Controlling a Humanoid Robot Arm for Grasping and Handling a Moving Object in 3-D Environment without Cameras

Ali Chaabani
National School of Engineering of Tunis, University of Manar, Tunisia
Email: chaabani.ali@gmail.com

Mohamed Sahbi Bellamine and Moncef Gasmi
Computer Laboratory for Industrial Systems, National Institute of Applied Sciences and Technology, University of Carthage, Tunisia
Email: aroussia@insat.rnu.tn, mcfgsm@yahoo.fr

Abstract—**Lot of researchers have worked on robotic grasping tasks on stationary or fixed objects, others have focused on objects in dynamic motion using cameras to record their images in order to treat them to estimate the position of the capture. This method is very difficult, requiring a lot of computing, image processing. Therefore, we should recall another simple method of handling. In addition, the majority of robotic arms available for humanoid control applications are complex and yet expensive. In this paper, we are going to detail the requirements to manipulating a humanoid robot arm with 7 degree-of-freedom to grasp and handle any moving objects in the 3-D environment in the presence or not of obstacles and without using the cameras. We used the OpenRAVE simulation environment and a robot arm equipped with the Barrett hand. We also describe a randomized planning algorithm capable of planning. This algorithm is an extension of RRT-JT that interleaves exploration using a Rapidly-exploring Random Tree with exploitation using Jacobian-based gradient descent to control a 7-DoF WAM robotic arm to avoid the obstacles, track a moving object, and grasp planning. We present results in which a moving mug is tracked, stably grasped with a maximum rate of success in a reasonable time and picked up by the Barret hand to a desired position.**

Index Terms—**grasping, moving object, trajectory planning, robot hand, obstacles**

I. 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.

In the field of robotics, many applications have been tailored towards servoing using visual information. The goal is to use information obtained from vision inside a servo loop to control a mobile manipulator [4], [5], [6]. These challenges are the major reason for a limited performance in the tracking and grasping process which can be solved via use of predictive algorithms. [7] 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. [10] proposed a method to grasp efficiently the objects and developed a system able to grasp industrial parts moving on a conveyor belt by controlling a 6DOF robot arm with a camera mounted on its gripper. [11] implemented a real time vision system with a single camera for identifying and intercepting several objects. [12] proposed a visual servo system for real-time tracking and grasping of a moving object and a parallel method was adopted to raise matching speed.

These researchers have recognized that the main problems in the visual servoing are to solve the delay introduced by image processing or the response of the robot system and resolve the target occlusion. These troubles are the major reason for a limited performance in the tracking and grasping process which can be solved through of the use of predictive algorithms. [13] 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. [14] present a tracking algorithm based

Manuscript received August 15, 2014; accepted December 15, 2014.

doi: 10.12720/joace.3.5.391-397

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. [15] 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. [16] presents a binocular eye-to-hand visual servoing system that is able to track and grasp a moving object in real time. In the tracking module, they use three linear predictors (one for each component of the three dimensions) to predict and generate the trajectory that will describe the 3D object position in the near future, therefore, their manipulator robot is able to track and grasp a moving object, even if the object is temporarily occluded. [17] 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 witch based on many calculated and estimation [18]. 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.

Following this introduction to the grasp planning problem, the solutions already published in the literature, and how my solution is unique. Section II is devoted to the detailed description of the Rapidly-Exploring Random Trees (RRT), and the transpose of the Jacobian is briefly given in Section III. The next section contains a description of the WAM™ arm. In Section V, some results are given. Section VI presents conclusions drawn from this work.

II. Rapidly-Exploring Random Trees (RRT)

In previous work [19], [20], researchers have tackled the motion planning problem by sampling some number of end effector poses from the goal regions and using inverse kinematics(IK) to find joint configurations which place the end effector at the sampled locations. These configurations are then set as goals for a randomized planner, such as an RRT or BiRRT [21], [22]. While often capable of solving the problem at hand, this approach is neither probabilistically complete nor efficient. The issue is that some number of samples from the goal regions are chosen a priori as goal configurations, and the planner is forced to use only these goals.

Another approach to planning with certain types of workspace goals is to explore the Configuration space (C-space (see Fig. 1) of the robot with a single search tree that uses heuristics to bias the exploration toward a goal region [23]. However, the goal regions and heuristics defined in [24] are highly problem specific and difficult to tune. Drumwright and Ng-Thow-Hing [25] employ a similar strategy of extending toward a randomly-generated IK solution for a workspace point. In [26], 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.

[27] 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 (Fig. 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.

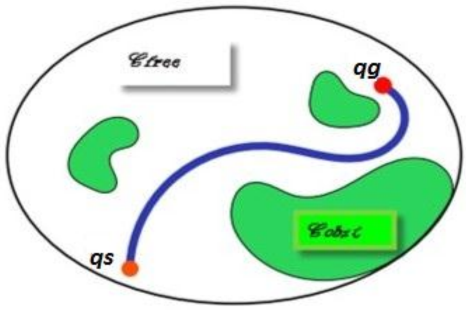

Figure 1. Configuration space(C-space)

III. 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 R3, we are interested in computing an extension in configuration space from q to wards x_g. Unfortunately, the mapping from Q to X is usually nonlinear and very expensive to compute. However, its derivative, called the Jacobian, is a linear map from the tangent space of Q to that of X, is expressed as $J\dot{q}=\dot{x}$, where $x \in X$ is the end-effector position (or pose) corresponding to q, and can be computed quickly. Ideally, to drive the end-effector to a desired configuration x_g, ($dx_g/dt \approx 0$: 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. In the absence of any obstacles, internal collisions, or joint limits, this simple controller is guaranteed to reach the goal. Unfortunately, in the absence of a closed form solution, the computation of the inverse of the Jacobian must be done numerically at each time step. An alternate approach, is to use the transpose of the Jacobian instead of the inverse. This results in a control law of the form $\dot{q}=KJ^{T}e$. 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 $e^{T}\dot{x} = ke^{T}JJ^{T}e \geq 0$. Since this is always positive, under our assumptions about obstacles, the controller is guaranteed to make forward progress towards the goal [27].

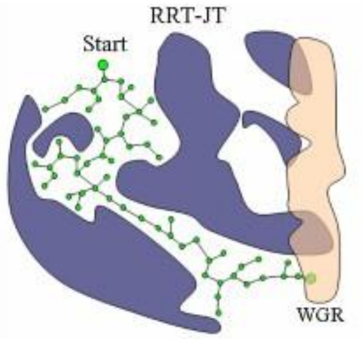

Figure 2. Depiction of the RRT-JT algorithm searching in C-space: from the start configuration to (WGRs). The blue regions are obstacles, the forward-searching tree is shown with green nodes, [18].

IV. THE WAM™ ARM

The WAM Arm is a highly dexterous backdrivable manipulator. It is the only commercially available robotic arm with direct-drive capability supported by Transparent Dynamics between the motors and joints, so its joint-torque control is unmatched and guaranteed stable. It is built to outperform today's conventional robots by offering extra ordinary dexterity, zero backlash, and near-zero friction. The WAM Arm is available in 3 main configurations, 4-DOF, 7- DOF, both with human-like kinematics, and 4-DOF with 3-DOF Gimbals. The joint ranges exceed those for conventional robotic arms [28].

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

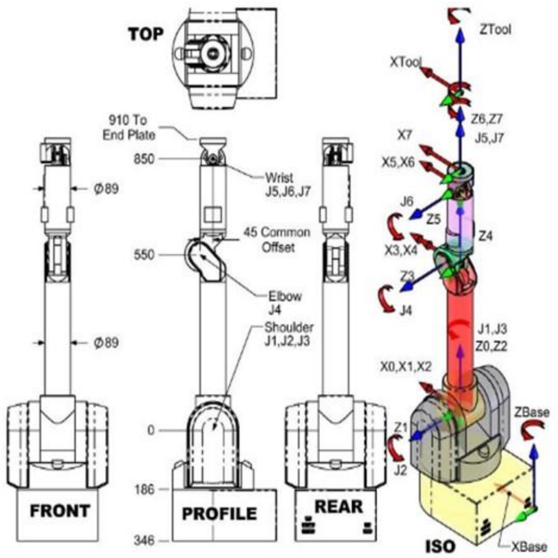

Figure 3. WAM 7-DOF dimensions and D-H frames, [30]

Fig. 3 shows the entire 7-DOF WAM system in the zero position. A positive joint motion is based on the right hand rule for each axis. The following equation of homogeneous transformation in Fig. 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
- α_{k-1}=angle between Z_{k-1} to Z_k was approximately X_{k-1}
- θ_k =angle between X_{k-1} to X_k was approximately Z_k

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

TABLE I. 7-DOFWAM FRAME PARAMETERS

K	a_k	α_k	d_k	θ_k
1	0	$-\pi/2$	0	θ_1
2	0	$\pi/2$	0	θ_2
3	0.045	$-\pi/2$	0.55	θ_3
4	−0.045	$\pi/2$	0	θ_4
5	0	$-\pi/2$	0.3	θ_5
6	0	$\pi/2$	0	θ_6
7	0	0	0.060	θ_7
T	0	0	0	

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

$$ {}^{0}T_{Tool} = {}^{0}T^{1}_{1}\, T^{2}_{2}\, T^{3}_{3}\, T^{4}_{4}\, T^{5}_{5}\, T^{6}_{6}\, T^{7}_{7}\, T_{Tool} $$

V. 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 Fig. 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.

A. *Grasping Object in the Environment without Obstacles*

1) Case study №1: Moving object with velocity V_1= 8mm/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 Barret hand closes its fingers and grasp the object.

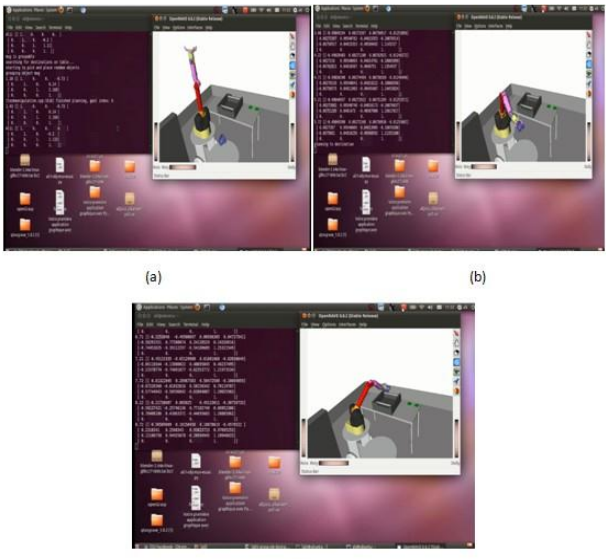

Figure 5. Successful grasping of a moving object

As shown in the image sequence of Fig. 5, the tracking and grasping of the object is achieved efficiently. Fig. 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, Fig. 5.b the object moves with the velocity *V1*= 8mm/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 Fig. 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'ones. So they determine the position of the moving object and select the shortest distance from its current position to the moving object.

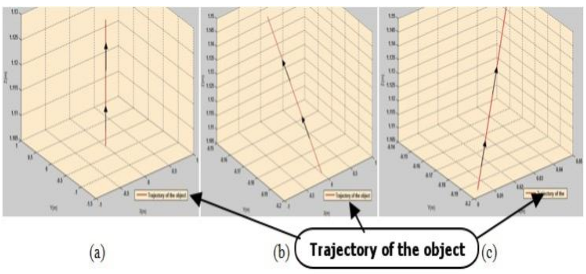

Figure 6. The trajectory of the object

Those tree figures represent the same trajectory of a moving object with the same velocity *V1* in a different dimension. Fig. 6.a illustrates the trajectory based on the Z axis, while Fig. 6.b illustrates the trajectory in the plane (Y, Z), and Fig. 6.c is in the space (X,Y,Z). The object moves in a straight line.

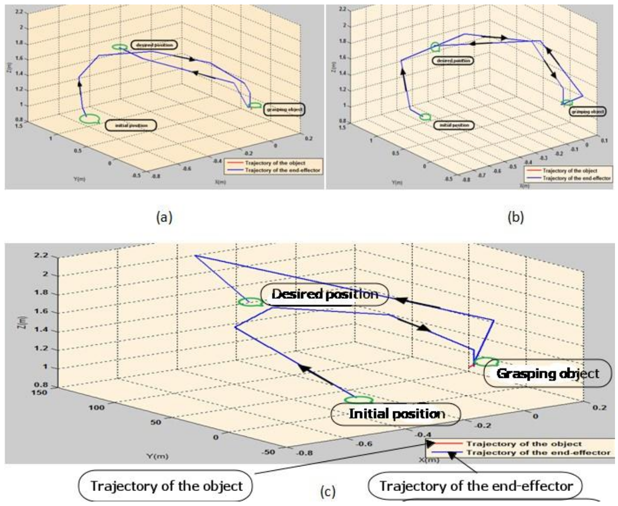

Figure 7. The trajectory of the end-effector and the object

Fig. 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 *V1*, Fig. 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 *V1*, and in Fig. 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 *V1*.

TABLE II. OBJECT MOVES WITH V_1

	according (Z)axis		**In (Y,Z)**		in(X,Y,Z)	
	T_{grasp} (s)	T_{end} (s)	Tgrasp (s)	T_{end} (s)	Tgrasp (s)	Tend **(s)**
$test_1$	2.91	9.94	3.99	10.08	3.22	**8.27**
$test_2$	2.40	6.55	2.55	6.28	2.81	6.49
$test_3$	3.75	6.8	3.21	8.15	4.17-11.3	15.26

The Table II 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 test 3 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.

2) Case study №2: Moving object with velocity V2=4V1:

Fig. 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*, Fig. 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*, Fig. 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*.

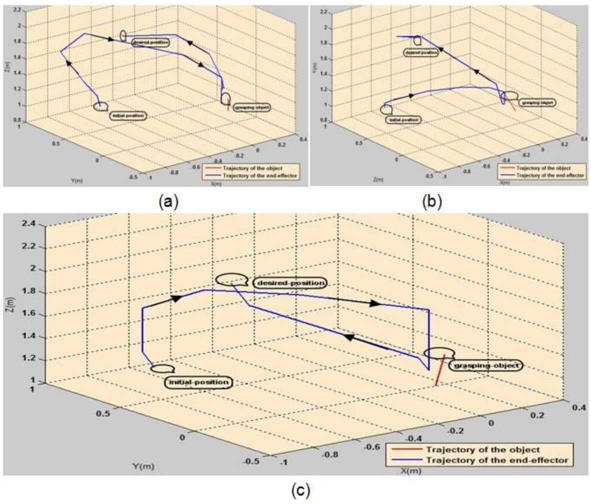

Figure 8. The trajectory of the end-effector and the object

TABLE III. OBJECT MOVES WITH V_2

	according (Z)axis		in(Y,Z)		in(X,Y,Z)	
	Tgrasp (s)	Tend(s)	Tgrasp (s)	Tend (s)	Tgrasp (s)	Tend (s)
$test_1$	3.89	8.57	2.9	7.54	3.75	8.73
$test_2$	4.07	9.93	3.05	8.57	3.02	8.18
$test_3$	3.51	11.4	3.48	7.48	3.21	11.8

The Table III presents results separately of the time for grasping the moving object which moves with velocity *V2*=4*V1* 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 nigh 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.

B. Grasping Object in the Presence of Obstacle

1) Case study №1: Moving object with velocity V_1= 8mm/s in the presence of obstacle

As shown in the image sequence in Fig. 9, the tracking and the grasping of the item is achieved efficiently. Fig. 9.a shows that the hand of the robot keeps a distance from it, the Barret hand and the object are in the initial position, Fig. 9.b the object moves with the velocity *V1*= 8mm/s and the robot moves to the position of the object's centroid, avoids the obstacle, opens the fingers, closes them back and finally grasps it. In Fig. 9.c the robot picks it up while avoiding obstacle and in Fig. 9.d the robot takes it to a determined position.

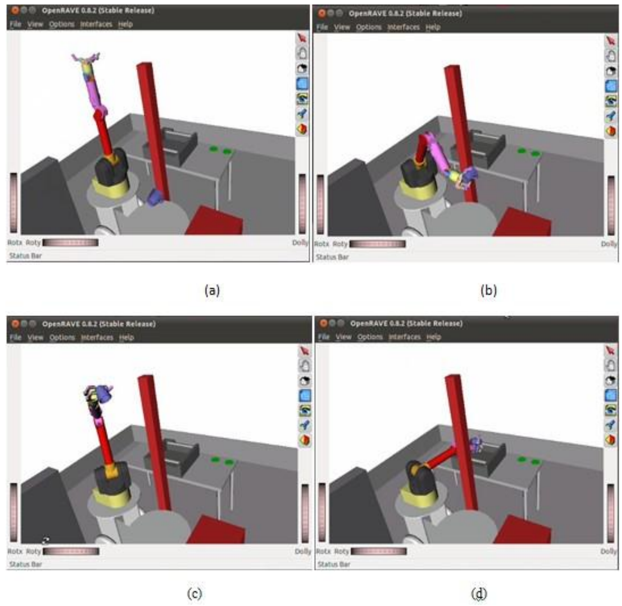

Figure 9. Successful grasping of a moving object while avoiding obstacle

To capture a moving object safety without collision and to lift it up stably without slippage, the end effector needs to be controlled while considering the relation between its position, the moving object's position and the obstacle one's. It determines the position of the moving object and of the obstacle (in the middle between the object and the end-effector) and select the shortest distance from its current position, while avoiding obstacle in the environment.

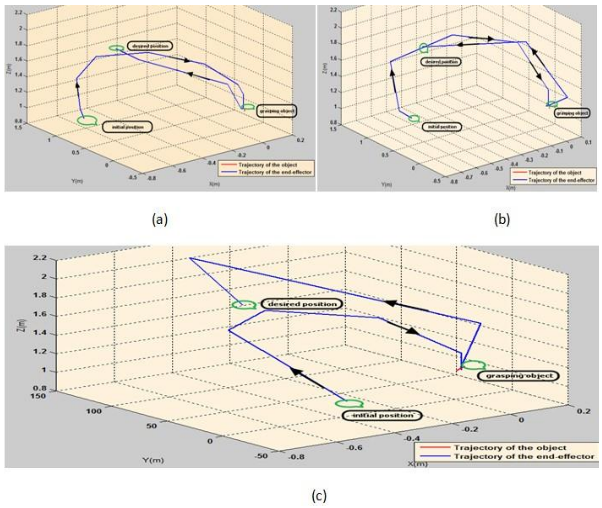

Figure 10. The trajectory of the end-effector and the object

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

TABLE IV. OBJECT MOVES WITH V_1 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)
$test_1$	2.45	8.91	3.11	7.63	5.16	11.38
$test_2$	2.95	9.08	2.83	9.33	2.91	7.07
$test_3$	3.18	9.74	3.03	7.6	4.24	8.77

The Table IV presents the results of the time for grasping the moving object while avoiding obstacle, and the time to move it to the desired position, as always it moves with velocity *V1*. Times are nigh in all tests. The direction of the object's movement affects on the time grasping (T_{grasp})and on the time to move it to desired position (T_{end}).

2) Case study №2: Moving object with velocity V_2=4V_1 in the presence of obstacle

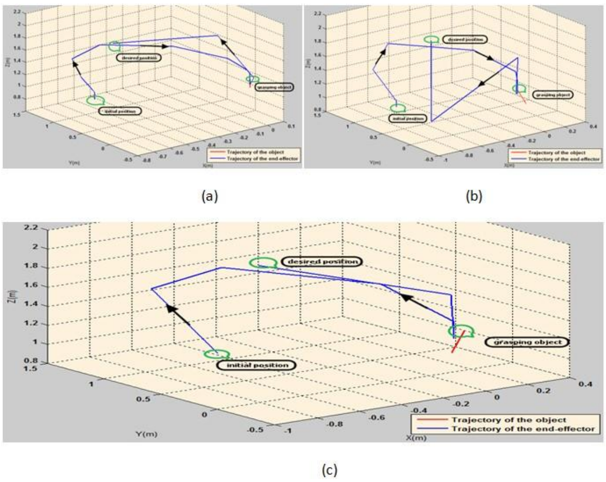

Figure 11. The trajectory of the end-effector and the object

Fig. 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 , Fig. 11.b represents the curves of the second test: the robot grasps the object in time T_{grasp}= 2.74 s, the object moves in the plane (Y, Z) with velocity $V_{2,}$ in Fig. 11.c the curves of the second test: the robot grasps the object in time T_{grasp}= 2.42 s, the movement is in space(X,Y,Z) with velocity $V_{2.}$

TABLE V. 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)	Tgrasp (s)	T_{end} (s)	T_{grasp} (s)	T_{end} (s)
$test_1$	2.96	9.47	3.58	9.57	3	9.51
$test_2$	2.43	8.18	2.74	7.6	2.42	7.16
$test_3$	2.37	6.87	2.63	8.13	2.54	7.35

The Table V shows the results of the time for grasping the moving object which moves with velocity *V2*=4*V1* 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 close but slightly higher. Therefore, increasing the speed affects on the time of grasping the moving object, even the direction of the object's movement affects on the time of holding, we note that in the presence of obstacles the times are slightly higher than in their absence.

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

VI. CONCLUSION

So far, we have presented a simulation of grasping a moving object with different velocities in terme to deplace 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 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 for grasping the moving object which moves with different velocity while the obstacles are absent and present, and also the time to put this object in a desired position. Firstly, the object moves with velocity *V1* Second the object moves with velocity *V2*=4 *V1*. The proposed algorithm successfully grasp the moving object in a rational time and put it in a desired position.

Times are nigh in the most of the tests. The presence of obstacles, increasing the speed of grasping the object. The direction of the object's movement affects the time of grasping the object and the time to put the object in a determined position. The times recorded in the presence of the obstacle are slightly higher than recorded in the absence of the obstacle.

In this article, we proposed an algorithm for grasping a moving object in the presence of a fixed obstacle. Future work will aim at improving the grasping in the presence of a movable obstacles.

REFERENCES

[1] F. Ruggiero, "Grasp and manipulation of objects with am ulti-fingered hand in unstructured environments," Ph. D thesis, Universita Deglistudi Di Napoli (2010).

[2] M. C. L. T. A. Leper and K. 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. 25, no. 3, 2013.

[4] H. Faster, "A robot ping pong player: Optimized mechanics, high performance 3dvision, and intelligent sensor control," *Robotersysteme*, pp. 161170, 1990.

[5] M. S. Kim, "Robot visual servo through trajectory estimation of amoving object using kalman filter emerging intelligent

computing technology and applications," D. S.Huang, et al., ed. Springer Berlin /Heidel- bergvol.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," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1990, pp. 604-609.

[8] A. Kiva, "On adaptive vision feed back control of robotic manipulators," in *Proc. IEEE Conf. Decision and Control2*, 1991, pp. 1883–1888.

[9] B. M. J. Canny, "Easily computable optimum grasps in2-dand 3-d," in *Proc. IEEE International Conference on Robotics and Automation*, 1994, pp. 739–747.

[10] N. T. Nomura, "Integrated visual serving system to grasp industrial parts moving on conveyer by controlling 6 dof arm," in *Proc. the IEEE International Conference on Systems, Man, and Cybernetics*, 2000, pp. 1768–1775.

[11] M. C. D. S. E. A. Sen, G., "Identification and prediction of a moving object using realtime global vision sensing," in *Proc. the 20th IEEE Instrumentation and Measurement Technology Conference*, vol. 2, 2003, pp. 1402–1406.

[12] J. Z. Ge, "Areal time stereo visual servoing for moving object grasping based parallel algorithms," in *Proc. IEEE Conference on Industrial Electronics and Applications*, 2007, pp. 2886 289.

[13] J. B. Balkenius, "Event prediction and object motion estimation in the development of visual attention," in *Proc. Fifth International Workshop on Epigenetic*, 2005.

[14] A. B. S. Yeoh, "Accurate realtime object tracking with linear prediction method," in *Proc. International Conferenceon Image Processing*, vol. 3, 2003.

[15] Z. K. Matas, "Learning efficient linear predictors for motion estimation," in *Proc. 5 Indian Conference on Computer Vision, Graphics and Image Processing*, Springer-Verlag, Madurai, India, 2006.

[16] F. Pacheco and J. Binocular, "Visual tracking and grasping of a moving object with a 3 dtrajectory predictor," in *Proc. International Journal of Applied Research and Technology*, vol. 7, no. 3, 2009.

[17] J. W. P. S. J. Kim, and J. Lee, "Implementation of tracking and capturing a moving object using a mobile robot," *International Journal of Control Automation and Systems*, vol. 3, p.444, 2005.

[18] D. Berenson, "Manipulation planning with workspace goal regions," The Robotics Institute, Carnegie Mellon University, USA, 2009.

[19] J. K. M. Stilman, J. U. Schamburek, and T. Asfour, "Manipulation planning among movable obstacles," *IROS*, 2007.

[20] K. K. Y. Hirano and S. Yoshizawa, "Image-based object recognition and dexteroushand/arm motion planning using rrts for grasping in cluttered scene," *IROS*, 2005.

[21] S. LaValle and J. Kuffner, "0Rapidly-exploring random trees: Progress and prospects," *WAFR*, 2000.

[22] S. M. LaValle. Planning algorithms. [Online]. Available: http://planning.cs.uiuc.edu/(2006).

[23] R. D. D. Bertram, J. Kuffner, and T. Asfour, "An integrated approach to inverse kinematics and path planning for redundant manipulators," *ICRA*, 2006.

[24] E. Drumwright and 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.

[25] D. F. M. V. Weghe and S. Srinivasa, "Randomized path planning forredundant manipulators without inverse kinematics," *Humanoids*, 2007.

[26] S. D. Berenson, Pittsburgh and by the National Science Foundationunder GrantNo.EEC-0540865 (2009).

[27] M. V. Weghe, "Randomized path planning for redundant manipulators without inverse kinematics,"

[28] E. L. Damian, "Grasp planning for object manipulation by an autonomous robot," Master's thesis, National Institute of Applied Sciences of Toulouze, 2006.

[29] M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control*, 1989.

[30] WAM Arm User's Guide. [Online]. Available: www.barrett.com

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

Moncef GASMI was born in Tunis,Tunisia, in 1958. He received respectively, from the Tunis National School of Engineering (ENIT), the Principal Engineering Diploma in Electrical Engineering in 1984, 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 is Professor and Director of the Informatics Laboratory for Industrial Systems (LISI) at the National Institute of Applied Sciences and Technology (INSAT). His domain of interests is related to the modeling, analysis and control of complex systems.