Available online at www.sciencedirect.com

ScienceDirect

Procedia Manufacturing 11 (2017) 99 – 106

Procedia MANUFACTURING

27th International Conference on Flexible Automation and Intelligent Manufacturing, FAIM2017, 27-30 June 2017, Modena, Italy

Minimum distance calculation for safe human robot interaction

Mohammad Safeea, Nuno Mendes, Pedro Neto*

University of Coimbra, Department of Mechanical Engineering POLO II, 3030-788 Coimbra, Portugal

Abstract

The ability of efficient and fast calculation of the minimum distance between humans and robots is vitally important for realizing a safe human robot interaction (HRI), where robots and human co-workers share the same workspace. The minimum distance is the main input for most of collision avoidance methods, HRI, robot decision making, as well as robot navigation. In this study it is presented a novel methodology to analytically compute of the minimum distance between cylindrical primitives with spherical ends. Such primitives are very important since that there geometrical shape is suitable for representing the co-worker and the robots structures. The computational cost of the minimum distance between n cylinders is of order $O(n^2)$. In this study QR factorization is proposed to achieve the computational efficiency in calculating the minimum distance mutually between each pair of cylinders. Experimental tests demonstrated the effectiveness of the proposed approach.

Peer-review under responsibility of the scientific committee of the 27th International Conference on Flexible Automation and Intelligent Manufacturing

Keywords: Minimum distance; Robot; Safe Human Robot Interaction;Collision Avoidance.

1. Introduction

The presence of robots around us will become a reality in the near future. However, they need to become safe in the way they interact with us. Thus, safety is a major concern in collaborative robotics, since robots and humans will coexist and share the same workspace. A collaborative robot shall be capable in real-time of acquiring sensor data

* Corresponding author. Tel.: +351 239 790 700; fax: +351 239 790 701.
E-mail address: pedro.neto@dem.uc.pt

Peer-review under responsibility of the scientific committee of the 27th International Conference on Flexible Automation and Intelligent Manufacturing
doi:10.1016/j.promfg.2017.07.157

related to the robot surrounding environment. These data are usually used to calculate the proximity between the robot and the obstacles/humans in the surrounding environment. This proximity is best measured by the minimum distance, which is the main input for most of the algorithms related to: collision avoidance, collaboration/interaction, robot decision making, as well as mobile-robots guidance and navigation.

Capturing accurate information from multi-layer sensor systems is still difficult to achieve. To improve accuracy existing sensing systems rely on markers. In addition, to compute the minimum distance the obstacles/humans need to be approximated by primitive geometries in a non-trivial process. There are different methods to compute the minimum distance between primitive geometries, which depend on the geometry itself.

In this study a novel methodology for computing the minimum distance between geometric primitives, as cylinders, is proposed. The cylinders are chosen because they give a good approximation for representing humans, robots and objects in general. The method is analytical and utilizes QR factorization. This is particularly important because an accurate representation of a robot environment by geometric primitives requires a relatively large number of geometric elements with continuous and real-time computing related with the minimum distance between all the static and dynamic elements in the environment. For example, for a robotic environment represented by n geometric primitives, it will requires $n(n-1)/2$ invocations of minimum distance calculation. Although the number of calculations can be reduced by ruling out by hand the unnecessary calculations by specifying the object's between which collisions cannot occur, still the number of remaining calculations is considerable, which emphasis the fact that a fast algorithm is important for efficient implementation of intelligent robotic systems.

2. State of the art

Several researchers proposed different methods for representing humans and robots geometry. Convex polyhedrons were proposed in [1] for representing two PUMA 560 manipulators. In [2] ellipses and spheres were used to represent the robot and obstacles. A computationally efficient way to represent the robot and obstacles is to use primitive shapes [3], [4]. A similar convention was proposed in [5]. In [6] a humanoid is represented by cylinders since that such representation allows for analytical and efficient calculation of the minimum distance which is used to perform self-collision avoidance. In [7] GPU was used to calculate the minimum distance between objects that are represented by meshes, such method gives precise representation of objects, nonetheless this method is very hard to implement. The skeletal algorithm proposed in [8], represents a framework for self-collision avoidance in humanoids, in this method the robot is represented by spheres and cylinders. A robot represented by twelve bounding boxes (mainly cylinders) was proposed in [4]. This representation demonstrated vital for an offline path planning algorithm proposed by the authors. An advanced collision map for performing (point to point motion) PTP with collision avoidance capability in a robotic cell with two robotic manipulators is presented in [9]. Each robot is represented by four cylinders, giving a best fit and a tight representation of the robot. From the previous studies, Figure 1, it can be concluded that the choice of the geometric primitive to represent the elements in a given environment is important for the accuracy of the representation and the computational cost required to compute the minimum distance.

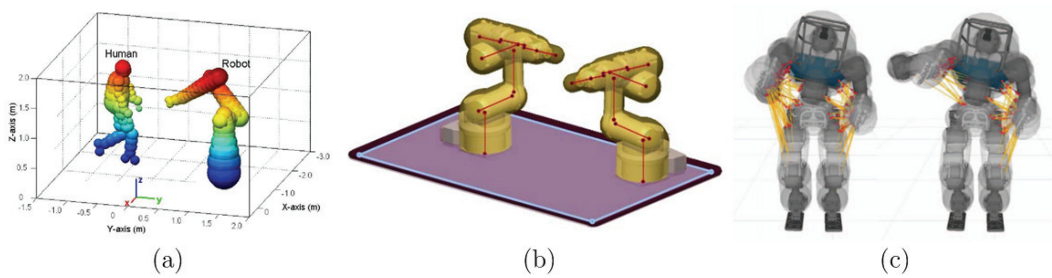

Figure 1 (a) Robot and human represented by spheres [16], (b) robots represented by segments of lines with spheres/cylinders swept onto them [3], and (c) humanoid robot represented by cylinders [6].

After the geometric representation it is required to effectively compute the minimum distance between such geometric primitives, in continuous and real-time. Several researchers have proposed solutions for this purpose. Chapter 3 of [8] describes a method for minimum distance computation between two cylinders with spherical ends. Also in [10] the authors presented an algebraic method for this matter. Another method for computing the minimum distance between cylinders with flat ends was proposed in [11]. Nevertheless, the aforementioned methods are lengthy because they consider the different configurations in which two cylinders might collide with each other. In [11] seventeen different configurations are considered and in [10] nine different configurations are considered. A method to determine the minimum distance between multiple known (geometry, position, orientation, and configuration) and multiple unknown objects within a camera image is in [12]. The distance is estimated by searching for the largest expansion radius where the projected model does not intersect the object areas classified as unknown in the camera image. A method for computing the minimum translational distance based on the Gilbert-Johnson-Keerthi algorithm between two spherically extended polytopes is introduced in [13]. A novel method to evaluate distances between dynamic obstacles using multiple depth cameras is in [14]. A depth-space oriented discretization of the Cartesian space is introduced (representing the workspace monitored by a depth camera), including occluded points.

3. Representation of objects

For realizing a safe collaborative robotic cell, with collision avoidance capability, the software shall implement an accurate and efficient way to represent the physical objects numerically, this is best achieved by using cylinders, while cylinders give a good representation of the human body, and the structure of the robot, yet due to their geometry it is efficient to calculate the minimum distance between cylinders in an analytical, accurate, way.

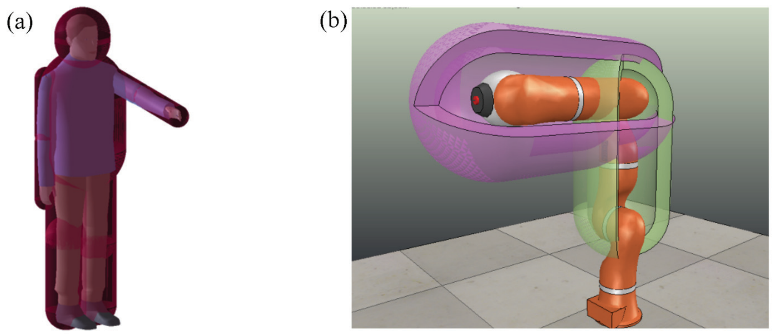

Figure 2 (a) Worker covered by cylinders (b) KUKA iiwa robotic manipulator represented by cylinders.

4. Analytical solution for minimum distance between cylinders

It is presented in this section a novel method for calculating the minimum distance between two cylinders, the proposed method gives analytical solution and is simple to implement. In Figure 3 it is shown two line segments representing the axes of two cylinders and their associated common normal. Each segment can be defined by two vectors in base frame, one at the beginning of the segment and the other at the end of that segment. Let's designate the position vectors defining the end points of the primitive segment of the first cylinder by p_1 and u_1, and the position vectors defining the end points of the primitive segment of the second cylinder by p_2 and u_2. Then, we can define two vectors $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ as:

$$\boldsymbol{s}_1 = \boldsymbol{u}_1 - \boldsymbol{p}_1 \quad (1)$$

And

$$\boldsymbol{s}_2 = \boldsymbol{u}_2 - \boldsymbol{p}_2 \quad (2)$$

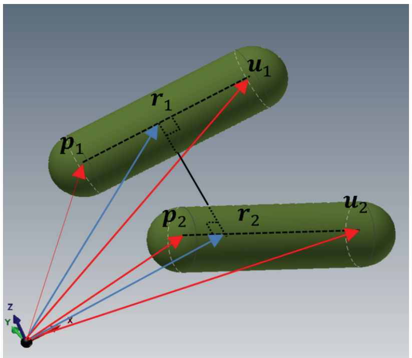

Figure 3 Minimum distance between two line segments representing two cylinders

We considered two points of interest on the two primitive segments. Those points are represented relative to base frame by two vectors, $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$, while the parameterized equation of $\boldsymbol{r}_1$ is:

$$\boldsymbol{r}_1 = \boldsymbol{p}_1 + \boldsymbol{n}_1\lambda_1 \quad (3)$$

And the parameterized equation of $\boldsymbol{r}_2$:

$$\boldsymbol{r}_2 = \boldsymbol{p}_2 + \boldsymbol{n}_2\lambda_2 \quad (4)$$

While λ_1 and λ_2 are scalar parameters, those parameters have a value in the range from zero to one, when the points they represent are confined in-between the two ends of the primitive segment.

The difference vector $\Delta\boldsymbol{r}$, between $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ is:

$$\Delta\boldsymbol{r} = \boldsymbol{r}_2 - \boldsymbol{r}_1 \quad (5)$$

When the two line segments, L_1 and L_2, are not parallel, the problem of calculating the minimum distance and their associated points renders to a minimization problem of the norm of vector $\Delta\boldsymbol{r}$:

$$\min(\Delta\boldsymbol{r}) = \min(|\boldsymbol{p}_2 + \boldsymbol{n}_2\lambda_2 - (\boldsymbol{p}_1 + \boldsymbol{n}_1\lambda_1)|) \quad (6)$$

Thus, the minimum distance calculation between two line segments reduces to the following optimization problem:

$$\min(|\Delta\boldsymbol{r}|) = \min\left(\left|[\boldsymbol{n}_2 \quad -\boldsymbol{n}_1]\begin{bmatrix}\lambda_2\\ \lambda_1\end{bmatrix} + (\boldsymbol{p}_2 - \boldsymbol{p}_1)\right|\right) \tag{7}$$

Or in a more abstract form:

$$\min(\Delta\boldsymbol{r}) = \min(\mathbf{A}\boldsymbol{x} - \boldsymbol{y}) \tag{8}$$

Where matrix $\mathbf{A}$ is the concatenation of vectors $\boldsymbol{n_2}$ and $-\boldsymbol{n}_1$.

And vector $\boldsymbol{y}$ is:

$$\boldsymbol{y} = (\boldsymbol{p}_1 - \boldsymbol{p}_2) \tag{9}$$

We can rewrite the optimization function in the following equivalent quadratic form:

$$\min(f) = \min\left((\mathbf{A}\boldsymbol{x} - \boldsymbol{y})^T(\mathbf{A}\boldsymbol{x} - \boldsymbol{y})\right) \tag{10}$$

And the problem can be viewed as minimizing the previous function, subject to the following constrains:

$$0 < x_1 < 1 \tag{11}$$

And

$$0 < x_2 < 1 \tag{12}$$

Where x_1 and x_2 are the components of the vector $\boldsymbol{x}$.

We can reformulate the function f by performing $\mathbf{QR}$ factorization on matrix $\mathbf{A}$. Then, the optimization function can be rewritten:

$$f = (\mathbf{QR}\boldsymbol{x} - \boldsymbol{y})^{\mathrm{T}}(\mathbf{QR}\boldsymbol{x} - \boldsymbol{y}) \tag{13}$$

While $\mathbf{Q}$ is 3×2 orthogonal matrix, and $\mathbf{R}$ is 2×2 upper triangular matrix. The previous function can be manipulated by taking advantage of the fact that $\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{1}_2$. Thus, after manipulation and fixing we find that minimizing (10) is equivalent to the following:

$$\min(f) = \min\left((\mathbf{R}\boldsymbol{x} - \mathbf{Q}^{\mathrm{T}}\boldsymbol{y})^T(\mathbf{R}\boldsymbol{x} - \mathbf{Q}^{\mathrm{T}}\boldsymbol{y})\right) \tag{14}$$

Or we can rewrite

$$\min(f) = \min(\boldsymbol{u}^T\boldsymbol{u}) \tag{15}$$

While $\boldsymbol{u}$ is given by

$$\boldsymbol{u} = \mathbf{R}\boldsymbol{x} - \mathbf{Q}^{\mathrm{T}}\boldsymbol{y} \tag{16}$$

Using equation (16) the square region of feasible solutions is transformed into a parallelogram, so equation (15) signify that the solution to the optimization problem is the point of the parallelogram, Figure 4, closest to the origin. According to the relative position of the parallelogram and the origin, we can distinguish two different situations. In the first the origin is inside the transformed region, so the closest point of this region to the origin is the origin itself.

The second case occurs when the origin is outside the transformed region, in this case the solution is the point of the transformed region's boundary closest to the origin. Let $\boldsymbol{u_{min}}$ be the point of the transformed region closest to the origin, then the solution of the optimization problem in (15) is

$$\boldsymbol{x_{min}} = \mathbf{R}^{-1}(\boldsymbol{u_{min}} + \mathbf{Q}^{\mathrm{T}}\boldsymbol{y}) \tag{17}$$

Then the minimum distance between the two cylinders d_{min} is calculated from

$$d_{min} = |\mathbf{A}\boldsymbol{x_{min}} - \boldsymbol{y}| - \rho_1 - \rho_2 \tag{18}$$

While ρ_1 and ρ_2 are the radius of the first and the second cylinders respectively.

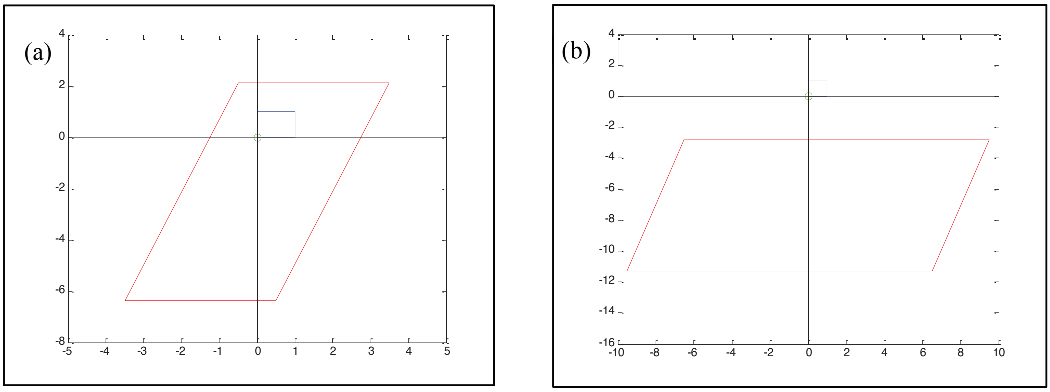

Figure 4 Region of feasible solutions for the quadratic optimization problem before applying the transformation blue, and after the transformation red, in (a) the origin is inside the transformed region, in (b) the origin is outside the transformed region.

5. Experimental results

The proposed method was tested in a simulation of a collaborative robotic cell in which the robot avoids collisions with the human co-worker, Figure 5. The simulation was implemented in the virtual reality simulator VREP, the control was implemented in MATLAB, the robot used is 6 DOF Fanuc manipulator, and is programmed to perform pick and place operation. In this system the robot and the co-worker are approximated by cylinders, using the proposed method the system finds the closest point of the robot to the co-worker and accordingly performs online trajectory readjustment of the end-effector using the potential field method [15], so that the collision avoidance between the co-worker and the robot is achieved. The distance between the human and robot is analytically computed using the proposed method. In these experiments the human was approximated by one cylinder while the robot was approximated by 2 cylinders.

Figure 5 Demonstrates a sequence of photos of a VREP simulation for safe robotic cell, minimum distance calculation was performed using the proposed method.

6. Conclusion

This study presented a new method for calculating the minimum distance between cylinders with spherical ends. The proposed method is simple to implement, and gives an analytical solution to the problem. The minimum distance calculation problem was reformulated as a bounded variable optimization problem. For solving the optimization problem we performed an affine-transformation on the region of feasible solutions. This transformation was deduced from **QR** factorization, so that the feasible solutions region becomes a parallelogram, in other words remains convex. We showed that the solution to the optimization problem corresponds to the point of the parallelogram region closest to the origin. The effectiveness of the proposed method was demonstrated successfully in simulation, the minimum distance is used as input for a potential-field based collision avoidance algorithm as shown in the Experimental results section.

Acknowledgements

This research was supported by Portugal 2020 project DM4Manufacturing POCI-01-0145-FEDER-016418 by UE/FEDER through the program COMPETE2020.

References

[1] C. Chang, M. J. Chung, and B. H. Lee, "Collision avoidance of two general robot manipulators by minimum delay time," Syst. Man Cybern. IEEE Trans., vol. 24, no. 3, pp. 517–522, 1994.

[2] S. I. Choi and B. K. Kim, "Obstacle avoidance control for redundant manipulators using collidability measure," Robotica, vol. 18, pp. 143–151, 2000.

[3] P. Bosscher and D. Hedman, "Real-time collision avoidance algorithm for robotic manipulators," 2009 IEEE Int. Conf. Technol. Pract. Robot Appl. TePRA 2009, pp. 113–122, 2009.

[4] F. Ouyang and T. Zhang, "Virtual velocity vector-based offline collision-free path planning of industrial robotic manipulator," 2015.

[5] F. Shadpey, "Force control and collision avoidance strategies for kinematically redundant manipulators," Concordia University, 1997.

[6] C. Fang, A. Rocchi, E. M. Hoffman, N. G. Tsagarakis, and D. G. Caldwell, "Efficient self-collision avoidance based on focus of interest for humanoid robots," in IEEE-RAS International Conference on Humanoid Robots, 2015, vol. 2015–Decem, pp. 1060–1066.

[7] K. B. Kaldestad, S. Haddadin, R. Belder, G. Hovland, and D. A. Anisi, "Collision avoidance with potential fields based on parallel processing of 3D-point cloud data on the GPU," in 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 3250–3257.

[8] R. V Patel and F. Shadpey, Control of Redundant Robot Manipulators: Theory and Experiments, 1st ed. Springer Publishing Company, Incorporated, 2009.

[9] A. Y. Afaghani and Y. Aiyama, "On-line collision avoidance between two robot manipulators using collision map and simple Escaping method," in Proceedings of the 2013 {IEEE/SICE} International Symposium on System Integration, {SII} 2013, Kobe, Japan, December 15-17, 2013, 2013, pp. 105–110.

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

[11] D. Biermann, R. Joliet, and T. Michelitsch, "Fast distance computation between cylinders for the design of mold temperature control systems," 2008.

[12] S. Kuhn and D. Henrich, "Fast vision-based minimum distance determination between known and unkown objects," in IEEE International Conference on Intelligent Robots and Systems, 2007, pp. 2186–2191.

[13] E. J. Bernabeu and J. Tornero, "Hough transform for distance computation and collision avoidance," IEEE Trans. Robot. Autom., vol. 18, no. 3, pp. 393–398, 2002.

[14] F. Flacco and A. De Luca, "Real-time Computation of Distance to Dynamic Obstacles with Multiple Depth Sensors," IEEE Robot. Autom. Lett., vol. 3766, no. JANUARY, pp. 1–1, 2016.

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

[16] L. Balan and G. M. Bone, "Real-time 3D Collision Avoidance Method for Safe Human and Robot Coexistence," in 2006 IEEERSJ International Conference on Intelligent Robots and Systems, 2006, pp. 276–282.