View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Biblioteca Digital do IPB

Emerald Insight

Industrial Robot: An International Journal

Multiple Manipulators Path Planning using Double A*

Pedro Tavares Pedro Costa José Lima A. Paulo Moreira

Article information:

To cite this document:
Pedro Tavares Pedro Costa José Lima A. Paulo Moreira , (2016),"Multiple Manipulators Path Planning using Double A*", Industrial Robot: An International Journal, Vol. 43 Iss 6 pp. -

Permanent link to this document:
http://dx.doi.org/10.1108/IR-01-2016-0006

Downloaded on: 03 September 2016, At: 01:28 (PT)

References: this document contains references to 0 other documents.

To copy this document: permissions@emeraldinsight.com

UAEU

Access to this document was granted through an Emerald subscription provided by emerald-srm:232579 []

For Authors

If you would like to write for this, or any other Emerald publication, then please use our Emerald for Authors service information about how to choose which publication to write for and submission guidelines are available for all. Please visit www.emeraldinsight.com/authors for more information.

About Emerald www.emeraldinsight.com

Emerald is a global publisher linking research and practice to the benefit of society. The company manages a portfolio of more than 290 journals and over 2,350 books and book series volumes, as well as providing an extensive range of online products and additional customer resources and services.

Emerald is both COUNTER 4 and TRANSFER compliant. The organization is a partner of the Committee on Publication Ethics (COPE) and also works with Portico and the LOCKSS initiative for digital archive preservation.

*Related content and download information correct at time of download.

Downloaded by United Arab Emirates University At 01:28 03 September 2016 (PT)

Multiple Manipulators Path Planning using Double A*

Pedro Tavares[1,2], José Lima[2,3], Pedro Costa[1,2], and António Paulo Moreira[1,2]

FEUP - Faculty of Engineering of University of Porto, Portugal
[2] INESC TEC - INESC Technology and Science formely INESC Porto
Porto, Portugal
[3] IPB - Polytechnic Institute of Bragana, Portugal

Abstract. Purpose - Streamlining automated processes are currently undertaken by developing optimization methods and algorithms for robotic manipulators. In this paper a new approach to improve streamlining of automatic processes is presented. This new approach allows for multiple robotic manipulators commonly found in the industrial environment to handle different scenarios, thus providing a high flexibility solution to automated processes.
Design/ Methodology / Approach - The developed system is based on a spatial discretization methodology capable of describing the surrounding environment of the robot, followed by a novel path planning algorithm. Gazebo was the simulation engine chosen and the robotic manipulator used was the universal robot 5 (UR5). The proposed system was tested using the premises of two robotic challenges: EuRoC and Amazon Picking Challenge.
Findings The developed system was able to identify and describe the influence of each joint in the Cartesian space; and it was possible to control multiple robotic manipulators safely regardless of any obstacles in a given scene.
Practical implications - This new system was tested in both real and simulated environments and data collected showed that this new system performed well in real life scenarios, such as EuRoC and Amazon Picking Challenge.
Originality / Value The new proposed approach can be valuable in the robotics field with applications in various industrial scenarios, as it provides a flexible solution for multiple robotic manipulators path and motion planning.

Keywords: Path Planning, A*, ROS, Configuration Space, Kinematics, Industrial Environment

1 Introduction

Robotic manipulators have attracted substantial interest from the engineering community both from the research point of view and the need for technological development. Moreover it has attracted substantial interest from the industrial community, whose focus is on translating this technology to the final customer, enabling its use to maximize the efficiency of automated processes. The main goal of these solutions is the optimization of manufacturing processes (Chen et al. (2009)). These robot applications range from transportation of equipment, pick and place operations to dangerous and inaccessible tasks (Hvilshoj & Bogh (2011), Kim (2012)); while maintaining profitable margins in small areas and with small workloads (Bischoff et al. (2010), Mikael et al. (2008)).

The most desired characteristic of present and future robots is the ability to become fully independent. To date many approaches have been consider to address the need for intelligent task planning in a robot with the aim to mimic the human behaviour. To this regard path planning of a given task is key to all robotic approaches, as it allows for the characterization of the sequence of configurations of a robot at each moment, in order to reach the target destination. For robotic manipulator arms this sequence of configuration is related to the state of each joint. A correct path planning has several benefits such as minimizing the execution time and the effort required to the robot. Furthermore, it is also important to weight on the utility of a multi-robot solution for a given industrial environment. The use of several robots, integrated either within an environment with human operators or other robots, is likely to increase the efficiency of a process and to minimize the total operation time of a given task. In a survey on Multi-Robot coordination, Yan et al. (2013) discussed motivating factors that support the use of multiple manipulator robotic systems. In that review, some of the discussed advantages associated to multi-robot systems included:

- Better spatial distribution.
- Improved robustness of methodology resulting from data fusion and information sharing among the robots.
- Improved fault-tolerance that can benefit from information redundancy.
- Better system reliability, flexibility, scalability and versatility. Robots with diverse abilities can be combined together to deal with complex tasks.

Current robotic solutions in use in the industry lack flexibility to autonomously react to changes. Most of those solutions were implemented without using a modular approach. Consequently, a minor change in the required task could lead to a complete restructuring of the source code that controls that robot. A possible solution to overcome some of the current limitations of available robots rely on the use of the Robotic Operative System (ROS). In short, ROS is a set of software libraries and tools that can be used to build modular robot applications due to its open-source drivers and state of the art algorithms. ROS allows the development of simple algorithms that when combined assure a modular solution to any complex problem, thus adding flexibility to the overall system. In this paper, it will be presented a generic approach with enough flexibility to be potentially applicable to any object handling scenario in robotics using the ROS platform. It will also be presented a new path planning algorithm and a methodology for multi-robot applications. The main aim of the proposed approach is to be completely adaptable to any task.

2 Related Work

The ever-growing development of the robotics field is associated with the scientific and technological development of both software and hardware. Current robotic solutions using manipulators arms are focused on pick and place operations. Typically these operations are important for object handling in a given dynamic environment and need to be conducted in a safe manner. Examples of applications include the packing of medical drugs, automation systems in the food industry (Chua et al. (2003)) and the automation of production lines in the automotive industry (Jiang et al. (2010), Scholer et al. (2015)).

Since pick and place operations are one of the key applications for manipulators arms, these operations have been extensively studied. Mattone, Campagiorni & Galati (2000) and Mattone, Divona & Wolf (2000) described the two fundamental principles of such operations: (1) sensing, detecting and classifying the objects to be handled and (2) gripping the identified objects. These two principles are relevant as they provide both feedback and actuation for the system. However when developing a robotic system, another key aspect is the motion planning. In that regard, Hwang & Ahuja (1992) surveyed the work on gross-motion planning, explaining the key steps in motion planning; while Barraquand & Latombe (1991) proposed a new approach to robot path planning that consisted of building and searching a graph that connected the local minima of a potential function defined over the robot's configuration space. Later, Ralli & Hirzinger (1994) refined that algorithm increasing its efficiency and decreasing its estimated execution time. Although these studies marked the beginning of autonomous and motion controlled manipulator arms, a crucial evolution was the integration of obstacles avoidance algorithms proposed by Yao et al. (2008).

While designing a robotic system using manipulator arms, it is also necessary to consider a control tier that regulates the system movement and minimizes the errors throughout the operation. In 2002, Son suggested a learning algorithm that together with a fuzzy optimal process was able to facilitate pick and place approaches. Moreover Son also demonstrated that it was possible to avoid jamming by using measured force and moment information (Son (2002, 2011)).

Another problem with growing relevance within the research and engineering community focus on the coordination of multiple systems. Several contributions have been made to this field that provided insights on different aspects of this problem. Peng & Akella (2005) presented a solution to coordinate several robots by using kinodynamic constraints and specified paths. Subsequent to this work, Sun (2010) identified the main problem of multiple robotic manipulators systems: maintaining kinematic relationships amongst robots. Furthermore, Fei et al. (2004) proposed an approach to real-time collision-free motion planning of dual-arm reconfigurable robots. Years later, Saha & Isto (2007) developed an algorithm to control two robotic manipulators to handle deformable linear objects.

Despite the tremendous utility of automated robotic systems and the previously published work in the field, there is still room for improvement, as many robots do not have the flexibility to adjust to different tasks and/or its programming code is not implemented using building-block approaches such as ROS. DeMarco et al. (2011) and Cousins (2010) contributions prove the importance of modular implementations, in particular the ROS framework. Recently, the use of the ROS framework has also led to crucial developments in motion planning algorithms, most notably with the development of the software *MoveIt!*. This is a state-of-the-art software for mobile manipulators that covers motion planning, kinematics and navigation, among others. However, these software and the ROS framework have some limitations. Periodically, the planned trajectories determined are uneven and disconcerted. Therefore, combining the modular development of ROS with a robust motion algorithm would be beneficial.

Recently, Madsen et al. (2015) evaluated the use of autonomous manipulation technology in a real world industrial manufacturing environment. The benefits of using robotic solution in the industrial environment range from increased efficiency to minimized costs and reduced operating times. However Madsen et al. point several aspects in need of further research, before the technology can be made available to the wide industrial world. These included: robustness, safety, standardization, and robot and workstation re-configurability.

3 Robotic Manipulator System Architecture

Flexible robotic manipulator applications can be defined as a multi-tier problem. Typically we can consider three tiers: scene recognition, planning and actuation. This architecture allows to define a simple and universal method for all solutions that require robotic manipulators. Furthermore it has enough flexibility that allows to consider multi-robot systems and any given environment.

ROS as stated before allows a modular development of any system, therefore, its use allows to efficiently divide all tiers of our architecture into several simple applications. Moreover it opens a channel of communication between tiers using services, topics or actions, which allows to create a data flow, valuable to the robotic system.

Some considerations on each tier are presented below.

3.1 Tier 1

A key step to guarantee efficiency in any robotic system is the recognition of the surrounding environment. Therefore, initially, our general approach identifies and creates a 3D model of the world based on information provided by the sensors inherit to the system. This is crucial and allows for identification of all objects in a given scene, using a dictionary of objects class and a decision tree. The usage of both dictionary of objects' class and the decision tree allows for the classification of any object in the scene using key characteristics associated to them such as eccentricity, area, volumetric information and colour. Moreover to achieve a higher number of classifications and considering computational time, adding tolerance levels for those characteristics showed to be beneficial. This allows to completely identify all scenes and objects present in the scene, while providing important information to be used by the robot motion planning.

3.2 Tier 2

Tier 2 of the proposed architecture is related to path and motion planning. Once created the 3D model of the surrounding environment to the robot, it is important to discretize it accordingly to the robot's configuration space, since it is within this configuration space that we can define the robot's movement. Additionally it is also possible to define obstacles in that configuration space which will allow a best fitted path sequence from the planning algorithm.

In order to complete the goals of tier 2, there is another topic to be handled which is the robot kinematics, especially inverse kinematics. Since that modern robotic manipulators have several degrees of freedom, any required goal position has more than one solution. Thus, within our architecture, besides inverse kinematics there is also an intelligent algorithm able to determine the best robot configuration.

Having defined the desired joint values, there is the need to apply path planning algorithms. The proposed planner will be further exploited in section 4.2. Essentially this planner is able to provide a sequence of positions that the robot should assume in order to smoothly reach the goal position. In order to validate each step, direct kinematics will be used, as it will allow to check if the estimated pose match the pre-defined one.

3.3 Tier 3

The last tier of the proposed architecture is responsible for the actuation and control of the robot. Essentially, this tier relies on a control cycle. Throughout the all process there are required continuous corrections due to the errors inserted by both recognition system and the robot's actions. By using this control cycle it is possible to minimize the uncertainties associated to each operation and as a result decrease the risk of failure and increase the efficiency of the global solution. Within this tier, the robot must also cope with grasping, gripping and placement of parts, possible tool change and robot movement.

Coordinating all 3 tiers at each iteration of the control cycle proved to be beneficial since a dynamic management of the operation prevents possible errors due to a continuous state update. Although all these tiers must be considered and validated within a robotic system, the focus of this paper is tier two, related to path and motion planning regarding multi-robot systems.

4 Industrial Manipulators Path Planning

Throughout this section it will be described the main principles of our proposed methodology. This work adds onto the double A* algorithm for multiple industrial manipulators previously developed (Tavares et al. (2016)). The first step is related to the inverse kinematics, used to define the desired position in robot joints; and the second step considers the methodology used to command the robot in order to reach that set of joint values.

4.1 Configuration Space and Kinematics

Once detected the surrounding environment of the robot, it is necessary to define a configuration space for that robot. Thus, we propose a method to transform the 3D space into a discrete configuration space (Figure 1).

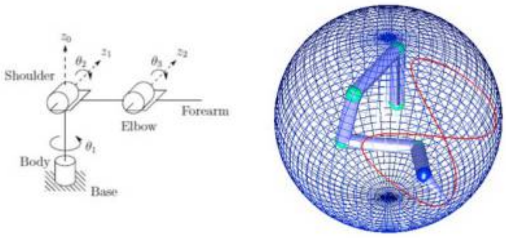
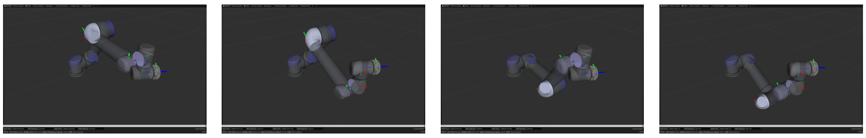


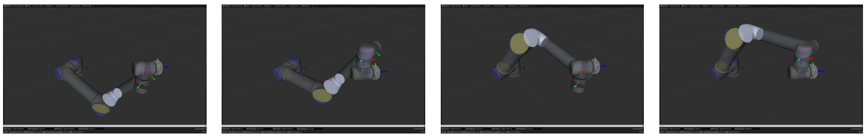
Fig. 1. Left panel: the architecture of an anthropomorphic robot; and right panel: an example of a configuration space for that manipulator Spong et al. (2006)

Path planning algorithms try to minimize the path between two points. However, to do so, it is necessary to define a configuration space of all possible outputs that the robot can produce. Since that the robot is controlled by joint position, it is beneficial to define a configuration space as a joint space. This was proved beneficial in previous works, such as, in the work of Zarubin et al. (2012). Typically, manipulator robots have six degrees of freedom. Thus, the discretization of the configuration space is frequently a problem due to the lack of memory. For example, the definition of one degree within the interval $[-\pi, \pi]$ would result in approximately $2.2 * 10^{15}$ possible configurations. Therefore it is not viable to store all those configurations and the properties associated to each one (e.g. availability, index, the joints state and others). Consequently, to overcome this memory problem, the precision of the discretization had to be reduced. Our solution to cope with this problem was to divide the path planner approach in two phases. The first one would be an approach phase which would place the robot near its desired position and then a precise phase to lead the robot to its final configuration, assuring a minimal error.

The kinematics associated to the selected robot allows one to associate the current state of the robot with a pose. For a robotic manipulator, a configuration q can be defined as a set of values that define the current position of the robot. This way, when considering manipulators with N joints (such as the Universal Robot 5 (UR5)), we can transform a three dimensional problem onto a N dimensional but well defined problem, where each configuration state matches a single point in the robot workspace. However there is still one consideration be had. Considering the most common manipulator (anthropomorphic manipulator) there are usually eight different. At each pose (position and orientation), the robotic arm can assume different configuration based on the shoulder, elbow and wrist configuration (Figure 2).



SL-EU-WD SL-EU-WU SL-ED-WD SL-ED-WD



SR-ED-WU SR-ED-WD SR-EU-WU SR-EU-WD

Fig. 2. UR5 different configurations; SL - shoulder left, SR - shoulder right, EU - elbow up, ED - elbow down, WU - wrist up, WD - wrist down

Considering the kinematics of a robot, we can use the DH method proposed by Denavit & Hartenberg (1955). By using this method we can define a transformation from the origin to the end-effector of the robot as a product of N-1 homogeneous transformations between the N joints of the manipulator (Equation 1).

$$T_N^0 = T_1^0 * T_2^1 * T_3^2 * ... * T_N^{N-1} \tag{1}$$

Each transformation can be defined as a translation and a rotation between joints. Thus, we can define a transformation matrix, based on the four DH parameters (a_i;α_i;d_i;θ_i) as presented in Equation 2.

$$\begin{bmatrix} R_y^x & P_y^x \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} cos(\theta_i) & -sin(\theta_i) * cos(\alpha_i) & sin(\theta_i) * cos(\alpha_i) & a_i * cos(\theta_i) \\ sin(\theta_i) & cos(\theta_i) * cos(\alpha_i) & -cos(\theta_i) * cos(\alpha_i) & a_i * sin(\theta_i) \\ 0 & sin(\alpha_i) & cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

As mentioned before, the robot of choice for this project was the UR5. For this robot the DH parameters can be found in Table 1 below.

Table 1. UR5 DH Parameters

Joint	a_i	α_i	d_i	θ_i
1	0	π / 2	d_1	θ_1
2	-a_2	0	0	θ_2
3	-a_3	0	0	θ_3
4	0	$\pi/2$	d_4	θ_4
5	0	-$\pi/2$	d_5	θ_5
6	0	0	d_6	θ_6

4.2 Double A* Implementation

Once discretization of all the surrounding space has been successfully achieve, a path planner algorithm can be applied. One of the key contributions of our proposed system is a new path planning algorithm based on the A* family. As mentioned before, the lack of memory denies the possibility of a high global discretization. However the A* star family requires a discretization graph to be used by the planner. As such we implemented a double planner, with two phases (1) approach phase; (2) precision phase. In the first phase, the configuration space was divided into 20 fragments, reducing the storage space requires of all configurations. The second planner adds precision to the system. This phase is based in the same fundamental idea of phase 1, however it is only applied when nearer the destination of interest. Thus, the range of possible configurations for the path planning are reduced and the division of each configuration can be more precise. This way it is assure that the target position can be rapidly and efficiently achieved.

Another key step is to define a data structure to use throughout the algorithm. As such, a unique index was created that can identify all possible configurations. We also included in that index a property that classifies each configuration availability as free space or obstacle. As such, when an object is inserted in the robot's space, it is possible to define its position as impossible to reach. There were also fields that allow to store the previous configuration while performing the algorithm, thus allowing to identify the correct sequence of configurations at the end of the algorithm. We also applied a heuristic rule, which identifies the best next probable step. This was achieved by using a cost function that considers the distance between the target position and the surrounding configurations. To save all this intermediate steps we used a heap based structure named Open List, in which every configuration in the neighbourhood to the current position was inserted, with the aim to beneficiate the path planning strategy. These configurations are ordered in the list in terms of relevance accordingly to the heuristic rule.

The process of finding neighbours is ran as a cycle where configurations near the current were identified and stored. This cycle stops when the target configuration is found. At that point, the intermediate configurations are retrieved and used to transform indexes onto robot joint states, so that the robot can be sent to reach those states and reach the final configuration required. This algorithm is key for the proposed system, as it assures the correct guidance of the robot between any given point, thus assuring a well-defined path plan. This algorithm was initially proposed in our original work (Tavares et al. (2016)).

Our proposed methodology described here allow for both development of a path planning algorithm and description of configuration space with high enough flexibility necessary for multi manipulator systems approaches (further detail in sections below).

5 Proposed Multi Manipulator System

We have tested the proposed algorithm described above in both single and multi-robot systems. We found that the configuration of space discretization was critical for multi-robot systems and, in order to control multiple robots separately, the usage of the ROS Framework proved to be beneficial. Our results showed that launching several ROS nodes that control individually each robot and report their state to the remaining nodes was necessary to adequately cope with multi-robot system. Furthermore, our proposed approach considered each robot as an obstacle for the remaining ones. To circumvent the memory usage issue, we defined the robot as a set of points. For anthropomorphic manipulators, such as the UR5, it is rather intuitive to select those points (Figure 3).

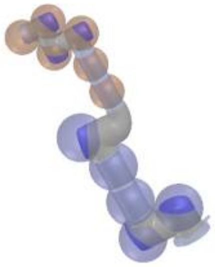


Fig. 3. UR5 Discretization

These points describe the key sections of the robot. First, this approach considers the initial and final point (base and end-effector tip). Then, it adds one point for each joint and finally the major connections (associated to the shoulder and elbow of the manipulator) are split into three positions (equally spaced throughout the joint connection).The main advantage of this approach is related to memory usage and reduced computational time. By considering the robot to be a set of spheres, as shown in Figure 3, it is possible to define a given robot as a set of eight points. Consequently, in each iteration, there are only eight events that need to be handled or updated and that represent the full extent of the neighbour robots.

In this project, we considered two UR5 spaced by 0.3 meters. Each robot considers the other as an obstacle by creating zones of occupation centred in the point discretization approach mention before. Each robot was programmed with the path planning algorithm presented in section 4.2. Another feature added to this multi-robot system was the insertion of a cost function that evaluated which of the robots was the most suitable to move at each instance. This addition is fairly intuitive as it is logical to opt for the robot that would be nearest to an object (or position) of interest. We defined the cost function (J) as a weighted sum of the distance between robot and desired position, the possibility of reaching such position and the related joints effort. Thus, the cost function is defined as:

$$J = w_1 * ||f_{pos} - i_{pos}|| + w_2 * (reaching\_Impossibility) + w_3 * joint\_effort \quad (3)$$

f_{pos} and i_{pos} are related to the desired and current position, respectively. The $reaching\_Impossibility$ is simply a boolean which states if the robot is able to reach the desired pose and the last parameters is quite obvious to be the total effort for the robot joints. Moreover each weight (w_1,w_2,w_3) has different values related to their importance. In order to choose the ideal manipulator for each task it was used a minimization criteria of the cost function. Thus, it is also fairly intuitive that the weight w_2 is defined as a large value. Then, it was preferred to give a greater importance to the distance parameter in order to minimize the operation execution time.

Finally, as proved before by Cousins (2010), the ROS framework is ideal for both single or multi manipulator systems. Therefore, a configuration file similar to the one used by the PR2 robot was created for this project.

6 Experimental Validation

The double A* Star algorithm was tested by defining multiple positions for each robot displaying similar behaviour to the one expected for a single manipulator. Examples are exhibited in Tables 2 and 3. Note that the two robot were symmetrically separate by 0.3 meters. As such, the cartesian position for the robots' bases were (x,y,z) - (0;-0.15;0) and (0;0.15;0).

Table 2. Pose for robot 1: (x,y,z,roll,pitch,yaw) - (-0.2,-0.3,0.4, 0, 0, 0) - phase 1 at the left and phase 2 at the right

Step	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	0	0	0	0	0	0
2	0	0	-20	20	20	20
...	...	...	...	...	...	...
8	0	0	-140	40	80	60

Step	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	1.1	0	-141.1	41.1	80	78.9
2	2.2	1.1	-142.2	42.2	81.1	78.9
...	...	...	...	...	...	...
10	10	3.3	-143.3	47.8	88.9	78.9

Table 3. Pose for robot 2: (x,y,z,roll,pitch,yaw) - (-0.2,0.3,0.25, π, 0, 0) - phase 1 at the left and phase 2 at the right

Step	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	-60	20	-160	20	80	140
2	-80	40	-140	0	60	160
...	...	...	...	...	...	...
7	-80	40	-140	-100	60	160

Step	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
1	-78.9	41.1	-138.9	-100	77.8	178.9
2	-77.8	42.2	-138.9	-98.9	76.7	178.9
...	...	...	...	...	...	...
9	-71.1	50	-138.9	-92.2	70	178.9

Then to validate the configuration space methodology, it was defined a target pose that requires one robot to cross the other. As such it was only provided the Cartesian position and orientation given in Euler angle. Then the implemented algorithm run a set of possible configuration until it found the best-fitted solution for the required pose. Figure 4 illustrates the movement of the robot for this example situation.

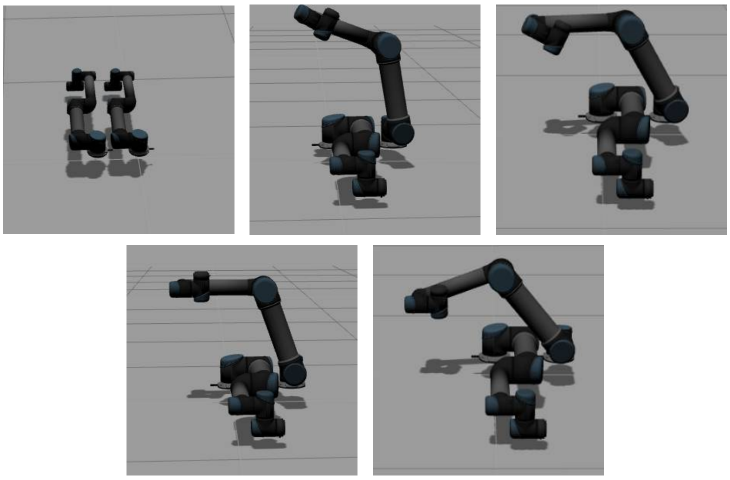


Fig. 4. Robot movement evolution considering crossing situations.

Finally, the cost function was tested, considering three hypothetical cases: both robot lying down, both robots on the same quadrant and one robot crossing another. Some random poses were processed using Matlab and the examples are displayed below in a graphical manner (Figure 5).

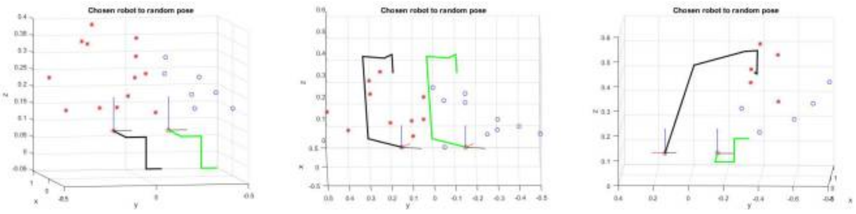


Fig. 5. Targets reachable for robot1 (blue) and robot2 (red). Robot1 is represent using green and robot2 using black.

The used weights were: w_1 - 1 ; w_2 - 100 ; w_3 - 0.75. Important to mention that both distance input and joint effort were normalize considering a maximum. For the distance input, it was the maximum length achieved by the robot (0.81

meters) and for the joint effort it was the sum of all joints interval ($6*2*\pi$). The value of w_2 is the highest as it is the parameter that states the required pose reachability. If the robot could not reach, this weight assure a high cost. Then the distance parameter associated to the weight w_1 must be higher that the joint effort parameter, since that we are more interested in minimizing distance over effort. Considering both maximums and the worst case joint space discretization of 20 degrees, w_3 has to be three quarters of w_1 which sets the threshold of 0.75 cm change in distance equivalent to 20 degrees in joint effort.

7 Discussion and Future Perspectives

Here we proposed a flexible robotic architecture that can be applied to multi-robot systems and that can be valuable in both the industrial and the research world.

This work adds onto our previously developed double A* planner. One of the key progresses made in this study is related with a discretization approach of the configuration space, which was achieved by considering a robot as a set of connecting spheres. Using this method we were able to successfully define each robot as a part of an obstacle area on the other robots. Furthermore, adding the developed double A* planner proves to be as highly efficient in multi-robot systems as it was in single robot systems. Another feature that contributed for a high efficient system was the cost function minimization strategy that allowed to choose the correct robot for each desired position.

Although this system shows interesting results, there are still some issues to be addressed. The approach phase of the developed planner still needs to be smoother. Crossing the current approach phase of the double A* planner with a mitigation algorithm that considers the Cartesian space is still a viable idea to be explored in future work. Additionally, the proposed solution was only tested considering two robots and it would be interesting to expand this system to a larger number of robots and different kinds of manipulator arms.

In conclusion, the present solution formalizes a methodology to be applied in multi-robot applications. The handling of the space discretization and the inclusion of the cost function showed to be beneficial to assure security and efficiency to the intended application range. The path planner produces as reliable results in multi-robots systems as in a single manipulator system. The present solution is a valuable tool to the engineering and robotics research community.

Acknowledgements

The research leading to these results has received funding from the project "NORTE-01-0145-FEDER-000020" which is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

Bibliography

Barraquand, J. & Latombe, J.-C. (1991), 'Robot motion planning. a distributed representation approach', *International Journal of Robotics Research* **10**(6), 628–649.

Bischoff, R., Kurth, J., Schreiber, G., Koeppe, R., Albu-Schaeffer, A., Beyer, A., Eiberger, O., Haddadin, S., Stemmer, A., Grunwald, G. & Hirzinger, G. (2010), The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing, *in* 'Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)', pp. 1–8.

Chen, H. P., Wang, J. J., Zhang, G., Fuhlbrigge, T. & Kock, S. (2009), 'High-precision assembly automation based on robot compliance', *International Journal of Advanced Manufacturing Technology* **45**(9-10), 999–1006.

Chua, P. Y., Ilschner, T. & Caldwell, D. G. (2003), 'Robotic manipulation of food products - a review', *Industrial Robot-an International Journal* **30**(4), 345–354.

Cousins, S. (2010), 'Ros on the pr2', *Ieee Robotics & Automation Magazine* **17**(3), 23–25.

DeMarco, K., West, M. E. & Collins, T. R. (2011), An implementation of ros on the yellowfin autonomous underwater vehicle (auv), *in* 'OCEANS 2011', pp. 1–7.

Denavit, J. & Hartenberg, R. S. (1955), *A kinematic notation for lower pair mechanisms based on matrices.*

Fei, Y., Fuqiang, D. & Xifang, Z. (2004), 'Collision-free motion planning of dual-arm reconfigurable robots', *Robotics and Computer-Integrated Manufacturing* **20**(4), 351 – 357.

Hvilshoj, M. & Bogh, S. (2011), '"little helper" - an autonomous industrial mobile manipulator concept', *International Journal of Advanced Robotic Systems* **8**(2), 80–90.

Hwang, Y. K. & Ahuja, N. (1992), 'Gross motion planning - a survey', *ACM Computing Surveys* **24**(3).

Jiang, X., Koo, K. M., Kikuchi, K., Konno, A., Uchiyama, M. & Ieee (2010), 'Robotized assembly of a wire harness in car production line', *Ieee/Rsj 2010 International Conference on Intelligent Robots and Systems (Iros 2010)* pp. 490–495.

Kim, J.-h. (2012), 'Automated medicine storage and medicine introduction/discharge management system'. US Patent 8,281,553.

Madsen, O., Bgh, S., Schou, C., Andersen, R. S., Damgaard, J. S., Pedersen, M. R. & Krger, V. (2015), 'Integration of mobile manipulators in an industrial production', *Industrial Robot: An International Journal* **42**(1), 11–18.

Mattone, R., Campagiorni, G. & Galati, F. (2000), 'Sorting of items on a moving conveyor belt. part i. a technique for detecting and classifying objects', *Robotics and Computer-Integrated Manufacturing* **16**(2-3), 73–80.

Mattone, R., Divona, M. & Wolf, A. (2000), 'Sorting of items on a moving conveyor belt. part 2: performance evaluation and optimization of pick-and-place operations', *Robotics and Computer-Integrated Manufacturing* **16**(2-3), 81–90.

Mikael, H., Erik, H. & Mats, J. (2008), Robotics for sms - investigating a mobile, flexible, and reconfigurable robot solution, *in* '39th International Symposium on Robotics, ISR 2008', pp. 56–61.

Peng, J. & Akella, S. (2005), 'Coordinating multiple robots with kinodynamic constraints along specified paths', *International Journal of Robotics Research* **24**(4), 295–310.

Ralli, E. & Hirzinger, G. (1994), Fast path planning for robot manipulators using numerical potential fields in the configuration space, Vol. 3, pp. 1922–1929.

Saha, M. & Isto, P. (2007), 'Manipulation planning for deformable linear objects', *IEEE Transactions on Robotics* **23**(6), 1141–1150.

Scholer, M., Vette, M. & Rainer, M. (2015), 'A lightweight robot system designed for the optimisation of an automotive end-off line process station', *Industrial Robot: An International Journal* **42**(4), 296–305.

Son, C. (2011), 'Intelligent robotic path finding methodologies with fuzzy/crisp entropies and learning', *International Journal of Robotics and Automation 2011* **26**(3), 323–336.

Son, C. M. (2002), 'Optimal control planning strategies with fuzzy entropy and sensor fusion for robotic part assembly tasks', *International Journal of Machine Tools & Manufacture* **42**(12), 1335–1344.

Spong, M. W., Hutchinson, S. & Vidyasagar, M. (2006), *Robot modeling and control*, John Wiley & Sons, Hoboken, NJ.

Sun, D. (2010), *Synchronization and Control of Multiagent Systems*, 1st edn, CRC Press, Inc., Boca Raton, FL, USA.

Tavares, P., Lima, J. & Costa, P. (2016), 'Double a* path planning for industrial manipulators', *Advances in Intelligent Systems and Computing* **418**, 119–130.

Yan, Z., Jouandeau, N. & Cherif, A. A. (2013), 'A survey and analysis of multi-robot coordination', *International Journal of Advanced Robotic Systems* **10**.

Yao, L., Ding, W., Chen, Y. & Zhao, S. (2008), 'Obstacle avoidance path planning of eggplant harvesting robot manipulator', *Nongye Jixie Xuebao/Transactions of the Chinese Society of Agricultural Machinery* **39**(11), 94–98.

Zarubin, D., Ivan, V., Toussaint, M., Komura, T. & Vijayakumar, S. (2012), Hierarchical motion planning in topological representations., *in* 'Robotics: Science and Systems'.