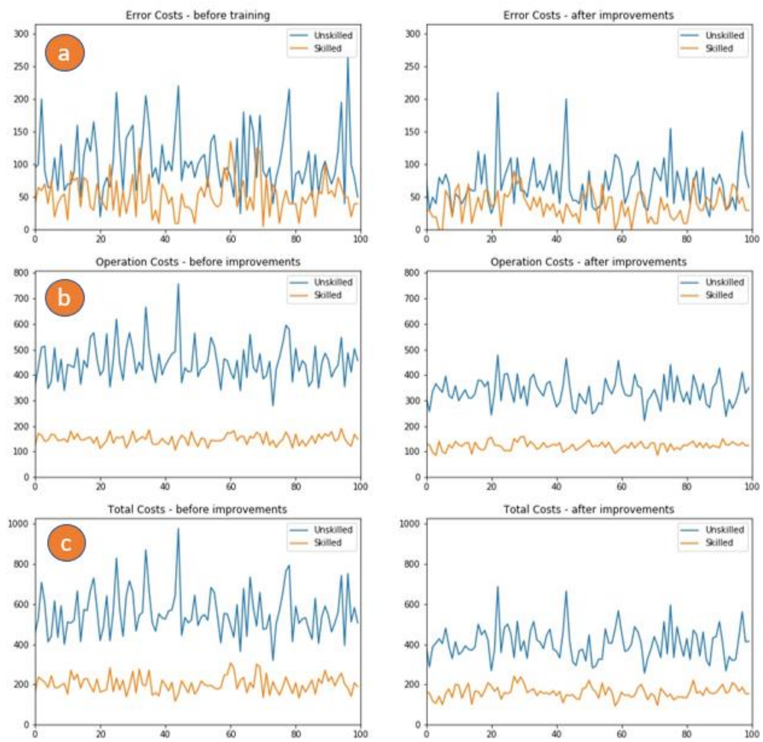

Figure 7: L-R - Before, After improvements: Cost comparison

7. Conclusions

1. This report discusses the design, modelling and evaluation of a multi-agent simulator for a flow junction workflow
2. We design the simulation using HUB-CI logic, where humans, parts, trays/kits are considered as individual agents with specific attributes (qualities).
    a. Human agents are modelled with specific task performance and error rates related to their skill level
    b. Skill level is a relative term which depends on the set of tasks they perform. Hence, it can also be considered as a skill-gap
3. We propose two improvements to the baseline workflow of the flow junction:
    a. Avatar/AR based tutoring of operators: From preliminary data analysis, lack of skilled operators showed significant correlation with error occurrences in the workflow. Thus, we envision the use of an AR-based tutor for machine/physical tasks which uses the operators cognitive and physical learning abilities to improve their own task times
    b. IoT based MHS design: In order to improve the awareness/reduce cognitive load of the operators, we consider an IoT based kit design where the relevant kits, locations employ IoT based LED's/lighting to signify the relation between parts, kits and locations. This is aimed towards reducing error rates of operators working in manual scanning, placing and location tasks
4. HUB-CI logic is used to integrate these two improvements as modular additions to the workflow, ensuring workflow continuity and optimized interactions between the workflow and improvements
5. Based on the simulation results, the following can be observed:
    a. HUB-CI Level 1A (Avatar only) provides statistically significant improvements to operation costs. The degree of the improvements depends on the skill level of the worker - the maximum benefits are provided to lower skilled workers, reducing skill gap to maximum extent. Higher skilled workers showed improvements in performance but not to the same extent as lower skilled workers.
    b. HUB-CI Level 1B (IoT/MHS) shows statistically significant results in reducing error rates (and consequently, error costs). However, it does not produce significant results on the operation costs
    c. HUB-CI Level 2 protocol showed the higher overall reduction in total costs, showing a 36% reduction in costs for lower skilled workers, and 18% reduction in costs for higher skilled workers
6. We can thus show quantifiable improvements to the workflow by our proposed improvements design and HUB-CI as a workflow integrator.

8. Limitations

1. The assumptions made about the proposed learning curve models for avatar, IoT based improvements need to be validated through real-world experiments.
2. Additional experiments with avatar need to be done to assess the relation of the tutoring and the cognitive and physical learning abilities - the real nature of their relation has not been researched/evaluated in real-world scenarios.
3. The proposed IoT design, while hypothesized to improve the awareness of the operator, need to be validated by use of physical experiments - to also confirm the relation between awareness and error rates.
4. We have not considered costs such as training cost of avatar, implementation and maintenance cost of IoT/MHS. These need to be considered in the future once clearer estimates of the costs can be provided.

9. Acknowledgements

This work is supported by the Production, Robotics, and Integration Software for Manufacturing & Management (PRISM) Center at Purdue University; and NSF project Grant# 1839971, "FW-HTF: Collaborative Research: Pre-Skilling Workers, Understanding Labor Force Implications and Designing Future Factory Human-Robot Workflows Using a Physical Simulation Platform".

Special thanks to Mr. Doug Mansfield, President of Kirby Risk Manufacturing.

10. References

[1] A. Ascó, "An Evolutionary Algorithm and operators for the Airport Baggage Sorting Station Problem," Soft Comput., vol. 23, no. 20, pp. 10055–10083, 2019.

[2] Anizar, I. Siregar, E. Putri, I. Maududi, and W. Habibi, "Lighting Quality Improving Work Thoroughness of Sorting Operators," IOP Conf. Ser. Mater. Sci. Eng., vol. 722, p. 12009, 2020.

[3] L. Alboteanu, G. Manolea, and F. Ravigan, "Automatic sorting and handling station actuated by pneumatic drive," Ann. Univ. Craiova, Electr. Eng. Ser., no. 1, pp. 1–8, 2018.

[4] Sellers, C.J., and Nof, S.Y., "Part Kitting in Robotic Facilities," Material Flow, 3(1-3), 1986, 163-174.

[5] Sellers, C.J., and Nof, S.Y., "Performance Analysis of Robotic Kitting Systems," Robotics and Computer-Integrated Manufacturing, 1989, 6(1), 15-24.

[6] A. C. Caputo, P. M. Pelagagge, and P. Salini, “Modelling human errors and quality issues in kitting processes for assembly lines feeding,” Comput. Ind. Eng., vol. 111, pp. 492–506, 2017.
[7] M. A. Corakci, “An evaluation of kitting systems in lean production,” Univ. Coll. Borås Sch. Eng., 2008.
[8] E. H. Grosse, C. H. Glock, M. Y. Jaber, and W. P. Neumann, “Incorporating human factors in order picking planning models: Framework and research opportunities,” Int. J. Prod. Res., vol. 53, no. 3, pp. 695–717, 2015.
[9] A.-C. Falck, R. Örtengren, and M. Rosenqvist, “Assembly failures and action cost in relation to complexity level and assembly ergonomics in manual assembly (part 2),” Int. J. Ind. Ergon., vol. 44, no. 3, pp. 455–459, 2014.
[10] J. Lindblom and P. Thorvald, “Towards a framework for reducing cognitive load in manufacturing personnel,” Adv. Cogn. Eng. neuroergonomics, vol. 11, pp. 233–244, 2014.
[11] Y. Zhang, Z. Guo, J. Lv, and Y. Liu, “A framework for smart production-logistics systems based on CPS and industrial IoT,” IEEE Trans. Ind. Informatics, vol. 14, no. 9, pp. 4019–4032, 2018.
[12] W. Ding, “Study of smart warehouse management system based on the IOT,” Adv. Intell. Syst. Comput., vol. 180 AISC, pp. 203–207, 2013.
[13] H. Sang Ko and S. Y. Nof, “Design and application of task administration protocols for collaborative production and service systems,” Int. J. Prod. Econ., vol. 135, no. 1, pp. 177–189, 2012.
[14] G. Liu, W. Yu, and Y. Liu, “Resource management with RFID technology in automatic warehouse system,” IEEE Int. Conf. Intell. Robot. Syst., pp. 3706–3711, 2006.
[15] A. Musa and A.-A. A. Dabo, “A review of RFID in supply chain management: 2000--2015,” Glob. J. Flex. Syst. Manag., vol. 17, no. 2, pp. 189–228, 2016.
[16] M. Mohammad Bagheri, “Intelligent and Adaptive Tutoring Systems: How to Integrate Learners,” Int. J. Educ., vol. 7, no. 2, p. 1, 2015.
[17] Y. Cao, X. Qian, T. Wang, R. Lee, K. Huo, and K. Ramani, “An Exploratory Study of Augmented Reality Presence for Tutoring Machine Tasks,” in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, 2020, pp. 1–13.
[18] G. Westerfield, A. Mitrovic, and M. Billinghurst, “Intelligent Augmented Reality Training for Motherboard Assembly,” Int. J. Artif. Intell. Educ., vol. 25, no. 1, pp. 157–172, 2015.
[19] P. Devadasan, H. Zhong, and S. Y. Nof, “Collaborative intelligence in knowledge based service planning,” Expert Syst. Appl., vol. 40, no. 17, pp. 6778–6787, 2013.
[20] P. O. Dusadeerungsikul et al., “Collaboration requirement planning protocol for HUB-CI in factories of the future,” Procedia Manuf., vol. 39, pp. 218–225, 2019.

[21] Jaber, M. Y., and Glock, C. H., “A learning curve for tasks with cognitive and motor elements,” Computers & Industrial Engineering, vol. 64, pp. 866-871, 2013.
[22] Duffey, R. B., and Saull, J. W., “Managing and predicting technological risk and human reliability: A new learning curve theory,” Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, vol. 222, no. 2, pp. 245-254, 2008.
[23] Huang, G., Qian, X., Wang, T., Patel, F., Sreeram, M., Cao, Y., Ramani, K., Quinn, A. “AdapTutor: an Adaptive Tutoring System for Machine Tasks using Augmented Reality”. Submitted to ACM CHI Conference on Human Factors in Computing Systems (CHI 2021).
[24] Sreeram, M. Human-in-the-loop of Cyber-Physical Agricultural Robotic Systems. Master’s Thesis, Purdue University, 2020.

[25] Sreeram, M., and Nof, S.Y., *Flow Junction Research*, PRISM Report, Purdue University, Dec. 1, 2020. (Revised as PRISM Research Memo 2022-P2 in July 2022.)