

# Motion Planning and Controlling Algorithm for Grasping and Manipulating Moving Objects in the Presence of Obstacles

Ali Chaabani<sup>1</sup>, Mohamed Sahbi Bellamine<sup>2</sup> and Moncef Gasmi<sup>3</sup>

<sup>1</sup>National School of Engineering of Tunis, University of Manar, TUNISIA

<sup>2,3</sup>Computer Laboratory for Industrial Systems, National Institute of Applied Sciences and Technology, University of Carthage, TUNISIA

## **ABSTRACT**

*Many of the robotic grasping researches have been focusing on stationary objects. And for dynamic moving objects, researchers have been using real time captured images to locate objects dynamically. However, this approach of controlling the grasping process is quite costly, implying a lot of resources and image processing. Therefore, it is indispensable to seek other method of simpler handling... In this paper, we are going to detail the requirements to manipulate a humanoid robot arm with 7 degree-of-freedom to grasp and handle any moving objects in the 3-D environment in presence or not of obstacles and without using the cameras. We use the OpenRAVE simulation environment, as well as, a robot arm instrumented with the Barrett hand. We also describe a randomized planning algorithm capable of planning. This algorithm is an extent of RRT-JT that combines exploration, using a Rapidly-exploring Random Tree, with exploitation, using Jacobian-based gradient descent, to instruct a 7-DoF WAM robotic arm, in order to grasp a moving target, while avoiding possible encountered obstacles. We present a simulation of a scenario that starts with tracking a moving mug then grasping it and finally placing the mug in a determined position, assuring a maximum rate of success in a reasonable time.*

## **KEYWORDS**

*Grasping, moving object, trajectory planning, robot hand, obstacles.*

## **1. INTRODUCTION**

The problem of grasping a moving object in the presence of obstacles with a robotic manipulator has been reported in different works. There have been many studies on grasping motion planning for a manipulator to avoid obstacles [1], [2], [3]. One may want to apply a method used for mobile robots, but it would cause a problem since it only focuses on grasping motion of robot hands and since the configuration space dimension is too large. Motion planning for a manipulator to avoid obstacles, however, which takes account of the interference between machine joints and obstacles, has been extensively studied in recent years and now has reached a practical level. Grasping operations in an environment with obstacles are now commonly conducted in industrial applications and by service robots.

Many robotic applications have been designed to use the concept of “Planning Using Visual Information”, i.e. control a given robot manipulator via a “servo loop” that use real world images to take decisions [4], [5], [6]. At this point, the use of predictive algorithms, as the core of the robot servo, tend to offer a better performance in the tracking and grasping process. [7]

International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), Vol.3, No. 3/4, November 2014  
developed a system to grasp moving targets using a static camera and precalibrated camera-manipulator transform. [8] proposed a control theory approach for grasping using visual information. [9] presented a system to track and grasp an electric toy train moving in an oval path using calibrated static stereo cameras.

The major challenges encountered within the visual servoing are mainly, how to reduce the robot grasp response, considering the delay introduced by images processing, and how to resolve target occlusion, when obstacles may obstruct the potential way to the object. Predictive algorithms construct one of the best approaches to escape such performance limits, and ensuring a smart tracking and grasping process. [10] use a prediction module which consists of a linear predictor with the purpose of predicting the location that a moving object will have and thus generate the control signal to move the eyes of a humanoid robot, which is capable of using behavior models similar to those of human infants to track objects. [11] present a tracking algorithm based on a linear prediction of second order solved by the Maximum Entropy Method. It attempts to predict the centroid of the moving object in the next frame, based on several past centroid measurements. [12] represent the tracked object as a constellation of spatially localized linear predictors which are trained on a single image sequence. In a learning stage, sets of pixels whose intensities allow for optimal prediction of the transformations are selected as a support for the linear predictor. [13] Implementation of tracking and capturing a moving object using a mobile robot.

The researchers who use the visual servoing system and the cameras for grasping moving object find many difficulties to record images, to treat them, because of a lot computing and image processing and also who use the predictive algorithms find a problem in the complexity of algorithms which based on many calculated and estimation [14]. In this research we want to grasp a moving object with limited motion velocity. This can be done by determining desired position for the object, the robot moves and aligns the end effector with the object and reaches towards it. This paper presents a motion planning and controlling an arm of a humanoid robot for grasping and manipulating of a moving object without cameras. We used an algorithm to control the end effector pose (position and orientation) with respect to the pose of objects which can be moved in the workspace of the robot. The proposed algorithm successfully grasped a moving object in a reasonable time.

After introducing the grasp object problem, a discussion is made to distinguish the actual solution among the others published in the literature. A description of the Rapidly-Exploring Random Trees (RRT) is detailed in section 2. Then, in section 3, we give a brief overview of the transpose of the Jacobian. The next section contains a description of the WAM<sup>TM</sup> arm. In Section 5 contains the Robot Dynamics. In Section 6, some results are given. Section 7 presents conclusions drawn from this work.

## **2. RAPIDLY-EXPLORING RANDOM TREES (RRT)**

In previous work [15],[16], approaching the motion planning problem was based on placing the end effector at pre-configured locations, computed using the inverse kinematics (IK) applied to some initial samples taken from the goal region. These locations are then set as goals for a randomized planner, such as an RRT or BiRRT [17], [18]. The solution presented by this approach remain unfinished because of the miss considered probabilistic aspect. The issue is that the planner is forced to use numbers priori chosen from the goal regions.

Another way to tackle the grasp planning, certain types of workspace goals, is to explore the configuration space of the robot with a heuristic search tree, and try to push the exploration toward one goal region [19]. Nevertheless, the goal regions and heuristics presented in [20] are

International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), Vol.3, No. 3/4, November 2014  
 highly problem specific to generalize and tricky to adjust. Drumwright and Ng-Thow-Hing [21] employ a similar strategy of extending toward a randomly-generated IK solution for a workspace point. In [22], Vande Weghe et al. present the RRT-JT algorithm, which uses a forward-searching tree to explore the C-space and a gradient-descent heuristic based on the Jacobian-transpose to bias the tree toward a work-space goal point.

[23] present two probabilistically complete planners: an extension of RRT-JT, and a new algorithm called IKBiRRT. Both algorithms function by interleaving exploration of the robot's C-space with exploitation of WGRs(Workspace Goal Regions). The extended RRT-JT (Figure 2) is designed for robots that do not have such algorithms and is able to combine the configuration space exploration of RRTs with a workspace goal bias to produce direct paths through complex environments extremely efficiently, without the need for any inverse kinematics.

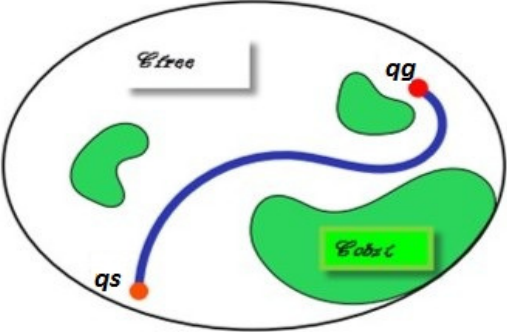


Figure1. Configuration space(C-space)

### 3. USING THE JACOBIAN

Given a robot arm configuration  $q \in Q$  (the configuration space) and a desired end effector goal  $x_g \in X$ , where  $X$  is the space of end effector positions  $R^3$ , we are interested in computing an extension in configuration space from  $q$  to wards  $x_g$ . Although the mapping from  $Q$  to  $X$  is often nonlinear and hence expensive to deduct, its derivative the Jacobian, is a linear map from the tangent space of  $Q$  to that of  $X$ , that can be computed easily ( $J\dot{q}=\dot{x}$ , where  $x \in X$  is the end effector position (or pose) corresponding to  $q$ ). Ideally, to drive the end effector to a desired configuration  $x_g$ , ( $d x_g / dt \approx 0$ : object moves slowly) we could compute the error  $e(t) = (x_g - x)$  and run a controller of the form  $\dot{q} = KJ^{-1}e$ , where  $K$  is a positive gain. This simple controller is capable to attain the target without considering any possible barriers or articulation limits. However this turn into a complex controller, where the inverse of the Jacobian must be done at each time step. To escape this expensive approach, we use alternatively the transpose of the Jacobian and the control law fall into the form of  $\dot{q} = KJ^T e$ . The controller eliminates the large overhead of computing the inverse by using the easy-to-compute Jacobian instead. It is easy to show that, under the same obstacle-free requirements as the Jacobian inverse controller, the Jacobian transpose(JT) controller is also guaranteed to reach the goal. The instantaneous motion of the end effector is given by  $\dot{x} = J\dot{q} = J(KJ^T e)$ . The inner product of this Instantaneous motion with the error vector is given by  $e^T \dot{x} = ke^T J J^T e \geq 0$ . As this is always positive, under our assumptions with obstacles, we may ensure that the controller will be able to make onward progress towards the target[24].

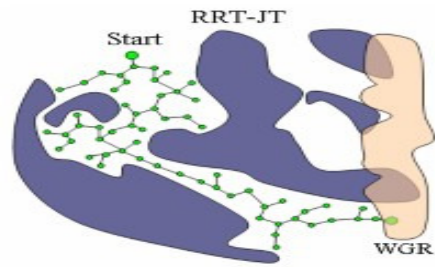


Figure 2. Depiction of the RRT-JT algorithm searching in C-space: from the start configuration to (WGRs). The forward-searching tree is shown with green nodes, the blue regions are obstacles, [14].

#### 4. THE WAM™ ARM

The WAM arm is a robotic back drivable manipulator. It has a stable joint-torque control with a direct-drive capability. It offers a zero backlash and near zero friction to enhance the performance of today's robots. It comes with three main variants 4-DoF, 7- DoF, both with human-like kinematics, and 4-DoF with 3-DoF Gimbals. Its articulation ranges go beyond those for conventional robotic arms [25].

We use WAM 7-DoF Arm with attached Barrett Hand.

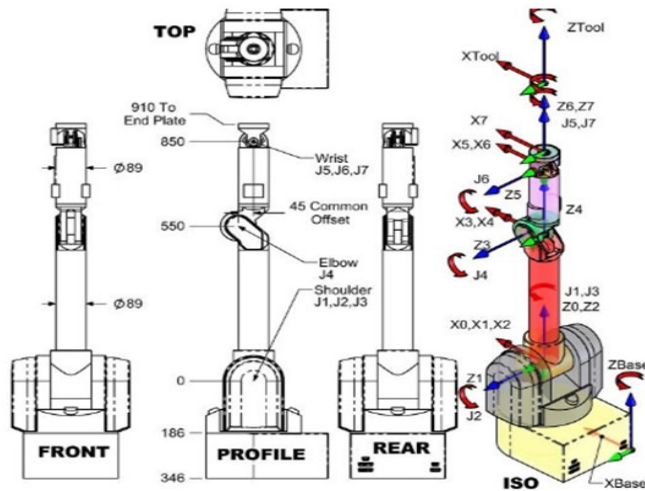


Figure 3. WAM 7-DoF dimensions and D-H frames, [26].

Figure 3. presents the whole 7-DoF WAM system in the initial position. A positive joint motion is on the right hand rule, for each axis. The following equation of homogeneous transformation in Figure 4 is used to determine the transformation between the axes K and K-1.

$${}^{k-1}T_k = \begin{bmatrix} \cos\theta_k & -\sin\theta_k \cos\alpha_k & \sin\theta_k \sin\alpha_k & a_k \cos\theta_k \\ \sin\theta_k & \cos\theta_k \cos\alpha_k & -\cos\theta_k \sin\alpha_k & a_k \sin\theta_k \\ 0 & \sin\alpha_k & \cos\alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 4. D-H generalized transform matrix

- $a_{k-1}$  = the distance from  $Z_{k-1}$  to  $Z_k$  measured along  $X_{k-1}$
- $d_k$  = the distance from  $X_{k-1}$  to  $X_k$  measured along  $Z_k$
- $\alpha_{k-1}$  = angle between  $Z_{k-1}$  to  $Z_k$  was approximately  $X_{k-1}$
- $\theta_k$  = angle between  $X_{k-1}$  to  $X_k$  was approximately  $Z_k$

The Table 1 contains the parameters of the arm with 7-DoF

Table 1. 7-DoF WAM frame parameters

$K$	$a_k$	$\alpha_k$	$d_k$	$\theta_k$
1	0	$-\pi/2$	0	$\theta_1$
2	0	$\pi/2$	0	$\theta_2$
3	0.045	$-\pi/2$	0.55	$\theta_3$
4	-0.045	$\pi/2$	0	$\theta_4$
5	0	$-\pi/2$	0.3	$\theta_5$
6	0	$\pi/2$	0	$\theta_6$
7	0	0	0.060	$\theta_7$
T	0	0	0	

As with the previous example, we define the frame  ${}^7T_{Tool}$  for our specific end effector. By multiplying all of the transforms up to and including the final frame, we determine the forward kinematics for any frame on the robot. To determine the end tip location and orientation we use the following equation:

$${}^0T_{Tool} = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 T_{Tool}$$

## 5. ROBOT DYNAMICS

The dynamic simulation uses the Lagrange equations to get the angular acceleration from the torque of each joint.

First, I computed the body Jacobian of each joint  $J_i$  corresponding to  $M_i$ , where  $M_i$  is the  $i$ th joint's inertia matrix. So the manipulator inertia matrix  $M(\theta)$  can be calculated as

$$M(\theta) = \sum_{i=1}^n J_i^T(\theta) M_i J_i(\theta)$$

Also calculate potential part,

$$P(\theta) = \sum_{i=1}^n m_i g h_i(\theta)$$

International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), Vol.3, No. 3/4, November 2014  
 Second, I computed the torques of each joint using Lagrange equation, which is,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = \tau$$

where  $\tau$  represents the torque of the joint and,

$$L(\theta, \dot{\theta}) = \frac{1}{2} \dot{\theta}^T M(\theta) \dot{\theta} - P(\theta)$$

After expanding the components of the equation, the equation is as following:

$$M(\theta)\ddot{\theta} + H(\theta, \dot{\theta})\dot{\theta} + G(\theta, \dot{\theta}) = \tau$$

where  $H(\theta, \dot{\theta})$  is the Coriolis and centrifugal force term and  $G(\theta, \dot{\theta})$  is the gravity term and  $\theta = [\theta_1 \theta_2 \theta_3 \theta_4 \theta_5 \theta_6 \theta_7]^T$  and  $\dot{\theta} = \frac{d\theta}{dt}$ ,  $\ddot{\theta} = \frac{d^2\theta}{dt^2}$ .

## 6. RESULTS AND ANALYSIS

To demonstrate and illustrate the proposed procedure, we present an example which the robot is equipped with a 7-DoF arm (see Figure 3) and a three-fingered Barrett hand (in fact in each time there are three tests: test1, test2 and test3). The goal is to follow a moving model mug, stably holding it, pick it up and move it to the desired position while avoiding the existing obstacles. The mug was moving in a straight line trajectory in the space with velocity range 8-32 mm/s. The initial positions of the end effector were (-0.730m, 0.140m, 2.168m) and those of the moving object were (-0.005m, -0.200m, 1.105m). In order to grasp the moving object stably and move it, the robot hand reaches the object than it closes its fingers.

### 6.1. Grasping object in the environment without obstacles

#### 6.1.1. Object moves with velocity $V=8\text{mm/s}$ :

The transformation equations used to update the manipulator's joints until the distance between the end effector and the moving object almost equal to zero. Once the position of the contact is achieved, the Barrett hand closes its fingers and grasps the object.

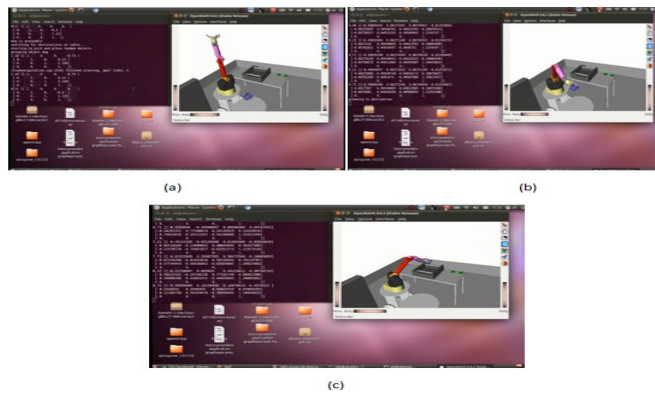


Figure 5: Successful grasping of a moving object

As mentioned in Figure 5, the grasping of the object is done successfully. Figure 5.a show that the hand of the robot keeps at a distance from the object, the Barret hand and the object are in the initial position, Figure 5.b the object moves with the velocity  $V_I = 8\text{mm/s}$  and the robot moves to the position of the centroid of the object, opens the fingers, closes the fingers and finally grasps the object. In Figure 5.c the robot picks up the object and moves it to the desired position.

To capture the moving object safety and to lift it up stably without slippage, the end effector needs to be as controlled as the relation between their position and the object's ones. So they determine the actual position of the moving object, and pick the closest distance between the end effector and the target.

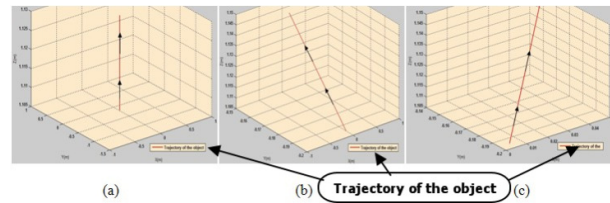


Figure 6: The trajectory of the object

Those three figures represent the same trajectory of a moving object with the same velocity  $V_I$  in a different dimension. Figure 6.a illustrates the trajectory based on the Z axis, while figure 6.b illustrates the trajectory in the plane(Y,Z), and figure 6.c is in the space(X,Y,Z). The object moves in a straight line.

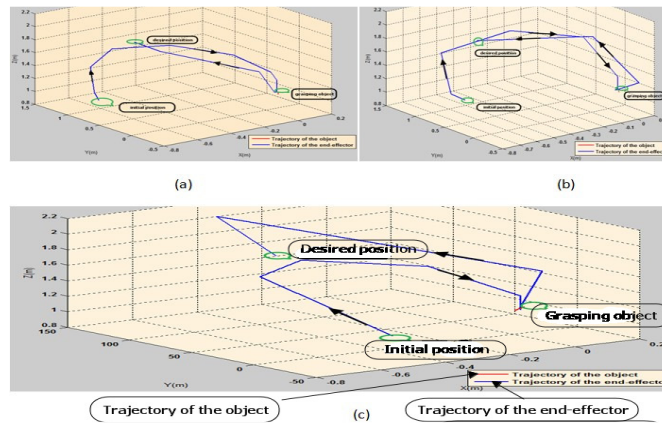


Figure 7: The trajectory of the object and the end effector

Figure 7.a illustrates the curves of the third test: the robot grasps the object in time  $T_{grasp} = 3.75$  s, which moves according to the Z axis with velocity  $V_I$ , Figure 7.b represents the curves of the first test: the robot grasps the object in time  $T_{grasp} = 3.99$  s, which moves in the plane(Y,Z) with velocity  $V_I$ , and in Figure 7.c the curves of the second test: the robot grasps the object in time  $T_{grasp} = 2.81$  s, which moves in the space(X,Y,Z) with velocity  $V_I$ .

Table 2. Object moves with  $V1$

	according (Z)axis		in(Y,Z)		in(X,Y,Z)	
	$T_{gras}$	$T_{end}(s)$	$T_{grasp}($	$T_{end}(s)$	$T_{grasp}($	$T_{end}(s)$
test	2.91	9.94	3.99	10.08	3.22	8.27
test	2.40	6.55	2.55	6.28	2.81	6.49
test	3.75	6.8	3.21	8.15	4.17-11.3	15.26

The table 2 provides the results in separately; the time for grasping the moving object and, the time to move the object to the desired position, the object moves with velocity  $V1$ . Times are nigh in the different test. In test3 where the object moves in the space, we note two times to grasping: the first grasping attempt fails, the robot does a second grasp and it succeeds.

**6.1.2. Object moves with velocity  $V2=4V1$ :**

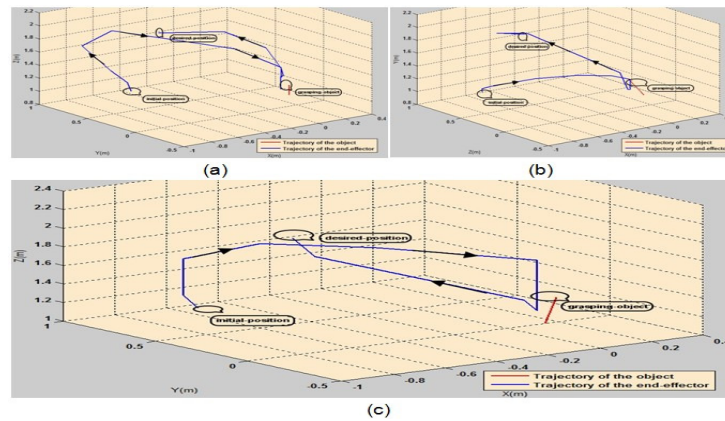


Figure 8: The trajectory of the end effector and the object

Figure 8.a the curves of the second test: the robot grasps the object in time  $T_{grasp}= 4.07$  s, the object moves according to the Z axis with velocity  $V2$ , Figure 8.b the curves of the third test: the robot grasps the object in time  $T_{grasp}= 3.48$  s, it moves in the plane (Y, Z) with velocity  $V2$ , Figure 8.c the curves of the second test: the robot grasps the object in time  $T_{grasp}= 3.02$  s, the latter moves in the space(X,Y,Z) with velocity  $V2$ .

Table 3. OBJECT moves with  $V2$

	according (Z)axis		in(Y,Z)		in(X,Y,Z)	
	$T_{grasp}(s)$	$T_{end}(s)$	$T_{grasp}(s)$	$T_{end}(s)$	$T_{grasp}(s)$	$T_{end}(s)$
test <sub>1</sub>	3.89	8.57	2.9	7.54	3.75	8.73
test <sub>2</sub>	4.07	9.93	3.05	8.57	3.02	8.18
test <sub>3</sub>	3.51	11.4	3.48	7.48	3.21	11.8

The table 3 presents results separately of the time for grasping the moving object which moves with velocity  $V2=4V1$  and the time to move the object to the desired position.



If we increase the velocity of the object, we see that the results are high but slightly higher. Therefore, increasing the speed affects on the time of grasping the moving object, even the direction of movement of the object affects on the time of grasping.

As shown in the tables, our algorithm successfully picked it up 100% of the time, and our robot successfully grasped the objects. We demonstrate that the robot is able to grasp the moving object in a reasonable time.

## 6.2 . Grasping object in the presence of one obstacle in the environment:

### 6.2.1. Object moves with velocity $V_I = 8\text{mm/s}$ :

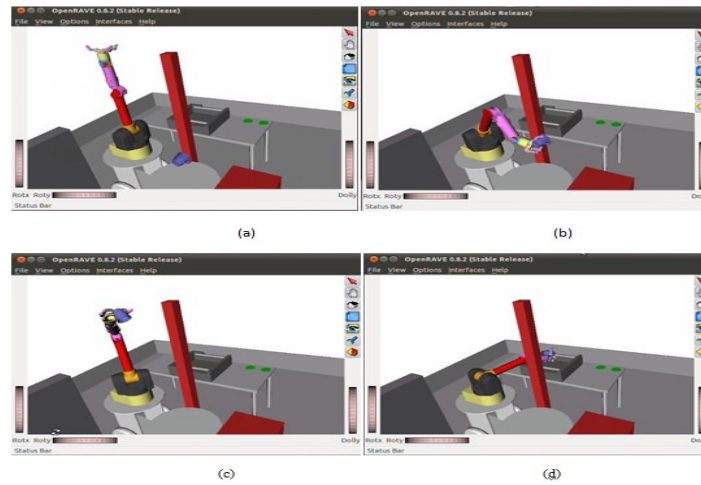


Figure 9. Grasps a moving object while avoiding obstacle with success.

As presented in Figure 9, the grasping of the object is done successfully. In Figure 9.a, the Barret hand and the object are in their initial locations. The object moves with a velocity of 8mm/s. In Figure 9.b, the robot reach the centroid of the object, while keep avoiding any encountered obstacles, and finally grasps the object and closes the fingers. In figure 9.c the robot pick up the object while avoiding obstacle and in figure 9.d the robot moves the object to a desired position.

To capture a moving object safety without collision and to lift up the object stably without slippage, the end effectors needs to be controlled while considering the relation between the position of the end effectors, the position of the moving object and the position of the obstacle. The end effectors determines the position of the obstacle (in the middle between the object and the end effectors) and the moving object and selects the shortest distance from its current position to the moving object, while avoiding obstacle in the environment.

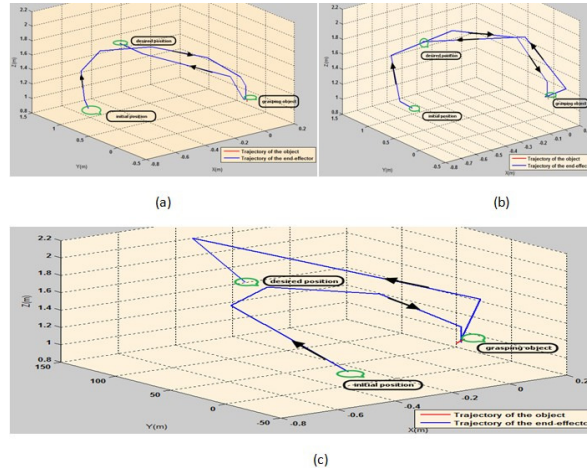


Figure 10. The trajectory of the end effector and the object

Figure 10.a illustrates the curves of the first test: the robot grasps the object in time  $T_{grasp} = 2.45$  s, which moves according to the Z axis with velocity  $V1$ , figure 10.b illustrates the curves of the third test: the robot grasps the object in time  $T_{grasp} = 3.03$  s, which moves in the plane(Y,Z) with velocity  $V1$ , figure 10.c illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp} = 2.91$  s, which moves in the space(X,Y,Z) with velocity  $V1$ .

Table 4.object moves with  $V1$  in the presence of obstacle

	according		in(Y,Z)		in(X,Y,Z)	
	$T_{grasp}(s)$	$T_{end}(s)$	$T_{grasp}(s)$	$T_{end}(s)$	$T_{grasp}(s)$	$T_{end}(s)$
test1	2.45	8.91	3.11	7.63	5.16	11.38
test2	2.95	9.08	2.83	9.33	2.91	7.07
test3	3.18	9.74	3.03	7.6	4.24	8.77

The table 4 presents results separately of the time for grasping the moving object which moves with velocity  $V1$  while avoiding obstacle, and the time to move the object to the desired position. Times are high in the different test. The direction of movement of the object affects on the time grasping ( $T_{grasp}$ ) and on the time to move the object to desired position ( $T_{end}$ ).

### 6.2.2. Object moves with velocity $V2=4V1$ :

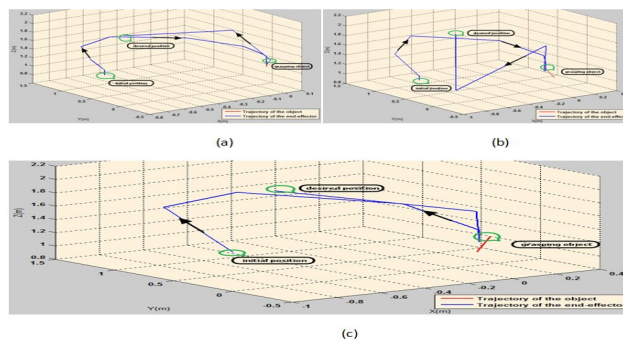


Figure 11. The trajectory of the end effector and the object

Figure 11.a illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp}=2.43$  s, which moves according to the Z axis with velocity  $V_2$ , figure 11.b illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp}=2.74$  s, which moves in the plane (Y, Z) with velocity  $V_2$ , figure 11.c illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp}=2.42$  s, which moves in the space(X,Y,Z) with velocity  $V_2$ .

Table 5. object moves with  $V_2$  in the presence of obstacle

	according (Z)axis		in(Y,Z)		in(X,Y,Z)	
	$T_{grasp}(s)$	$T_{end}(s)$	$T_{grasp}(s)$	$T_{end}(s)$	$T_{grasp}(s)$	$T_{end}(s)$
test <sub>1</sub>	2.96	9.47	3.58	9.57	3	9.51
test <sub>2</sub>	2.43	8.18	2.74	7.6	2.42	7.16
test <sub>3</sub>	2.37	6.87	2.63	8.13	2.54	7.35

The table 5 presents results separately of the time for grasping the moving object which moves with velocity  $V_2=4V_1$  while avoiding obstacle and the time to move the object to the desired position.

If we increase the velocity of the object, we see that the results are high but slightly higher. Therefore, increasing the speed affects on the time of grasping the moving object, even the direction of movement of the object affects on the time of grasping, we note that in the presence of obstacles the times are slightly higher than in the absence of obstacles.

As shown in the tables, our algorithm successfully picked it up 100% of the time, and our robot successfully grasped the objects. We demonstrate that the robot is able to grasp the moving object in a reasonable time. The times recorded in the presence of the obstacle are slightly higher than recorded in the absence of the obstacle.

### 6.3. Grasping object in the presence of two obstacles in the environment:

In the presence of obstacles, we plan a path in 7-DoF configuration space that takes the end effector from the starting position to a goal position, avoiding obstacles. For computing the goal orientation of the end effector and the configuration of the fingers, we used a criterion that attempts to minimize the opening of the hand without touching the object being grasped or other nearby obstacles. Finally finds a shortest path from the starting position to possible target positions.

6.3.1. Two obstacles between the object and the desired position(see Figure 12):

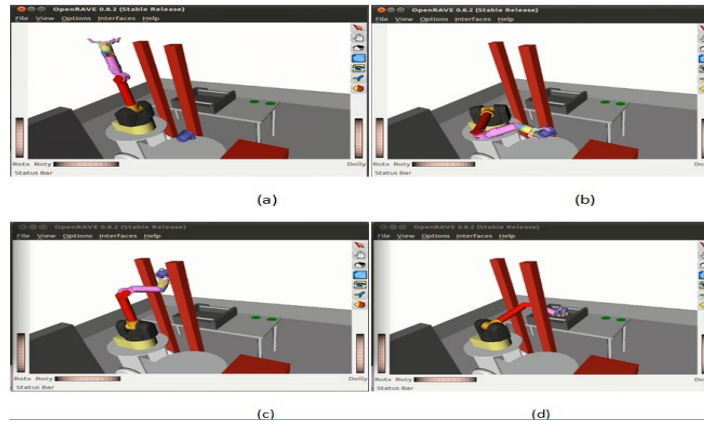


Figure 12. Two obstacles between the object and the desired position

The presence of two obstacles between the object and the desired position affects on the time to move the object to the desired position.

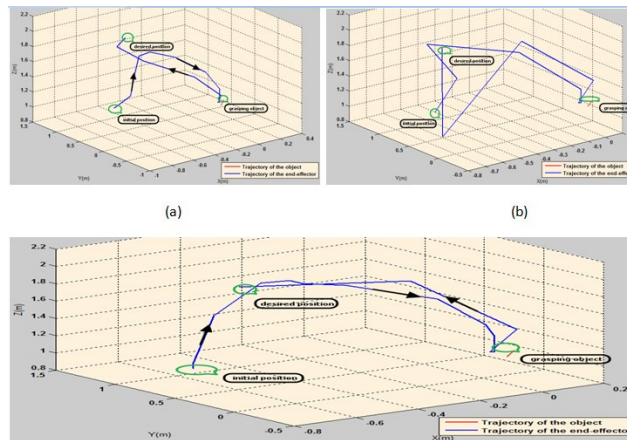


Figure 13. The trajectory of the end effector and the object while avoiding two obstacles between the object and the desired position

Figure 13.a illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp}= 3.67$  s and  $T_{end}= 8.99$  s, which moves according to the Z axis with velocity  $V_2$ , figure 13.b illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp}= 2.93$  s and  $T_{end}= 8.83$  s, which moves in the plane (Y, Z) with velocity  $V_2$ , figure 13.c illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp}= 3.12$  s and  $T_{end}= 10.62$  s, which moves in the space(X,Y,Z) with velocity  $V_2$ .

6.3.2. Two obstacles between the arm and the object(see Figure 14):

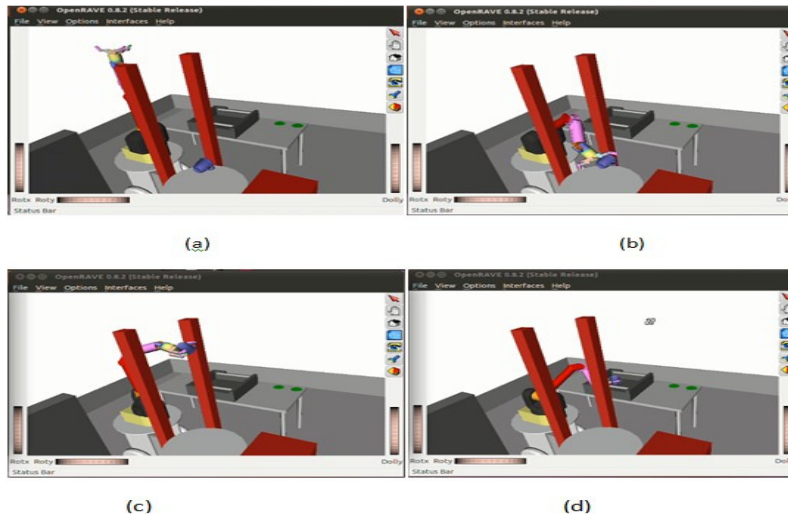


Figure 14. Two obstacles between the arm and the object

The presence of two obstacles between the arm and the object affects on the time to grasp the moving object.

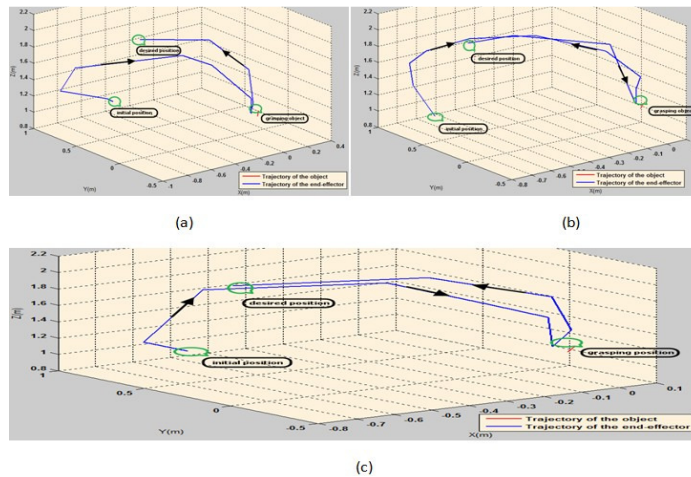


Figure 15. The trajectory of the end effector and the object while avoiding two obstacles between the arm and the object

Figure 15.a illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp}= 3.21$  s and  $T_{end}= 7.93$  s, which moves according to the Z axis with velocity  $V2$ , figure 15.b illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp}= 3.37$  s and  $T_{end}= 8.98$  s, which moves in the plane (Y, Z) with velocity  $V2$ , figure 15.c illustrates the curves of the second test: the robot grasps the object in time  $T_{grasp}= 3.47$  s and  $T_{end}= 7.50$  s, which moves in the space(X,Y,Z) with velocity  $V2$ .

### 6.4. Grasping object in the presence of three obstacles in the environment:

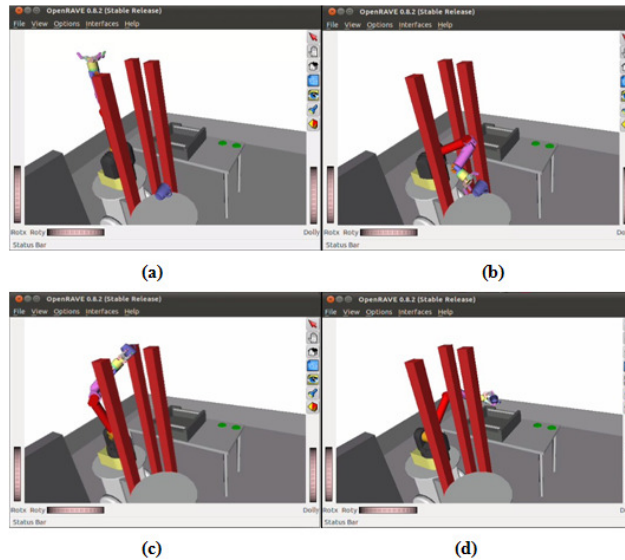


Figure 16. WAM Arm avoids three obstacles

In figure 16, the robot successfully grasping the item. Figure 16.a shows that the hand of the robot keeps a distance from the object, the Barret hand and the object are in the initial position, in figure 16.b the object moves with the velocity  $V_I = 8\text{mm/s}$  and the robot moves to the position of the centroid of the object, it chooses the shortest way to the object between the two obstacles, opens the fingers, closes them and finally grasps it in  $T_{grasp} = 2.46\text{s}$ . In figure 16.c the robot picks it up while avoiding obstacle and in figure 16.d the robot moves it to the desired position in  $T_{end} = 6.68\text{s}$ .

Table 6. Object moves with  $V_I$  in the presence of three obstacles

	according (Z) axis		in(Y,Z)		in(X,Y,Z)	
	$T_{grasp}(s)$	$T_{end}(s)$	$T_{grasp}(s)$	$T_{end}(s)$	$T_{grasp}(s)$	$T_{end}(s)$
test <sub>1</sub>	2.38	7.14	3.20	8.43	2.88	9.08
test <sub>2</sub>	2.56	7.88	2.50	7.30	2.46	6.68
test <sub>3</sub>	2.13	7.20	2.50	8.20	2.67	7.04

The table 6 presents results of the time for grasping the moving object which moves with velocity  $V_I$  while avoiding three obstacles and the time to move it to the desired position.

If we increase the number of the obstacles, the time of grasping the object was reduced. The choice of the trajectory by the robot is reduced. The robot chooses the shortest trajectory to the object.

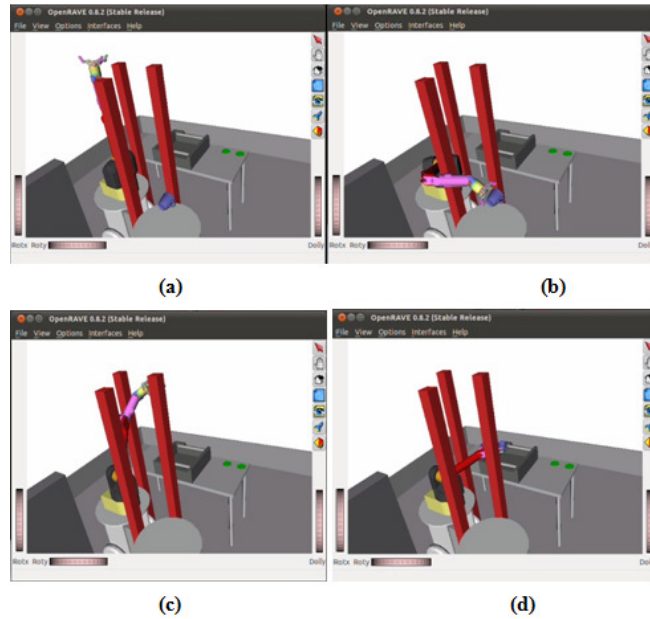


Figure 17. WAM Arm avoids three obstacles

As shown in the image sequence of figure 17, the robot successfully grasping the item. Figure 17.a shows that the hand of the robot keeps a distance from the object, the Barret hand and the object are in the initial position, in figure 17.b the object moves with the velocity  $V1= 8\text{mm/s}$  and the robot moves to the position of the centroid of the object, opens the fingers, closes them and finally grasps it in  $T_{grasp}=2.78\text{s}$ . In figure 17.c the robot picks it up while avoiding obstacle, it chooses the shortest way to the object between the two obstacles and in figure 17.d the robot moves it to the desired position in  $T_{end}= 6.90\text{s}$ .

## 7. CONCLUSIONS

So far, we have presented a simulation of grasping a moving object with different velocities in terms of displacing it to a desired position while avoiding obstacles using the 7-DoF robotic arm with the Barret hand in which we involve the RRT algorithm. In fact, this algorithm allows us to overcome the problem of the inverse kinematics by exploiting the nature of the Jacobian as a transformation from a configuration space to workspace.

We set forth separately the time of grasping the moving object shifting with different velocity in the presence and the absence of obstacles and the time to put this object in a desired position. Firstly, it moves with velocity  $V1$ . Secondly, it moves with velocity  $V2=4 V1$ . The proposed algorithm successfully holds the moving object in a rational time putting it in the aim station.

Times are high in the different test. The presence of obstacles increases the speed of grasping the object. The direction of movement affects on the time of holding the item and on the time to put it in the desired position. The times recorded in the presence of the obstacle are slightly higher than recorded in the absence of them.

In this article, we have studied an algorithm dedicated for grasping a moving target while trying to escape a fixed obstacle. A future work will tend to enhance the present algorithm, with introduction of movable obstacles.

## REFERENCES

- [1] F.Ruggiero, Grasp and manipulation of objects with an multi-fingered hand in unstructured environments, Ph.D. Thesis, UNIVERSITA DEGLI STUDI DI NAPOLI (2010).
- [2] T.Adam Leper, Kayden Hsiao, Strategies for human-in-the-loop robotic grasping, Robot Manipulation and Programming (2012).
- [3] K.Nagase, Y.Aiyana, Grasp motion planning with redundant dof of grasping pose, Journal of Robotics and Mechatronics Vol.25No.3 (2013).
- [4] H. Fester, A robot ping pong player: optimized mechanics, high performance 3d vision, and intelligent sensor control, Robotersysteme pp. 161170 (1990).
- [5] M.-S.Kim, Robot visual servo through trajectory estimation of moving object using kalman filter emerging intelligent computing technology and applications, D.-S.Huang, et al., Eds., ed: Springer Berlin / Heidelberg vol.5754(2009) pp.1122–1130.
- [6] F.Husain, Real time tracking and grasping of a moving object from range video(2013).
- [7] N.Hushing, Control of a robotic manipulator to grasp a moving target using vision, IEEE Int. Conf. Robotics and Automation (1990) 604-609.
- [8] A. Kiva, on adaptive vision feedback control of robotic manipulators, in: Proc. IEEE Conf. Decision and Control 2, 1991, pp.1883–1888.
- [9] B.M. J. Canny., Easily computable optimum grasps in 2-d and 3-d, In Proc. IEEE International Conference on Robotics and Automation (1994) pages 739–747.
- [10] J.B.Balkenius, C., Event prediction and object motion estimation in the development of visual attention, Proceedings of the Fifth International Workshop on Epigenetic (2005).
- [11] A.-B. S.Yeoh, P., Accurate real-time object tracking with linear prediction method, Proceedings of International Conference on Image Processing Volume 3 (2003).
- [12] Z.-K.Matas, J., Learning efficient linear predictors for motion estimation, Proceedings of 5 Indian Conference on Computer Vision, Graphics and Image Processing, Springer-Verlag, Madurai, India (2006).
- [13] J.-w. P. Sang-joo Kim, J. Lee., Implementation of tracking and capturing a moving object using a mobile robot, International journal of control automation and systems 3(2005)444.
- [14] D. Berenson, Manipulation Planning with Workspace Goal Regions, The Robotics Institute, Carnegie Mellon University, USA (2009).
- [15] J. K. M. Stilman, J.-U. Schamburek, T. Asfour, Manipulation planning among movable obstacles, in IROS (2007).
- [16] K.K.Y.Hirano, S.Yoshizawa, Image-based object recognition and dexterous hand/arm motion planning using rrt\* for grasping in cluttered scene, in IROS (2005).
- [17] S. LaValle, J. Kuffner, Rapidly-exploring random trees: Progress and prospects, in WAFR (2000).
- [18] S.M. LaValle, Planning algorithms, Cambridge, U.K.: Cambridge University Press, available at <http://planning.cs.uiuc.edu/>(2006).
- [19] R.D.D.Bertram, J.Kuffner, T.Asfour, An integrated approach to inverse kinematics and path planning for redundant manipulators, in ICRA (2006).
- [20] E. Drumwright, V.Ng-Thow-Hing, Toward interactive reaching in static environments for humanoid robots, in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (2006) pp.846–851.
- [21] D.F.M. Vande Weghe, S.Srinivasa, Randomized path planning for redundant manipulators without inverse kinematics, in Humanoids (2007).
- [22] S.S.S.Dmitry Berenson, Pittsburgh and by the National Science Foundation under Grant No. EEC-0540865 (2009).
- [23] M.V.Weghe, Randomized path planning for redundant manipulators without inverse kinematics.
- [24] E.L.Damian, Grasp planning for object manipulation by an autonomous robot, Master's thesis, National Institute of Applied Sciences of Toulouse (2006).
- [25] M.W.Spong, M.Vidyasagar, Robot dynamics and control (1989).
- [26] WAM Arm User's Guide, [www.barrett.com](http://www.barrett.com)



## Authors

Ali CHAABANI was born in Kairouan, Tunisia, in 1987. He received, from the Sfax National School of Engineering (ENIS), the Principal Engineering Diploma in Computer science in 2011, from the National Institute of Applied Sciences and Technology (INSAT) the Master of Computer science and Automation in 2013. Now, he is a PhD student in Tunis National School of Engineering (ENIT) and researcher in Informatics Laboratory for Industrial Systems (LISI) at the National Institute of Applied Sciences and Technology (INSAT).



Mohamed Sahbi BELLAMINE is an assistant professor in the National Institute of Applied Sciences and Technology, University of Carthage. He is a member of Laboratory of Computer Science of Industrials Systems (LISI). His domain of interests is Human-Robot Interaction, Social & Sociable robots.



Moncef GASMI was born in Tunis, Tunisia, in 1958. He received the Principal Engineering Diploma in Electrical Engineering in 1984 from the Tunis National School of Engineering (ENIT), the Master of Systems Analysis and Computational Treatment in 1985, the Doctorate in Automatic Control in 1989 and the State Doctorate in Electrical Engineering in 2001. Now, he teach as a Professor and Director of the Informatics Laboratory for Industrial Systems (LISI) at the National Institute of Applied Sciences and Technology (INSAT). His interest are about

