OPTIMISED COLLISION-FREE TRAJECTORY AND CONTROLLER DESIGN FOR ROBOTIC MANIPULATORS

A. Seddaoui and C. M. Saaj

Surrey Space Centre, University of Surrey, GU2 7XH Guildford, United Kingdom
(a.seddaoui, c.saaj)@surrey.ac.uk

ABSTRACT

Path planning and collision avoidance are two crucial interconnected algorithms used to perform desired tasks for both fixed and moving base manipulators. In this paper, the collision-free trajectory generation algorithm presented for capturing a stationary target is a newer form of the state-of-the-art method using cycloids. It is further optimised to minimise the distance between the obstacle and the manipulator. A control algorithm using the Computed Torque Control to accurately track the pre-designed trajectory is discussed. The simulations results presented in this paper verify the efficacy and robustness of the controller based on the non-linear dynamical model of the robotic manipulator. The optimised collision-free trajectory resulted in smooth joint displacement, velocity and acceleration. Moreover, the algorithm presented can be applied to ground based and space based stationary or moving robotic manipulators.

Key words: Collision-free Trajectory; Control; Robotic Manipulator.

1. INTRODUCTION

Robotic manipulators are very widely used in terrestrial applications and they are increasingly needed for both orbital and planetary missions. The design of fixed-base and moving base manipulators is different when it comes to the significance of the dynamic coupling between the arm and the supporting base. When fixed-base manipulators are considered, the designer has to take into account path constraints. These constraints are points on the trajectory that the end-effector has to go through to achieve the path. Whereas when moving base manipulators are considered, the path is constrained by the amount of reaction forces applied on the base which need to be minimal. The larger is the joint displacement the higher are the forces, hence the need for optimisation. Another issue of high importance for both fixed and moving base manipulators is the presence of obstacles, which highlights the need for a collision-free path planning algorithm.
Various techniques for trajectory planning for both fixed and moving base manipulators have been developed, such as the Artificial Potential Fields (APF) approach where the moving point representing the end-effector, is driven away from high potential areas (obstacles) towards low potential areas tracing the desired path until it reaches the lowest potential (goal point). This approach was used on a fixed-base industrial robot in [1] and a space moving base manipulator by utilising the improved version of the APF in [2]. The latter algorithm was combined with the inverse kinematics based on the Generalised Jacobian Matrix (GJM) developed by Umetani and Yoshida in [3]. In the same context, an Ant Colony algorithm was presented to avoid trapping into local optimum which is one of the APF common issues [4]. As mentioned previously, the challenge when designing trajectories for moving base manipulators is to minimise the reaction forces on the base due to the dynamic coupling. Dubowsky and Torres developed a graphical tool called the Enhanced Disturbance Map (EDM) to make the manipulator move along a zero disturbance line, also called reaction-less trajectory [5]. They defined hot areas, where small movements of the manipulator result in large disturbances, and cool areas also called zero disturbance paths where reaction forces are minimal. These techniques proved to be efficient; however, they do not allow real time implementation since they require full preliminary knowledge of the manipulator's surrounding environment.
Polynomials also proved to be an effective method to design trajectories for both terrestrial fixed base and space moving base manipulators [6, 7]. Papadopoulos et al. used a polynomial to design trajectories in the configuration space i.e. the generalised coordinates are the joints displacement angles [7]. In the same context, Zhang optimised a polynomial function for minimisation of the attitude changes due to the reaction forces generated during the path planning execution [8]. Huang presented a method for path planning that adds constraints on the joint displacement and velocity to limit the impact forces at the capture point i.e. final position in the Cartesian space [9]. Xu later improved this method using polynomials to find smooth trajectories at the joint displacement, velocity and acceleration level [10]. The latter algorithm developed and applied on space moving base manipulators, was also used on terrestrial fixed base manipulators [11, 12]. This proves the efficacy of polynomials in path planning for both fixed and moving base manipulators.

Although there are various techniques to design the trajectory of the manipulator, it is often challenging to control the motion of the end-effector precisely and to avoid obstacles using classical controllers, such as the Proportional Derivative Controller. The issues are mainly due to the inaccuracies of the model used to design the controller or the limitation of linear controllers, which are used for real-time control of highly non-linear manipulator system.
The algorithm presented in this paper is an improved version of the state-of-the-art method that uses a fifth order polynomial to design a smooth trajectory between the initial and final position while avoiding obstacles using four harmonic functions. The harmonic functions are made of four quarters of cycloidal equations [12]. This method allowed the manipulator to avoid the obstacle but was not power efficient since the actuated joints followed a large deviation during the execution of the obstacle avoidance algorithm. Hence, it was further optimised using a newer version of the harmonic function, to achieve obstacle avoidance. The new version of the algorithm is based on a function that consists of the companion curve of a cycloid, which is a continuous function over the obstacle avoidance portion of the Cartesian and joint space trajectories. Furthermore, a nonlinear Computed Torque Control is designed to track the desired trajectory.

2. SYSTEM DESIGN AND MODELLING

The system consists of an N Degrees of Freedom (DoF) fixed-base manipulator with revolute joints. The vector of generalised coordinates $\boldsymbol{q}$ is equal to the vector of the revolute joints displacement $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_n]$. The equation of motion can be derived using the Lagrange approach and takes the following matrix form [13]:

$$\boldsymbol{\tau} = \mathbf{D}(\theta)\ddot{\theta} + \mathbf{C}(\theta, \dot{\theta})\dot{\theta} + \boldsymbol{G}(\theta), \tag{1}$$

where:
$\boldsymbol{\tau}$ is the Nx1 torque vector.
$\mathbf{D}(\theta)$ is the NxN symmetric positive finite matrix, representing the inertia of the system.
$\mathbf{C}(\theta, \dot{\theta})$ is the NxN matrix, representing the Coriolis and centrifugal effects.
$\boldsymbol{G}(\theta)$ is the Nx1 gravity vector.

3. COLLISION-FREE TRAJECTORY PLANNING

3.1. Path Planning

The initial position of the end effector is always taken as the current position at time t_0 and the final position is the desired grasping point. The path between the two positions is made of one or multiple trajectory segments. The trajectory segment, in this paper, is planned in the joint space to attain the desired Cartesian pose and is defined using a polynomial function satisfying initial and final conditions on the joints displacement, velocity and acceleration:

$$\left.\begin{array}{rclcrcl} \theta_i(0) & = & \theta_{i_0} & and & \theta_i(t_f) & = & \theta_{i_f} \\ \dot{\theta}_i(0) & = & 0 & and & \dot{\theta}_i(t_f) & = & 0 \\ \ddot{\theta}_i(0) & = & 0 & and & \ddot{\theta}_i(t_f) & = & 0 \end{array}\right\}. \tag{2}$$

In order to satisfy the above conditions, a fifth order polynomial is needed for the i^{th} joint:

$$\theta_i = b_5 t^5 + b_4 t^4 + b_3 t^3 + b_2 t^2 + b_1 t + b_0, \tag{3}$$

where b_i, for $i = 0, 1, ..., 5$, are coefficients of the polynomial to be determined by solving the polynomial equation for all six conditions of Eq. (2) which gives the following coefficients:

$$b_0 = \theta_{i_0}, \tag{4}$$
$$b_1 = b_2 = 0, \tag{5}$$
$$b_3 = \frac{10}{t_f^3}\left(\theta_f - \theta_0\right), \tag{6}$$
$$b_4 = -\frac{15}{t_f^4}\left(\theta_f - \theta_0\right), \tag{7}$$
$$b_5 = \frac{6}{t_f^5}\left(\theta_f - \theta_0\right). \tag{8}$$

3.1.1. Time normalisation

Let T be the period of the execution of the trajectory from $t = 0$ to $t = t_f$ and the normalised time $\tau = \frac{t}{T}$ where $\tau \in [0, 1]$. Then θ_i becomes:

$$\theta_i(\tau) = \theta_{i_0} + \left(\theta_{i_f} - \theta_{i_0}\right) P(\tau), \tag{9}$$
$$P(\tau) = \left(6\tau^5 - 15\tau^4 + 10\tau^3\right), \tag{10}$$

where P is the fifth order polynomial function that describes the trajectory between the initial position of the end-effector to a desired final position while satisfying conditions of Eq. (2).

3.2. Obstacle Avoidance

In 2011, an algorithm for obstacle avoidance using four harmonic equations was developed, where the obstacle avoidance segment of the trajectory was divided into four segments equal in time, then the length of the whole path was minimised to find the shortest path [12]. However, the results showed that the obstacle avoidance segment was not as optimal as needed for a minimal power consumption. Hence, the method proposed in this paper is an improvement of the previous method, which results in a novel optimal obstacle avoidance algorithm. The new method is based on one function describing the companion curve of a cycloid.

The method stipulates that during the obstacle avoidance period T', the polynomial function described by Eq (9) becomes as follows:

$$\theta(t) = \theta_i + (\theta_f - \theta_i)(P(\tau) + P'(t)). \quad (11)$$

Here, $P'(t)$ is the companion curve function for obstacle avoidance and is equal to:

$$P'(t) = A\left(1 - \cos\frac{2\pi}{T'}t\right), \quad (12)$$

where A and T' are the amplitude and the period of the function respectively; these are parameters that depend on the size of the obstacle.

3.2.1. Algorithm Optimisation

In the presence of one or multiple obstacles, the optimisation process involves finding the shortest distance, defined by the user, between the manipulator and the obstacle. It is performed by adjusting the amplitude of the companion curve function in Eq. (12) according to the distance d_0. The optimisation is achieved by solving the following equations:

$$d_0 = min\left\{f_i(X,Y,Z) : X,Y,Z \in \mathbb{R}^3\right\}, \quad (13)$$

and:

$$f_i = \sqrt{(X - X_i')^2 + (Y - Y_i')^2 + (Z - Z_i')^2}, \quad (14)$$

where $[X,Y,Z]^T$ and $[X_i', Y_i', Z_i']^T$ are the position of the manipulator at each time step and the positions of n obstacles respectively with $i = 1, 2, ..., n$.
The pseudo-code of the algorithm in the presence of one or multiple obstacles is as follows, where P_0, P_f and P are the initial, final and current positions of the manipulator:

$t = t_0$
$loop:$ $from$ P_0 to P_f
$t = t + \Delta t$
$P = P + \Delta P$
if $obs_1 = true$
$Solve$ $Eq.$ (12), (13), (14) for f_1
.
.
.
$elseif$ $obs_i = true$
$Solve$ $Eq.$ (12), (13), (14) for f_i
$else$
$Continue$
if $P = P_f$
$Close$

4. CONTROL METHOD

The Computed Torque Control (CTC) is a non-linear controller based on the dynamic model of the system. The dynamic model is used to cancel the nonlinearities by dynamically decoupling the system through a nonlinear feed-forward compensation. Then, small proportional and derivative gains are used in the feedback loop for the generalised system. The corresponding control block diagram is depicted in Fig. 1 and the standard CTC control law is as in [14, 15]:

$$\begin{aligned} \boldsymbol{\tau}' &= \ddot{\boldsymbol{\theta}} + \mathbf{K_v}\dot{\tilde{\boldsymbol{\theta}}} + \mathbf{K_p}\tilde{\boldsymbol{\theta}}, \\ \boldsymbol{\tau} &= \mathbf{D}(\theta)\boldsymbol{\tau}' + \mathbf{C}(\theta,\dot{\theta})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\theta). \end{aligned} \quad (15)$$

The overall control law is:

$$\boldsymbol{\tau} = \mathbf{D}(\theta)(\ddot{\boldsymbol{\theta}} + \mathbf{K_v}\dot{\tilde{\boldsymbol{\theta}}} + \mathbf{K_p}\tilde{\boldsymbol{\theta}}) + \mathbf{C}(\theta,\dot{\theta})\dot{\boldsymbol{\theta}} + \boldsymbol{G}(\theta). \quad (16)$$

Table 1. DH Parameters of the 6 DOF Manipulator

θ_i	α_i (degree)	a_i (mm)	d_i (degree)	Range (degree)
θ_1	90	0	160	-150 to +150
θ_2	0	250	0	-60 to +60
θ_3	90	160	0	+60 to +155
θ_4	-90	0	160	-160 to +160
θ_5	90	0	0	-90 to 90
θ_6	0	0	72 + Gripper	-200 to +200

5. SIMULATIONS

In order to verify the behaviour of the mathematical model, for the manipulator, under the Computed Torque Control while following the designed collision-free trajectory, a 6 DoF manipulator was used. Simulations were conducted in the Matlab environment. The physical parameters of the robot are shown in Tab. 1.

As mentioned in Section 3.2, the parameters A and T' of Eq. (12) depend on the size of the obstacle. These parameters were chosen to be fixed for the first round of simulations.
Using equations (9), (11), (12), (13) and (16), the manipulator was able to accurately follow the designed trajectory and avoid the obstacle detected as depicted in Fig. 2. The applied torques at the joint level were in the limit of the robot actuators parameters and are shown in Fig. 7. It is noticed that the manipulator's trajectory (Fig. 2) during the obstacle avoidance is not optimal, which means that high torques are required to move the joints (Fig. 7), hence, a considerable amount of power is consumed. This is clearly shown in Fig. 3, 4 and 5 representing respectively the desired displacement, velocity and acceleration compared to the ones resulting from the obstacle avoidance algorithm. In order to overcome this problem, one can think of changing the values of A and T' of equation (12) parameters to optimise the companion curve based obstacle avoidance algorithm.

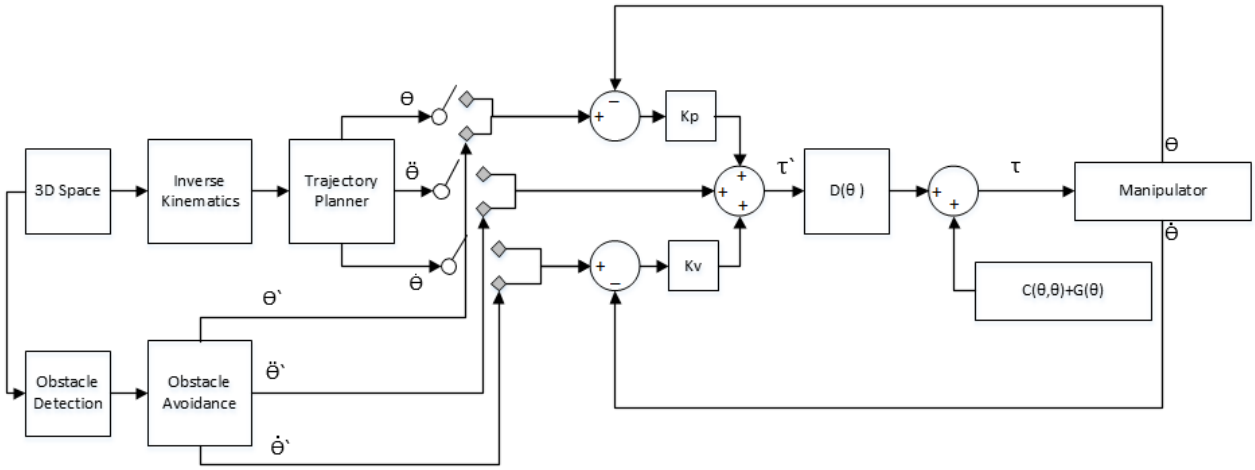

Figure 1. Block diagram describing the execution of the obstacle avoidance algorithm under CTC control

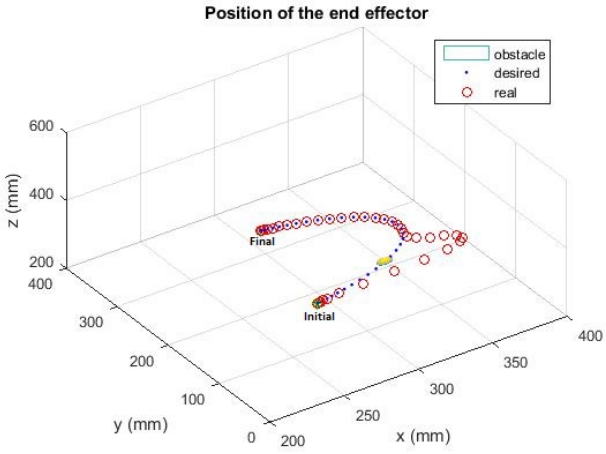

Figure 2. Non optimised Cartesian trajectory during obstacle avoidance

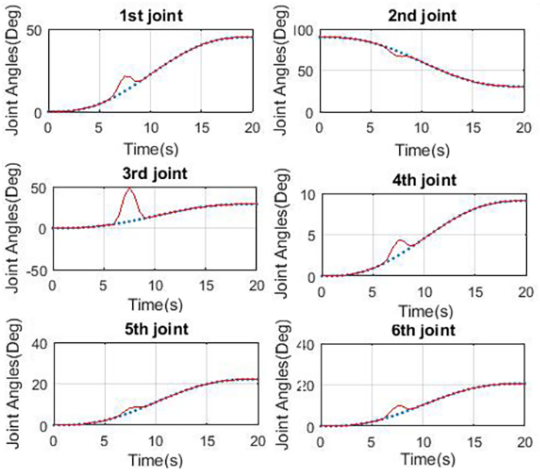

Figure 3. Joint trajectory during obstacle avoidance before optimisation

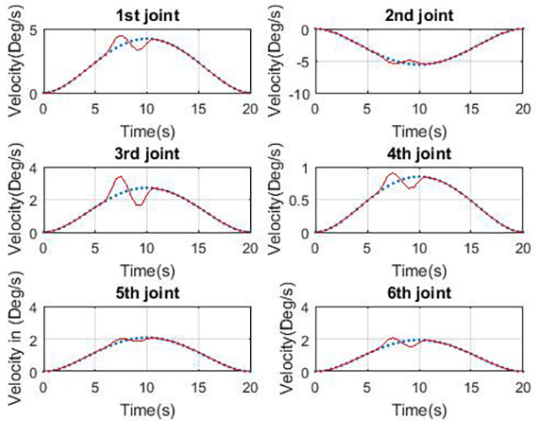

Figure 4. Joint velocity during obstacle avoidance before optimisation

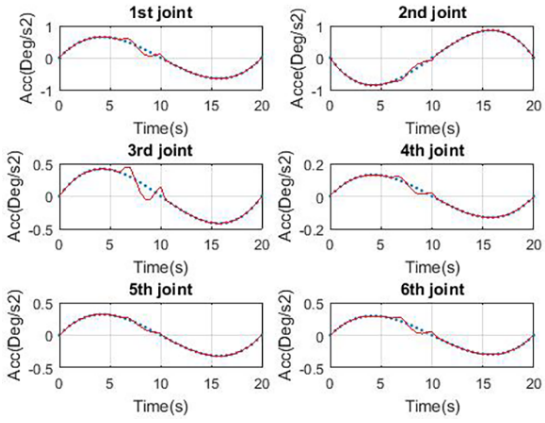

Figure 5. Joint acceleration during obstacle avoidance before optimisation

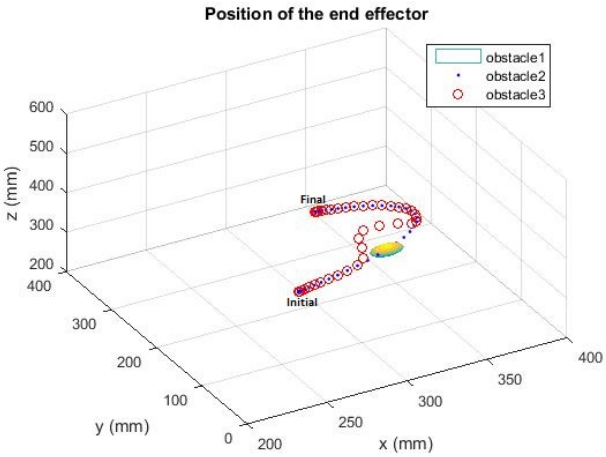

Figure 6. Optimised Cartesian trajectory during obstacle avoidance

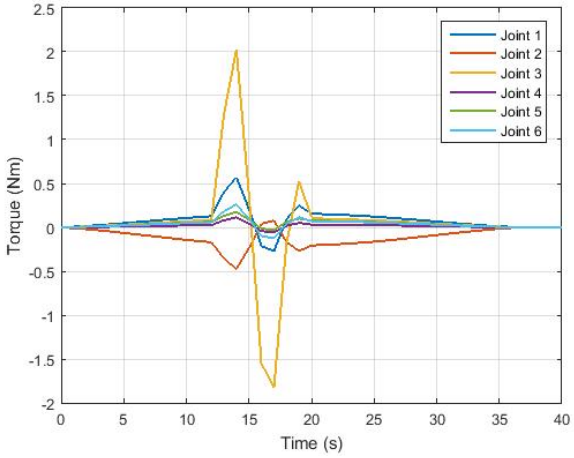

Figure 7. Joint torques before optimisation

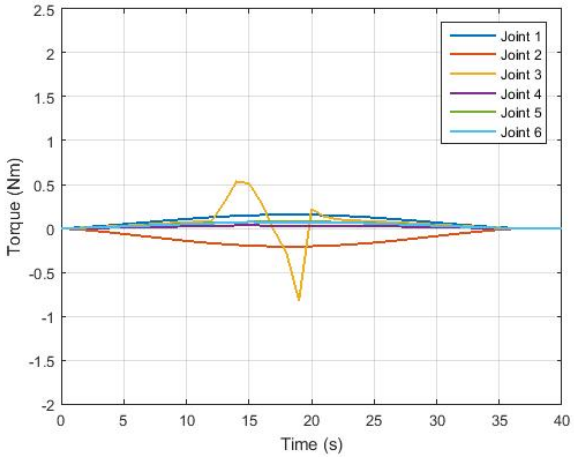

Figure 8. Joint torques after optimisation

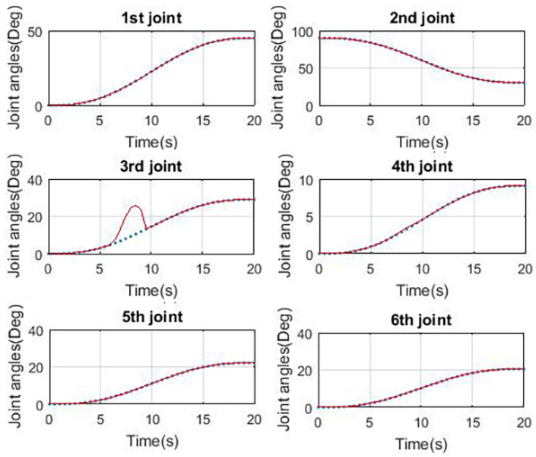

Figure 9. Optimised joint trajectory during obstacle avoidance

5.1. Optimisation

As mentioned above, the simulation results for the 6 DOF manipulator presented, show that a second stage of optimisation is required to reduce the power consumed at the joint level. For the obstacle avoidance segment, the objective function to minimise is the distance d between each step of the trajectory and the obstacle, as shown in equation (14). The aim is to always keep the end-effector at a desired minimum distance d_0 from the obstacle.
The optimal Cartesian obstacle avoidance trajectory is depicted in Fig. 6 along with the torques applied at each joint shown in Fig. 8. The displacement of the joints, after optimisation, is shown in Fig. 9. These results indicate that depending on the position of the obstacle and trajectory designed, the amplitude A of the cycloid companion function changes automatically to keep the manipulator at a constant distance d_0 from the obstacle.
An inspection of the power consumed during the execution of the obstacle avoidance algorithm shows the performance of the optimised algorithm. A comparison between the non-optimised and the optimised algorithm in terms of power consumption is shown in Fig. 10 and 11 respectively. This validates the need to optimise the trajectory followed by the manipulator in order to conserve on-board power.
As mentioned in Section 3.2, the amplitude and period of the obstacle avoidance function depends on the size of the obstacle. The results in Fig 12 show the trajectory followed by the manipulator in the presence of three heterogeneous obstacles, i.e. obstacles of different sizes and shapes. This proves the efficacy of th algorithm to avoid multiple obstacles.

From the results above, one can see that polynomials offer the flexibility to design trajectories given the initial and final positions, without preliminary knowledge of the surrounding environment. The obstacle avoidance algorithm presented in this paper is enabled only when an obstacle is detected along the path using vision sensors such

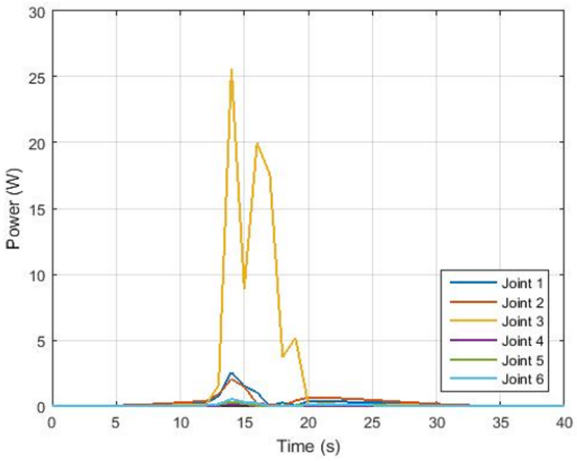

Figure 10. Power consumed during the obstacle avoidance before optimisation

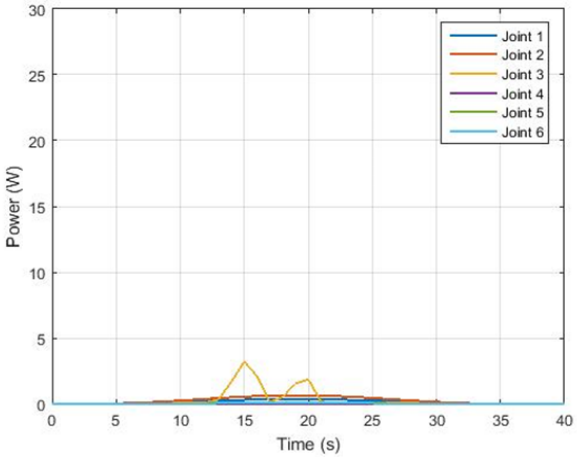

Figure 11. Power consumed during the obstacle avoidance after optimisation

as LiDARs and cameras. This real time process is crucial for autonomous missions. The optimised algorithm guarantees minimum power consumption but needs further development when dealing with moving targets and moving obstacles.

6. CONCLUSION

The series of results presented in this paper prove that the new algorithm for collision-free optimal trajectory offers smooth displacement of joints, velocity and acceleration in the presence of one or more obstacles. Furthermore, the algorithm enables the designer to choose how far the manipulator has to deviate from the original trajectory to efficiently perform the obstacle avoidance process. This enhanced performance is achievable while continuously tracking the trajectory with minimal on-board power. In addition to saving power, when dealing with space moving base manipulators, if the joints are highly actuated the reaction forces on the base are larger and the algorithm developed in this paper clearly shortens the trajectories traversed by each joint. It can hence be applied to space based stationary or moving robotic manipulators. Future research will focus on validating the path planning and control algorithms presented in this paper for space free-flying and free-floating robots used for debris removal and on-orbit servicing missions. When designing trajectories for space moving base manipulators, the algorithm is adapted to cope with the motion of both the target spacecraft and the eventual obstacles. For this purpose, a nonlinear controller for the nonlinear mathematical model representing the dynamics of the system will be developed.

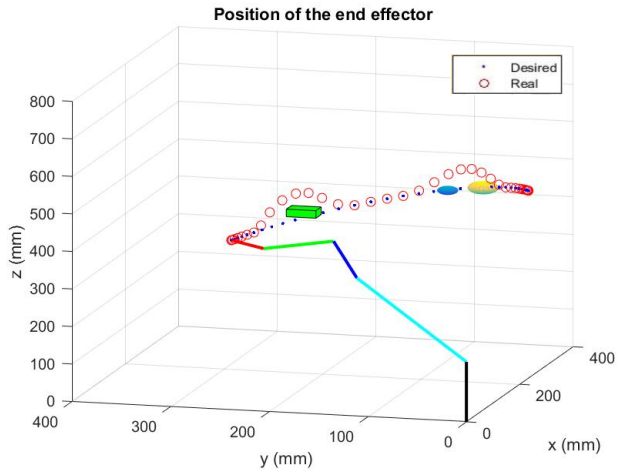

Figure 12. Cartesian trajectory for multiple heterogeneous obstacle avoidance

ACKNOWLEDGEMENT

The authors wish to acknowledge the Algerian Space Agency, UK Space Agency and Surrey Satellite Technology Ltd for funding this study.

REFERENCES

1. Oussama Khatib. (1986). Real-time obstacle avoid--ance for manipulatros and mobile robots. *The international journal of robotics research 5(1)*, 90 -98.
2. S Liu, Q Zhang, and D Zhou. (2014). Obstacle avoidance path planning of space manipulator based on improved artificial potential field method. *Journal of The Institution of Engineers (India): Series C 95(1)*, 31–39.
3. Yoji Umetani and Kazuya Yoshida. (1989). Resolved motion rate control of space manipulators with generalized jacobian matrix. *IEEE Transactions on robotics and automation 5(3)*, 303–314.

4. Dongsheng Zhou, Lan Wang, and Qiang Zhang. (2016). Obstacle avoidance planning of space manipulator end-effector based on improved ant colony algorithm. *SpringerPlus* 5(1), 509.

5. Steven Dubowsky and Miguel A Torres. (1991). Path planning for space manipulators to minimize spacecraft attitude disturbances. In *Robotics and Automation Proceedings., 1991 IEEE Interational Conference*. pp 2522–2528.

6. Cyrille Froissart and Pierre Mechler. (1993). Online polynomila path planning in Cartesian space for robot manipulators. *Robotica 11(03),* 245–251.

7. Evangelos Papadopoulos, Ioannis Tortopidis, and Kostas Nanos. (2005). Smooth planning for free-floating space robots using polynomials. *In Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. pp 4272–4277

8. Long Zhang, Qingxuan Jia, Gang Chen, and Hanxu Sun. (2015). Pre-impact trajectory planning for minimizing base attitude disturbance in space manipulator systems for a capture task. *Chinese Journal of Aeronautics 28(4),* 1199–1208.

9. Panfeng Huang, Jianping Yuan, Yangsheng Xu, and Rong Liu. (2006). Approach trajectory planning of space robot for impact minimization. *Information acquisition, IEEE International Conference*. pp 382–387.

10. WF Xu, Bin Liang, Cheng Li, YS Xu, and WY Qiang, Xu,. (2007). Path planning of free-floating robot in Cartesian space using direct kinematics. *International Journal of Advanced Robotic Systems 4(1),* 17–26.

11. Yasuhiro Wada, Yuichi Kaneko, Eri Nakano, Rieko Osu, and Mitsuo Kawato, Wada. (2001). Quantitative examinations for multi joint arm trajectory planning using a robust calculation algorithm of the commanded torque change trajectory. *Neural Networks 14(4),* 381–393.

12. Soheil S Parsa, Juan A Carretero, and Roger Boudreau. (2011). Length optimized smooth obstacle avoidance for robotic manipulators. *Transactions of the Canadian Society for Mechanical Engineering 35(4),* 505–514.

13. Mark W Spong and Mathukumalli Vidyasagar. (2008). *Robot dynamics and control.* John Wiley & Sons.

14. Alex Ellery. (2000). *An introduction to space* robotics, Springer-Verlag New York, Inc.

15. John J Craig. (2005). *Introduction to robotics: mechanics and control*, volume 3. Pearson Prentice Hall Upper Saddle River.