




Case Study

Adaptive threshold for robot manipulator collision detection using fuzzy system

Omar Abdelaziz^{1,2,4}  · Minzhou Luo^{1,3,4} · Guanwu Jiang^{1,4} · Saixuan Chen^{1,4}

Received: 18 April 2019 / Accepted: 27 January 2020 / Published online: 1 February 2020
© Springer Nature Switzerland AG 2020

Abstract

The topic of collision detection was recently addressed in the literature due to the demand of safe human–robot collaboration. Several methods have been proposed for this task; some of these methods use external sensors while others do not (sensor-less). The classical sensor-less method is attained by setting a threshold to detect the change in torque/current. This method is simple and applicable for real-time implementation and does not require any prior information on the robot dynamical model. However, any wrong choices on the threshold value might lead up to undesirable performance. Therefore, in this paper we propose an evolved solution using the classical method while operating the fuzzy logic system. The enhanced system results in an adaptive collision threshold based on the variation of the robots speed and acceleration. The system is presented in this paper along with the experimental setup and the implementation steps.

Keywords Collision detection · Human–robot interaction · Flexible robot manipulator · Fuzzy system

1 Introduction

The market need for cooperative robots is rapidly increasing due to its ability to reduce the risk of injury and allowing safe interaction with human [1, 2]. The coexistence and interaction of human in the robot workspace are not suitable in many of the existing robotics applications, particularly the robot manipulators for several reasons. One of the most significant reasons is the safety factor [3–6].

Therefore, topics such as collision avoidance and collision detection are getting attention from researchers in robotics field. The collision detection is different from collision avoidance, where collision avoidance refers to algorithms that move the robot into configurations that avoid collisions. Obstacles are known or detected before contact is made between the robot and the object (or the robot itself). Collision avoidance is a preventative technique. Collision avoidance methods generally use vision or models

of the robot and the environment to identify when the robot's proposed motion will lead to a collision [7–9], while collision detection refers specifically to methods that recognize when a collision has occurred, which is the focus of this work. However, conventional robots technologies are far from responding to this need.

In order for the robots to be capable to interact and collaborate with humans, it should consider certain requirements in the mechanical structure design, its hardware, and the control schemes [10]. These robots typically require lightweight, compliant, and flexible mechanical structures. To achieve the flexibility and compliance requirements, some researchers proposed utilizing special actuators [11], e.g., the variable impedance actuators (VIA), series-elastic actuators (SEAs), and variable-stiffness actuators (VSAs) [12–14]. However, the most commonly used device to achieve joint compliance and flexibility is the harmonic drive. The harmonic

✉ Omar Abdelaziz, o0m_ziz@yahoo.com | ¹Hefei Institute of Physical Science, University of Science and Technology of China, Hefei, China. ²Egyptian Russian University, Badr City, Cairo, Egypt. ³Hehai University, Changzhou, China. ⁴Institute of Intelligent Manufacturing Technology, Jiangsu Industrial Technology Research Institute, Nanjing 211800, China.



drive is a flexible transmission that produces stiffness behavior, as a spring, between the motor and the link. It is widely used in serial manipulators due to negligible backlash, compact design, and a high torque-to-weight ratio, in addition to its ability to achieve the desired flexibility in robot joints [15]. Moreover, for the robot to be able to collaborate with humans physically there should be a certain control strategy to detect the physical collisions and interactions [16, 17]. In this direction, there are several methods used for determining these actions. The most effective, fast, and simplest method of detecting the collision is by adding external sensors such as using force sensors, accelerometers, and tactile sensors [18–24]. Although this method can detect the collision force accurately, it can just detect the collision where the sensor is located; moreover, it is an expensive solution. In order to cope with the problems of utilizing external sensors, there are other collision/interaction detection techniques without using external sensors (sensor-less) proposed in the literature [25–30]. These methods are able to extract more information from a physical collision and efficiently estimate the actual joints torque due to collision, but it required reliable information on full dynamic model, or at least the gravity model as in [31, 32], which might not be available for many of the industrial robots. There are several efficient methods proposed for collision detection in [33] with using torque sensors, and without such as the classical approach. The sensor-less classical approach for collision detection based on recognizing abnormal variations of the joints motors current is treated as actuation faults [34]. This method does not require information about the dynamic model; it requires setting a threshold value for the change in torque/current. This method is simple to be implemented in real time. The threshold value can influence the sensitivity of the collision detection. For instance, if a low threshold is chosen, then false signals for collision could be detected due to a rapid change in the trajectory speed and/or acceleration. On the other side, collision might not be detected in case of choosing high threshold value.

In this work, we decided to improve this simple method by making the threshold adaptive to the change in the speed and acceleration using the fuzzy logic control. The fuzzy system is built based on our knowledge from observing the robot's readings during operation. The UR3 robot, which is used in this work, is one of the popular human collaborative robots in the market [35]. A review for the robot design and its forward and inverse kinematics analysis are presented in [36, 37]. The robot has a TCP/IP interface that allows access to some readings in real time such as joints actual position, actual speed, and actual current. In this work, the actual current

readings during the trajectory at the different speeds and accelerations are used.

We carried out extensive series of experiments by exciting several trajectories, with different speeds and acceleration, and recorded these data. We found from experiments that the current change rate ($\Delta i = i_t - i_{t-1}$), where t is the sample number, is usually smooth. This rate in normal operation, without collision, varies relatively to the change in velocity and acceleration, while during collision the current change rate (Δi) shows abnormal variation. Thus, a fuzzy control is designed for estimating the threshold value, as an output, using the speed and acceleration as inputs. The membership functions for the inputs and the output are based on the information acquired by experiments. By implementing the fuzzy controller, the threshold became more adaptive and it prevents false collision signals.

This paper is arranged as follows; Sect. 2 explains the methodology for collision detection. Section 3 presents the experimental setup including the implementation steps. Section 4 shows the results and provides relevant discussion. Finally, in Sect. 5 a conclusion is given.

2 Methodology

The method utilized in this work for detecting collision is based on observing the joints current change rate ($\Delta i = i_t - i_{t-1}$), where in normal operation, without collision, the absolute value ($|\Delta i|$) should not exceed the threshold ϵ_{th} . Once ($|\Delta i| > \epsilon_{th}$) for any of the joints, this means that collision had occurred, and for an elementary safety reaction the robot should stop. The excessive value of ($|\Delta i|$) above ϵ_{th} in case of collision is due to the torques/moments produced at the joints from the collision force. An illustrated example is shown in Fig. 1. In Fig. 1a, the force is applied on link 2 and it is

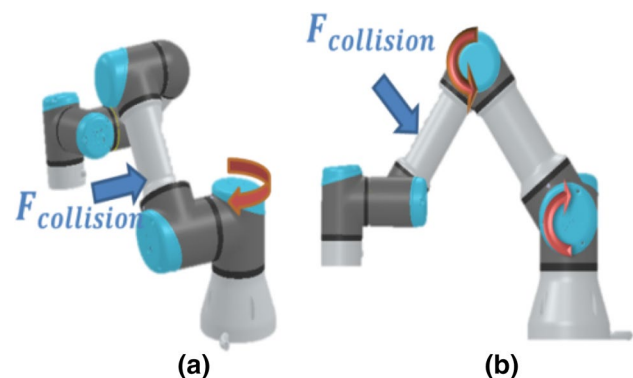


Fig. 1 Collision force produces moments at the joints in two different examples (a) and (b)

perpendicular to the axes of rotation of joint1, hence the highest moment/torque is produced in joint 1. In Fig. 1b, the force is applied on link 3 and it is perpendicular to the axes of rotation of joints 2 and 3, hence the moments/torques produced in these joints are higher than those of the other joints. We used this information for detecting collision. Experiments were carried out by running a trajectory two times: first without applying any external collision force and the second with applying external collision force. The sequence of photographs in Figs. 2 and 3 demonstrates the experiment, where Fig. 2

is taken while the trajectory is running without collision force, and Fig. 3 is taken under collision force.

Figure 4a shows the joints positions in time during the trajectory. In Fig. 4b–d, the joints readings of the current (i) and the current change ($|\Delta i|$) during the trajectory are shown, where the plot in red is the reading of the data during applying external collision forces, and the plot in blue is the reading of the data without applying any external forces; in this figure, the first three joints readings are presented. It is clearly obvious from the figure



Fig. 2 Experiment demonstration without collision force



Fig. 3 Experiment demonstration under collision force

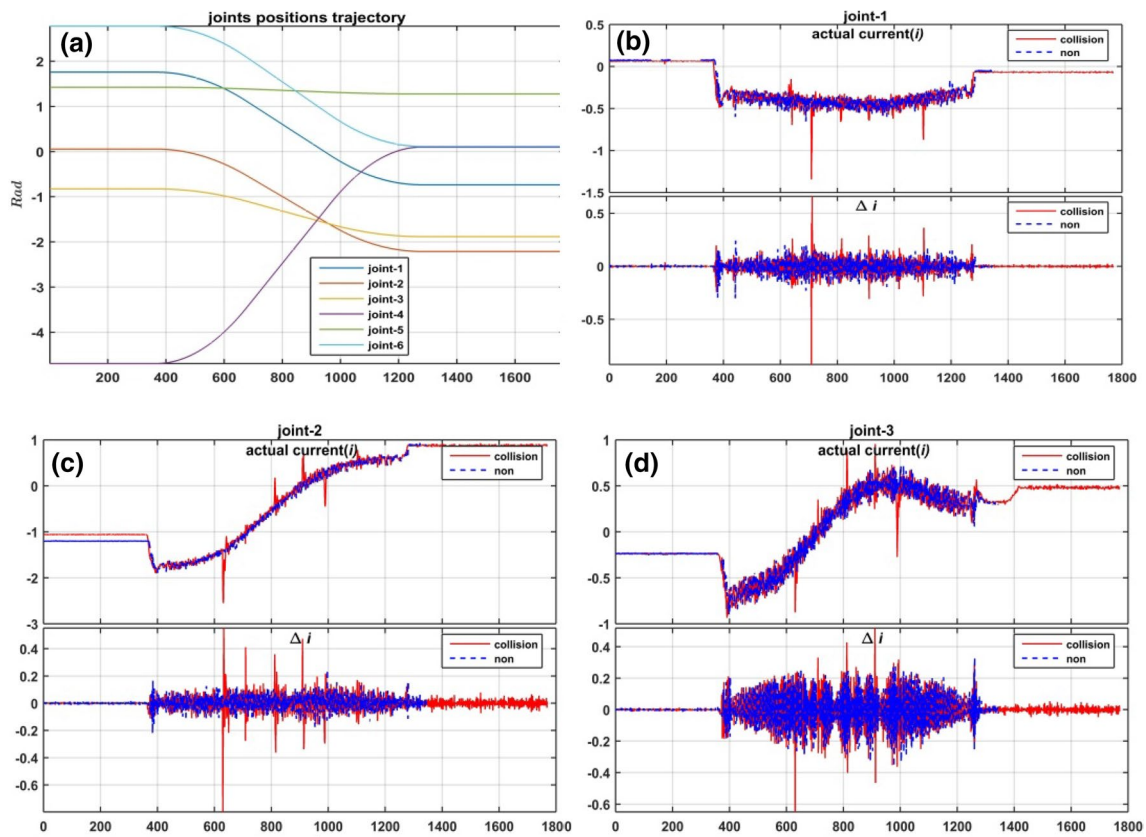


Fig. 4 **a** Joints positions for the excited trajectory over time; **b–d** the joints current readings (i) and the current change (Δi) during the trajectory; first with applying collision “in red” and second without collision “in blue”

that collision forces induce excessive currents in joints and appear as spikes.

So the method mainly depends on setting threshold value ϵ_{th} that could detect the excessive current change ($|\Delta i|$) such that it could detect false signals in case of choosing low ϵ_{thr} or it might not be able to detect the collision if high value is chosen. For these reasons, we carried out extensive series of experiments by performing several trajectories with different speeds and acceleration to get an approximate estimation of (Δi). This information is used to build the fuzzy controller, such that by giving the value of the speed and acceleration we get the proper ϵ_{th} . The proposed collision detection algorithm is shown in Fig. 5.

2.1 Fuzzy logic control

Fuzzy control provides a formal methodology for representing, manipulating, and implementing a human’s heuristic knowledge about how to control a system [38]. It came out for reasons to cope with the disadvantages of conventional control, whereas most conventional techniques require either an analytical model or an experimental model. Fuzzy logic control is particularly

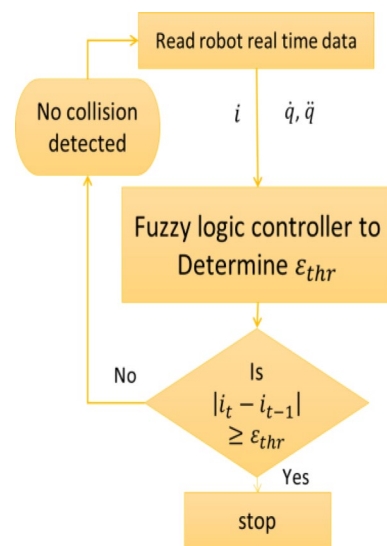


Fig. 5 Collision detection algorithm

suitable for complex and ill-defined processes. The fuzzy controller has four main components: (1) The “rule-base” holds the knowledge, in the form of a set of rules, of

how to control the system. (2) The inference mechanism evaluates which control rules are relevant at the current time and then decides what the input to the plant should be. (3) The fuzzification interface simply modifies the inputs so that they can be interpreted and compared to the rules in the rule-base. And (4) the defuzzification interface converts the conclusions reached by the inference mechanism into the inputs to the plant. The fuzzy controller components are shown in Fig. 6, where the joint speed (\dot{q}) and acceleration (\ddot{q}) represent the inputs, and the threshold ϵ_{th} represents the output. The next section shows the implementation steps.

3 Experimental setup

The first step is performing several trajectories with different speeds (\dot{q}) and acceleration (\ddot{q}), and recording these data in real time. From these data, we determined the maximum reading of (Δi) for every different pair of speed and acceleration. The maximum value of (Δi) is set to be the reference value for the threshold ϵ_{th} . In fact the relation between (Δi) to the change in \dot{q} , \ddot{q} is very noisy, and we cannot build ϵ_{th} estimation on a filtered signal. So we took the highest value of (Δi) within a range for several trajectories. For instance, we defined a membership function (MF) for the speed calling it "low," and similarly for the acceleration. The low MF of the speed takes absolute values from 0 to 0.15 rad/s as 100% low, and the acceleration takes values from 0 to 0.2 rad/s² as 100% low. Within the range of low speed and low acceleration, we get the highest value for Δi in all the experiments and set it as the reference for collision threshold. Based on this idea, we designed the fuzzy controller for each joint. In this paper, the steps for joint1 only are shown, as it can be implemented in the same way for the rest joints.

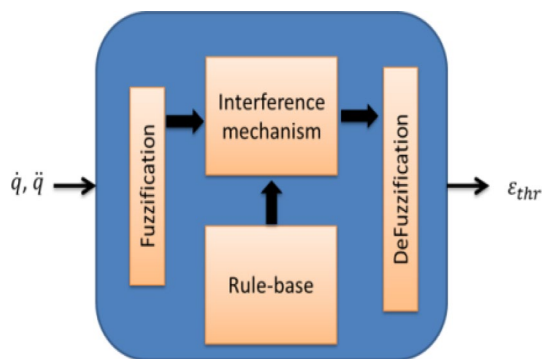


Fig. 6 Fuzzy control architecture

3.1 Implementation steps

The implementation steps are described as follows;

- (a) *Examine all the recorded trajectories data* Determine the relation between speed, acceleration, and (Δi). Then the highest value for (Δi) among the different classes of \dot{q} and \ddot{q} is identified. This step is shown in Fig. 7, where the red circles show the maximum value within the specified ranges of the speed and acceleration.
- (b) *Classify the speed and acceleration as follows* The speed has 4 classes {low, medium, high, very high}, and the acceleration has 5 classes {low, medium, high, very high, extremely high}, the ranges for each class are shown in
- (c) Table 1. The classification was decided based on our experience, such that the ranges considered high, low, or medium are determined according to our application requirements.
- (d) *Fuzzyfication* Is the process that converts or transforms the measured inputs called crisp values, into the fuzzy linguistic values used by the fuzzy interference mechanism. The inputs MF functions are shown in Fig. 8; Fig. 8a shows the speed and (b) shows the acceleration.

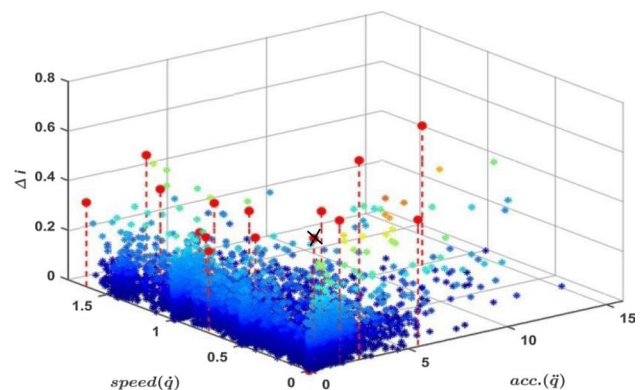


Fig. 7 Relation between speed ($|\dot{q}|$), acceleration ($|\ddot{q}|$), and (Δi)

Table 1 Speed and acceleration classifications

Speed classification	Range (rad/s)	Acceleration classification	Range (rad/s ²)
Low	0–0.25	Low	0–0.4
Medium	0.25–0.75	Medium	0.4–0.1
High	0.75–1.5	High	1–2
v-high	1.5–5	v-high	2–5
		X-high	5–15

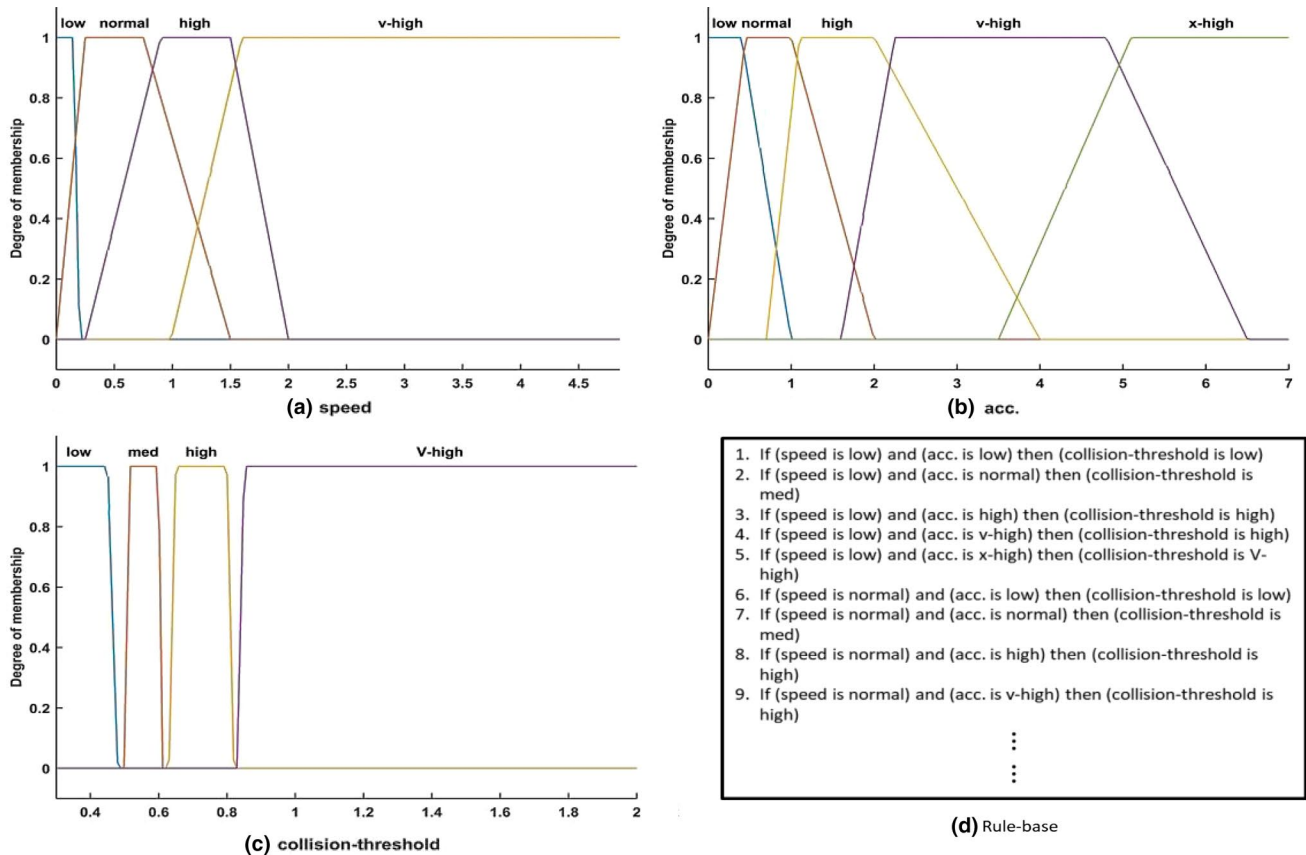


Fig. 8 The fuzzy system; **a** and **b** are the input membership functions for speed and acceleration, **c** is the output membership function collision threshold, and **d** rule-base

- (e) *Define the output membership function* Designed based on the maximum (Δi), where it is set to be bit higher than the maximum (Δi) for the different speeds and accelerations ranges. The output MF for the collision threshold ϵ_{th} is shown in Fig. 8c.
- (f) *Set the Rule-base* A collection of the expert control rules (knowledge) needed to achieve the control goal, as shown in Fig. 8d
- (g) *Defuzzification* Is the process that converts the result of fuzzy inference mechanism into the required crisp value. The correlation-minimum inference method proposed by Mamdani [39] is applied for fuzzy inference mechanism, and the center of area method [40] is used to estimate the value of ϵ_{th} . The surface view for the whole fuzzy system is shown in Fig. 9.

The next section provides the results and some discussion related to the system implementation.

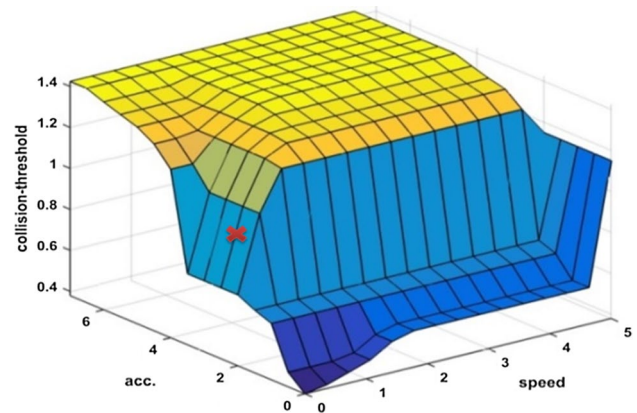


Fig. 9 Surface view for the collision threshold estimation fuzzy controller

4 Results and discussion

The relation in Fig. 7, for more than 50,000 observations, shows that the values (Δi) are not smooth and contain noise, and we could not easily get a linear relation for

the change in \dot{q} and \ddot{q} . And the maximum values in red are more probably noisy readings. Due to this, we could not build the estimation of collision threshold on the majority of (Δi) readings, which are shown as a bulk in the blue color. Instead we took the maximum noisy data. Therefore, an extensive large number of experiments observations is required, as to get the maximum possible value for (Δi) within a specified range of \dot{q} and \ddot{q} , so that in normal operation under any noisy circumstances, without collision, the value (Δi) should not exceed the collision threshold ε_{th} . And this will ensure that if the value of (Δi) exceeds the ε_{th} , it means that collision had occurred.

The surface view of the system in Fig. 9 shows the resulting threshold relative to the different ranges of \dot{q} and \ddot{q} , which ensures the value of ε_{th} adaptive to the change in \dot{q} and \ddot{q} ; moreover, it shows that the value of ε_{th} at collision will never be below the value of (Δi) within the same range of \dot{q} and \ddot{q} in normal operation. For instance, if we examine the observation with black cross mark shown in Fig. 7 at acceleration about 2.8642 (rad/s²) and the speed about 0.8166 (rad/s²), we can find that (Δi) is about 0.3185. And by using the same values of \dot{q} and \ddot{q} to check the fuzzy system output, from the surface view in Fig. 9 as a red cross mark, it can be found that the collision threshold ε_{th} is about 0.7–0.8, which is obviously higher than ($\Delta i = 0.3185$) the value at normal operation. This means that the fuzzy system adjusted the threshold ε_{th} to a value higher than the measured Δi at the same speed and acceleration.

The output value from the fuzzy system ε_{th} represents the sensitivity of the robot in detecting collision, and we are satisfied with this sensitivity level. However, we can change the collision sensitivity by doing some manual adjustments to the input and output membership functions values of the fuzzy system to get different collision thresholds ε_{th} .

Finally, this system is implemented in real time by developing a python program on a PC connected to the robot through the TCP/IP interface. The python language has several useful modules that can be used in robotics applications, such as the SciKit-Fuzzy toolbox for SciPy module [41], which can be used for implementing the fuzzy system.

5 Conclusion

In this paper, we showed a modified way for detecting collision within the robot manipulator by developing a fuzzy system. The proposed method is based on detecting the variation and setting a threshold value above this value meaning that the collision had occurred. From the recorded data results of the different trajectories, we

showed the behavior of at the different ranges of \dot{q} and \ddot{q} . Then we showed the implementation steps to obtain an adaptive threshold value using the fuzzy logic system. As a result, this paper contributes in achieving the collision detection with high performance by overcoming the false signal detection which might occur due to using fixed threshold values in the classical method.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

- Haddadin S, Albu-Schäffer A, Hirzinger G (2009) Requirements for safe robots: measurements, analysis and new insights. *Int J Robot Res* 28:1507–1527. <https://doi.org/10.1177/0278364909343970>
- Bicchi A, Peshkin MA, Colgate JE (2008) Safety for physical human–robot interaction. In: Siciliano B, Khatib O (eds) *Springer handbook of robotics*. Springer, Berlin, pp 1335–1348
- Maurtua I, Ibarguren A, Kildal J et al (2017) Human-robot collaboration in industrial applications: safety, interaction and trust. *Int J Adv Robot Syst* 14:1–10. <https://doi.org/10.1177/1729881417716010>
- Lasota PA, Fong T, Shah JA (2017) A survey of methods for safe human–robot interaction. *Found Trends Robot* 5:261–349. <https://doi.org/10.1561/23000000052>
- Haddadin S, Albu-Schäffer A, De Luca A, Hirzinger G (2008) Collision detection and reaction: a contribution to safe physical human–robot interaction. In: 2008 IEEE/RSJ international conference on intelligent robots and systems, IROS. IEEE, Nice, France, pp 3356–3363
- Alhaddad AY, Cabibihan J-J, Hayek A, Bonarini A (2019) Safety experiments for small robots investigating the potential of soft materials in mitigating the harm to the head due to impacts. *SN Appl Sci* 1:476. <https://doi.org/10.1007/s42452-019-0467-7>
- Safeea M, Mendes N, Neto P (2017) Minimum distance calculation for safe human robot interaction. *Proc Manuf* 11:99–106. <https://doi.org/10.1016/J.PROMFG.2017.07.157>
- Mohammed A, Schmidt B, Wang L (2016) Active collision avoidance for human–robot collaboration driven by vision sensors. *Int J Comput Integr Manuf* 00:1–11. <https://doi.org/10.1080/0951192X.2016.1268269>
- Flacco F, Kroger T, De Luca A, Khatib O (2012) A depth space approach to human-robot collision avoidance. In: 2012 IEEE international conference on robotics and automation. IEEE, pp 338–345
- Lamy X, Colledani F, Geffard F, et al (2009) Achieving efficient and stable comanipulation through adaptation to changes in human arm impedance. In: *Proceedings—IEEE international conference on robotics and automation*. IEEE, Kobe, Japan, pp 265–271
- Braun DJ, Petit F, Huber F et al (2013) Robots driven by compliant actuators: optimal control under actuation constraints. *IEEE Trans Robot* 29:1085–1101. <https://doi.org/10.1109/TRO.2013.2271099>

12. Ham R, Sugar T, Vanderborght B et al (2009) Compliant actuator designs. *IEEE Robot Autom Mag* 16:81–94. <https://doi.org/10.1109/MRA.2009.933629>
13. Pratt GA, Williamson MM Series elastic actuators. In: *Proceedings 1995 IEEE/RSJ international conference on intelligent robots and systems. Human robot interaction and cooperative robots*. IEEE Computer Society Press, pp 399–406
14. Schiavi R, Grioli G, Sen S, Bicchi A (2008) VSA-II: a novel prototype of variable stiffness actuator for safe and performing robots interacting with humans. In: *2008 IEEE international conference on robotics and automation*. IEEE, pp 2171–2176
15. Taghirad HD, Belanger PR (1996) An experimental study on modelling and identification of harmonic drive systems. In: *Proceedings of 35th IEEE conference on decision and control*. IEEE, Kobe, Japan, pp 4725–4730
16. Heinzmann J, Zelinsky A (2003) Quantitative safety guarantees for physical human–robot interaction. *Int J Robot Res* 22:479–504. <https://doi.org/10.1177/02783649030227004>
17. De Santis A, Siciliano B, De Luca A, Bicchi A (2008) An atlas of physical human–robot interaction. *Mech Mach Theory* 43:253–270. <https://doi.org/10.1016/J.MECHMACHTHEORY.2007.03.003>
18. Lu S, Chung JH, Velinsky SA (2005) Human–robot collision detection and identification based on wrist and base force/torque sensors. In: *Proceedings—IEEE international conference on robotics and automation*. IEEE, Barcelona, Spain, pp 3796–3801
19. Lumelsky VJ, Cheung E (1993) Real-time collision avoidance in teleoperated whole-sensitive robot arm manipulators. *IEEE Trans Syst Man Cybern* 23:194–203. <https://doi.org/10.1109/21.214777>
20. Yamada Y, Shin K, Tsuchida N, Komai M (1993) A tactile sensor system for universal joint sections of manipulators. *IEEE Trans Robot Autom* 9:512–517. <https://doi.org/10.1109/70.246064>
21. Fritzsche M, Elkmann N, Schulenburg E (2011) Tactile sensing: a key technology for safe physical human robot interaction. In: *Proceedings of the 6th international conference on human–robot interaction*. ACM Press, Lausanne, Switzerland, pp 139–140
22. Bartolozzi C, Natale L, Nori F, Metta G (2016) Robots with a sense of touch. *Nat Mater* 15:921–925
23. Ulmen J, Cutkosky M (2010) A robust, low-cost and low-noise artificial skin for human-friendly robots. In: *Proceedings—IEEE international conference on robotics and automation*. IEEE, Anchorage, Alaska, USA, pp 4836–4841
24. Sim O, Oh J, Lee KK, Oh JH (2018) Collision detection and safe reaction algorithm for non-backdrivable manipulator with single force/torque sensor. *J Intell Robot Syst Theory Appl* 91:403–412
25. Lee SD, Kim MC, Song JB (2015) Sensorless collision detection for safe human–robot collaboration. In: *IEEE international conference on intelligent robots and systems*. IEEE, Hamburg, Germany, pp 2392–2397
26. De Luca A, Mattone R (2005) Sensorless robot collision detection and hybrid force/motion control. In: *Proceedings—IEEE international conference on robotics and automation*. Barcelona, Spain, pp 999–1004
27. Morinaga S, Kosuge K (2003) Collision detection system for manipulator based on adaptive impedance control law. In: *2003 IEEE international conference on robotics and automation (Cat. No.03CH37422)*. IEEE, pp 1080–1085
28. De Luca A, Albu-Schäffer A, Haddadin S, Hirzinger G (2006) Collision detection and safe reaction with the DLR-III lightweight manipulator arm. In: *IEEE international conference on intelligent robots and systems*. IEEE, Beijing, China, pp 1623–1630
29. Erden MS, Tomiyama T (2010) Human-intent detection and physically interactive control of a robot without force sensors. *IEEE Trans Robot* 26:370–382. <https://doi.org/10.1109/TRO.2010.2040202>
30. Chen S, Luo M, He F (2018) A universal algorithm for sensorless collision detection of robot actuator faults. *Adv Mech Eng* 10:168781401774071. <https://doi.org/10.1177/1687814017740710>
31. Chen S, Luo M, Jiang G, Abdelaziz O (2018) Collaborative robot zero moment control for direct teaching based on self-measured gravity and friction. *Int J Adv Robot Syst*. <https://doi.org/10.1177/1729881418808711>
32. Geravand M, Flacco F, De Luca A (2013) Human-robot physical interaction and collaboration using an industrial robot with a closed control architecture. *IEEE Int Conf Robot Autom* 2013:4000–4007. <https://doi.org/10.1109/ICRA.2013.6631141>
33. Haddadin S (2011) Towards safe robots: approaching Asimov’s 1st law. *Springer Tracts in Advanced Robotics*, pp 1–352. <https://doi.org/10.1007/978-3-642-40308-8>
34. Suita K, Yamada Y, Tsuchida N, et al (1995) A failure-to-safety “Kyozon” system with simple contact detection and stop capabilities for safe human-autonomous robot coexistence. In: *Proceedings of 1995 IEEE international conference on robotics and automation*. IEEE, pp 3089–3096
35. Universal Robots (2017) UR3 robot. <https://www.universal-robot.com/products/ur3-robot/>. Accessed 14 May 2018
36. Abdelaziz O, Luo M, Jiang G, Chen S (2019) Multiple configurations for puncturing robot positioning. *Int J Adv Robot Expert Syst*. <https://airccse.com/jares/papers/1419jares01.pdf>. Accessed 1 Feb 2020
37. Chen S, Luo M, Abdelaziz O, Jiang G (2017) A general analytical algorithm for collaborative robot (cobot) with 6 degree of freedom (DOF). In: *Proceedings of the 2017 IEEE international conference on applied system innovation: applied system innovation for modern technology, ICASI 2017*. IEEE, Sapporo, Japan, pp 698–701
38. Passino KM, Yurkovich S (1998) *Fuzzy control*. Addison-Wesley, Boston
39. Mamdani EH (1977) Application of fuzzy logic to approximate reasoning using linguistic synthesis. *IEEE Trans Comput C* 26:1182–1191. <https://doi.org/10.1109/TC.1977.1674779>
40. Lee CC (1990) Fuzzy logic in control systems: fuzzy logic controller. I. *IEEE Trans Syst Man Cybern* 20:404–418. <https://doi.org/10.1109/21.52551>
41. Scikit-Fuzzy skfuzzy 0.2 docs—skfuzzy v0.2 docs. In: *pythonhosted.org*. <https://pythonhosted.org/scikit-fuzzy/>. Accessed 25 Jul 2018

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.