Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

SIMULATION BASED CYCLE TIME PREDICTION FOR ROBOT WELDING

Seongho Cho[1], Donguk Kim[1], Sangchul Park[1]

[1]Dept. of Industrial Engineering, Ajou University, Suwon, SOUTH KOREA

ABSTRACT

This paper introduces methodologies aimed at predicting the cycle time of robotic arm spot welding operations, which are essential for vehicle body assembly process plans. Predicting the cycle time of robot arm is crucial for process plan, as it is closely linked to overall production efficiency and safety considerations. However, it is common for companies in the vehicle body assembly industry to rely on rough estimates for cycle time prediction. We propose methodologies that ensure ease of use and accuracy based on simulation data to cope with this problem. This paper provides an overview of the overall process of each methodology, including data collection and model construction. Additionally, experiments are conducted to compare the performance of each methodology, with results indicating that our proposed approach outperforms conventional methods. Through this research, we found the potential for the development of advanced methods applicable in the industry.

1 INTRODUCTION

As customer demands continue to escalate, the importance of delivery adherence is increasingly emphasized within production systems. In this context, accurately predicting and managing the cycle time of processes is essential for production efficiency and customer satisfaction (Kuhl and Krause 2019). Especially in the field of vehicle body assembly, a process plan including the cycle time of each operation, is drafted during the product design phase (Sundaram and Fu 1988). Through this, the production cycle and maximum production volume of the product were predicted. However, if incorrect information is included in the process plan, it can lead to several issues. Failure to meet product delivery deadlines may result in sales losses, and the production efficiency of other products utilizing shared facilities may decrease. Therefore, the accurate drafting of process plans plays a crucial role in preventing such issues and enhancing the efficiency of the entire production process (Assid et al. 2020).

Process plan for vehicle body assembly, typically includes a method sheet with a gantt chart depicting the processes and their respective cycle times (Kumar et al. 2014), as shown in Figure 1. The chart includes processing equipment cycle times, such as the time taken by workers to move parts, the clamping time of jigs, and the welding time of robot arms, along with other preparation processes for the operation. When creating the gantt chart within the process plan, the calculation method for equipment cycle time is determined based on the standard unit process time table for each equipment. For example, if the standard unit process time for spot welding operations using robot arms is 3.3 seconds per point, the cycle time for the robot is calculated by multiplying 3.3 seconds by the number of spot-welding points allocated to the robot, included in the gantt chart. These standard unit process time is derived empirically based on experience, serving as an indicator for rough predictions.

# Process	OPERATION DESCRIPTION	H/T	M/T
	# S03	24	34
1	H/W F/PROOF CHECK	6	
	1) 62719-H1933	(6)	
2	#2 WORKER PART LOAD	8	
	1) LS-32719-H1841	4	
	2) RS-32719-H1842	4	
3	JIG CLAMP		3
4	SAFETY SYSTEM OPERATION		1
5	#1 ROBOT SPOT WELD(9 POINTS)		29
6	JIG UNCLAMP		1
7	#2 WORKER PART UNLOAD	10	
	1) LS-32719-S1824	5	
	2) RS-32719-S1824	5	

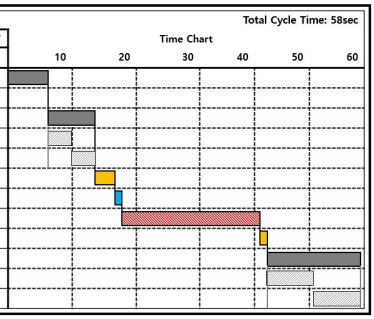

Figure 1: Gantt chart for body assembly process plan.

However, since the method using standard unit process time table roughly estimates the actual cycle time of the equipment, the process plan produced through this method may lack precision and potentially cause unexpected issues in the actual production process. Particularly when multiple robots are concurrently executing tasks within a single work cell, there exists a potential risk of accidents such as interference or collisions between robots. These incidents can result in production delays, equipment damage, safety concerns, highlighting the critical importance of precise prediction of the actual cycle time to proactively mitigate such occurrences (Touzani et al. 2022).

In this context we aim to focus on predicting spot welding operations cycle time utilizing robotic arms, which is one of the most prominent processes in the field of body assembly (Pellegrinelli et al. 2017). However, due to the nature of robotic arms, various factors such as the departure angle, motor acceleration and deceleration, and the approach angle to the welding position affect the operation (Huang et al. 2020). Therefore, predicting cycle time for the process poses a challenge.

Traditionally, the method for predicting the cycle time of robotic arms have involved breaking down the robot's actions into several unit actions and roughly estimating the cycle time based on these (Nof and Lechtman 1982). This method mirrors the standard unit process time table approach. However, precision issues have led to research on predicting cycle time through the rotation time of the motors driving the robot arm or through the movement profile of the end effector (Piazzi and Visioli 2000).

Companies that are not robotic arm manufacturers face limitations in acquiring information about the motors or movement profiles. Even if they obtain such information, they may encounter difficulties in analyzing it. In our research, to overcome limitations in information acquisition, we utilize a commercialized robot simulator for data acquisition. Subsequently, we perform cycle time prediction using the acquired information through two different approaches; 1) Cycle time prediction based on mathematical calculation and 2) Cycle time prediction utilizing a deep neural network.

The rest of this paper is structured as follows: Section 2 provides a brief description of the simulator used for data acquisition. Section 3 introduces the method of predicting cycle time through mathematical approaches utilizing simulation data. Section 4 introduces the deep neural network approach and the data preprocessing methods required for it. Section 5 provides the outcomes of experiment for prediction methodology. Finally, Section 6 concludes the paper with a summary of the results obtained from this research.

2 SIMULATION FOR DATA ACQUISITION

In this paper, both methods proposed require information about robot arm movement. For this purpose, we utilized the ABB Robotics' Robot Studio OLP (off-line program) (Connolly 2009). This program allows users to create a virtual welding robot workstation, as shown in Figure 2 (Gui 2022). In the created virtual workstation, the program enables simulation of robot arm movements akin to an OLP. Additionally, it provides continuous positional information of the end effector and continuous angle information of robot joints during simulation. Furthermore, it offers continuous data regarding the electric current and voltage applied to the motors.

This paper chooses the IRB6660 model as the target robot, commonly used in actual vehicle body assembly factories. Using information from the vehicle body assembly factories, we construct a virtual workstation. Through this workstation, we can acquire information, solely concentrating on joints and positions, while disregarding considerations related to electric current and voltage, to be utilized in our methods. To achieve this, preparations are made by configuring the Signal Analyzer in Robot Studio to extract only the necessary information.

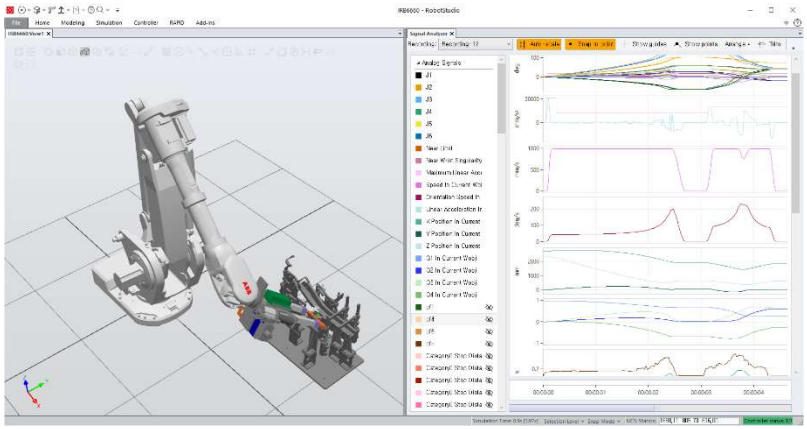

Figure 2: Virtual welding robot workstation using robot studio.

3 CYCLE TIME PREDICTION USING MATHEMATICAL APPROACH

3.1 Calculating End Effector Movement Time

For the first approach, we initially consider the composition of the welding operation of the robot arm. The welding operation of the robot arm consists of a combination of the robot arm's movement time and the welding time of the end effector. In this paper, we aim to primarily focus on the movement time of the end effector for predicting the cycle time of robot arm welding operations.

To obtain the end effector movement time, three-dimensional trajectory information of the end effector's path and the configured movement speed information of the robot are required. The trajectory of the end effector can be derived from the continuous positional data acquired through simulation. Robot Studio, which provides this information, offers position updates along the path at intervals of 0.024 seconds. Using this trajectory data, we can calculate the node-to-node movement time $T_{(n,\, n+1)}$ included in the robot arm welding program.

The node-to-node movement time $T_{(n,\, n+1)}$ is shown in equation (1), where n represents the index of the sequence within the robot arm welding program. $p_{(i,\, x)}$ represents the x-coordinate of the end effector along the path from sequence n to sequence $n+1$, where i and $i+1$ intervals correspond to the 0.024 seconds interval of the path data from Robot Studio. $p_{(i,\, y)}$ and $p_{(i,\, z)}$ represent the y-coordinate and z-coordinate, respectively, of the end effector at the same point in time as $p_{(i,\, x)}$. Lastly, V represents the configured movement speed information of the robot.

$$T_{(n,n+1)} = \frac{\sum_i \sqrt{|p_{i+1,x}-p_{i,x}|^2+|p_{i+1,y}-p_{i,y}|^2+|p_{i+1,z}-p_{i,z}|^2}}{V} \tag{1}$$

Through this method, we can derive the movement time of the end effector, a component of the welding operation of the robot arm.

3.2 Method for Calculating Cycle Time Values

The method for deriving the cycle time of the robot arm welding operation can be obtained based on the equation provided in Section 3.1, along with additional calculations. The equation in Section 3.1 calculates only the movement of the end effector during the robot arm welding operation. However, we intend to perform additional calculations for the welding operation after the movement of the end effector. The additional factor we need to consider is the welding time. Typically, the mechanical complexity of end effector for welding process is relatively low compared to other equipment in the production system. Furthermore, the time required for a spot-welding operation is known to be around 1 second (Aslanlar et al. 2008). Therefore, we have set the welding time to a constant one-second interval per point.

The calculated cycle time of the robot arm welding operation CT_c is shown in equation (2), where N represents the total number of sequences within the robot arm welding program, and p_s represents the total number of welding points within the robot arm welding program. Ultimately, by utilizing Equation 2, we can calculate the cycle time of the robot arm welding operation, including both the end effector movement time and the welding time. Through these detailed numerical calculations, we can predict cycle time more accurately than the conventional method.

$$CT_c = \sum_{n=0}^{N-1} T_{(n,n+1)} + 1 \times p_s \tag{2}$$

4 CYCLE TIME PREDICTION USING DEEP NEURAL NETWORKS

4.1 Data Collection for Training

For the second approach, we first focus on the characteristics of the deep neural network models. Deep neural networks (DNNs) or artificial intelligence (AI) are prominent methods for data-driven prediction, utilizing explanatory variables and response variables within the data to train and construct models, which are then used to forecast the target. A distinctive characteristic of DNN models is their significant reliance on data. Therefore, in our research, the process of collecting train data before constructing the model is conducted as an initial step.

We generate 100 weldable points in the virtual welding robot workspace previously constructed. Each weldable point is one of the points within the CAD file of the part where the robot arm's end effector can reach without collision. Using these 100 points, we create bidirectional movement robotic arm operation programs from point to point. Subsequently, simulations are conducted, resulting in the acquisition of a total of 19,800 point to point trajectory data. Table 1 represents information showing only the robot's joint data among the point-to-point trajectory data, indicating the robot's state at each moment during the simulation. In this table, The column "Fine" indicates the event that occurs when the robot reaches the current target point.

Table 1: Example of point to point trajectory data.

Time(s)	J1(˚)	J2(˚)	J3(˚)	J4(˚)	J5(˚)	J6(˚)	Fine
0.000	19.56	-22.19	5.44	-8.33	63.25	21.92	o
0.024	19.30	-22.27	5.47	-8.22	63.15	22.27	
0.048	18.91	-22.32	5.57	-8.20	62.86	23.29	
…							
1.968	14.19	-31.12	24.27	-33.24	31.78	99.75	
1.992	14.74	-30.90	24.87	-36.87	31.06	100.11	
2.016	15.30	-30.67	25.47	-40.50	30.33	100.47	o
2.040	15.85	-30.44	26.07	-44.12	29.61	100.83	

However, since each trajectory data consists of continuous data from point to point, their sizes vary. Due to this issue, it is necessary to convert the continuous trajectory data into discrete data. For this purpose, we used fine data in trajectory data. Using this event data, we construct a train dataset containing only information about the starting and ending points. This not only brings the advantage of uniformizing the data size but also provides benefits for users, allowing them to predict the movement time of the end effector solely based on the departure and arrival positions without the need for the entire trajectory data. The constructed train dataset, as shown in Table 2, consists of explanatory variables including six-axis angles for both the starting and ending points, along with the three-dimensional Cartesian coordinates and quaternion rotation information of the end effector. Additionally, for our research purposes, the movement time is constituted as the response variable.

The primitive construction of the train dataset, which is fundamentally essential for deep neural networks, has been accomplished. However, to enhance the learning process, data normalization is necessary. For this purpose, information required for Z-score normalization of the entire dataset is initially stored (Patro 2015). Subsequently, to evaluate the train results, a separated test dataset is required. Therefore, the entire dataset is divided into a 3:7 ratio, creating separate train and test datasets. Finally, normalization is performed on each dataset using the previously stored information for Z-score normalization. Ultimately, the construction of the requisite train and test datasets for the deep neural network has been completed.

Table 2: Discrete simulation data for train dataset.

Input Feature	Joint Values	Start joint 1	Arrive joint 1
		Start joint 2	Arrive joint 2
		Start joint 3	Arrive joint 3
		Start joint 4	Arrive joint 4
		Start joint 5	Arrive joint 5
		Start joint 6	Arrive joint 6
	Quaternion Values	Start quaternion 1	Arrive quaternion 1
		Start quaternion 2	Arrive quaternion 2
		Start quaternion 3	Arrive quaternion 3
		Start quaternion 4	Arrive quaternion 4
	Position Values	Start position X	Arrive position X
		Start position Y	Arrive position Y
		Start position Z	Arrive position Z
Output Feature	Time	Movement Time	

4.2 Building a DNN Model for Movement Time Prediction

Having completed the construction of the train data, our next step is to build a DNN for training. Considering the balance between the performance of the DNN and the risk of overfitting, we design the model structure using a model stacking ensemble approach (Akyol 2020). This structure involves two types of DNNs: a base DNN and an assemble DNN. Additionally, to enhance the expressive power of the model, we design both DNNs with four or more hidden layers, recognizing that deeper layers contribute to higher representational capacity (Thomas et al. 2017).

When designing the DNN architecture, special attention was given to selecting suitable activation functions for the hidden layers, considering the characteristics of the training data. The chosen activation functions and the structure of the DNN are outlined in Table 3 and Table 4. Since all explanatory variables in the training dataset include both positive and negative values, tanh and leaky ReLU functions were selected for the initial data processing of the base DNN, capable of handling both positive and negative data. Additionally, in the assemble DNN that integrates the outputs of the base DNN, two layers of leaky ReLU

were used to accommodate negative outputs from the base DNN, followed by ReLU functions for non-linearity processing.

Table 3: Activation function for hidden layers.

Base DNN	Input Parameters	18
	Layer 1	128 neurons, “Tanh”
	Layer 2	256 neurons, “Tanh”
	Layer 3	256 neurons, “Leaky ReLU”
	Layer 4	128 neurons, “Leaky ReLU”
	Output Parameters	64

Table 4: Activation function for hidden layers

Assemble DNN	Input Parameters	64×number of Base DNN(3)
	Layer 1	384 neurons, “Leaky ReLU”
	Layer 2	192 neurons, “Leaky ReLU”
	Layer 3	96 neurons, “ReLU”
	Layer 4	48 neurons, “ReLU”
	Output Parameters	1

The overall structure of the model devise in this manner is illustrated in Figure 3. For training the neural network architecture designed in this research, the normalized train data is divided into three equal parts and passed through three base DNNs to obtain three output tensors. Subsequently, these three output tensors are stacked in arbitrary order to construct meta-data. Finally, the constructed meta-data is fed into the assembled DNN to derive the final prediction values.

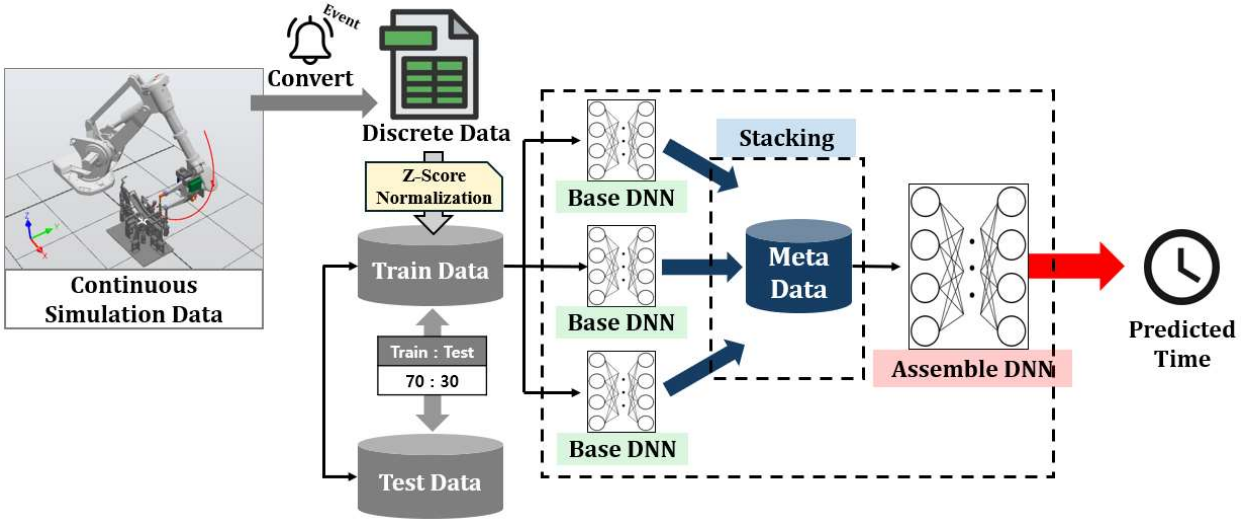

Figure 3: Overall structure and data flow of deep neural network.

4.3 Training the Movement Time Prediction Model

With the completion of data construction and artificial neural network design, we aim to proceed with model training. The hyperparameters for training are set as follows: a learning rate of 0.02, 200 epochs, and the AdaDelta optimization function. To facilitate a straightforward comparison between actual and

predicted movement times, the MAE (mean average error) function is chosen as the loss function, as illustrated in Table 5.

Table 5: Hyperparameters for deep neural network.

Hypermeters	Values
Epochs	200 epochs
Loss function	Mean absolute error
Optimization function	AdaDelta function
Learning rate	0.02

Considering the nature of artificial neural networks, the model's performance varies with each epoch during training. To acquire the best-performing model, evaluations are conducted on the test dataset at the end of each epoch. Figure 4 illustrates the MAE values observed throughout all epochs with providing more detailed view of the MAE values from the 20th epoch onwards. As depicted in both figures, among the 300 epochs, the model trained at the 294th epoch exhibited the most favorable performance with an MAE of 0.0302 seconds. Consequently, this model is selected as the predictive model for estimating movement times in subsequent experimental phase.

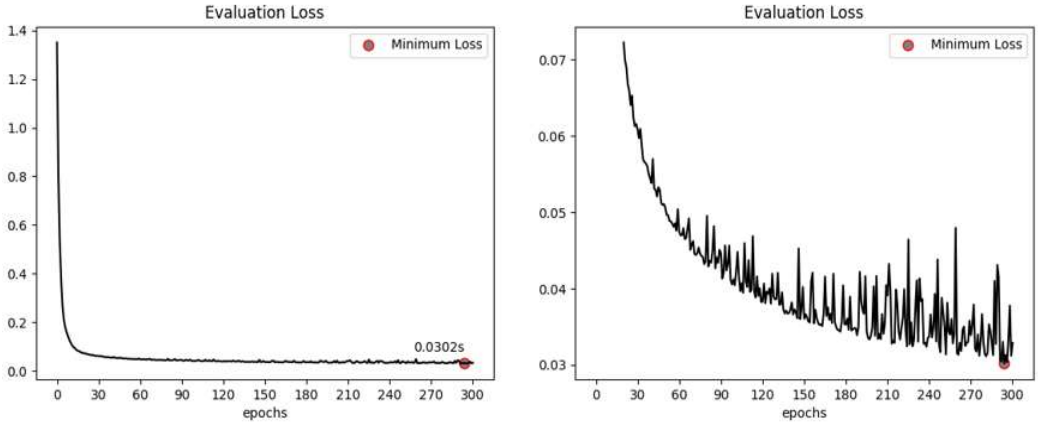

Figure 4: MAE graph during training.

4.4 Method for Calculating Predicted Cycle Time

The method for deriving the cycle time of the robot arm welding operation using DNN closely resembles the Equation in Section 3.2. We have constructed a DNN specifically for approximating the moving time; therefore, there is no need for an equation regarding moving time, like Equation 1 used in Equation 2. However, our constructed DNN is intended solely for predicting the movement time of the end effector, so we need to calculate the welding time additionally. As mentioned in Section 3.2, welding time is typically 1 second, so by calculating 1 second per welding point, we can compute the cycle time of the robot arm welding operation.

The cycle time of the robot arm welding operation using DNN , denoted as CT_p is shown in Equation 3, where PT_n represents the predicted movement time obtained through the DNN. By utilizing Equation 3, we can obtain the predicted cycle time of the robot arm welding operation, including both the end effector movement time and the welding time.

$$CT_p = \sum_{n=0}^{N} PT_n + 1 \times p_s \tag{3}$$

5 EXPERIMENTS FOR PREDICTION METHODOLOGIES

We conducted experiments to compare the performance of the three methods, as follows.

- Methods to be compared in the experiment.

1. cycle time using standard unit process time table,
2. cycle time using mathematical approach,
3. cycle time using deep neural network.

The experiment compares the cycle time prediction performance for 30 robot welding operation programs. To conduct the experiment, it is necessary to determine the total number of welding points for each program. For this purpose, 30 real numbers are extracted from a normal distribution with a mean of 12.1 and a standard deviation of 5.30, referencing the process plan of the REINF ASSY-SIDE COMPLETE vehicle body part. These real numbers are rounded to select the total number of welding points for each program. Subsequently, the robot welding operation programs are generated in the previously constructed virtual welding robot workspace, tailored to the selected number of welding points.

The Figure 5 depicts visual representations of the experimental outcomes. The upper graph showcases the cycle time of robot welding programs derived from actual simulations, contrasting them with those acquired through three distinct methods, and the lower graph demonstrates the disparities between the actual cycle times and the predictions made by the three methods. The Table 6, which follows, provides a statistical summary of the errors for the three methods displayed in the lower graph of Figure 5.

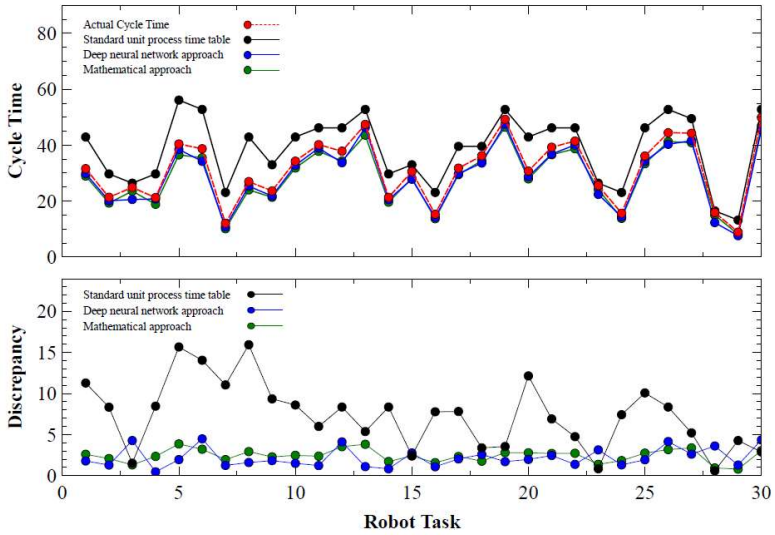

Figure 5: Experimental outcomes.

Table 6: Statistical summary of each method.

	Standard unit process time table (sec)	Mathematical approach (sec)	Deep neural network approach (sec)
Mean	7.362	2.431	2.204
Standard deviation	4.012	0.779	1.143
Min value	0.588	0.795	0.459
Max value	15.948	3.862	4.491

To facilitate a practical comparison of the experimental results, we have defined two performance indices: 1) Accuracy and 2) Usability. As expected, the performance index of accuracy refers to the average percentage error between the actual cycle time and the predicted cycle time. The performance index of usability indicates the proportion of welding operation programs, out of the total 30, with an accuracy exceeding 90%. This index assesses the effectiveness of achieving high accuracy.

Table 7 presents the performance indices for the experimental results. As evident from Table 5, both the mathematical approach and the deep neural network method exhibited superior accuracy and usability compared to the method utilizing the standard unit process time table.

Table 7: Result of performance indices.

	Accuracy (%)	Usability (%)
Standard unit process time table	72.201	23.333
Mathematical approach	91.782	83.333
Deep neural network approach	92.250	86.667

6 CONCLUSION

In this study, we compare three methods for predicting the cycle time of welding operations conducted by robotic arms in the body assembly process plan: using a standard unit process time table, employing a mathematical approach, and utilizing deep neural networks. To employ the mathematical approach, we separate the movement time of the end effector from the welding time. Moreover, we utilize continuous simulation data to calculate the movement time of the end effector based on distance. Additionally, we incorporate the welding time to determine the cycle time of the robotic arm welding operation. Furthermore, to leverage the deep neural network approach, we collect training data through simulations and construct a model designed for ease of use by transforming continuous data into event-based discrete data. Subsequently, we train the model to facilitate prediction of the cycle time of the robotic arm welding operation.

We found that the conventional method based on standard unit process time tables, while simple, suffers from the drawback of low accuracy. In contrast, prediction using mathematical approaches and deep neural networks demonstrate remarkable accuracy of over 90% and outstanding usability of over 80%. This represents an improvement of over 20% in accuracy and 60% in usability compared to the conventional method. This confirms that both methods offer approaches with sufficiently low error rates for practical application. While there isn't a significant performance difference observed between the mathematical approach and prediction using deep neural networks, it's important to note that prediction utilizing deep neural networks offers practical applicability due to its capability to operate solely based on the departure and arrival positions of the robotic arm without requiring the entire trajectory data, unlike the mathematical approach.

Through this research, we found the potential for advanced method to establish a cycle time in the field of vehicle body assembly that ensures both ease of use and accuracy when creating process plans. Furthermore, it is expected that superior models can be developed through continuous data collection and training. We expect that this will simplify the verification stage of the process plan, leading to cost-saving benefits.

ACKNOWLEDGMENTS

This work was supported by the Ministry of Trade, Industry & Energy (MOTIE, Korea) of the Republic of Korea (P0024562).

REFERENCES

Assid, M., Gharbi, A., & Hajji, A. "Production control of failure-prone manufacturing-remanufacturing systems using mixed dedicated and shared facilities." *International Journal of Production Economics*, 224, 107549, 2020.

Akyol, K. "Stacking ensemble based deep neural networks modeling for effective epileptic seizure detection." *Expert Systems with Applications*, 148, 113239, 2020.

Aslanlar, S., Ogur, A., Ozsarac, U., & Ilhan, E. "Welding time effect on mechanical properties of automotive sheets in electrical resistance spot welding." *Materials & Design*, 29(7), 1427-1431, 2008.

Connolly, C. "Technology and applications of ABB RobotStudio." *Industrial Robot: An International Journal*, 36(6), 540-545, 2009.

Gui, W. "Simulation design of welding robot workstation based on RobotStudio." In *ISMSEE 2022; The 2nd International Symposium on Mechanical Systems and Electronic Engineering* (pp. 1-5). VDE, 2022.

Huang, Z., Li, F., & Xu, L. "Modeling and simulation of 6 DOF robotic arm based on gazebo." In *2020 6th International Conference on Control, Automation and Robotics (ICCAR)* (pp. 319-323). IEEE, 2020.

Kuhl, J., & Krause, D. "Strategies for customer satisfaction and customer requirement fulfillment within the trend of individualization." *Procedia CIRP*, 84, 130-135, 2019.

Kumar, S. S., & Kumar, M. P. "Cycle time reduction of a truck body assembly in an automobile industry by lean principles." *Procedia Materials Science*, 5, 1853-1862, 2014.

Nof, S. Y., & Lechtman, H. "Robot time and motion provides means for evaluating alternative work methods." *Industrial Engineering*, 14, 38-48, 1982.

Patro, S. G. O. P. A. L., & Sahu, K. K. "Normalization: A preprocessing stage." *arXiv preprint arXiv:1503.06462*, 2015.

Pellegrinelli, S., Pedrocchi, N., Tosatti, L. M., Fischer, A., & Tolio, T. "Multi-robot spot-welding cells for car-body assembly: Design and motion planning." *Robotics and Computer-Integrated Manufacturing*, 44, 97-116, 2017.

Piazzi, A., & Visioli, A. "Global minimum-jerk trajectory planning of robot manipulators." *IEEE Transactions on Industrial Electronics*, 47(1), 140-149, 2000.

Sundaram, R. M., & Fu, S. S. "Process planning and scheduling—a method of integration for productivity improvement." *Computers & Industrial Engineering*, 15(1-4), 296-301, 1988.

Thomas, A. J., Petridis, M., Walters, S. D., Gheytassi, S. M., & Morgan, R. E. "Two hidden layers are usually better than one." In *Engineering Applications of Neural Networks: 18th International Conference, EANN 2017, Athens, Greece, August 25–27, 2017, Proceedings* (pp. 279-290). Springer International Publishing, 2017.

Touzani, H., Séguy, N., Hadj-Abdelkader, H., Suárez, R., Rosell, J., Palomo-Avellaneda, L., & Bouchafa, S. "Efficient industrial solution for robotic task sequencing problem with mutual collision avoidance & cycle time optimization." *IEEE Robotics and Automation Letters*, 7(2), 2597-2604, 2022.

AUTHOR BIOGRAPHIES

SEONGHO CHO received a bachelor's degree (2023) in industrial engineering, Ajou University, Korea. He is now a master's student in industrial engineering, Ajou University, Korea. He is interested in simulation-based digital manufacturing systems, production scheduling and automated material handling system. His email address is phillip4321@ajou.ac.kr

DONGUK KIM received a bachelor degree (2018) in industrial engineering and a master degree (2021) in industrial engineering, Ajou University, Korea. He is now a Ph.D candidate in industrial engineering, Ajou University, Korea. He is interested in simulation-based scheduling and planning, digital manufacturing, and PHM for plant. His email address is dek1603@ajou.ac.kr

SANGCHUL PARK was granted his B.S.(1994), M.S.(1996) and Ph.D.(2000) degrees in industrial engineering, Korea Advanced Institute of Science and Technology(KAIST). He is a professor in Department of industrial engineering, Ajou University, Republic of Korea, since 2004. He is interested in modeling and simulation, combat simulation for defense, and digital manufacturing system. His email address is scpark@ajou.ac.kr