Journal of Computer Science and Cybernetics, V.xx, N.x (2025), 1-
DOI: 10.15625/1813-9663/21613

COLLISION DETECTION FOR COBOTS: A NEW EFFECTIVE ALGORITHM

VAN-HUNG NGHIEM[1,2,*], SUBRATA CHOWDHURY[3]

[1] *Graduate University of Science and Technology, Vietnam Academy of Science and Technology (VAST), 18 Hoang Quoc Viet Street, Cau Giay District, Ha Noi City, Viet Nam*
[2] *University of Technology - Logistics of Public Security (UTLPS), 4 Lac Long Quan Street, Thuan Thanh Town, Bac Ninh Province, Viet Nam*
[3] *Dept. of Computer Science and Engineering, Sreenivasa Institute of Technology and Management Studies(A), Chittoor, Andhra Pradesh, India*

Abstract. Collaborative robots (Cobots) can interact with humans, simultaneously promoting the advantages of both humans and cobots to increase work efficiency. Cobots operate friendly and interact with humans because they are programmed to detect collisions safely. Therefore, the requirement to accurately and quickly detect collisions of cobot arms is a topic that attracts the attention of many researchers. This paper proposes an approach applying a supervised machine learning technique, SVR (Support Vector Regression), to detect collisions with CURA6 cobot arms. A real-world experiment (the Intema dataset) was used to validate the proposed method.

Keywords. Collision detection, cobot, 6-DoF.

1. INTRODUCTION

Cobots are robots that can work simultaneously with human [1]. Unlike industrial robots, cobots are designed to ensure safe interaction with humans as well as collaborative working environments. So, cobots are made of lightweight materials and designed with a shape that simulates the human arm. Cobots work in coordination with humans to increase production efficiency or overcome technical limitations without the need for physical protective devices such as shields, nets, and protective fences like industrial robots [2].

Because cobots work with human in a defined collaborative workspace, they should have a collision detection function that can detect collisions in man-cobot interaction. There have been many publications on the collision detection problem with the research object being the cobot arm [3–6]. A typical example is the 6-DoF (Six Degrees of Freedom) cobot arm designed to perform precise and safe tasks with humans. Figure 1 describes the Cooperative Universal Robotic Assistant 6 (CURA6) cobot arm by Intema [3].

This paper proposes using SVR (Support Vector Regression) with a suitable feature vector to detect collisions of CURA6 cobot arm. The advantages of the proposed method

*Corresponding author.
E-mail addresses: nghiemvanhung1985@gmail.com (N.V.Hung); subrata895@gmail.com (S.Chowdhury).

are not requiring additional external sensors, not requiring friction compensation, and having low computational cost.

The main contributions of the paper are as follows:

- Propose using a new and robust feature vector - x (consists the monitored signal with suitable time sampling - details in Section 3).
- Propose using a continuous output filter before labeling collision index.

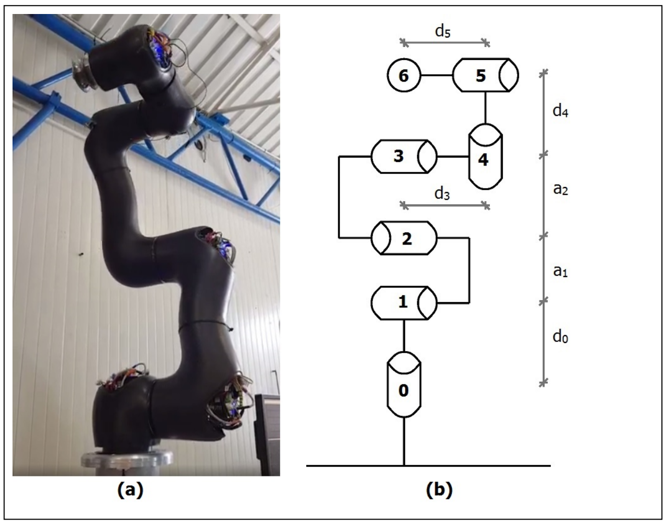

Figure 1: (a) The CURA6 cobot at Intema's lab; (b) Model of the CURA6 cobot [3]

The structure of this paper is organized as follows: Section 1 introduces cobot and collision detection for cobot. Section 2 introduces related research works. In Section 3, we provide an overview of cobot dynamics, and describe the proposed collision detection method. The experimental results and discussions are presented in Section 4. Finally, Section 5 provides conclusions and future research directions.

2. RELATED WORK

Haddadin et al. [7] published a comprehensive study on the problems of cobot collision. The collision problems include: collision detection phase, collision isolation phase, collision identification phase, collision classification phase and collision reaction phase (Figure 2 illustrates phases of the collision problems). This work focuses on the collision detection problem, which is the first step. This step will orient other tasks in the collision problems. The collision detection methods on cobot arms can be divided into (1) using machine learning and (2) not using machine learning. The group of methods (2) often relies on tactile surface sensors (sensor skins) covering the outside of the cobot [8,9]. However, tactile surface sensors are often expensive, not durable, and can be damaged when subjected to strong or repeated collisions. Machine learning-based methods [3,5,7] typically estimate the external joint torques due to collisions and then compare them with a set of pre-determined threshold values to check whether a collision has occurred. Direct torque measurement would be ideal, but torque sensors can be expensive, an alternative is to estimate the joint torque from current measurements at the joint actuators.

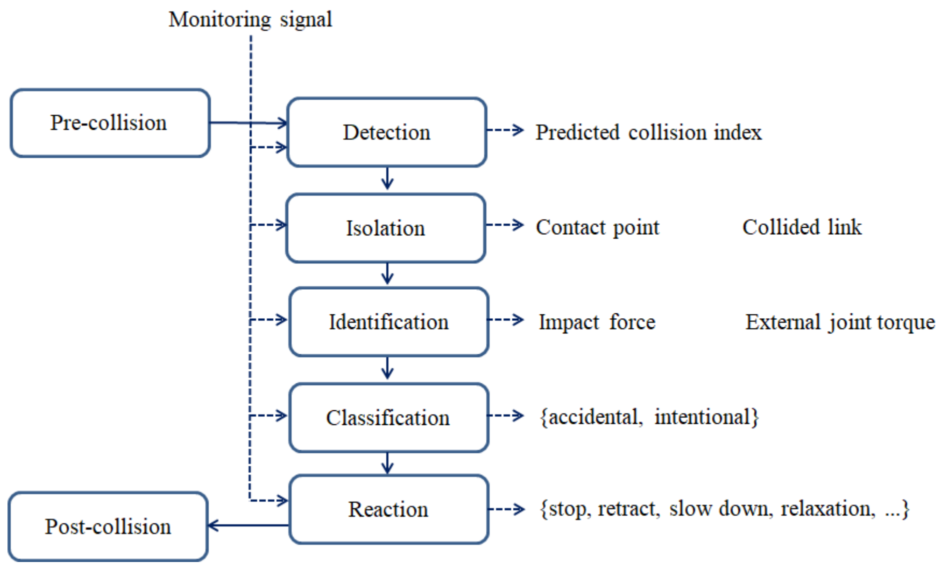

Figure 2: Overview diagram of the phases of the collision problem [7]

Nowadays, collision detection methods using machine learning have become popular [3,5]. With the condition of collecting enough diverse and good-quality data, machine learning methods can overcome the uncertainty of parameters in the dynamics model as well as unmodeled effects such as friction and measurement noise.

Sharkawy et al. [10] used Multilayer Feedforward Neural Network (M-FNN) with joint position and velocity errors as input, and estimated the external joint torque without using cobot dynamics (but also required manual adjustment of collision thresholds for each joint), while [11] considered the measured joint torques and gravitational force vectors of the manipulator. However, these methods were only tested with two- and three-joint movements of a 7-DoF cobot manipulator. Zhang et al. [12] train an FNN with 36-dimensional input signals extracted from both the time and frequency domain features of joint torque sensor measurements. For full-body humanoid cobots, the research group of Narukawa [13] proposes an improved one-class support vector machine (OC-SVM) technique, in which the input is considered to be the moment point vector of the humanoid cobot and the force and torque measurements for each limb.

Recently, Heo et al. [14] proposed a method for fast collision detection of a 6-DoF cobot arm using a one-dimensional convolutional neural network (1D CNN) with size (where the time window size is denoted) in which the estimated external joint torques are input; the output is the binary collision detection result (True, False). Park et al. [5] focused on solving the collision detection problem of the Doosan M0609 cobot arm. Meanwhile, Czubenko and his team [3] proposed a combination of several different neural networks such as AR, RNN, CNN-LSTM, and MC-LSTM to accurately detect collisions of CURA6 cobot arm. However, the common limitation of the methods proposed by Park and Czubenko is the dependence on motor friction, especially Coulomb friction, which is generally difficult to estimate.

There have been many studies on current intensity prediction models in recent years and several techniques have been used in building current intensity prediction models. A widely used technique for current intensity prediction is the Support Vector Regression (SVR) method. This is a method developed from the Support Vector Machine (SVM) algorithm. The basic idea of SVR is to map the input space to a multidimensional feature space where linear regression can be applied (which is ineffective if linear regression is applied directly). The characteristic of SVR is that it gives us a sparse solution; that is, to build a regression function, we do not need to use all the data points in the training set. The points that contribute to building the regression function are called Support Vectors. The classification of a new data point will depend only on the support vectors.

This paper proposes to apply a supervised learning method - SVR - with a suitable feature set, to detect collisions of CURA6 cobot arm based on current measurements along with the cobot dynamics model. Heo [14] trained a 1D CNN with a 66-dimensional signal stack consisting of cobot sensor measurements including estimated external joint torques as input and output the detected binary decomposition of the signal. The validation was performed only on the cyclic movements of the 6-DoF manipulator. Our proposed method improved the learning-based approach to detect cobot stress points using only current sensor measurements similar to its dynamic model. For the collision detection algorithm, we considered SVR while simultaneously using energy estimators and baseline models of the robot hardware without friction terms. Therefore, our approach has the advantage of not requiring threshold tuning and matching points. The proposed method has the advantage of not requiring any other additional external sensors, compared to [12] which requires joint torque sensors at the joints. The proposed method does not require friction compensation. The computational cost in the proposed method is significantly less for both training and inference (4-dimensional input for SVR) while the method of Heo [14] has 66-dimensional input.

3. PROPOSED METHOD

3.1. Structure of the cobot arm

The main components of a CURA6 cobot [3] include: base-component, rotating-component or translational joints, links between joints and the cobot arm, also known as the end effector. Intema's CURA6 cobot arm is compact with a payload of 5000 grams and a wide operating radius of 1200 mm. The CURA6 cobot can perform complex tasks with high precision requirements such as handling and contacting objects that are easily deformed or easily torn or broken under operating conditions with very strict technical requirements.

Symbols in Table 1:

a_i: distance parameter from axis Z_{i-1} to axis Z_i measured along the axis X_i.

α_i: parameter of axis rotation Z_{i-1} around axis X_i parallel or coincident with axis Z_i.

d_i: distance parameter from axis X_{i-1} and axis X_i measured along the axis Z_{i-1}.

θ_i: rotation angle parameter of axis X_{i-1} around axis Z_{i-1} parallel or coincident with axis X_i.

3.2. Dynamic equations

The dynamic equations defined for the CURA6 cobot are

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

Table 1: Denavit - Hartenberg (D-H) parameters of the CURA6 cobot [3]

i	Distance a_i	Parameter α_i	Distance d_i	Parameter τ_i	Parameter θ_i
0	0	$\frac{\pi}{2}$	0.105	τ_0	θ_0
1	0.4	0	-	τ_1	θ_1
2	0.4	0	-	τ_2	θ_2
3	0	$\frac{\pi}{2}$	0.220	τ_3	θ_3
4	0	$-\frac{\pi}{2}$	0.200	τ_4	θ_4
5	0	0	0.140	τ_5	θ_5

where $q(t) = [q_0(t), q_1(t), ..., q_{n-1}(t)] \in \mathbb{R}^n$ includes the location of each joint ($n = 6$), $\dot{q}(t) \in \mathbb{R}^n$ and $\ddot{q}(t) \in \mathbb{R}^n$ represent the velocity and acceleration vectors respectively. $M(q) \in \mathbb{R}^{n\times n}$ is the cobot inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n\times n}$ is the Coriolis matrix, $g(q) \in \mathbb{R}^n$ is gravity, $\tau_F \in \mathbb{R}^n$ represents friction and $\tau_m \in \mathbb{R}^n$ is the joint moment vector. Note that matrix $M(q)$ is symmetric; $\tau_m = K_i i_m$ and $K_i \in \mathbb{R}^{n\times n}$ is the amplification matrix and $i_m \in \mathbb{R}^n$ is the intensity motor. When a collision occurs, equation (1) becomes

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

where $\tau_{ext} \in \mathbb{R}^n$ is the external joint torque caused by collisions or other external forces acting on the cobot.

Matrix $\dot{M}(q) - 2C(q, \dot{q})$ is obliquely symmetrical, with $\dot{M}(q)$ being the derivative symbol of $M(q)$ over time (details in [15]).

The signals need to be normalized in the range [0,1] before being processed as input. The output without considering friction is described by the equation

$$r = K_O\left(\rho(t) - \int_0^t \left(\tau_m - \hat{\beta} + r\right) ds - \rho(0)\right), \tag{3}$$

with

$$\dot{\hat{\rho}} = \tau_m - \hat{\beta}(q, \dot{q}) + r, \tag{4}$$

and

$$\dot{r} = K_O\left(\dot{\rho} - \dot{\hat{\rho}}\right), \tag{5}$$

Under ideal conditions, the dynamic relationship between $r(t)$ and external joint torque τ_{ext} is

$$\dot{r} = K_O\left(\tau_{ext} - \tau_F - r\right). \tag{6}$$

3.3. Collision detection method

Based on the SVR method [16], we research and develop a collision detection method based on our feature extraction scheme (Figure 3) for the CURA6 cobot arm.

Convention: A time-varying f-dimensional signal $s(t) \in \mathbb{R}^f$ sampled at the sampling interval t_I over a time window of size t_W, number of samples is $N + 1 = \frac{t_W}{t_I} + 1 \in \mathbb{N}$.

SVR takes the absolute value of the output in (3) as input.

Feature vector includes converting the signal into a vector form and assigning labels to the features. To sample over time, through experimentation, we set time sampling $t_I = 7$ ms and $t_W = 70$ ms. After sampling the output of the momentum observer and transforming it into a matrix $R \in \mathbb{R}^{n\times(N+1)}$, value of rows 1, 2, 3 and 4 of R to form a single vector, resulting in feature vectors $x\left(t\right)$ as below

$$x = \begin{bmatrix} r_{J_1} & r_{J_2} & r_{J_3} & r_{J_4} \end{bmatrix}^T \in \mathbb{R}^{4(N+1)}, \tag{7}$$

where $r_{J_i} \in \mathbb{R}^{1(N+1)}$ represents the value of the ordered row i of R. Note that row i of R represents the sampled momentum observation output of the joint i. Compared with using the signals of all joints, the size of the feature vector x is smaller, reducing the amount of computation for SVR.

SVR training: We propose to use the radial basis function [17] as below

$$K\left(x, x^{'}\right) = \exp\left(\frac{\left\|x - x^{'}\right\|^2}{2\sigma^2}\right), \tag{8}$$

SVR hyperparameter is $\sigma = 3$ in (8).

Collision detection sensitivity adjustment: The output of the trained SVR is a scalar value normalized to a unit which is then thresholded and converted to the corresponding binary collision. The output value is close to 0 when no external force is applied and increases as the estimated external torque of joints 1–4 increases and decreases as the external torque decreases. The larger the estimated external torque and the more joints affected by the external force, the larger the SVR output value.

We filter the output to reduce false positives (*false alarms*). Collisions are only reported if the output of the collision detection method is consistently *true* for t_c ms. The output filter eliminates false positives (false alarms with long durations in the $1 - t_c$ ms range) which makes collision detection more efficient. We set $t_c = 0.3$ ms for the SVR method.

We programmed and tested the proposed method in Python.

3.4. Evaluation method

The confusion matrix evaluates the classifier's performance by comparing real and predicted outputs (Figure 3). Four indicators including *Accuracy*, *Sensitivity*, *PPV* (Positive Predictive Value), and *Specificity* are utilized to evaluate the performance of the proposed method.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}, \tag{9}$$

$$Sensitivity = \frac{TP}{TP + FN}, \tag{10}$$

$$PPV = \frac{TP}{TP + FP}, \tag{11}$$

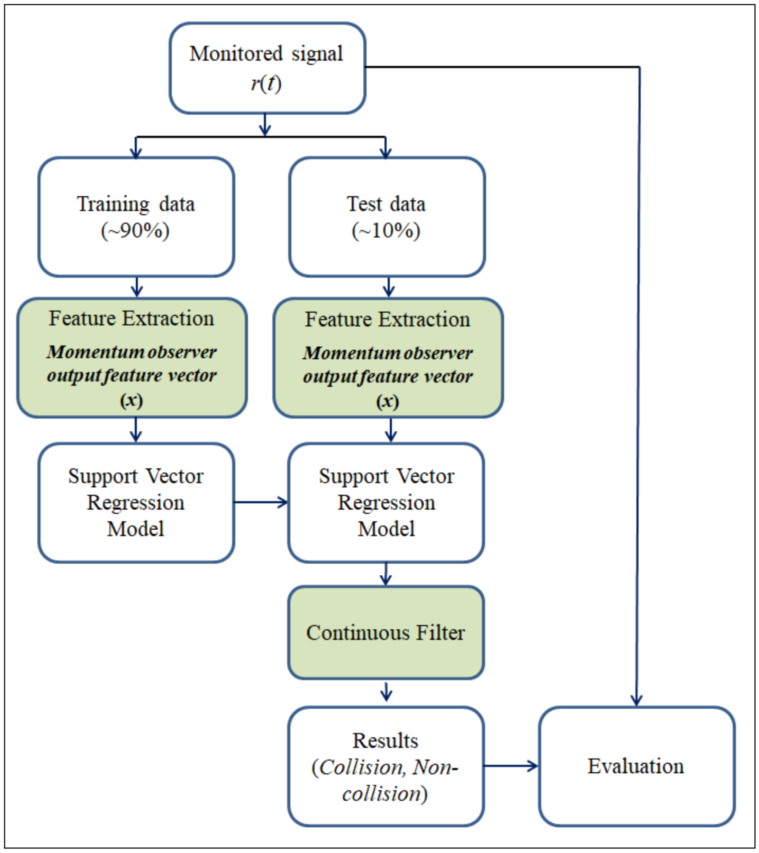

Figure 3: Constructive process of the proposed SVR method

$$Specificity = \frac{TN}{TN + FP}, \tag{12}$$

where TP - True Positive: A collision happened, the model predicted it happened, FP - False Positive: A collision didn't happen, the model predicted it happened, FN - False Negative: A collision happened, the model predicted didn't happen, TN - True Negative: A collision didn't happen, the model predicted didn't happen.

4. RESULTS AND DISCUSSIONS

Drawing from the theoretical framework and the model presented, this section conducts experiments on various datasets and assesses the efficiency of the suggested method by utilizing standard metrics in collision detection and predicting relationships between methods.

4.1. Experiment data

We 15 prepared random cobot motion slices as training data, each slice containing 10000 samples (7 minutes of motion). The speed of each joint is safely designed by the solver to be within 25% of the maximum motor speed. There are 03 slices collected without the cobot payload and the remaining 12 slices with payload (in grams) from the following list {4039, 2757, 2298, 1282, 1016, 782}, with 02 slices for each payload (approximate payload range is from 500 grams to 4000 grams). Of which, training data (90%) and test data (10%). This dataset is downloaded from Gitlab [3]. The dataset consists of two sub-datasets, the 1.6 threshold dataset [A] and the 2.0 threshold dataset [A].

4.2. Results

This section presents the experimental outcomes, as well as discussions and comparisons with baseline or recently published studies, to highlight the advantages and disadvantages of the proposed approach. In particular, it encompasses the experimental findings related to collision detection and the prediction of relationships between objects.

Table 2: Confusion matrix: Threshold 1.6 [A], SVR model. A: max V 10%, B: max V 20%, C: max V 30%, D: max V 40%, E: max V 50%, F: max V 60%

(A) max V 10%	**R/a Collision (10%)**	**R/a Non-co (10%)**
Predicted Collision (10%)	60	29
Predicted Non-co (10%)	5	113
(B) max V 20%	**R/a Collision (20%)**	**R/a Non-co (20%)**
Predicted Collision (20%)	59	40
Predicted Non-co (20%)	6	135
(C) max V 30%	**R/a Collision (30%)**	**R/a Non-co (30%)**
Predicted Collision (30%)	57	76
Predicted Non-co (30%)	8	136
(D) max V 40%	**R/a Collision (40%)**	**R/a Non-co (40%)**
Predicted Collision (40%)	60	65
Predicted Non-co (40%)	5	158
(E) max V 50%	**R/a Collision (50%)**	**R/a Non-co (50%)**
Predicted Collision (50%)	59	90
Predicted Non-co (50%)	6	180
(F) max V 60%	**R/a Collision (60%)**	**R/a Non-co (60%)**
Predicted Collision (60%)	59	87
Predicted Non-co (60%)	6	167

Table 2 shows the confusion matrix in different max V of Threshold 1.6 [A]. All cases have 65 actual collisions. The number of non-collision observations varies.

In 207 observations (Table 2.A), the results of the proposed method are: 60/65 collision observations detected correctly, 113/142 non-collision observations detected correctly; only 34 observations were mistaken (5 collision observations were mistakenly predicted as non-collision, and 29 non-collision observations were mistakenly predicted as collision).

In 240 observations (Table 2.B), the results of the proposed method are: 59/65 collision observations detected correctly, 135/175 non-collision observations detected correctly; only 46 observations were mistaken (6 collision observations were mistakenly predicted as non-collision, and 40 non-collision observations were mistakenly predicted as collision).

In 277 observations (Table 2.C), the results of the proposed method are: 57/65 collision observations detected correctly, 136/212 non-collision observations detected correctly; 84 observations were mistaken (8 collision observations were mistakenly predicted as non-collision, and 76 non-collision observations were mistakenly predicted as collision).

In 288 observations (Table 2.D), the results of the proposed method are: 60/65 collision observations detected correctly, 158/223 non-collision observations detected correctly; 70 observations were mistaken (5 collision observations were mistakenly predicted as non-collision, and 65 non-collision observations were mistakenly predicted as collision).

In 335 observations (Table 2.E), the results of the proposed method are: 59/65 collision observations detected correctly, 180/270 non-collision observations detected correctly; 96 observations were mistaken (6 collision observations were mistakenly predicted as non-collision, and 90 non-collision observations were mistakenly predicted as collision).

In 319 observations (Table 2.F), the results of the proposed method are: 59/65 collision observations detected correctly, 167/254 non-collision observations detected correctly; 93 observations were mistaken (6 collision observations were mistakenly predicted as non-collision, and 87 non-collision observations were mistakenly predicted as collision).

In summary, we can see that with threshold 1.6 [A], the results in collision detection were good, but the results in non-collision detection are not as good as expected.

Table 3: Confusion matrix: Threshold 2.0 [A], SVR model. A: max V 10%, B: max V 20%, C: max V 30%, D: max V 40%, E: max V 50%, F: max V 60%

(A) max V 10%	**R/a Collision (10%)**	**R/a Non-co (10%)**
Predicted Collision (10%)	51	3
Predicted Non-co (10%)	14	113
(B) max V 20%	**R/a Collision (20%)**	**R/a Non-co (20%)**
Predicted Collision (20%)	50	4
Predicted Non-co (20%)	15	135
(C) max V 30%	**R/a Collision (30%)**	**R/a Non-co (30%)**
Predicted Collision (30%)	53	10
Predicted Non-co (30%)	12	136
(D) max V 40%	**R/a Collision (40%)**	**R/a Non-co (40%)**
Predicted Collision (40%)	56	10
Predicted Non-co (40%)	9	159
(E) max V 50%	**R/a Collision (50%)**	**R/a Non-co (50%)**
Predicted Collision (50%)	56	16
Predicted Non-co (50%)	9	180
(F) max V 60%	**R/a Collision (60%)**	**R/a Non-co (60%)**
Predicted Collision (60%)	59	12
Predicted Non-co (60%)	6	167

Table 3 shows the confusion matrix in different max V of Threshold 2.0 [A]. All cases have 65 actual collisions. The number of non-collision observations varies. Table 3.A shows the results of the proposed method when max V 10% and Threshold 2.0 [A].

In 181 observations, there are: 51/65 collision observations detected correctly, 113/116 non-collision observations detected correctly; only 17 observations were mistaken (14 collision observations were mistakenly predicted as non-collision, and 3 non-collision observations were mistakenly predicted as collision).

Table 3.B shows the results of the proposed method when max V 20% and Threshold 2.0 [A]. In 204 observations, there are: 50/65 collision observations detected correctly, 135/139 non-collision observations detected correctly; only 19 observations were mistaken (15 collision observations were mistakenly predicted as non-collision, and 4 non-collision observations were mistakenly predicted as collision).

Table 3.C shows the results of the proposed method when max V 30% and Threshold 2.0 [A]. In 211 observations, there are: 53/65 collision observations detected correctly, 136/146 non-collision observations detected correctly; 22 observations were mistaken (12 collision observations were mistakenly predicted as non-collision, and 10 non-collision observations were mistakenly predicted as collision).

Table 3.D shows the results of the proposed method when max V 40% and Threshold 2.0 [A]. In 234 observations, there are: 56/65 collision observations detected correctly, 159/169 non-collision observations detected correctly; 19 observations were mistaken (9 collision observations were mistakenly predicted as non-collision, and 10 non-collision observations were mistakenly predicted as collision).

Table 3.E shows the results of the proposed method when max V 50% and Threshold 2.0 [A]. In 261 observations, there are: 56/65 collision observations detected correctly, 180/196 non-collision observations detected correctly; 25 observations were mistaken (9 collision observations were mistakenly predicted as non-collision, and 16 non-collision observations were mistakenly predicted as collision).

Table 3.F shows the results of the proposed method when max V 60% and Threshold 2.0 [A]. In 244 observations, there are: 59/65 collision observations detected correctly, 167/179 non-collision observations detected correctly; 18 observations were mistaken (6 collision observations were mistakenly predicted as non-collision, and 12 non-collision observations were mistakenly predicted as collision).

In summary, we can see that with threshold 2.0 [A], the results in non-collision detection were good, but the results in collision detection are not as good as expected.

4.3. Evaluations and discussions

The performance results obtained by the proposed method (calculated by (9), (10), (11), (12)), compared with Czubenko [3] (using the same experimental data), show similar results (see details in Table 4).

In summary, we can see that with Threshold 2.0 [A] the performance of the proposed method when detecting non-collisions is good (see Specificity values), the performance of the proposed method when detecting collisions is a bit lower (see Sensitivity values). This is in contrast to the Threshold 1.6 [A] case.

From the results obtained, it can be seen that the intensity threshold affects the detection of collision/no collision. In this experiment, the Threshold of 2.0 [A] detects current changes worse (so collision detection is worse), the analysis results tend to assume that there is no intensity change. And the number of non-collision observations in the dataset is very large, so it leads to more correctly prediction of non-collision. On the contrary, the Threshold of 1.6 [A]) is more sensitive in detecting intensity changes, so collision detection is better. For this reason, the non-collision observations tend to misunderstood as collision.

Table 4: Evaluation of results and comparison with Czubenko [3]

maxV	**Threshold**	**Sensitivity**		**PPV**		**Specificity**		**Accuracy**	
		Ours	[3]	Ours	[3]	Ours	[3]	Ours	[3]
10%	1.6 [A]	0.923	0.908	0.674	0.670	0.796	0.796	0.836	0.831
20%		0.908	0.923	0.596	0.606	0.771	0.777	0.808	0.817
30%		0.877	0.862	0.429	0.427	0.642	0.646	0.697	0.697
40%		0.923	0.954	0.480	0.492	0.709	0.713	0.757	0.767
50%		0.908	0.908	0.396	0.399	0.667	0.670	0.713	0.716
60%		0.908	0.908	0.404	0.404	0.657	0.657	0.708	0.708
10%	2.0 [A]	0.785	0.800	0.944	0.945	0.974	0.974	0.906	0.912
20%		0.769	0.754	0.926	0.942	0.971	0.978	0.907	0.907
30%		0.815	0.800	0.841	0.852	0.932	0.938	0.896	0.896
40%		0.862	0.877	0.848	0.851	0.941	0.941	0.919	0.923
50%		0.862	0.892	0.778	0.795	0.918	0.923	0.904	0.916
60%		0.908	0.908	0.831	0.831	0.933	0.933	0.926	0.926

The SVR-based collision detection method detected collisions of the 6-DoF CURA6 cobot arm effectively. The advantage of the proposed method is that it only requires measurements of current sensors along with a dynamic model of the cobot; there is no need to model or determine the frictional moments in the joints. The advantage of the SVR-based method proposed is that it is suitable for the context of limited training data.

The test scenarios are limited because the study relies on public datasets provided by the research community. Furthermore, the rationale behind the dataset design is unclear. For instance, although the number of collisions per scenario is fixed, the non-collisions instances are varied. Compared with the research results in [3] and [14], our method is better in the context of handling the impact of different loads. However, the validation for unknown random loads remains a difficult problem [18–22]. Furthermore, the problem of collision detection of two or more cobot arms is complex and needs further research [23–30]. For mass-produced collaborative cobots, another practical issue is that the processes required for effective collision detection must be replicated on a large scale. These will be interesting topics for future research.

5. CONCLUSION

The proposed SVR-based collision detection technique effectively detected collisions of the CURA6 6-DoF cobot arm. The comparison results show that the performance of the proposed method is similar to the existing method. However, the proposed method is suitable for limited data contexts. The main advantage of the proposed method is that it only requires measurements of current sensors along with a dynamic model of the cobot; there is no need to model or determine the frictional moments in the joints.

The proposed method is better than existing methods in the context of handling the impact of different loads. However, the validation for unknown random loads remains a difficult problem. Furthermore, the problem of collision detection between two or more cobot arms is complex and requires further research. For mass-produced collaborative cobots, another practical issue is that the processes required for effective collision detection must be replicated on a large scale. These will be open topics for future research.

ACKNOWLEDGMENT

We express our gratitude to the Intema's Research Group in Gdańsk and the Graduate University of Science and Technology - VAST, for their assistance with this study.

REFERENCES

[1] W. Li, Y. Hu, and Y. Zhou, "Safe human - robot collaboration for industrial settings: a survey," *Journal of Intelligent Manufacturing*, vol. 35, pp. 2235–2261, 2024. [Online]. Available: https://link.springer.com/article/10.1007/s10845-023-02159-4

[2] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957415818300321

[3] M. Czubenko and Z. Kowalczuk, "A simple neural network for collision detection of collaborative robots," *Sensors*, vol. 21, no. 12, pp. 4235–4254, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/12/4235

[4] Y. Jiang, Y. Wu, X. Wu, and Z. B., "Research on collision detection of collaborative robot using improved momentum-based observer," *IAS Global Conference on Emerging Technologies*, pp. 1–6, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10149985

[5] K. M. Park, J. Kim, J. Park, and F. C. Park, "Learning-based real-time detection of robot collisions without joint torque sensors," *Robotics and Automation Letters*, vol. 6, no. 1, pp. 103–110, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9237113

[6] D. Zurlo, T. Heitmann, M. Morlock, and A. De-Luca, "Collision detection and contact point estimation using virtual joint torque sensing applied to a cobot," *International Conference on Robotics and Automation (ICRA)*, pp. 7533–7539, 2023. [Online]. Available: https://ieeexplore.ieee.org/document/10160661

[7] S. Haddadin, A. De-Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1292–1312, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8059840

[8] A. Cirillo, F. Ficuciello, C. Natale, S. Pirozzi, and L. Villani, "A conformable force/tactile skin for physical human–robot interaction," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 41–48, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7346450

[9] M. Strohmayr, H. Wörn, and G. Hirzinger, "The DLR artificial skin step i: Uniting sensitivity and collision tolerance," *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 1012–1018, 2013. [Online]. Available: https://ieeexplore.ieee.org/document/6630697

[10] A. N. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, "Neural network design for manipulator collision detection based only on the joint position sensors," *Robotica*, vol. 38, no. 10, pp. 1737–1755, 2020. [Online]. Available: https://doi.org/10.1017/S0263574719000985

[11] A. Sharkawy, P. N. Koustoumpardis, and N. Aspragathos, "Human–robot collisions detection for safe human–robot interaction using one multi-input–output neural network," *Soft Comput.*, vol. 24, no. 9, pp. 6687–6719, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s00500-019-04306-7

[12] Z. Zhang, K. Qian, B. W. Schuller, and D. Wollherr, "An online robot collision detection and identification scheme by supervised learning and bayesian decision theory," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1144–1156, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9109713

[13] K. Narukawa, T. Yoshiike, K. Tanaka, and M. Kuroda, "Real-time collision detection based on one class SVM for safe movement of humanoid robot," *Proc. IEEE-RAS 17th Int. Conf. Humanoid Robot*, pp. 791–796, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8246962

[14] Y. J. Heo, D. Kim, W. Lee, H. Kim, J. Park, and W. K. Chung, "Collision detection for industrial collaborative robots: A deep learning approach," *IEEE Robot. and Autom. Lett.*, vol. 4, no. 2, pp. 740–746, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8613933

[15] K. Sagar, V. Ramadoss, D. Zlatanov, and M. Zoppi, "Storm: Screw theory toolbox for robot manipulator and mechanisms," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7233–7240, 2020. [Online]. Available: https://dl.acm.org/doi/10.1109/IROS45743.2020.9340991

[16] H. Huang, X. Wei, and Y. Zhou, "An overview on twin support vector regression," *Neurocomputing*, vol. 490, no. 7, pp. 80–92, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231222003101

[17] A. Apsemidis, S. Psarakis, and J. M. Moguerza, "A review of machine learning kernel methods in statistical process monitoring," *Computers Industrial Engineering*, vol. 142, no. 1, 106376, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0360835220301108

[18] C. Bröhl and J. Nelles, "Tam reloaded: A technology acceptance model for human-robot cooperation in production systems," *Proceedings of the HCI International 2016 - Posters' Extended Abstracts*, vol. 617, pp. 97–103, 2016. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-40548-3_16

[19] V. Weistroffer and A. Paljic, "A methodology to assess the acceptability of human-robot collaboration using virtual reality," *Proceedings of the the 19th ACM Symposium on Virtual Reality Software and Technology*, pp. 39–48, 2013. [Online]. Available: https://dl.acm.org/doi/10.1145/2503713.2503726

[20] E. Roesler, J. Meerwein, J. Krueger, and L. Onnasch, "Beyond the default: The effects of adaptable robot speed in industrial human-robot interaction," *HRI'24: Companion of the*

2024 ACM/IEEE International Conference on Human-Robot Interaction, pp. 896–900, 2024. [Online]. Available: https://dl.acm.org/doi/10.1145/3610978.3640578

[21] E. Mariotti, E. Magrini, and A. De-Luca, "Admittance control for human-robot interaction using an industrial robot equipped with a f/t sensor," *Proceedings of the 2019 Int. Conf. Robot. Automat.*, pp. 6130–6136, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8793657

[22] E. Magrini and A. De-Luca, "Human-robot coexistence and contact handling with redundant robots," *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 4611–4617, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8206331

[23] C. C. Chan and C. Tsai, "Collision-free speed alteration strategy for human safety in human-robot coexistence environments," *IEEE Access*, vol. 8, pp. 80 120–80 133, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9072145

[24] C. Gaz, E. Magrini, and A. De-Luca, "A model-based residual approach for human-robot collaboration during manual polishing operations," *Mechatronics*, vol. 55, pp. 234–247, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957415818300369

[25] S. Moe, K. Y. Pettersen, and J. T. Gravdahl, "Set-based collision avoidance applications to robotic systems," *Mechatronics*, vol. 69, pp. 199–215, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957415820300799

[26] C. Miyata, K. Chisholm, J. Baba, and M. Ahmadi, "A limb compliant sensing strategy for robot collision reaction," *IEEE/ASME Trans. Mechatron.*, vol. 21, no. 2, pp. 674–682, 2016. [Online]. Available: https://ieeexplore.ieee.org/document/7312975

[27] N. M. Ceriani, A. M. Zanchettin, P. Rocco, A. Stolt, and A. Robertsson, "Reactive task adaptation based on hierarchical constraints classification for safe industrial robots," *IEEE/ASME Trans. Mechatron.*, vol. 20, no. 6, pp. 2935–2949, 2015. [Online]. Available: https://ieeexplore.ieee.org/document/7102768

[28] E. Villagrossi, L. Simoni, M. Beschi, N. Pedrocchi, A. Marini, L. M. Tosatti, and A. Visioli, "A virtual force sensor for interaction tasks with conventional industrial robots," *Mechatronics*, vol. 50, pp. 78–86, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0957415818300163

[29] Z. Kowalczuk and M. Czubenko, "Intelligent decision-making system for autonomous robots," *Int. J. Appl. Math. Comput. Sci.*, vol. 21, no. 4, pp. 671–684, 2011. [Online]. Available: https://doi.org/10.2478/v10006-011-0053-7

[30] T. Varhegyi, M. Melik-Merkumians, M. Steinegger, G. Halmetschlager, and G. Schitter, "A visual servoing approach for a six degrees-of-freedom industrial robot by RGB-D sensing," *Proceedings of the OAGMARW Joint Workshop Vision, Automation and Robotics*, pp. 74–79, 2017. [Online]. Available: https://doi.org/10.3217/978-3-85125-524-9-14

Received on September 27, 2024
Accepted on November 28, 2024