동력시스템공학회지 제29권 제5호 pp. 3-14 2025년 10월 ISSN 2713-8429(Print) ISSN 2713-8437(Online)
Journal of Power System Engineering https://doi.org/10.9726/kspse.2025.29.5.003
Vol. 29, No. 5, pp. 3-14, October 2025

# Investigating Collision Detection Techniques in Six-Degree-of-Freedom Collaborative Robots

Duc-Anh Pham*, Jeong-Uk Lee*, Do-Yeong Jeong** and Seung-Hun Han**†

(Received 25 Novemebr 2024, Revised 24 October 2025, Accepted 24 October 2025)

**Abstract :** In the contemporary era of advanced technology, Collaborative Robots, known as Cobots, have emerged as a highly promising domain of research and application. Cobots represent a category of robots endowed with the capacity for direct interaction with human counterparts within shared working environments. Their design philosophy centers around harnessing the synergies between human and robotic capabilities, thereby augmenting work efficiency while concurrently ensuring a secure and productive work environment. A pivotal facet of Cobots pertains to their innate ability to operate in a secure and human-friendly manner. This is achieved through the implementation of autonomous collision detection mechanisms, enabling immediate cessation of operation to mitigate potential harm to humans. This attribute assumes particular significance when Cobots and humans collaborate within the same physical workspace. Our research endeavors are concentrated on the enhancement of performance and reliability within Cobots' collision detection systems. To this end, we propose the utilization of two supervised machine learning methodologies, specifically Support Vector Machine Regression (SVMR) and 1D Convolutional Neural Network (1D CNN), to bolster the precision and speed of collision detection for the CURA6 robotic arm-based on Intema's CURA6 dataset. The findings of this study are poised to significantly augment the operational capabilities of Cobots, thereby reducing the risk of accidents in industrial and manufacturing settings.

**Key Words :** Collision Detection Mechanisms, Support Vector Machine Regression (SVMR), 1DConvolutional Neural Network (1D CNN), Supervised Machine Learning

## 1. Introduction

The field of collaborative robotics,[1)] driven by the Industry 4.0 revolution, is evolving as Cobots integrate into industrial automation. Cobots, designed to work alongside humans, adhere to safety standards like ISO 10218[2)] and ISO/TS 15066:2016, ensuring safe human-machine collaboration and considering the social impacts of this interaction.

This study focuses on collision detection in 6-DoF Cobots.[3-6)] It highlights the CURA6 robot (Fig. 1) by Intema,[3)] designed for safe human interaction. Cobots use lightweight materials and incorporate features such as safe stop and force limitation to ensure safety.[2)]

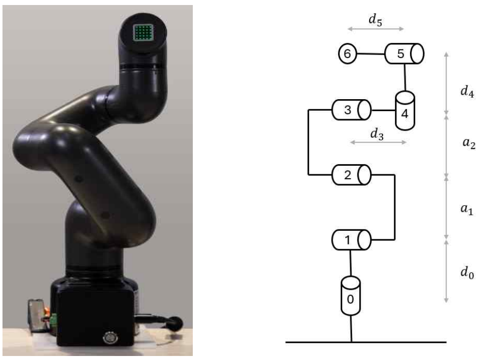
Collision detection in collaborative robots has been extensively researched with various approaches

**† Seung-Hun Han(https://orcid.org/0000-0002-9125-7439) : Professor, Department of Mechanical System Engineering, Gyeongsang National University.
E-mail : shhan@gnu.ac.kr, Tel : 055-772-9105
*Duc-Anh Pham(https://orcid.org/0000-0002-0822-2199), Jeong-Uk Lee(https://orcid.org/0000-0001-6918-255X) : Doctoral student, Department of Mechanical System Engineering, Gyeongsang National University.
**Do-Yeong Jeong(https://orcid.org/0009-0007-1461-9671) : Master student, Department of Mechanical System Engineering, Gyeongsang National University.

addressing different aspects of safety and performance. Traditional collision detection methods can be broadly categorized into two main approaches: non-machine learning and machine learning-based techniques.[3,5,7]

Non-machine learning approaches[8,9] primarily rely on external sensors and threshold-based detection mechanisms. Cirillo et al. developed conformable force/tactile skin systems for physical human-robot interaction, while Strohmayr et al. introduced the DLR artificial skin combining sensitivity with collision tolerance. However, these methods face significant limitations including high implementation costs, complex sensor integration, and susceptibility to environmental noise and wear.

Machine learning approaches have gained prominence due to their ability to learn complex patterns and adapt to various scenarios. Sharkawy et al.[10] proposed neural network designs for manipulator collision detection using only joint position sensors, eliminating the need for torque sensors. Zhang et al.[11] developed an online collision detection scheme combining supervised learning with Bayesian decision theory. Narukawa et al.[12] implemented real-time collision detection based on one-class SVM for humanoid robots. More recently, Heo et al.[5] advanced 1D CNN applications for 6-DoF robotic arms, demonstrating reduced reliance on external sensors and friction compensation.




(a) Photo of the CURA6 robot

(b) Model of the CURA6 robot

Fig. 1 Photo and model of CURA6 robot's

Despite these advances, existing methods still face several limitations: (1) dependency on accurate friction modeling which is time-consuming and varies with operating conditions, (2) requirement for multiple external sensors increasing system complexity and cost, (3) limited adaptability to varying payload conditions, and (4) trade-offs between detection sensitivity and false positive rates.

The proposed method addresses these limitations by: (1) eliminating the need for complex friction modeling through advanced machine learning techniques, (2) utilizing only motor current measurements and robot dynamic models, reducing sensor requirements, (3) demonstrating superior performance under varying load conditions, and (4) achieving better balance between detection accuracy and computational efficiency through optimized SVMR and 1D CNN implementations.

The study proposes using SVMR and 1D CNN for collision detection, offering a cost-effective and efficient solution by eliminating external sensors. This research contributes to enhancing safety and productivity in collaborative robotics, underscoring its importance in modern manufacturing.

The paper is structured to cover foundational robot dynamics (Section 2), methodology (Section 3), experimental results (Section 4), and conclusions with future directions (Section 5), ensuring comprehensive insights for experts and general readers.

## 2. Experimental setup and method

The CURA6 robot, developed by Intema, exemplifies advanced collaborative robotics, particularly within Industry 4.0. Designed for precision and delicate operations, CURA6 integrates seamlessly into industrial environments, showcasing a sophisticated understanding of human-robot interaction.

Key components of the CURA6 include its base, rotational and translational joints, connecting links, and a highly precise end-effector, or “robot hand” Notably, the CURA6 is compact and lightweight, weighing only 5,000 g, with an operational radius of 1,200 mm. This design enables the robot to perform complex tasks, particularly with fragile objects prone to deformation or breakage under stress.

The kinematics of CURA6 are defined using Denavit-Hartenberg (D-H) parameters,[3] following the modified DH convention (Craig notation), which include $a_{i-1}$ (link length along $X_{i-1}$ axis), $\alpha_{i-1}$ (link twist around $X_{i-1}$ axis), $d_i$ (link offset along $Z_i$ axis), and $\theta_i$ (joint angle around $Z_i$). These parameters establish the robot's spatial configuration, range of motion, and operational capabilities, underpinning its versatility and precision in industrial applications.

The D-H parameters are essential to the CURA6 robot’s precision and efficiency, enabling accurate control of its movements for diverse industrial applications. These parameters enhance the robot’s adaptability, making it ideal for tasks ranging from handling delicate materials to complex assembly operations.

The CURA6 also exemplifies the integration of advanced engineering with ergonomic design, mimicking the human arm to provide intuitive and safe human-robot interaction. This fosters a collaborative work environment, enhancing both efficiency and safety. The CURA6 represents a significant advancement in collaborative robotics, combining lightweight, compact design with sophisticated kinematic capabilities. Its precision and versatility make it a key player in the evolving industrial automation sector.

Table 1 D-H parameters table of the CURA6 robot

| Joint $i$ | Distance $a_i$ (m) | Rotation Angle $\alpha_i$ (°) | Distance $d_i$ (m) | Torque $\tau_i$ (Nm) | Rotation Angle $\theta_i$ (°) |
|---|---|---|---|---|---|
| 1 | 0 | $\frac{\pi}{2}$ | 0.105 | $\tau_1$ | $\theta_1$ |
| 2 | 0.4 | 0 | 0 | $\tau_2$ | $\theta_2$ |
| 3 | 0.4 | 0 | 0 | $\tau_3$ | $\theta_3$ |
| 4 | 0 | $\frac{\pi}{2}$ | 0.220 | $\tau_4$ | $\theta_4$ |
| 5 | 0 | $-\frac{\pi}{2}$ | 0.200 | $\tau_5$ | $\theta_5$ |
| 6 | 0 | 2 | 0.140 | $\tau_6$ | $\theta_6$ |

Table 2 Material properties of SCP10

| | |
|---|---|
| Young’s modulus (GPa) | value-added10 |
| Poisson’s ratio | 0.3 |
| Yield Strength (MPa) | 433 |
| UTS (MPa) | 460 |

The dynamic of the CURA6 robot are governed by the following equation:

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)+\tau_F=\tau_m \tag{1}$$

Where, $q(t)=[q_0(t),q_1(t),\ldots,q_{n-1}(t)]INR^n$ represents the joint positions, where $n=6$. $\dot{q}(t)$ and $\ddot{q}(t)$ denote the velocity and acceleration vectors, respectively.

$M(q)INR^{n\times n}$ is the symmetric inertia matrix.

$C(q,\dot{q})INR^{n\times n}$ is the Coriolis matrix.

$g(q)INR^n$ represents the gravitational force.

$\tau_F INR^n$ is the frictional force.

$\tau_m INR^n$ is the joint torque vector

The torque $\tau_m$ is further defined as $\tau_m=K_i i_m$, where $K_i INR^{n\times n}$ is the amplification matrix, and

$i_m INR^n$ is the motor current.

When the robot encounters external forces, the dynamics equation modifies to include the external torque $\tau_{ext} INR^n$:

$$M(q)\ddot{q}+C(q,\dot{q})\dot{q}+g(q)+\tau_F=\tau_m+\tau_{ext} \tag{2}$$

This extension allows the analysis of the robot's response to collisions or external forces, crucial for understanding environmental interactions.

Additionally, the research explores the properties of the matrix $\dot{M}(q)-2C(q,\dot{q})$, noting its skew-symmetric nature $\dot{M}(q)$ represents the time derivative of $M(q)$. This skew-symmetric property highlights internal force and torque counterbalances within the robot, which is essential for dynamic control.

The relation $\dot{M}(q)=C(q,\dot{q})+C^T(q,\dot{q})$ is also examined:

$$\dot{M}(q)=C(q,\dot{q})+C^T(q,\dot{q}) \tag{3}$$

This research segment emphasizes the intricate dynamics of the CURA6 robot, focusing on the impact of external torques and time-dependent changes in the inertia matrix, vital for designing responsive and adaptive robotic systems. For further in-depth analysis, refer.[12,13)]

The total momentum $\rho$ of the robot is a fundamental concept, encompassing both linear and rotational dynamics, essential for understanding and controlling the robot's interactions and task performance. This comprehensive measure accounts for the robot's mass distribution, component velocities, and angular momentum, crucial for predicting maneuverability, stability, and responsiveness to external forces.

Defining total momentum provides insights into the robot's ability to perform precise movements in tasks like delicate assembly or navigating complex environments. Additionally, it aids in optimizing energy efficiency and ensuring smooth operations, forming the basis for further research in motion planning, control systems, and adaptive behavior algorithms.

The total momentum $\rho$ is defined as:

$$\rho=M(q)\dot{q}INR^n \tag{4}$$

The time derivative of $\rho$ is given by:

$$\dot{\rho}=\dot{M}(q)\dot{q}+\dot{M}(q)\ddot{q} \tag{5}$$

or equivalently:

$$\dot{\rho}=\tau_m+\tau_{ext}-\tau_F+C^T(q,\dot{q})\dot{q}-g(q) \tag{6}$$

This segment focuses on the dynamics of the momentum observer, incorporating the term $\beta(q,\dot{q})$ and the matrix $K_O$. The term $\beta(q,\dot{q})=g(q)-C^T(q,\dot{q})\dot{q}$ represents a function of the robot’s state variables, highlighting the gravitational forces $g(q)$ and the Coriolis/centrifugal forces $C^T(q,\dot{q})\dot{q}$. This formulation is critical for understanding the interactions of forces and torques acting on the robot.

The matrix $K_O=diag(k_{O,i})INR^{n\times n}$ is a crucial component in the momentum observer's dynamics, with each diagonal element $k_{O,i}$ serving as a tuning parameter that influences the observer's speed and accuracy in state estimation. The dynamics of the momentum observer are governed by:

$$\dot{\hat{p}}=\tau_m-\tau_F-\hat{\beta}(q,\dot{q})+r_m \tag{7}$$

$$\dot{r}_m=K_O(\dot{p}-\dot{\hat{p}}) \tag{8}$$

The total momentum p of the robot is a fundamental concept, encompassing both linear and rotational dynamics, essential for understanding and controlling the robot's interactions and task performance. Here, p represents the generalized momentum vector (not to be confused with density ρ ), which accounts for the robot's mass distribution, component velocities, and angular momentum.

Where $r_m(t) INR^n$ is the observer's output [13]. This output can be expressed as:

$$r_m = K_O\left(\rho(t) - \int_0^t \left(\tau_m - \tau_F - \hat{\beta} + r_m\right) ds - \rho(0)\right) \quad (9)$$

with $\rho = \widehat{M}(q)\dot{q}$ and $\hat{\beta} = \hat{g}(q) - \widehat{C}^T(q, \dot{q})\dot{q}$ where $\widehat{M}, \widehat{C}$ and $\hat{g}$ are estimated model parameters. To derive the relationship between the observer output and external torques, we substitute Equations (6) and (7) into Equation (8), assuming ideal modeling conditions where Ĝ(q) = G(q) and ĉ(q,q̇) = c(q,q̇). Under ideal modeling conditions and substituting the robot dynamics from Equation (6). Rearranging for the acceleration term:

Substituting this into the momentum observer equation and assuming ideal conditions (Ĝ=G and ĉ=c), the relationship between r̂ and the external joint torque $\tau_{text}$ becomes:

$$\dot{r}_m = K_O(\tau_{ext} - r_m) \quad (10)$$

This formulation allows $r_m(t)$ to estimate $\tau_{ext}$.

The detection of external joint torques involves comparing these estimates against user-defined thresholds $\epsilon_{cd} INR^n$ The collision detection function $cd: r_m(t) \rightarrow \{True, False\}$ is defined as:

$$\epsilon_{cd}(r_m(t)) = \begin{cases} True \text{ if } |r_m(t)| > \epsilon_{cd} \\ False \text{ if } |r_m(t)| \le \epsilon_{cd} \end{cases} \quad (11)$$

This binary function determines whether a collision has occurred by indicating if the estimated torque surpasses the predefined thresholds[7,14]. This approach enables the robotic system to distinguish between normal operational forces and potential collisions, triggering safety protocols if necessary. Such precise and efficient collision detection is vital for ensuring safety in environments with frequent human-robot interaction.

## 3. Proposed Method

This study utilized a dataset comprising fifteen distinct motion sequences of the CURA6 robot, each containing 10,000 samples, amounting to approximately seven minutes of motion per sequence. The robot's joint speed was controlled within 25% of the maximum motor speed to ensure safe operation. Three sequences were recorded with no load, while the remaining twelve sequences included loads ranging from 780g to 4,059 g, with two sequences per specified load. This dataset distribution aims to simulate real-world scenarios under varying conditions. For model training, 90% of the dataset was designated for learning and 10% for validation. The dataset is accessible in the public GitLab repository (https://gitlab.com/intemagdansk/cura6-dataset/-/tree/main).

Each sample's normalization is performed to fit within the [0,1] range, ensuring consistency. Unlike traditional approaches that require explicit friction modeling, our proposed method operates under simplified dynamic conditions without friction compensation. This design choice offers several advantages:

In practice, modeling errors are inevitable due to parameter uncertainties, unmodeled dynamics, and manufacturing tolerances. The proposed method's robustness to model uncertainties can be analyzed as follows:

When model parameters deviate from their true values, the estimated model parameters $\hat{G}(q)$ and $\hat{c}(q,\dot{q})$ in Equation (10) become:

$$\hat{G}(q) = G(q) + \triangle G(q) \tag{12}$$

$$\hat{c}(q,\dot{q}) = c(q,\dot{q}) + \triangle c(q,\dot{q}) \tag{13}$$

Where $\Delta G(q)$ and $\Delta c(q,\dot{q})$ represent modeling errors. Under these conditions, the observer output $\hat{r}$ becomes:

$$\hat{r} = \tau_{ext} + K_{o^{-1}}\left[\triangle G(q)\ddot{q} + \triangle c(q,\dot{q})\right] \tag{14}$$

The modeling error term $K_{o^{-1}}\left[TRIANGLEG(q)\ddot{q} + TRIANGLEc(q,\dot{q})\right]$ acts as additional noise in the collision detection system. However, the machine learning approaches (SVMR and 1D CNN) demonstrate inherent robustness to such uncertainties through:

1) Training Data Diversity: The dataset includes various loading conditions (780 g to 4059g), naturally incorporating model variations.

2) Feature Learning: Both SVMR and 1D CNN learn to distinguish between model uncertainties and actual collision events through pattern recognition.

3) Threshold Optimization: The detection thresholds are optimized considering the presence of model uncertainties in the training data.

The output is modeled by the dynamic response without friction as per the following equations:

$$r_m = K_O\left(\rho(t) - \int_0^t \left(\tau_m - \hat{\beta} + r\right) ds - \rho(0)\right) \tag{15}$$

with

$$\dot{\hat{\rho}} = \tau_m - \hat{\beta}(q,\dot{q}) + r \tag{16}$$

or

$$\dot{r} = K_O\left(\dot{\rho} - \dot{\hat{\rho}}\right) \tag{17}$$

and under conditions without friction, the relationship between the external torque $\tau_{ext}$ and the state variable $r(t)$ is given by:

$$\dot{r} = K_O\left(\tau_{ext} - \tau_F - r\right) \tag{18}$$

Advantages of Friction-Free Modeling:

1) Reduced Computational Complexity: Eliminates the need for complex friction parameter identification and real-time friction compensation.

2) Enhanced Robustness: Friction models are highly dependent on operating conditions (temperature, wear, lubrication), making friction-free approaches more stable across different environments.

3) Simplified Implementation: Reduces the number of parameters requiring calibration and maintenance.

4) Machine Learning Compensation: The SVMR and 1D CNN models implicitly learn to handle friction effects through training data, providing adaptive compensation without explicit modeling.

This approach contrasts with traditional methods where friction compensation requires: (1) extensive parameter identification procedures, (2) continuous model updating for changing conditions, and (3) complex nonlinear friction models that may introduce additional uncertainties.

This research extends previous methodologies such as Support Vector Machine Regression (SVMR)[15] and One-Dimensional Convolutional Neural Networks (1D CNN)[16,17] to develop a collision detection system for the CURA6 robotic arm. A time-varying signal $s(t) \in R^f$ was sampled at intervals $t_I$ over a time window $t_W$, yielding

$N+1=\frac{t_W}{t_I}+1$ samples. This setup captures the robot's dynamic response over time for model training and operation.

The SVMR model is adapted for regression analysis to estimate the likelihood of collisions based on observed movement. The CNN is employed for sequential pattern recognition, processing time-series data for anomalous patterns that may signify potential collisions. Combining these methods, the system enhances detection sensitivity to ensure safety in complex environments. Rigorous parameter adjustments, such as $t_W$ and $t_I$, were implemented to optimize the collision detection model, utilizing $f$-dimensional signals encompassing positions, velocities, and torques.

The SVMR model, using the absolute value from Equation (12) as input, designs a feature vector by transforming sampled signals. The authors set $t_W=80ms$ and $t_I=8ms$, converting observer outputs into a feature matrix $R\in R^{n\times(N+1)}$ and creating a feature vector $x(t)$:

$$x=[r_{J_1}\ r_{J_2}\ r_{J_3}\ r_{J_4}]^T INR^{4(N+1)} \tag{16}$$

The SRQ kernel function[18] is utilized for SVMR training, defined as:

$$K_{SRQ}(x,z)=\frac{1}{\left(1+\frac{\parallel x-z\parallel^2}{\sigma^2}\right)^2} \tag{17}$$

Parameters used are $C=1, \epsilon=0.02$, and $\sigma=3$. The SVMR model's sensitivity was calibrated, using a thresholded, normalized output for a binary collision index, enhancing operational reliability (Fig. 2).

The 1D CNN network inputs sampled signals within $t_W=50ms$ and $t_I=10ms$ intervals, forming matrix $R$, which transforms into an input feature matrix $Z(t)$ using Fourier or other transformations:

$$Z=\begin{bmatrix} | & | & & | \\ \left|r_{J_1}^T\right| & \left|r_{J_2}^T\right| & \cdots & \left|r_{J_6}^T\right| \\ | & | & & | \end{bmatrix}\in R^{n\times(N+1)} \tag{18}$$

The CNN architecture, consisting of convolutional layers with both 'Same' and 'Valid' padding, processes data through hierarchical feature extraction, followed by a fully connected layer for classification and collision prediction. This framework offers a refined mapping from raw signals to a binary collision output, aiding real-time applications.

The images (a) and (b) in the photo you uploaded both describe parts of a one-dimensional Convolutional Neural Network (1-D CNN) architecture.

Image (a) illustrates the transformation process of input data through multiple layers of 1-D convolutions, alternating between "Same" and "Valid" padding types, followed by a flatten and fully connected layer, and ending with a continuous filter to generate a collision index.

Image (b) details how a 1-D convolutional layer

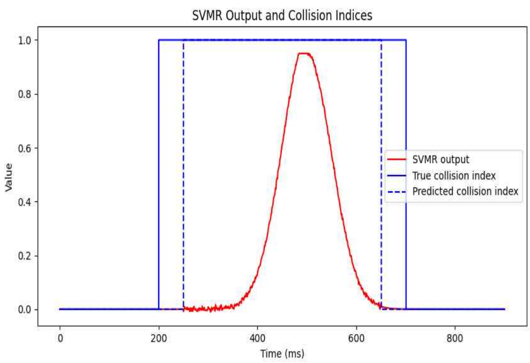
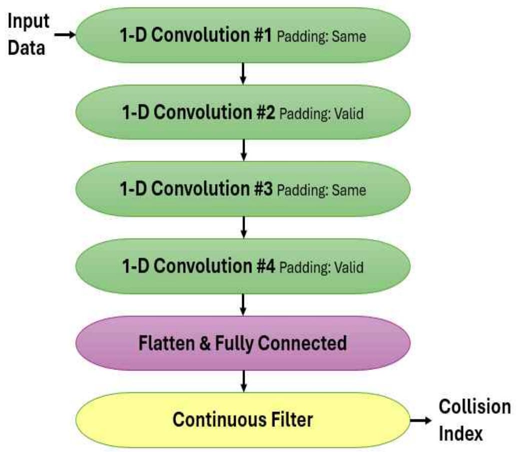


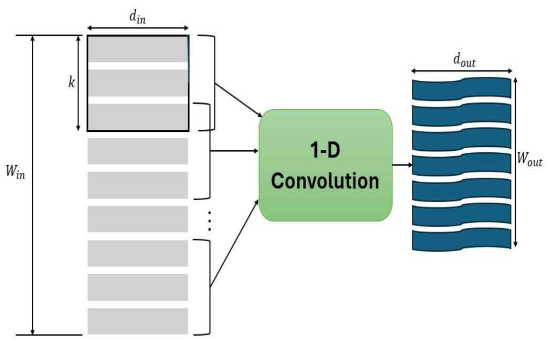

Fig. 2 Output of the SVMR and the Corresponding Collision Index

works, from an input of length d_in, through a filter of size k and width W_in, to produce an output of length d_out and width W_out.

To minimize false positives, an output filter ensures a collision alert is issued only if the detection algorithm maintains a 'True' state for at least $t_c$ milliseconds. Through extensive testing, $t_c = 3ms$ for SVMR and $t_c = 4ms$ for CNN yielded optimal detection reliability. This filtering effectively reduces spurious signals, enabling robust collision detection.




(a) 1D CNN Processing Flow




(b) Detail of 1D Convolutional Layer in CNN

Fig. 3 One-dimensional Convolutional Neural Network (1-D CNN) architecture

## 4. Experiments and Results

This study uses the Precision-Recall Area Under Curve (PRAUC) as a metric for evaluating collision detection performance, defined through Precision and Recall as follows:

$$Precison = \frac{TP}{TP+FP} \tag{19}$$

$$Recall = \frac{TP}{TP+FN} \tag{20}$$

Where $TP$ represents true positives, $FP$ false positives, and $FN$ false negatives, with collisions marked as positive events. High Precision minimizes false positives, while high Recall minimizes false negatives, and thus a high PRAUC value indicates an optimal balance between both.

First, we evaluated the PRAUC for Support Vector Machine Regression (SVMR) by varying the detection threshold $\alpha$. We observed how different $\alpha$ values affected Precision and Recall. For Convolutional Neural Network (CNN) performance, we varied the continuous output filter parameter, $t_c$, analyzing its impact on collision prediction accuracy.

In the SVMR analysis, we systematically adjusted $\alpha$ from -0.08 (predicting all labels as 1) to 0.98 (predicting all as 0). This incremental adjustment allowed for an extensive range of Precision and Recall values, captured in Figure 4a, depicting the SVMR Precision-Recall curve at different threshold settings.

For CNN, we explored $t_c$ values from 1 ms (no filtering) to 619 ms (maximum filtering). Notably, the filtering process only converts predictions of 1 to 0, limiting false positives but also constraining maximum Recall to below 1, as false negatives remain unaffected.

The results in Figure 4 illustrate key performance insights for SVMR and CNN. SVMR achieved a PRAUC of 83.79%, indicating its effectiveness in distinguishing between collision and non-collision events. In comparison, CNN, with optimized filter parameters, achieved a slightly higher PRAUC of 86.81%, demonstrating superior accuracy under the tested conditions. Both models effectively handled collision detection with different threshold and filter parameter configurations.

This research was implemented in Python, benefiting from its flexibility and extensive libraries for machine learning. The experimental setup addressed several challenges, including parameter uncertainties and unmodeled factors like friction and sensor noise. The methods exhibited high reliability and accuracy in collision prediction across varied conditions, underscoring their robustness and suitability for real-world applications.

Extensive computational performance testing was conducted to evaluate the real-time capabilities of both methods. The experiments were performed on an Intel i7-13700KF processor with 32GB RAM, measuring processing time across 1000 collision detection cycles. The results are summarized in Tables 3-5.

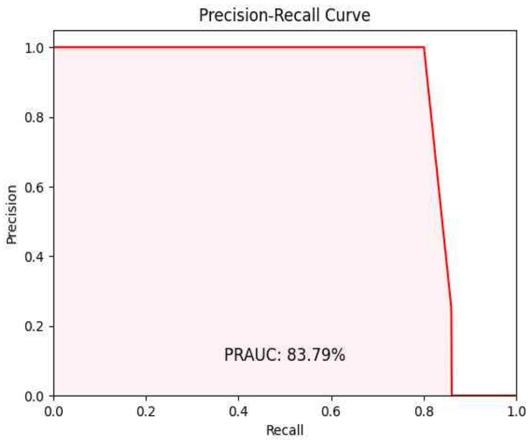



Fig. 4 Results of SVMR and CNN combined with PRAUC

Table 3 Average Processing Time Comparison

| Method | Average Time (ms) | Standard Deviation (ms) | Performance Improvement |
|---|---|---|---|
| SVMR | 12.3 | 1.8 | Baseline |
| 1D CNN | 3.1 | 0.7 | 9.2 ms (75% reduction) |

Table 4 Processing Time Under Different Load Conditions

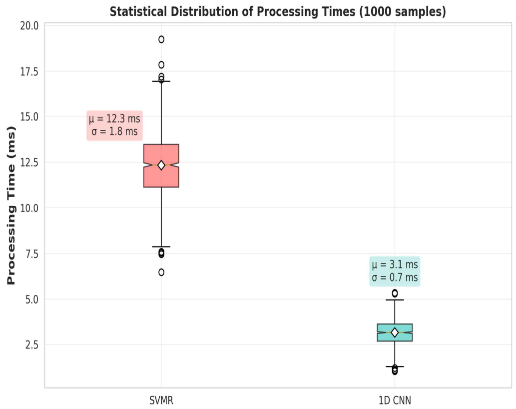
| Load Condition | 1D CNN (ms) | SVMR (ms) | Difference (ms) |
|---|---|---|---|
| No-load | 2.8 | 11.9 | 9.1 |
| Maximum load (4059g) | 3.4 | 12.7 | 9.3 |
| Variable load ($\sigma$ ) | 0.7 | 1.8 | - 0.9 |

Table 5 Processing Time Breakdown by Component

| Method | Component | Time (ms) |
|---|---|---|
| SVMR | Feature extraction | 4.2 |
| | SVM computation | 8.1 |
| | Total | 12.3 |
| 1D CNN | Forward propagation | 2.1 |
| | Post-processing | 1.0 |
| | Total | 3.1 |

The 1D CNN's superior speed stems from: (1) optimized matrix operations in convolutional layers, (2) parallel processing capabilities, and (3) reduced feature extraction overhead compared to SVMR's kernel computations.

Fig. 5 shows results of Statistical Distribution of Processing Times (1000 samples). Figure 6 shows load conditions comparison.




Fig, 5 Results of Statistical Distribution of Processing Times (1000 samples)

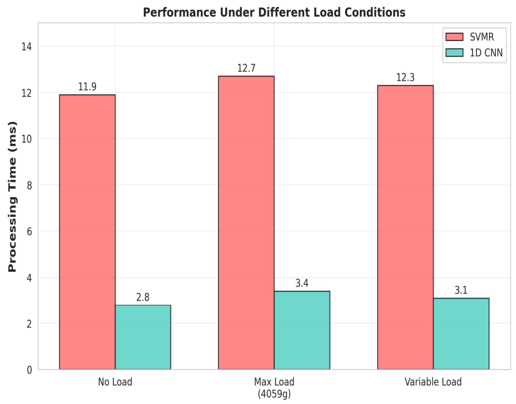



Fig. 6 Performance Under Different Load Conditions

## 5. Conclusions

Our developed collision detection method, combining Support Vector Machine Regression (SVMR) and One-Dimensional Convolutional Neural Network (1D CNN), has proven highly effective in identifying collision events in the 6-Degrees of Freedom (DoF) CURA6 robot arm. A key advantage of this method lies in its minimal sensor data requirements, relying solely on motor current measurements and the robot's dynamic model. This eliminates the need for complex friction modeling, which is often time-consuming in robotics.

The SVMR method requires adjustment of only one parameter, while the 1D CNN method requires no parameter adjustments, making it highly user-friendly. The 1D CNN approach is about 9 milliseconds faster than SVMR in collision detection, a critical advantage in applications requiring rapid response. However, SVMR is more suitable in data-scarce environments, whereas 1D CNN excels with abundant data, allowing adaptation to various scenarios.

Our approach shows significant improvement over previous studies, particularly in handling impacts with varying loads, enhancing the robot's adaptability and resilience. Nonetheless, challenges remain in validating the method for random and undefined loads, as well as in multi-arm robot scenarios, which require further exploration.

Looking forward, our research focuses on scaling these methods for mass-produced collaborative robots, ensuring they are reliable, easy to integrate, and cost-effective. We aim to refine these methods to accommodate a broader range of scenarios, including collaborative robots working alongside human operators. Additionally, we plan to delve deeper into machine learning and artificial intelligence to enhance the accuracy, speed, and reliability of collision detection. This research opens new avenues for improving the safety and efficiency of robotic systems, with promising applications across diverse fields.

## Acknowledgement

This research was supported by the Regional Innovation System & Education(RISE) program through the RISE Center, Gyeongsangnam-do, funded by the Ministry of Education(MOE) and the Gyeongsangnam-do Provincial Government, Republic of Korea.(2025-RISE-16-001).

## Author contributions

D. A. Pham; formal analysis, software, resources, data curation, writing—original draft preparation, data collected and analyzed, visualization, writing—review and editing; S. H. Han; supervision, project administration, writing—review and editing. J. U. Lee; formal analysis, data curation, writing—original draft preparation, data collected and analyzed, visualization. D. Y. Jeong; formal analysis, data curation, writing-original draft preparation, data collected and analyzed, visualization.

## References

1. H. Ayesha et al., 2023, "Control System Design and Methods for Collaborative Robots", Applied Sciences, 13(1), 675. (https://doi.org/10.3390/app13010675)
2. C. Franklin, 2017, "Standards revisions: robots and robotic systems: the current version of the Industrial Robot Safety Standard, ANSI/RIA R15. 06-2012, is a US national adoption of the ISO 10218-2011, Part 1, Robots, and Part 2, Robotic Systems. Look for new versions of these documents in the 2020 or 2021 timeframe. Also see information on collaborative robots, loading and unloading stations, end-effectors, and lockout and tagout", Control Engineering, 64(3), 42-44.
3. M. Czubenko and Z. Kowalczuk, 2021, "A simple neural network for collision detection of collaborative robots", Sensors, 21(12), 4235. (https://doi.org/10.3390/s21124235)
4. Y. Jiang et al., 2023, "Research on Collision Detection of Collaborative Robot using improved Momentum-based Observer", 2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET), 1-6.
5. Y. J. Heo et al., 2019, "Collision detection for industrial collaborative robots: A deep learning approach", IEEE Robotics and Automation Letters, 4(2), 740-746. (https://doi.org/10.1109/LRA.2019.2893400)
6. K. M. Park et al., 2020, "Learning-based real-time detection of robot collisions without joint torque sensors", IEEE Robotics and Automation Letters, 6(1), 103-110. (https://doi.org/10.1109/LRA.2020.3033269)
7. V. Villani, et al., 2018, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications", Mechatronics, 55, 248-266. (https://doi.org/10.1016/j.mechatronics.2018.02.009)
8. A. Cirillo et al., 2015, "A conformable force/tactile skin for physical human–robot interaction", IEEE Robotics and Automation Letters, 1(1), 41-48. (https://doi.org/10.1109/LRA.2015.2505061)
9. M. W. Strohmayr, H. Wörn and G. Hirzinger, 2013, "The DLR artificial skin step I: Uniting sensitivity and collision tolerance", 2013 IEEE International Conference on Robotics and Automation (ICRA), 5136-5141.
10. A. Sharkawy, P. N. Koustoumpardis and N. Aspragathos, 2020, "Neural network design for manipulator collision detection based only on the joint position sensors", Robotica, 38(10), 1737-1755. (https://doi.org/10.1017/S0263574719000985)
11. Z. Zhang et al., 2020, "An online robot collision detection and identification scheme by supervised learning and bayesian decision theory", IEEE Transactions on Automation Science and Engineering, 18(3), 1144-1156. (https://doi.org/10.1109/TASE.2020.2997094)
12. K. Narukawa et al., 2017, "Real-time collision detection based on one class SVM for safe movement of humanoid robot", 2017 IEEE-RAS 17th International Conference on Humanoid

Robotics (Humanoids), 786-791.
13. K. M. Lynch and F. C. Park, 2017, "Modern robotics", Cambridge University Press.
14. C. Gaz, E. Magrini and A. De Luca, 2018, "A model-based residual approach for human-robot collaboration during manual polishing operations", Mechatronics, 55, 234-247. (https://doi.org/10.1016/j.mechatronics.2018.02.014)
15. A. J. Smola and B. Schölkopf, 2004, "A tutorial on support vector regression", Statistics and Computing, 14, 199-222. (https://doi.org/10.1023/B:STCO.0000035301.49549.88)
16. Y. Lecun et al., 1998, "Gradient-based learning applied to document recognition", Proceedings of the IEEE, 86(11), 2278-2324. (https://doi.org/10.1109/5.726791)
17. D. Palaz, R. Collobert and M. Magimai Doss, 2013, "Estimating phoneme class conditional probabilities from raw speech signal using convolutional neural networks", arXiv preprint arXiv:1304.1018. (https://doi.org/10.48550/arXiv.1304.1018)
18. M. G. Genton, 2001, "Classes of kernels for machine learning: a statistics perspective", Journal of Machine Learning Research, 2(Dec), 299-312
19. A. Roy and S. Chakraborty, 2023, "Support vector machine in structural reliability analysis: A review", Reliability Engineering & System Safety, 233, 109126. (https://doi.org/10.1016/j.ress.2023.109126)
20. M. Krichen, 2023, "Convolutional neural networks: A survey", Computers, 12(8), 151. (https://doi.org/10.3390/computers12080151)