Transdisciplinary Engineering Methods for Social Innovation of Industry 4.0
M. Peruzzini et al. (Eds.)

doi:10.3233/978-1-61499-898-3-301

Human-Robot Collision Avoidance for Industrial Robots: A V-REP Based Solution

Mohammad SAFEEA[1] and Pedro Neto
University of Coimbra, Portugal

Abstract. In this study we revisit the problem of on-line human-robot collision avoidance for industrial manipulators, where the robot is operating in non-static surrounding with moving obstacles/humans. The objective is to leverage the power of concurrent simulation packages for validating human-robot collision avoidance algorithm. The simulations are carried out on a robotic cell in which a human co-worker and a robot share the same working area. The robot is required to perform a task while the co-worker is moving freely in the space around the robot. Experimental tests were conducted recurring to the real-time virtual-reality simulations carried out using the Virtual Robot Experimentation Platform (V-REP).

Keywords. Collision avoidance, Potential fields, Safety, HRI, V-REP.

Introduction

In the field of Human-Robot Interaction (HRI), safety is considered as a vital requirement for having robots working together with humans [1]. This concept applies equally to social and industrial robots. Both social and industrial robots assist humans in their daily lives and workplace. Industrial collaborative robots are key in the frame work of Industry 4.0, their use is becoming more popular and programming them is becoming more easier [2]. On the factory floor, safety issues are one of the main factors which can break down the boundaries that limit the direct contact between humans and robots. Nowadays, in the industrial environment, manipulators have to be separated behind guarding fences and humans are not allowed to enter into the working area of the robot. This is due to the fact that industrial manipulators are still blind to their surroundings and pose a fundamental danger to humans. Owing to industrial requirements, industrial manipulators exercise high inertia and operate at high speeds. Under these conditions any collision between a manipulator and a human could lead to severe injuries. To achieve the long-sought goal of having robots in human centered environments, human safety shall be guaranteed, and the possibilities of collision shall be eliminated. Thus, the subject of collision avoidance is one of the essential questions that need to be addressed for assuring the safety of the co-worker when interacting with a robot. Yet, collision avoidance algorithms are hard to develop, especially without the presence of a virtual-reality that gives a visual feedback to the designing engineers. This difficulty is mainly due to two factors: (1) the complex articulation presented by the robot and (2) the geometric complexity of the obstacles/surroundings in the industrial environment. Collision avoidance algorithms imply complex computations applied on large amount of variables, as such simulation-software come in handy for developing these complex algorithms. While simulation packages are becoming useful tools in the modelling of physical systems, their use is widespread in various fields of engineering disciplines. This fact is also evident in robotics. Owing to their complex

---

[1] Corresponding Author, Mail: ms@uc.pt.

articulations, robots are hard to program without having a simulator in hand. Currently, due to the increase in computational power, it is possible to simulate complex virtual models that highly replicate real scenarios. This fact is very important in collaborative robotics, and can be applied for the problem of real-time collision avoidance. In the last few years, the Virtual Experimentation Platform (V-REP) [3] demonstrated to be a powerful simulator that is designed specifically for simulating robots and robotic cells. V-REP simulator integrates friendly virtual-reality environment, and several libraries and calculation modules. Those modules include collision detection, minimum distance calculation, Path/Motion planning and several physics engines. In addition, the simulator provides several Application Programming Interfaces (API) libraries that make it possible to control simulations from external programs. Those programs can be developed using several programming languages including: MATLAB, Octave, Python, Java, C++, Lua and Urbi. This makes the simulator accessible to wide range of programmers and engineers. Given that V-REP does not implement a real-time collision avoidance capacity, this study proposes a real-time collision avoidance implementation for robotic manipulators using V-REP.

1. State of the art

Collision avoidance is a fundamental subject in robotics that has been studied heavily in the last three decades. Starting by the potential fields method [4] in which the robot is considered to be moving in a potential field. The minimum value of the potential field is centered at the goal point that the robot has to reach, while obstacles correspond to peaks in the field. In the topology of this field the robot rolls downhill towards the equilibrium state, or the minima of the field. Consequently, the robot finds a path to the goal that avoids the obstacles. After the introduction of the potential fields, several studies and methods have been proposed, lately, the depth space approach for collision avoidance for robotic manipulators was addressed in [5]. In this study a Kinect camera is used to capture the position of obstacles and afterwards the minimum distances between the detected obstacles and predefined control points on the structure of the robot are calculated. In such a case, when the minimum distance is less than a predefined constant value, repulsion vectors act on the robot, repelling it away from the obstacle. Another variant to the potential fields is the 3 Dimensional force field [6]. Virtual ellipsoids are used as envelopes that cover the links of the robot, such that when the ellipsoid is close to obstacles, a repulsive force is generated, repelling it farther from them. In [7] the authors presented the skeleton algorithm, a method for avoiding self-collisions for multi armed robots. In this method, the links of the robot are approximated by cylinders, repulsion forces act on those cylinders when they come close to collision. In [8] the authors presented a method for collision avoidance that relies on the circular fields. In this method the robot is considered to be a charged particle moving in a magnetic field. This magnetic field is generated due to a virtual current passing in the surface of the obstacle, in such scenario the robot keeps away from colliding with the obstacle. Apart from the potential fields method and their variants, researchers applied fluid mechanics principals to achieve collision avoidance for robots. In [9] the authors used the Circular Theorem from fluid dynamics for calculating the stream lines of a potential flow which corresponds to the collision free path of the robot.

In [10] the authors described an algorithm for collision avoidance for redundant manipulators. The algorithm operates at the kinematics (joint-velocities) level. At first, the minimum distance between the robot and the obstacle is calculated, if closer than a predefined safety distance, a motion component is assigned to the part of the robot closest to the obstacle, repelling the robot away from collision. The novelty in their approach was in the way the repulsion motion was defined. In such approach, the repulsion component is projected on the direction aligned with the closest distance between the robot and the obstacle. This way of defining the repulsion motion gave their algorithm better immunity against singularities. Nonetheless, the proposed method was built around the fact that the end-effector task is the primary task, or the task with the highest-level priority, while collision avoidance was treated in the null space of the Jacobian associated with the end-effector. This will cause the manipulator to fail in avoiding collision with the obstacle in some situations. For example, when there is a contradiction between the end-effector task, assigned the highest-level priority, and the collision avoidance task, assigned a lower-level of priority. A solution for this problem, more efficient and not requiring re-planning, is proposed in [11]. In this solution a coefficient is introduced into the control equation, and by changing the value of this coefficient the priority level between tasks can be switched smoothly.

From the state of the art review, it can be noticed that the collision avoidance problem has been studied extensively. Yet, the problem of integrating the collision avoidance algorithm into an industrial robotic cell had not been given much attention, as such in this study it is proposed a method to integrate the collision avoidance algorithm into a state machine for controlling the robotic cell, results are presented in simulation using V-rep.

2. Mathematical formulation

The collision avoidance algorithm used in this study is based on the potential fields method [4]. Using this method the robot is moving in a potential field, where the attraction vectors attract the end-effector towards the target, and vectors of repulsion repel the robot away from obstacles.

In this study the attraction vector acts on the end-effector and attracts it to the goal. This vector is a function of the error:

$$e = p_e - p_{goal} \tag{1}$$

Where e is the error vector between the end-effector and the goal point, p_e is the position vector of the end-effector, and p_{goal} is the position vector of the goal. After calculating the error vector, the attraction vector v_{att} can be calculated:

$$v_{att} = -K_p e - K_i \int e dt \tag{2}$$

Where K_p is the proportional term matrix and K_i the integral term matrix. Using inverse kinematics methodologies [12], the angular velocities $\dot{q}_{att}$ of the robot joints due to the attraction vector are calculated from:

$$\dot{q}_{att} = J^T \left(JJ^T + \lambda I \right)^{-1} v_{att} \tag{3}$$

Where J stands for the Jacobean of the robot associated with the Tool Center Point (TCP), the T symbol in the superscript stands for the matrix transpose operator, λ is a damping coefficient and I is the identity matrix.

For achieving collision avoidance, a repulsion vector shall act on the point of the robot closest to the obstacle. This vector repels the robot away from obstacles. The main input for calculating the repulsion vector is the minimum distance between the obstacle and the robot $d_{\min}$, where $d_{\min}$ can be calculated roughly using closed-form formula, after approximating the robot and the obstacles by primitive shapes [13], [14]. More precisely, the minimum distance can be calculated directly from V-REP which uses the mesh-objects of the robot and the obstacles for performing the calculation. After calculating the minimum distance, and the points of the robot and the obstacle associated with this minimum distance, the repulsion vector can be specified. The direction of the repulsion vector s is calculated:

$$s = \frac{p_r - p_o}{\left| p_r - p_o \right|} \tag{4}$$

Where p_r is the position vector of the point of the robot closest to the obstacle, and p_o is the point of the obstacle closest to the robot. After calculating the direction of the repulsion vector, its magnitude is calculated:

$$f_{rep} = \begin{cases} k_{rep}\left(\dfrac{d_0}{d_{\min}} - 1 \right) & \text{if } d_{\min} < d_0 \\ 0 & \text{if } d_{\min} > d_0 \end{cases} \tag{5}$$

Where f_{rep} is the magnitude of the repulsive force, k_{rep} a repulsion constant and d_0 the distance at which the repulsion vector is activated. The repulsion vector is calculated:

$$v_{rep} = f_{rep} s \tag{6}$$

And the angular velocities due to the repulsion vector are calculated:

$$\dot{q}_{rep} = J_{cp}{}^{T}\left(J_{cp} J_{cp}{}^{T} + \lambda I \right)^{-1} v_{e.att} \tag{7}$$

Where $\dot{q}_{rep}$ are the angular velocities due to the repulsion vector and J_{cp} the Jacobean associated with the point of the robot closest to the obstacle. Then, the total angular velocities are calculated:

$$\dot{q} = \dot{q}_{att} + \dot{q}_{rep} \tag{8}$$

Subjected to the angular velocities vector, the robot moves towards the goal while avoiding collisions with the obstacles.

3. Testing and results

In the context of the presented methodology and to validate the proposed framework several real-time virtual-reality simulations were performed. The simulations were carried out using MATLAB and V-REP. Consequently, the main computations were implemented in MATLAB (including attraction/repulsion vectors, direct/inverse kinematics, state machine logic, etc.). The real-time virtual-reality simulations were carried out in V-REP, where using the simulator a 3D representation of the scene is designed (including the co-worker, his motion path, the robot, the conveyor belts, and the object subject to manipulation operation). During the simulation V-REP takes care of minimum distance calculation and keeps track of the objects in the scene with an elegant graphical feedback.

Table 1. Test parameters. X and Y, for X and Y approaching directions, R for random approaching direction.

Test	Distance d_0 (m)	Approaching direction
1	1	X
2	0.75	Y
3	1	R

The two programs MATLAB and V-REP were connected together using sockets. In the presented simulations the robot used is 6 DOF FANUC M-710i industrial manipulator. During the simulation, the robot is idle or performing a task, while an obstacle is moving in a collision course with the robot. Statistical data of velocities and positions were collected and the results were analyzed. The simulations were run on Intel Core i7 laptop, with 4 GB of RAM. Three different tests were performed, Table 1. Test 1 and Test 2 were performed to measure the performance of the collision avoidance algorithm under different configurations. In Test 3 the algorithm is implemented in a typical robotic cell. In this cell the robot is performing a pick-and-place operation while avoiding collision with a co-worker moving randomly in the shared workspace. The video accompanying the tests can be found at (https://www.youtube.com/watch?v=pUa1ba559tk).

3.1. Test 1

In this test the robot is stationary in resting position and the human co-worker is approaching the robot from the front (along X direction), Figure 1. The co-worker walks towards the end-effector, the minimum distance between them decreases, and when it reaches a predefined safety distance the robot reacts by moving-away to avoid collision with the co-worker. After performing the real-time virtual-reality simulation of the aforementioned scene, the robot moves smoothly, and avoids collision with the co-worker successfully.

To quantify the response of the robot, several parameters where collected during the simulation:

1. The position of the end-effector expressed in Cartesian coordinates of the reference frame;
2. The position of a reference point on the co-worker expressed in Cartesian coordinates of the reference frame;
3. Minimum distance between the co-worker and the robot;

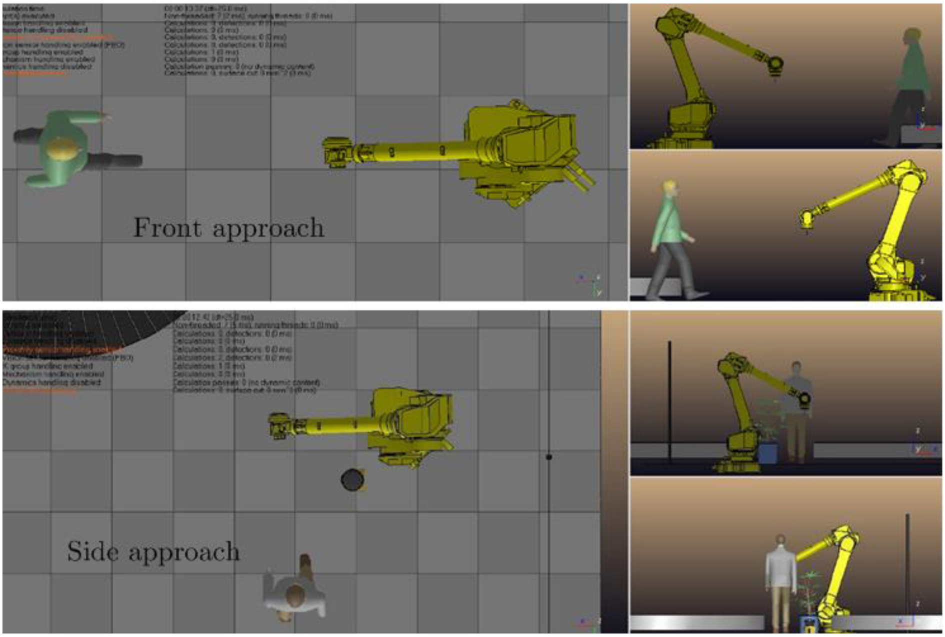

Figure 1. Layout of simulation scene for Test 1 and Test 2.

1. Velocity of the end-effector.

The response of the robot is analyzed using the plot in Figure 2. The figure shows the X coordinate of the end-effector, the X coordinate of the co-worker and the difference between them. We can notice from the plot that at the beginning of the simulation the robot is stationary in a resting position, while the co-worker is moving with a negative velocity of about 0.3 m/sec, along X direction towards the robot. When a predefined safe distance between the robot and the co-worker is reached, the robot reacts by moving away from the co-worker. When the co-worker changes its motion to the opposite direction, the robot moves back towards its resting position. The simulation cycle continue for several more periods and the same reaction was noticed in each period. Figure 3 shows the evolution of the minimum distance between the co-worker and the robot as function of time. Data pertaining to several simulation cycles were collected. The minimum distance reported is 0.75 meter. Figure 4 shows the X component of end-effector's velocity.

3.2. Test 2

In this test the co-worker approaches the robot from the side along Y direction, Figure 1. At the beginning the co-worker is 2.5 meters away from the end-effector and then he/she starts walking toward the robot, with an average velocity of 0.2 m/sec along the Y direction. Figure 5 shows that at the beginning the end-effector is stationary while the co-worker is walking towards the robot. By the time the safety distance between the robot and the co-worker, 0.7 meter, is reached the robot starts to react to avoid the collision with the co-worker. From the curve in Figure 5 we notice that the minimum Y distance reached between the end-effector and the co-worker is 0.52 meters. This approach by the side is different from the frontal approach in terms of robot's reaction, in which collision avoidance relies mainly on motion along axis 1 of the robot. Nevertheless, the results obtained are similar, the robot moves smoothly, and avoids collision with the co-worker successfully.

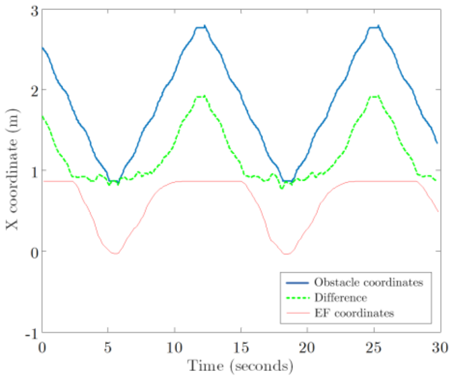

Figure 2. Test 1 results.

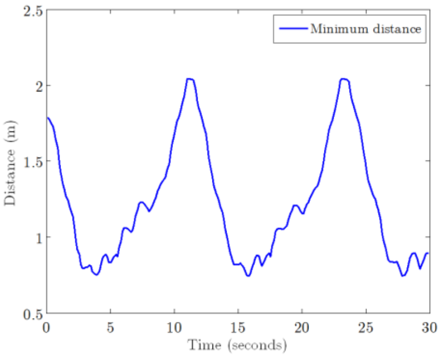

Figure 3. Minimum distance between robot and cowrker for Test 1.

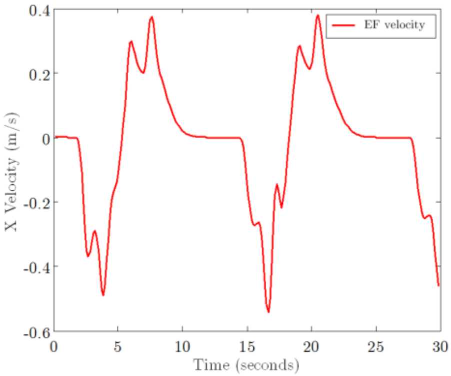

Figure 4. End-effector velocity for Test 1.

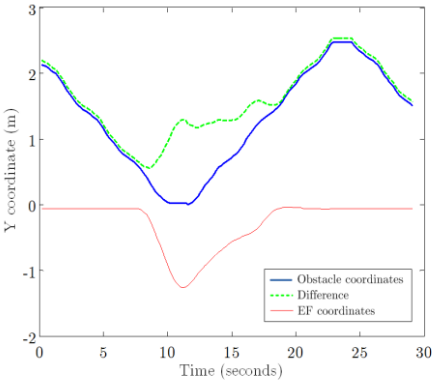

Figure 5. Test 2 results.

3.3. Test 3

For performing complex tasks while maintaining a collision avoidance capability, the collision avoidance algorithm can be implemented in a state machine architecture, Figure 6. Test 3 is a representation of this case. In this test the robot is manipulating an object from one conveyor to another, a common application for a pick-and-place task. It is required that the robot moves as close as possible to a predetermined nominal trajectory, while avoiding collision with human co-worker that shares the same workspace. For performing this test a state-flow control logic was developed based on state machine diagram in Figure 7. At any time the simulation progresses in one of several different states, operation logic is described:

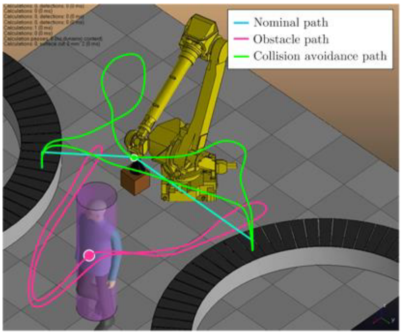

Figure 6. Test 3 motion trajectories

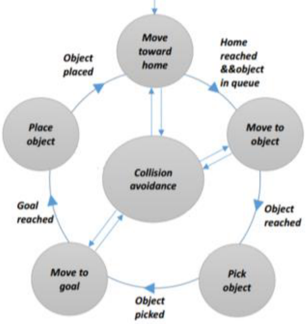

Figure 7. State machine for controlling a robotic cell with collision avoidance capability

1. The robot moves toward the home position;
2. Once the home position is reached, the robot moves along a predefined trajectory towards the object;
3. Once the object is reached the robot picks it up;
4. The robot moves along a predefined trajectory towards the goal position;
5. Once reached, the robot places the object at the goal position;
6. The robot moves back to home position;
7. At any state, whenever the co-worker is closer than some safety distance from the robot, the state will transit to collision avoidance mode, and the collision avoidance controller is activated.

Figure 6 shows motion trajectories of the robot and the co-worker. Figure 8 shows snapshots of the simulation progression. The simulation was performed successfully, the robot avoided collisions with the co-worker and the objectives of the method were satisfied.

Figure 8. Robotic cell simulation, Test 3.

4. Conclusion

In this study it is presented a V-REP simulation of a collision avoidance algorithm based on the artificial potential fields. The control process is based on two main principals: (1) generation of end-effectors motion as close as possible to a predefined trajectory, and (2) generation of collision avoidance motion from an estimated collision imminence measure. To estimate collision imminence, the minimum distance between the robot and the obstacles are calculated using V-REP. An attraction vector is calculated based on the error between the TCP point of the end-effector and the goal target. A repulsion vector is calculated based on the minimum distance between the robot and the obstacle. Simulation tests demonstrated the framework's reliability and computational efficiency. We concluded that the system is robust and able to achieve collision avoidance for collaborative robotics.

References

[1] M. S. S. Haddadin, Towards Safe Robots: Approaching Asimov's 1st Law, Springer-Verlag, Berlin Heidelberg, 2011.

[2] M. Safeea and P. Neto, KUKA Sunrise Toolbox: Interfacing Collaborative Robots with MATLAB, arXiv:1709.01438, 2017.

[3] E. Rohmer, S.P.N. Singh, and M. Freese, V-REP: A versatile and scalable robot simulation framework, in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.

[4] O. Khatib, Real-time Obstacle Avoidance for Manipulators and Mobile Robots, *Proceedings. 1985 IEEE Int. Conf. Robot. Autom.*, Vol. 2, 1986, pp. 90–98.

[5] F. Flacco, T. Kröger, A. De Luca and O. Khatib, A depth space approach to human-robot collision avoidance, *Proc. - IEEE Int. Conf. Robot. Autom.* , 2012, pp. 338–345.

[6] P. Chotiprayanakul, D. K. Liu, D. Wang and G. Dissanayake, A 3-Dimensional Force Field Method for Robot Collision Avoidance in Complex Environments, *24th Int. Symp. Autom. Robot. Constr.*, 2007, pp. 139-145.

[7] A. De Santis, A. Albu-Schäffer, C. Ott, B. Siciliano and G. Hirzinger, The skeleton algorithm for self-collision avoidance of a humanoid manipulator, in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 2007, DOI: 10.1109/AIM.2007.4412606.

[8] S. Haddadin, R. Belder and A. Albu-Schaeffer, Reactive motion generation for robots in dynamic environments, in *Proceedings IFAC 2011 World Congress, 28. Aug. - 02. Sep. 2011, Milano, Italy*, 2011.

[9] R. Soukieh, I. Shames and B. Fidan, Obstacle avoidance of non-holonomic unicycle robots based on fluid mechanical modeling, in *2009 European Control Conference (ECC),* 2009, pp. 3269–3274.

[10] L. Žlajpah and B. Nemec, Kinematic control algorithms for on-line obstacle avoidance for redundant manipulators, in *2002. IEEE/RSJ International Conference on Intelligent Robots and Systems,* Vol. 2, 2002, pp. 1898–1903.

[11] L. Žlajpah and T. Petrič, Obstacle avoidance for redundant manipulators as control problem, in S. Küçük (ed.) *Serial and parallel robot manipulators: kinematics, dynamics, control and optimization*, InTech, Rijeka, 2012, DOI: 10.5772/32651.

[12] S.R.S. Buss, Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods, *IEEE Transactions on Robotics and Automation,* Vol. 17(1), 2004, pp. 1–19.

[13] M. Safeea, N. Mendes and P. Neto, Minimum Distance Calculation for Safe Human Robot Interaction, *Procedia Manufacturing*, Vol. 11, 2017, pp. 99–106.

[14] P. Ennen, D. Ewert, D. Schilberg and S. Jeschke, Efficient Collision Avoidance for Industrial Manipulators with Overlapping Workspaces, *Procedia CIRP*, Vol. 20, 2014, pp. 62–66.