Motion Planning for Collision Avoidance via Cylindrical Models of Rigid Bodies

Pierre M. Larochelle *‡ *and John S. Ketchel †‡

* Associate Professor, pierrel@fit.edu

† Doctoral Candidate, jketchel@fit.edu

‡ Robotics and Spatial Systems Lab, Florida Institute of Technology, Melbourne, FL, USA

Abstract. This paper presents a novel methodology for motion planning to avoid collisions. Right circular cylinders are used to model the bodies of interest in the workcell. This algorithm uses line geometry and dual number algebra to exploit the geometry of right circular cylindrical objects to facilitate the detection of collisions. First, the rigid bodies are modelled with infinite length cylinders and an efficient necessary condition for collision is evaluated. If the necessary condition is not satisfied then the two bodies do not collide. If the necessary condition is satisfied then a collision between the bodies may occur and we proceed to the next stage of the algorithm. In the second stage the bodies are modelled with finite cylinders and a definitive necessary and sufficient collision detection algorithm is employed. The result is a straight-forward and efficient means of detecting collisions of cylindrically shaped bodies moving in three dimensions. This methodology is then employed in a traditional motion planning algorithm. A case study is included.

1 Introduction

Collision detection is vital for real world implementation of robotic mechanical systems. Collision detection assists in motion planning, real-time control, digital prototyping and motion simulation of these systems. Zsombor-Murray (1992) presents the visualization of the shortest distance between two lines in space. His constructive geometry and algebraic solutions to the problem motivated the work presented here. Xavier (2000) correctly states that failure to detect a collision is less acceptable than false positives, which can be further checked and that for the sake of speed exact or accurate collision detection is often sacrificed. Prior works on the collision detection problem have yielded many software packages such as: VEGAS(Johnson and Vance, 2001), V-Clip(S. Caselli and Mazzoli, 2002), RAPID(S. Caselli and Mazzoli, 2002), SOLID(S. Caselli and Mazzoli, 2002), I-Collide(J. Cohan and Ponamgi, 1995), V-Collide(T. Hudson and Manocha, 1997) and PQP(S. Caselli and Mazzoli, 2002). They vary in their modelling and in mathematical

*This material is based upon work supported by the National Science Foundation under grant No. #0422705. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

method for determining if a collision has occurred but none seek to exploit line geometry to facilitate collision detection as is done here.

In this paper we present a methodology for motion planning to avoid collisions that uses a novel algorithm for determining quantitatively if two bodies moving in three dimensional space collide. We present a two stage algorithm for determining if two bodies moving in collide. In the first stage, infinite length cylinders are used to model the objects, then line geometry is used to determine if the cylinders intersect. If these infinite cylinders do not intersect then the two bodies do not collide and no further testing is required. If the two infinite cylinders do intersect then further testing is necessary. We proceed to the second stage where cylinders of finite length are used to model the objects and they are tested to determine quantitatively if they collide.

The methodology presented here is general and can be used to detect collisions between any rigid bodies moving in three dimensions provided that the bodies are predominantly cylindrical in shape. We focus upon such bodies because i is commonly found in industrial robots and parallel kinematic machines.

2 Collision Detection

2.1 Infinite Cylinder Testing

Initially, each rigid body in the workcell is modelled by a cylinder of infinite length and finite radius. If the distance between the two cylinders is less than the sum of their two radii then the infinite cylinders have collided. Hence, if the actual finite cylindrical objects have collided it is *necessary* that the minimum distance between their associated infinite cylinders be less than the sum of their radii.

We use Plücker coordinates and dual vectors to represent the major axis of an infinite cylinder (McCarthy, 2000). Line S_1 can be defined by points $\vec{c}$ and $\vec{f}$ or by point $\vec{c}$ and direction vector $\vec{s}$ (see Fig. 1 and Eqs. 2.1 & 2.2).

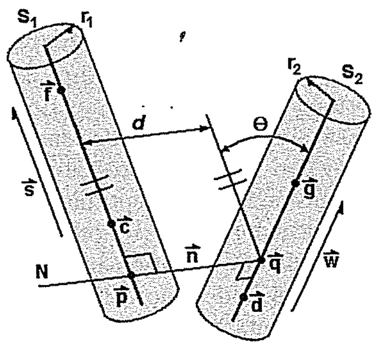

Figure 1. Infinite Cylinders

$$S_1 = \left(\frac{\vec{f}-\vec{c}}{\|\vec{f}-\vec{c}\|}, \vec{c} \times \frac{\vec{f}-\vec{c}}{\|\vec{f}-\vec{c}\|}\right) = (\vec{s}, \vec{c} \times \vec{s}) \tag{2.1}$$

$$S_2 = \Big(\frac{\vec{g}-\vec{d}}{\|\vec{g}-\vec{d}\|}, \vec{d} \times \frac{\vec{g}-\vec{d}}{\|\vec{g}-\vec{d}\|}\Big) = (\vec{w}, \vec{d} \times \vec{w}) \tag{2.2}$$

We also use the dual vector representation of the lines and dual vector algebra (see Fischer, 1995 and McCarthy, 2000) where ϵ^2=0.

$$\hat{S}_1 = (\vec{s}, \vec{c} \times \vec{s}) = a + \epsilon a^0 \text{ and } \hat{S}_2 = (\vec{w}, \vec{d} \times \vec{w}) = b + \epsilon b^0 \tag{2.3}$$

Line dot product:

$$\hat{S}_1 \cdot \hat{S}_2 = (a, a^0) \cdot (b, b^0) = \cos\theta - \epsilon d \sin\theta = \cos\hat{\theta} \tag{2.4}$$

Line cross product:

$$\hat{S}_1 \times \hat{S}_2 = (a, a^0) \times (b, b^0) = (\sin\theta + \epsilon d \cos\theta)\hat{N} = \sin\hat{\theta}\hat{N} \tag{2.5}$$

where $\hat{N}$ is the common normal line to $\hat{S}_1$ and $\hat{S}_2$. The above operations are useful for calculating the distance d and the angle θ between two lines. If $d\sin\theta \neq 0$ then the lines do not intersect ($d \neq 0$) and are not parallel ($\sin\theta \neq 0$). If $d\sin\theta = 0$ and $\cos\theta \neq 1$ then the lines intersect ($d = 0$) and are not parallel. If the $\cos\theta$ term of the dot product is equal to 1 then the lines are parallel and the resultant dual vector of the cross product will have a 0 real component. The cross product's dual component ($d\cos\theta$) will be 0 when the lines are identical. If $d\cos\theta \neq 0$ then the distance d can be calculated. Fig. 2 shows a detailed flow chart of the necessary condition for a collision. If the necessary condition is satisfied then additional testing is required.

2.2 Finite Cylinder Testing

If a possible collision has been detected by the infinite cylinder test then further testing is required to determine if an actual collision has occurred. The model is modified from cylinders of infinite length to cylinders of finite length. This changes the approach from testing lines to testing line segments. The finite cylinder testing procedure is presented in detail in Ketchel and Larochelle (2005).

3 Case Study

To demonstrate the application of the collision detection methodology presented above to robot motion planning we designed a robotic workcell (Fig. 3) for a pick and place application. This is a typical application found in the automotive industry. In this task the robot is to pick-up a dashboard, maneuver it into an auto body, place it, and then retraces its motion out of the auto body. The model of the workcell includes a robot, a tool, a workpiece (dashboard) and the drop location (automobile frame). The robot selected for the case study is the ABB6000 (Fig. 4) and it is modelled by 9 cylinders shown in shades of orange and grey of varying radii. The tool is modelled by 9 cylinders shown in shades of light grey of varying radii. The dashboard is modelled by 15 green cylinders of varying radii. The drop-off location is modelled by 25 red and blue cylinders.

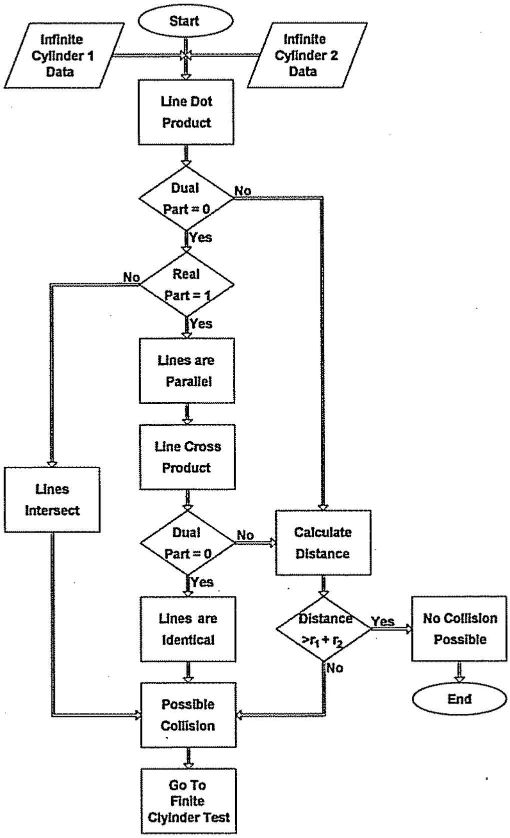

Figure 2. Infinite Cylinder Testing

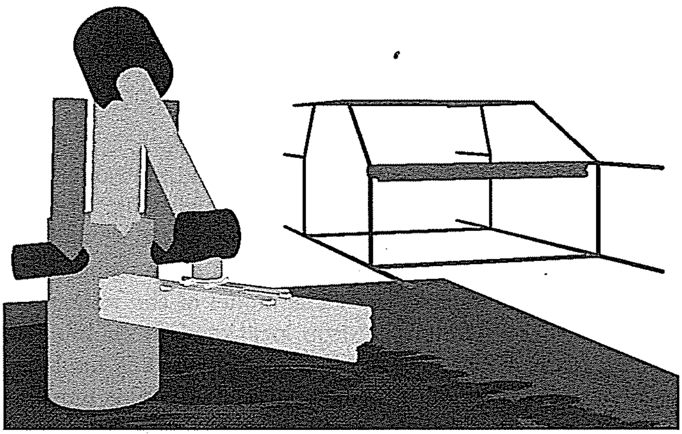

Figure 3. Workcell Model

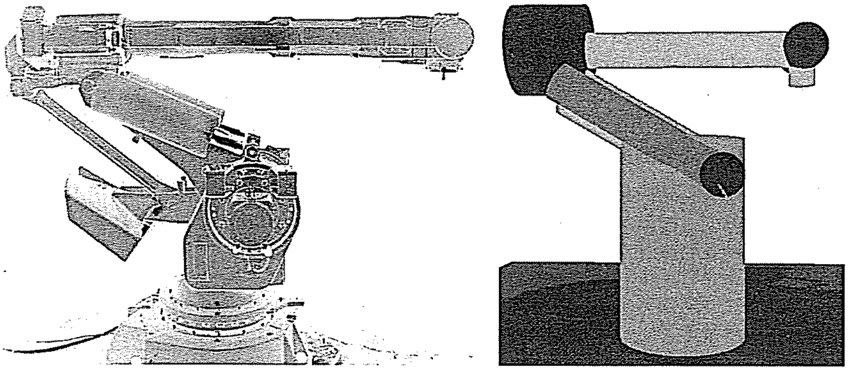

Figure 4. ABB 6000: Photo (From Directory (2005)) & Model

3.1 Robot Motion Planning

A simple motion planner was created to move the robot incrementally through the desired motion. Euler parameters for orientation (yaw, pitch and roll) and position (X, Y and Z) are used to describe the desired via points of the robot's motion. For a desired orientation there are two yaw, pitch and roll parameter sets that yield the same orientation. The motion planner enables the user to select which to use to represent the orientation. For each via point the finite difference between the current and next set of Euler parameters are found. This difference is divided by the number of steps for the desired move. The planner uses linear interpolation to step the motion incrementally from one via point to the next. Since this is not a real-time application the number of incremental steps between via points is prescribed by the user. The orientation parameters require that the shortest path is taken from one via point to the next. If the previous commanded yaw is 170 (deg) and the next commanded yaw is -170 (deg) the motion planner takes the shortest route; incrementing 20 (deg) instead of 340 (deg). The motion planner does this by checking if the difference between two sequential orientation parameters is > 180 (deg). If it isn't then the target via point is modified by adding or subtracting 360 (deg) until the difference is < 180 (deg). The position parameters are easily interpolated for smooth motion.

At each increment of motion planning every cylinder must be tested for a possible collision with every other cylinder. Since speed is important for the calculations, we look to reduce the number of tests that must be run for each incremental step. First, cylinders can not collide with themselves since they are rigid links. Second, if cylinder 5 has already been tested for a collision with cylinder 1 then testing cylinder 1 for a possible collision with cylinder 5 is redundant. The number of tests required here for each step is 1032 (in this model the tool and the dashboard can not collide). At each increment all 1032 tests are preformed using infinite cylinders to quickly check for possible collisions. Any infinite cylinders that are identified as possibly colliding are sent to finite cylinder testing to definitively determine if an actual collision has occurred.

3.2 Collision Case

A robot motion was planned for the task. A set of via points for the tool center point to follow and the number of intermediate steps were determined (Tbl. 1). At every incremental step all of the 1032 possible collisions were tested. Several collisions occurred. The first collision (shown in Tbl. 2) occurs between the 2^{nd} and 3^{rd} via points. Fig. 5 shows where the dashboard side (segment 56) collides with the car frame right windshield support (segment 16). Hence this motion is unsuitable and additional motion planning was performed.

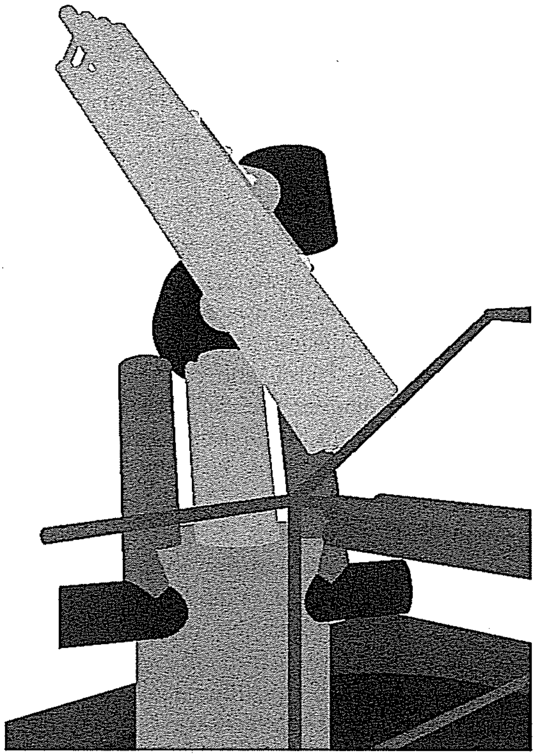

Figure 5. Collision of the Dashboard and Car Frame

3.3 Non-Collision Case

Here a collision free motion has been planned by altering the 2^{nd} and 3^{rd} via points as follows: X was decreased to 800, Z decreased to 950, and the pitch was increased to -120 to have the tool center point closer to the robot base and the dash rotated slightly more vertically. These alterations provide the dashboard clearance from the car frame. The final position of the dashboard places it within the car frame with $\frac{1}{2}$ (inch) clearance (Fig. 6).

Table 1. Collision Case Study - Via Points

X mm	Y mm	Z mm	Yaw deg	Pitch deg	Roll deg	# of Increments
500	−1400	0	0	−180	−45	10
900	−1200	1200	90	−135	0	20
900	−375	1200	90	−135	0	20
1300	−375	521.5	90	−135	0	20
2150	−375	521.5	90	−180	0	40
2150	−725	521.5	90	−180	0	40

Table 2. Collision Case Study - Output

Collision Type		Seg. No.		Seg. No.	
Two Hit		16		56	
X	Y	Z	Yaw	Pitch	Roll
900	−750	1200	90	−135	0
mm	mm	mm	deg	deg	deg
Axis 1	Axis 2	Axis 3	Axis 4	Axis 5	Axis 6
−28.08	−30.81	20.46	−84.53	62.41	−56.69
deg	deg	deg	deg	deg	deg

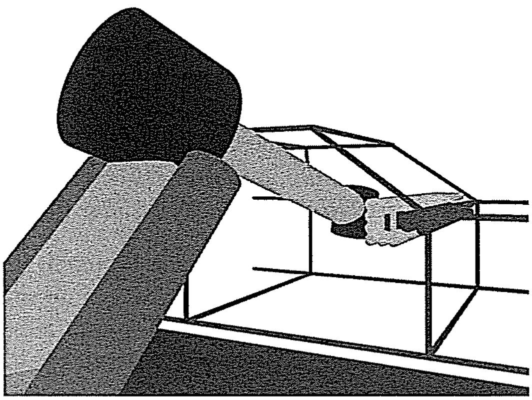

Figure 6. Successful Dashboard Placement

4 Conclusion

We have presented a novel methodology for motion planning to avoid collisions. Right circular cylinders were used to model all of the bodies of interest in the robot workcell. The proposed algorithm uses line geometry and dual number algebra to exploit the geometry of right circular cylindrical objects to facilitate the detection of collisions. The result is a straight-forward and efficient means of detecting collisions of cylindrically shaped bodies that is applicable to robot motion planning. An illustrative case study was presented.

Bibliography

Surplus Record Machinery & Equipment Directory. www.surplusrecord.com, 2005.

I. Fischer. *Dual-Number Methods in Kinematics, Statics, and Dynamics.* CRC Press LLC, 1995.

D. Manocha J. Cohan, M. Lin and M. Ponamgi. I-collide: An interactive and exact collision detection system for large-scale environments. In *Symposium on Interactive 3D Graphics*, 1995.

T. Johnson and J. Vance. The use of voxmap pointshell method of collision detection in virtual assembly methods planning. In *Proceedings of the ASME 2001 Design Engineering Technical Conferences and Computers and Information Conference*, number DET2001/DAC-21137. ACM Press, 2001.

J. Ketchel and P. Larochelle. Collision detection of cylindrical rigid bodies using line geometry. In *Proceedings of the ASME 2005 Design Engineering Technical Conferences and Computers and Information Conference*, number DETC05/MECH-84699, 2005.

J. M. McCarthy. *Geometric Design of Linkages.* Springer-Verlag, 2000.

M. Reggiani S. Caselli and M. Mazzoli. An experimental evaluation of collision detection packages for robot motion planning. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2002.

J. Cohen S. Gottschalk T. Hudson, M. Lin and D. Manocha. V-collide: Accelerated collision detection for vrml. In *Proceedings of the 2nd Symposium on VRML.* ACM Press, 1997.

P. Xavier. Implict convex-hull distance of finite-screw-swept volumes. In *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, 2000.

P.J. Zsombor-Murray. Spatial visualization and the shortest distance between two lines problem. In *Proceedings of the 5th ASEE International Conference ECGDG*, 1992.