

# Planning Movement of a Robotic Arm for Assembly of Products

Jose Ismael Ojeda Campaña

*Instituto Tecnológico de Estudios Superiores de Los Cabos, San José del Cabo 23400, México*

Received: April 02, 2015 / Accepted: April 16, 2015 / Published: April 25, 2015.

**Abstract:** This paper presents an algorithmic proposal of a movements' planner for a robotic manipulator. Its application is the planning of trajectories from wooden cuttings in a band saw. This algorithm is based on heuristics. A simulator was developed in order to allow the user to define the table's shape and figure that you want to trim. The simulator was developed in Java from Sun Microsystems, and next it will be applied in <sup>TM</sup> Cyberbotics Webots.

**Key words:** Electronic planner, robotic arm, heuristics.

## 1. Introduction

Many people have participated in the development of planners for robots manipulators. However, only a few of them have had successful applications in industry. Commonly, the main problem of motion planning or BMPP (basic motion planning problem) for its acronym in English, can be the following [1]: considering an initial and a desired final configuration, it has to find a path beginning with the initial configuration and ending in the final, and in the course from one configuration to another it must avoid crashing with obstacles. It is assumed that geometry and location of obstacles are completely known, that is, it is assumed a structured environment. Conventional methods of motion planning mainly deal with the basic motion problems. New approaches in motion planning are gathered in the C-space (configuration space), which it usually includes these two procedures: (a) first, it is defined the configuration space of objects and the free space, and then (b) it is found the collision-free path in the configuration space. Due to the high computational cost, the use of planners applying this

concept of C-space is limited for a few degrees of freedom, for example, from 2 to 3 degrees only [2, 3] This is one of the reasons of the limited use of movement planners in manipulators for applications in industry.

In the 1980s, artificial intelligence models tried to answer this problem. In general, these models used local information to develop a collision-free path. One of the most significant advantages of this new approach was that it allowed having reactive responses to the different events occurring in its environment. One of the issues unsettled was to avoid being trapped in local minima [4].

In the 1990s, several planning techniques for the C-space were exploited [5-8], the common feature of these planners is the discrete representation of the C-space, for example, the C-space is decomposed into grids or cells. At the same time, the connectivity of the C-space and implementation in algorithms of a collision detection/distance mechanism are discussed. Compared with previous methods of C-space, the computational time is highly reduced due to the discretization of the C-space. However, the computational time is still too high to be appreciated in real-time applications. For example, in the case of 4

---

**Corresponding author:** Jose Ismael Ojeda Campaña, M. Sc., research fields: artificial intelligence, fuzzy logic control, automatic control, and embedded system. E-mail: ismaelojeda@hotmail.com.

degrees of freedom manipulator, if each degree of freedom is quantified in 90 discrete levels, then the C-space of the robotic arm will be  $904 = 6.5 \times 10^7$  configurations and the manipulator of six unions will need more than 500 billions of configurations [9, 10], it indicates that for a six degrees of freedom manipulator, the C-space bitmap with a resolution of  $9^\circ$  and  $360^\circ$  in each union, would need about 512 MB of memory.

BMPP literature can be summarized as follows: for manipulators with four or fewer degrees of freedom, it is feasible to build and search the discretization of the C-space which is between several seconds and several minutes. Manipulators, with more than 4 degrees of freedom, take a long and exhaustive search for a solution by C-space. The problem gets worse by increasing the number of degrees of freedom. Simplifying properly the representation of the C-space or combining the method with a hierarchical search, with this might be feasible for some special applications [11].

This paper describes a motion planner for a 6-joint robotic arm, in which the application plans cut paths, based on how an expert solve that problem, that is, it develops the heuristics and all prior knowledge that the planner needs to achieve the task.

The first problem which arises is how the user delivers the information to the planner, for example: the geometric shape of the table and figure to be cut, the restrictions of the robotic arm and the environment where the task is developed.

The geometric form of the table is given in a list of vertices  $(x, y)$  which does not necessarily show an order. The algorithm of convex hull is applied to this list of vertices to obtain a convex figure for working. A convex figure is one in which all the given points are located either at the border or inside the polygon.

## 2. Heuristics

The heuristics which are developed are those used by the simulator. The heuristics were developed with the help of algorithms of computational geometry, trigonometry, and as an expert solves the problem.

### 2.1 Convex Hull

The convex hull of a set of points  $Q$  is the smallest polygon convex  $P$ , in which each point  $Q$  is in the shape of the polygon  $P$  or inside of it. The convex hull of  $Q$  is denoted by  $CH(Q)$  [12]. There are various convex hull algorithms. This paper is based on the algorithm known as “Jarvis March” or “Gift Wrapping” given that it has higher speed by its resolution:  $O(nh)$  times; where:  $n = \#$  total vertices and  $h =$  number of vertices of the convex hull figure [13].

Through the algorithm of convex hull, it is obtained the convex figure of the list of vertices provided by the user, a characteristic of convex figures is that drawing a straight line from the contour of the figure to the other end of the contour, this line will not leave the figure.

### 2.2 Sequence of Cutting.

The planner gets the cutting sequence based on leaving the largest possible table area, by doing that, it is obtained a greater surface in which the robotic arm can hold the piece. This heuristic is implemented by calculating the remnants of the table to make a specific cutting.

Given that both, the figure that will be cut and the table figure are irregular polygons, calculation of the area was implemented through the calculation of triangles between vertices, as described below [14]:

Choose arbitrarily one of the table vertices, and that is called vertex 1. Then, following in opposite clock direction, number the other vertices 2, 3, 4, ...,  $n$  of the table.

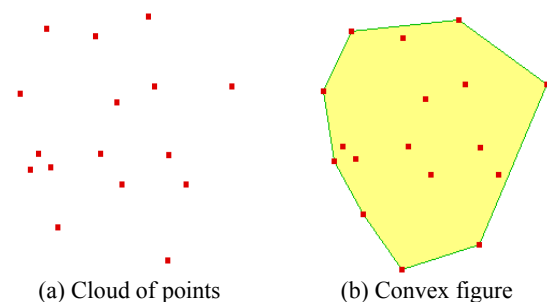


Fig. 1 Convex figure.

Then, all possible triangles that have in common the vertex 1 are formed. In other words, the triangle formed by 1, 2, 3 then 1, 3, 4, then 1, 5, 6 and so to the last vertex.

From each one, the area of the triangles formed by their determinants is calculated; the sum of all outcomes will be the area of the table.

To the full amount of determinants, there are terms that can be removed and an equation is obtained as a result to calculate the area of an irregular polygon; as shown in the following formula:

$$A(P) = \frac{1}{2} \cdot \left| \left( \sum_{i=1}^{l=n-1} x_i \cdot y_{i+1} - y_i \cdot x_{i+1} \right) + (x_n \cdot y_1 - y_n \cdot x_1) \right| \quad (1)$$

Fig. 2 shows an example about how different areas that are obtained from cutting are calculated. Each of these areas is calculated using the algorithm described above, and it is sorted in descending order. The cutting sequence in this particular example, it would be [A1, A2, A3].

Once the cutting sequence has been resolved, it is proceed to choose the point in which the table will be attached to. The selection criterion is getting the point where we have the better balance as possible. This point is obtained calculating the polygon centroid or centre of gravity (table).

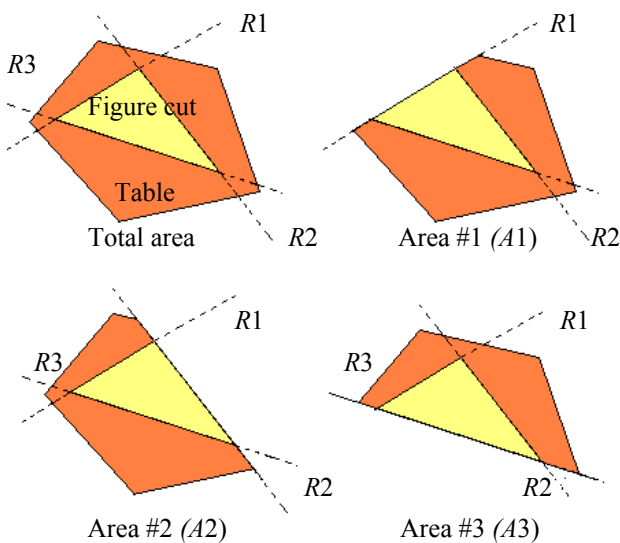


Fig. 2 Cutting sequence.

### 2.3 Centroid

The centroid is equivalent to the “centre of gravity” or “centre of mass” in a homogeneous element of specific constant weight. It is the imaginary place where it can be considered that all its weight is concentrated. The term centroid is applied to geometric figures, which do not have weight. The position of the centroid, assuming that the polygon is made of a material of uniform density is given below [15]. The area of the polygon is first calculated assuming XN is X0, in other words, that is a closed polygon. The following is the equation used to calculate the coordinates of the centre of gravity (Cx, Cy).

$$Cx = \frac{1}{6A} \sum_{i=0}^{N-1} (x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (2)$$

$$Cy = \frac{1}{6A} \sum_{i=0}^{N-1} (y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i) \quad (3)$$

where, (Cx, Cy) is the coordinate in which the polygon centre of gravity is located. For a convex figure, the centroid is always within the figure (Fig. 3).

Once the centre of gravity of the table is known, the point at which will be fastened the table to perform the alignment of the vertices of the figure to be cut on the saw band is selected. For this operation, we use heuristics of alignment.

### 2.4 Alignment

Alignment is performed by a rotation considering as origin a vertex of the figure to be cut or considering a point out of it. In the case of rotation per vertex it is calculated the angle that forms the straight of cutting to the axis of abscissas. The slope of the line through the

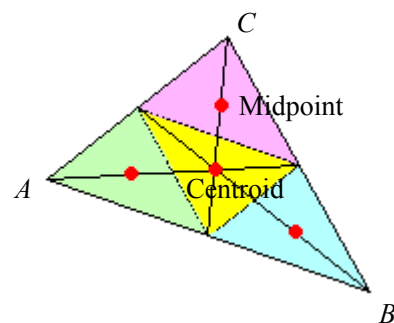


Fig. 3 Centre of gravity of a triangle.

points that form them is obtained, as we know, the slope is the tangent of the angle (Fig. 4).

According to mathematical theorems which state: “the sum of the internal angles of a triangle equals  $180^\circ$ ”, and “alternate internal angles of a line that intersect two parallel straight are equal”, the following equation is obtained:

$$\theta = 180^\circ - \beta - \alpha \tag{4}$$

where,  $\theta$  = angle to rotate the table;  $A =$  it is a right angle  $90^\circ = \pi/2$ ; and  $B = \text{arc tg}$  (slope of the line).

For rotation in which the origin is outside the figure, the distance between the point and the table has to be stated. For establishing the rotation, the angle is calculated through a right triangle.

We need to find the equation of the straight cutting to form the triangle. Once this equation is stated, we need to find the equation from line perpendicular to the straight cutting that passes through the point to attach. So, three points of the triangle are obtained as follows (Fig. 5):

- P1: Point in which the piece will be attached;
- P2: Point in which straight cutting and its perpendicular cross;
- P3: Intersection point between the straight cutting with one of the straights of the table.

Once the vertices of the triangle rectangle are defined, the angle  $\theta$  is calculated using the Pythagorean Theorem. After the piece is rotated, heuristic of optimal side is used.

### 2.5 Optimal Side

This heuristic determines the side and the point which will be attached the table through the generation of a triangle rectangle between the intersections of the cutting line and the table, centre of gravity, and the intersection between the cutting line and the perpendicular line that passes through the centre of gravity of the table (Fig. 6). To do this, the following algorithm is used:

- (1) Get the equation of the straight cutting;

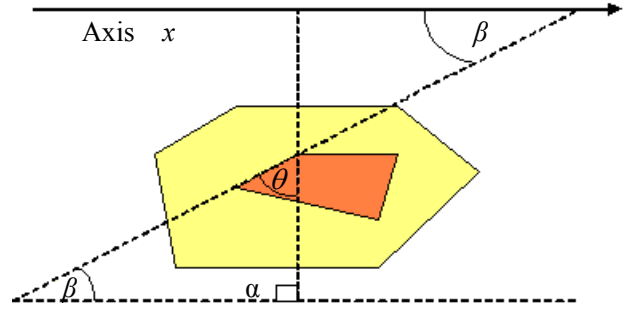


Fig. 4 Calculation of the rotation angle per vertex.

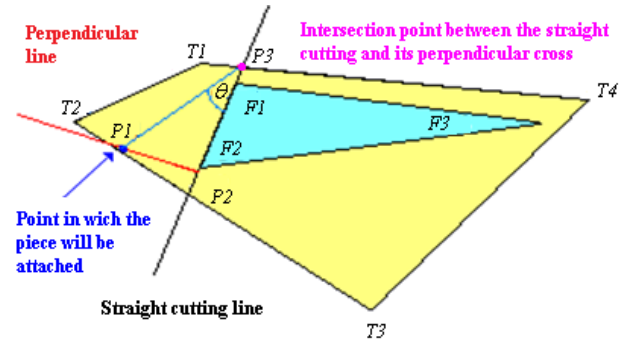


Fig. 5 Rotation on a point outside the cutting figure.

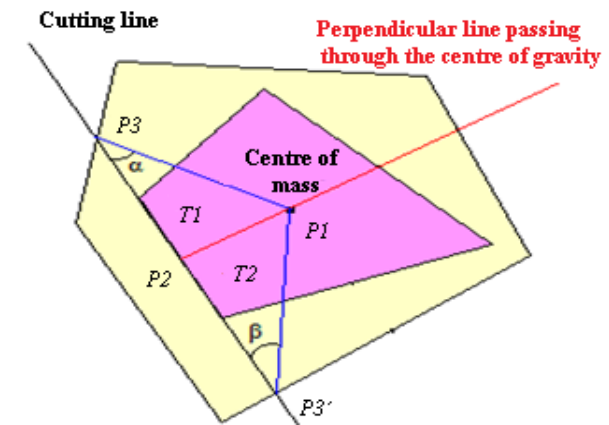


Fig. 6 Heuristic of optimal side.

- (2) Get the centre of gravity;
- (3) Get the equation of the line perpendicular to the cutting line that passes through the centre of gravity of the table;
- (4) Get the equation of the straights of the table which crosses with the cutting line. These must be at least two and at most four equations;
- (5) Once the above points, three points of the triangle are gotten:
  - P1. Centre of gravity of the table;
  - P2. Intersection of the cutting line and the

perpendicular line passing through the Centre of gravity;

- P3. Intersection of the cutting line and a line of the outline of the table.

The most parallel straight to the perpendicular line presents the smallest angle.

When the geometric problem of cutting paths has been resolved, the basic actions of the robotic arm on the situational calculus (Golog) are implemented (Fig. 7):

Prolog requires these basic actions must be expressed in the form of logical equations, therefore, the preconditions of actions were developed:

$$\begin{aligned}
 \text{poss}(\text{close}(\text{obj}_a), s) &\equiv \text{nextTo}((\text{obj}_b, s) \wedge (\text{obj}_a \neq \text{obj}_b)) \\
 \text{poss}(\text{grab}(\text{obj}), s) &\equiv \neg \text{grabbed}(\text{obj}) \\
 \text{poss}(\text{open}, s) &\equiv (\neg \text{opened}) \vee (\text{closed}) \\
 \text{poss}(\text{closed}, s) &\equiv (\neg \text{closed}) \vee (\text{opened}) \\
 \text{poss}(\text{up}, s) &\equiv (\neg \text{upper}) \vee (\text{down}) \\
 \text{poss}(\text{down}, s) &\equiv (\neg \text{lower}) \vee (\text{up}) \\
 \text{poss}(\text{left}, s) & \\
 \text{poss}(\text{right}, s) & \\
 \text{poss}(\text{forward}, s) & \\
 \text{poss}(\text{reverse}, s) & \\
 \text{poss}(\text{turn}, s) &\equiv \text{raised}(\text{obj})
 \end{aligned}$$

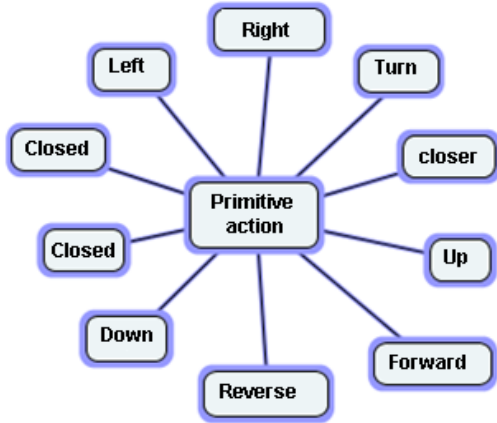


Fig. 7 Primitive actions.

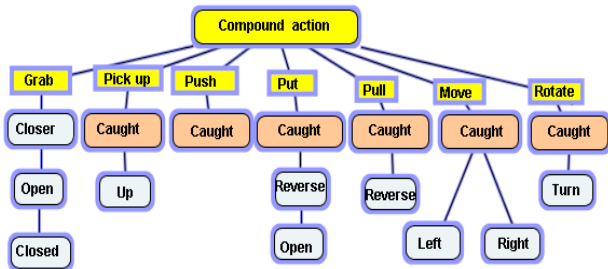


Fig. 8 Compound actions.

At the same time, compound actions were generated, which are a combination of one or more basic actions.

The successor state axioms are generated with the primitive and the compound actions:

$$\begin{aligned}
 \text{nextTo}(\text{obj}, \text{do}(a, s)) &\equiv (\neg \text{nextTo} \wedge a = \text{closer}(\text{obj})) \vee \\
 &\quad (\text{nextTo}(\text{obj}, s) \wedge a \neq \text{closer}(\text{obj})) \\
 \text{grabbed}(\text{obj}, \text{do}(a, s)) &\equiv (\neg \text{grabbed}(\text{obj}, s) \wedge a = \text{grab}(\text{obj})) \\
 \text{raised}(\text{obj}, \text{do}(a, s)) &\equiv (\text{grabbed}(\text{obj}, s) \wedge a = \text{up}(\text{obj})) \vee \\
 &\quad (\text{raised} \wedge a \neq \text{down} \wedge a \neq \text{put}) \\
 \text{opened}(\text{do}(a, s)) &\equiv (\text{closed} \wedge a = \text{open}) \vee (\text{opened} \wedge a \neq \text{close}) \\
 \text{closed}(\text{do}(a, s)) &\equiv (\text{opened} \wedge a = \text{close}) \vee (\text{closed} \wedge a \neq \text{open}) \\
 \text{upper}(\text{do}(a, s)) &\equiv (\text{lower} \wedge a = \text{up}) \vee (\text{upper} \wedge a \neq \text{down}) \\
 \text{lower}(\text{do}(a, s)) &\equiv (\text{upper} \wedge a = \text{down}) \vee (\text{lower} \wedge a \neq \text{up})
 \end{aligned}$$

With these logical equations, we can represent the knowledge of the manipulator.

### 3. Conclusions

Robotics manipulation is one of the most difficult tasks to achieve. It involves the performance of an abstraction of suitably generic and flexible actions to be used as a programming language. There is no, however, a universally accepted conceptualization that can be used as a reference. Therefore, it is necessary to fall back on design of heuristics that are appropriate to the problem of handling hand. We have developed a basic mechanism of manipulation that allows management of wooden tables for cutting that is employed in furniture manufacture. In addition, there is the basis of high-level actions that the robotic arm must use in creating cutting plans, based on the geometric shape of the figure to be cut. For this, we have studied a logical language based on situational calculus, since it has a more representational level akin to human conceptualization.

### References

- [1] Latombe, J. C. 1991. *Robot Motion Planning*. Dordrecht: Kluwer Academic.
- [2] Gouzenes, L. 1984. "Strategies for Solving Collision-Free Trajectories Problems for Mobile and Manipulator Robots." *International Journal of Robotics Research* 3 (4): 51-65.
- [3] Perez, T. L. 1987. "A Simple Motion Planning Algorithm for General Robot Manipulators." *IEEE Transactions on Robotics and Automation* 3 (3): 224-38.

- [4] Khatib, O. 1986. "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots." *International Journal of Robotics Research* 5 (6): 90-8.
- [5] Kondo, K. 1991. "Motion Planning with Six Degrees of Freedom by Multi Strategic Bi-directional Heuristic Freespace Enumeration." *IEEE Transactions on Robotics and Automation* 7 (3): 267-77.
- [6] Gupta, K. K. 1990. "Fast Collision Avoidance for Manipulator Arms: A Sequential Search Strategy." *IEEE Transactions on Robotics and Automation* 6 (6): 522-32.
- [7] Chalou, D., Gini, M., and Kumer, V. 1993. "Parallel Search Algorithms for Robot Motion Planning." Presented at the IEEE International Conference on Robotics and Automation, Atlanta, GA.
- [8] Kavraki, L. E., Svestka, P., Latombe, J. C., and Overmars, M. 1996. "Probabilistic Roadmaps for Path Planning in High Dimensional Configuration Spaces." *IEEE Transactions on Robotics and Automation* 12 (4): 566-80.
- [9] Chalou, D., Boley, D., Gini, M., Kumar, V., and Olson, C. 1998. "Parallel Search Algorithms for Robot Motion Planning." In *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, edited by Gupta, K., and Pobil, A. P. New York: John Wiley & Sons, Inc.
- [10] Chen, P. C., Hwang, Y., and Sandros, A. 1992. "Motion Planner with Performance Proportional to Task Difficulty." Presented at the IEEE International Conference on Robotics and Automation, France.
- [11] Gupta, K. 1998. "Overview and State-of-the-Art." In *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, edited by Gupta, K. and Pobil, A. P. New York: John Wiley & Sons, Inc.
- [12] Cormen, T. H., Leiserson, C. E., Rivest R. L., and Stein, C. 1998. *Introduction to Algorithms*. New York: Mc Graw Hill.
- [13] Convex Hull Algorithm. <http://www.cse.unsw.edu.au/~lambert/java/3d/giftwrap.html>.
- [14] Pastor, A., Escobar, D., Mayoral, E., and Ruiz, F. 2011. *Cultura General 2 Matematicas*. Spain: Paraninfo.
- [15] Hibbeler, R. C. 2007. *Mecánica de Materiales*. México: Pearson Educación.