OPTIMAL MOTION PLANNING OF A SPACE ROBOT WITH BASE DISTURBANCE MINIMIZATION

Eric Guiffo Kaigom, Thomas Josef Jung, and Jürgen Roßmann

Institute for Man-Machine Interaction, Ahornstraße 55, 52074 Aachen, Germany,
{kaigom,jung,rossmann}@mmi.rwth-aachen.de

ABSTRACT

The base attitude of a free floating space robot may change while performing a motion with its manipulator. This dynamic coupling complicates the motion planning of the space robot and must be taken into account in order to reach a target point with a desired end-effector pose. However, the free floating mode is useful, because of its energy efficiency, and with proper motion planning, its adequacy for smooth motions with minimized base disturbance. In this paper we propose a motion planning algorithm to steer the end-effector of a free floating space robot to a target pose with minimized base disturbance. The joint position and velocity limits are respected. The approach is based on the Reaction Null-Space and the Constrained Particle Swarm Optimization. Simulation results show the effectiveness of the planning method applied to a seven Degrees of Freedom manipulator.

Key words: Free Floating Space Robot; Motion Planning; Reactionless Maneuver; Constrained Particle Swarm Optimization.

1. INTRODUCTION

Space robots are expected to perform various tasks in future on-orbit servicing activities. Some of these may include the inspection of malfunctioning satellite devices, the repair and the refueling of satellites to extend their mission life as well as their capture and de-orbiting at their end-of-life.

The maneuvers and dynamic interactions between the space robot (chaser) and the satellite (target) involved in a capture task depend on multiple factors such as their proximity and their behaviour before and after docking attemps. For this reason, the capture may be divided into four phases [1]-[2]:

- *Free flying approach phase*: The chaser moves from its home orbit into the target orbit. After a status inspection, it approaches the target further with help of proximity sensors and small jet firings.
- *Free floating approach phase*: The thrusters are turned off. The base is permitted to translate and rotate in response to the manipulator motion. The chaser moves its end-effector toward the grasping point on the target in a smooth and accurate manner.
- *Impact phase*: The contact between the manipulator and the target is established. This results in an impact, which may cause a rebound.
- *Post impact phase*: The impact phase may be followed by disturbances such as attitude drift or components deformation.

The dynamic coupling between the base and the manipulator during the free floating approach phase has been well studied [3]. Without proper motion planning, it can cause the end-effector to miss the grasping point. In the absence of external forces and torques acting on the chaser, its linear and angular momentums are conserved. While the linear momentum conservation equations can be integrated (holonomic constraints), the angular momentum conservation equations cannot (nonholonomic constraints) (see [3] p. 174). Further, it has been

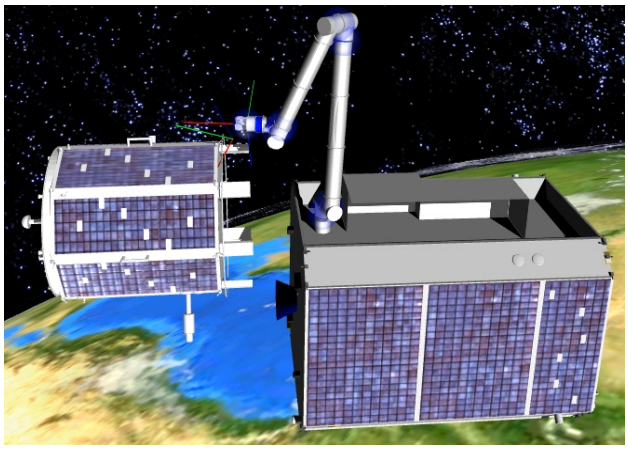

Figure 1. The simulation model. The free floating chaser (rhs) with minimized base disturbance moving its manipulator toward the target (lhs).

shown that in the case of zero momentum, the trajectory of the base attitude is dependent on the joint trajectory [4]. This means that for the same initial and final end-effector pose, the latter may be reached with different base attitudes depending on the path taken in the joint space. This result has been used by many authors to plan the motion of free floating space robots by using a polynomial joint trajectory to reach an end-effector pose with a desired base attitude (see [5] and references therein). With this approach, however, the base is subject to strong attitude disturbances during the entire motion. This could lead to a collision with the target or to communication pertubations while teleoperating [6].

To avoid collision with the target or for communication purposes, the attitude of the base has to be kept nearly constant in the free floating approach. In [7] the *Reaction Null-Space* (RNS) is used to characterize the joint velocity which locally minimizes the base reaction. Although this velocity is well defined, only [8], as far as we know, has addressed the generation of corresponding trajectories. The authors partition the joints into sets of redundancy one and then take advantage of the integrability of a one-dimensional distribution included into the distribution generated by the RNS basis to plan reactionless motions.

In this work we use polynomial time varying input factors to formulate the feasible (reactionless) joint velocity as linear combination of the basis vectors of the RNS. We then optimize the input parameters to steer the end-effector to a desired pose. The velocity profile parameterization allows the manipulator to be at rest at the beginning and at the end of the motion. The joint velocity limits are satisfied through a scaling factor. The Constrained Particle Swarm Optimization (C-PSO) with joint position limits as constraints is used to solve the motion planning problem.

The paper is organized as follows: Section 2 gives a brief introduction of the concept of reactionless motions from [7]. Feasible joint trajectories and their parameterization are formulated in section 3 and 4, respectively. The C-PSO algorithm is shortly presented in section 4 and, in addition, the problem of stagnation is addressed. The simulation results are shown in section 5. Finally, Section 6 is for conclusion.

2. REACTIONLESS MOTIONS

The angular momentum L of the chaser with a N Degrees of Freedom manipulator can be expressed as [7]:

$$L = \tilde{H}_b w_0 + \tilde{H}_{bm}\dot{q} \tag{1}$$

The terms $\tilde{H}_b w_0$ and $\tilde{H}_{bm}\dot{q}$ represent the angular momentum of the base and the *coupling* angular momentum between the base and the manipulator, respectively. The vector $w_0 \in \mathbb{R}^3$ is the base angular velocity and the vector $q \in \mathbb{R}^N$ contains the joint configuration values.

Assuming that the chaser is at rest at the beginning of the manipulator motion, it follows $L = 0 \in \mathbb{R}^3$. $\tilde{H}_b$ is a always invertible. Hence, if $\tilde{H}_{bm}\dot{q} = 0 \in \mathbb{R}^3$ during the manipulator motion, the attitude of the base will remain constant ($w_0 = 0$). It follows that for

$$\tilde{H}_{bm}(t)\dot{q}(t) = 0, \quad 0 \leq t \leq t_f \tag{2}$$

the attitude of the chaser is kept constant during the manipulator motion of length t_f.

The matrix $\tilde{H}_{bm}$ depends on the dynamic properties (mass, inertia tensor) of the chaser and its configuration q. In the absence of external forces, as in our case, it does not depend on the base attitude [8].

3. FEASIBLE TRAJECTORIES

The kind of restriction on the velocity profile as in (2) is called *pfaffian constraints* [11]. It is transformed in this section into the general expression of joint velocity profiles which do not disturb the base.

It is obvious that each solution $\dot{q}(t)$ of (2) must lie in the current right null space $\mathcal{N}_{\tilde{H}_{bm}}(t)$ of $\tilde{H}_{bm}(t)$. The singular value decomposition of $\tilde{H}_{bm}(t)$ provides an explicit way to extract a basis of $\mathcal{N}_{\tilde{H}_{bm}}(t)$ in the sense that the singular vectors corresponding to the vanishing singular values span $\mathcal{N}_{\tilde{H}_{bm}}(t)$. Assuming that $\tilde{H}_{bm}(t)$ does not loose rank, a basis of $\mathcal{N}_{\tilde{H}_{bm}}(t)$ is of dimension $k = N - 3$. It is worth noting that during the motion of the manipulator, the basis vectors of $\mathcal{N}_{\tilde{H}_{bm}}(t)$ are always dependent on the current dynamic properties of the chaser because $\tilde{H}_{bm}(t)$ is too.

In the rest of this paper, we use the set $\mathcal{B}^k_{nsb}(t) = \{\dot{q}^{nsb}_1(t), \cdots, \dot{q}^{nsb}_k(t)\}$ to refer to a basis of $\mathcal{N}_{\tilde{H}_{bm}(t)}$having k basis vectors $\dot{q}^{nsb}_i(t)$ with $i \in \{1, 2, \cdots k\}$. In [7], $\mathcal{B}^k_{nsb}(t)$ is called the RNS. Every linear combination

$$\dot{q}(t) = \alpha_1(t)\dot{q}^{nsb}_1(t) + \cdots + \alpha_k(t)\dot{q}^{nsb}_k(t) \tag{3}$$

of these basis vectors, where $\alpha_i(t) \in \mathbb{R}$, is also a solution of (2).

The right hand side of (3) represents the joint velocity profiles enabling the manipulator to move without disturbing the base attitude.

4. TRAJECTORIES PARAMETERIZATION

The joints velocity are parameterized as follows:

$$\dot{q}(t) = \frac{1}{g}\left(\frac{\beta_1(t)}{||\dot{q}^{nsb}_1(t)||}\dot{q}^{nsb}_1(t) + \cdots + \frac{\beta_k(t)}{||\dot{q}^{nsb}_k(t)||}\dot{q}^{nsb}_k(t)\right) \tag{4}$$

with the real number $g \neq 0$ and

$$\beta_i(t) = a_i t^{n+2} + b_i t^{n+1} + c_i t^n, \quad n \in \{1, 2, 3, \cdots\}, \quad a_i \neq 0. \tag{5}$$

According to (3)

$$\alpha_i(t) = \frac{\beta_i(t)}{g||\dot{q}_i^{nsb}(t)||}. \tag{6}$$

Furthermore, the following constraints have to be respected:

$$\dot{q}(t=0) = \dot{q}(t=t_f) = 0, \tag{7}$$
$$q_{min} \leq q(t) \leq q_{max}, \tag{8}$$
$$|\dot{q}(t)| \leq \dot{q}_{max}. \tag{9}$$

The first constraint (7) is satisfied if $\beta_i(t=0) = \beta_i(t=t_f) = 0$, that is

$$c_i = -(a_i t_f^2 + b_i t_f). \tag{10}$$

β_i is continuous on the closed bounded interval $D = [0, t_f]$. Therefore, β_i attains its maximum and minimum on D. Next we are looking for $\hat{t}$ which maximizes $|\beta_i(t)|$ in D. After taking the time derivative $\dot{\beta}_i$ of β_i with

$$\dot{\beta}_i(t) = t^{n-1}[(n+2)a_i t^2 + (n+1)b_i t + n c_i], \tag{11}$$

it is simple to show that for a_i, b_i and c_i such that the discriminant Δ of the quadratic polynomial inside the brackets in (11) becomes positiv, that is

$$\Delta = ((n+1)b_i)^2 - 4n(n+2)a_i c_i \geq 0, \tag{12}$$

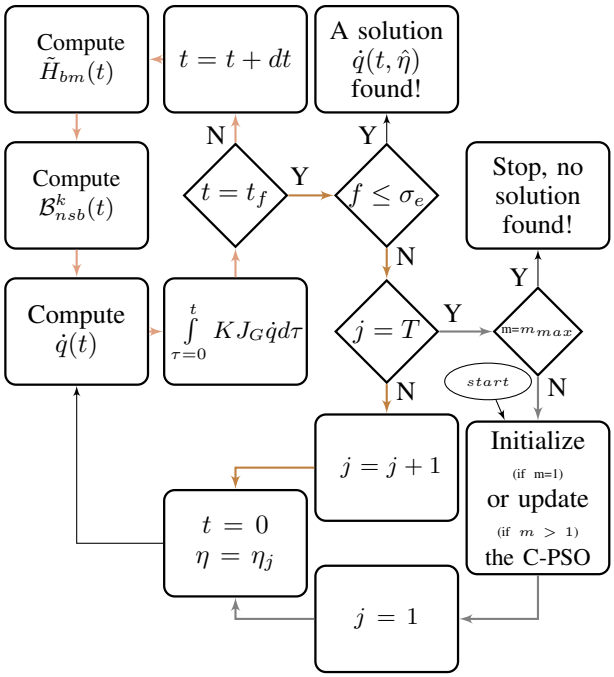

Figure 2. The flowchart of the proposed motion planning algorithm.

$\dot{\beta}_i$ has two ($\Delta = 0$) or three ($\Delta > 0$) roots

$$t_{i,0} = 0 \text{ and } t_{i,\pm} = \frac{-(n+1)b_i \pm \sqrt{\Delta}}{2(n+2)a_i}. \tag{13}$$

One of the $t_{i,\pm}$ corresponding to the value $\beta_{i,max} = \max\{|\beta_i(t = t_{i,+})|, |\beta_i(t = t_{i,-})|\}$ and lying in D is $\hat{t}$. So, the constraint (9) holds if we set

$$g \geq \frac{1}{\dot{q}_{max}} \sum_{i=1}^{k} \beta_{i,max}. \tag{14}$$

The motion planning is thus reduced to the constrained optimization problem of finding an optimal vector

$$\hat{\eta} = [a_1, b_1, \cdots, a_k, b_k]^T \tag{15}$$

such that

$$f(\hat{\eta}, t = t_f) = ||X_{des} - X(t = t_f)|| \leq \sigma_e, \tag{16}$$
$$q_{min} \leq q(t) \leq q_{max}, \tag{17}$$
$$\Delta \geq 0, \tag{18}$$

holds, with

$$X(t = t_f) = \begin{bmatrix} x_E \\ \Theta_E \end{bmatrix} = \int_{t=0}^{t_f} \dot{X} dt = \int_{t=0}^{t_f} KJ_G\dot{q}(\hat{\eta}, t)dt, \tag{19}$$

where σ_e, x_E and Θ_E are the accuracy threshold value, the position and orientation components of the end-effector pose X, respectively. The $(6 \times N)$-matrix J_G is referred to as the *generalized Jacobian Matrix* [7]. $K = \begin{bmatrix} I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & F^{-1}_{3\times3} \end{bmatrix}$ with $\dot{\Theta}_E = F^{-1}_{3\times3} w_E$ (see [4]). w_E is the angular velocity of the end-effector.

5. THE CONSTRAINED PARTICLE SWARM OPTIMIZATION

The Particle Swarm Optimization (PSO) is a stochastic search and optimization technique that has been proven to perform well in several optimization problems [12]. It is inspired by the social behavior and movement dynamic of bird swarms. Here, the particle swarm is a set of T optimization vectors η_j with $j \in \{1, \cdots, T\}$. The PSO consists of maximal m_{max} optimization steps of the position of each η_j in the space of possible (characterized by (18)) components of η_j. At least one of the η_j is expected to converge toward a position where (16) and (17) hold.

The next position of η_j with the current position $\eta_j(m)$ at the m-th optimization step, $m \in \{1, \cdots, m_{max}\}$, is determined by adding a displacement (called velocity) $\nu_j(m+1)$:

$$\eta_j(m+1) = \eta_i(m) + \nu_j(m+1) \tag{20}$$

with

$$\nu_j(m+1) = \omega\nu_j(m) + c_1 r_1(\eta_{j,pbest}(m) - \eta_j(m)) + c_2 r_2(\eta_{gbest}(m) - \eta_j(m)). \tag{21}$$

In (21) $\eta_{j,pbest}(m)$ is the current personal best position of η_j and $\eta_{gbest}(m)$ the global best position of the entire swarm. The weighting function ω acting on the last step velocity can be set as

$$\omega(m) = \omega_{max} - m\frac{\omega_{max} - \omega_{min}}{m_{max}} \tag{22}$$

to first allow a global search and with increasing optimization steps, a local search can be performed. r_1 and r_2 are random numbers uniformly distributed in $[0, 1]$. The real numbers c_1 (cognitive parameter), c_2 (social parameter) and the bounds ω_{min} and ω_{max} of ω can be chosen in such a way that each η_j converges to an asymptotically stable equilibrium position [10]. Note that this position is not necessarily a solution to the given optimization problem. It is only the exchange of experiences between the η_i which increases the convergence probability towards an $\hat{\eta}$ [12].

5.1. Constraints handling

The PSO does not handle constraints in its native form. To handle constraints in this work, only feasible (The constraint (17) must be satisfied.) $\eta_{gbest}(m)$ are taken into account, while preserving the exploration and exploitation behavior of the swarm. To do so, $\eta_{gbest}(m)$ is chosen as the best feasible $\eta_{j,pbest}(m)$, and the $\eta_{j,pbest}(m)$ are updated using the concept of *"constraint domination"* presented in [9]. Based on the amount of constraint violation, this technique enables the exploitation of "better unfeasible" optimization vectors to maintain the aforementioned behavior. The amount of constraint violation function $\Phi(\eta_j)$ is adapted in this work as follows:

$$\Phi(\eta_j) = \sum_{r=1}^{N} (\max(0, q^r_{\eta_j,max} - q^r_{max}) + \max(0, q^r_{min} - q^r_{\eta_j,min})) \tag{23}$$

In (23), q^r_{max} and q^r_{min} are the maximal and minimal allowed position of the r-th joint, $r \in \{1, \cdots, N\}$, while $q^r_{\eta_j,min}$ and $q^r_{\eta_j,max}$ are the encontered minimal and maximal position of the r-th joint, after following the trajectory corresponding to η_j, respectively.

5.2. Stagnation handling

The swarm stagnates if no improvements of $\eta_{j,pbest}$ and η_{gbest} are occurring. To solve this problem we integrate an alarm function which notifies the C-PSO algorithm when after a fixed number of optimization steps, the minimization change rate of the cost function remains lower than a given threshold near zero. In such a case the swarm is randomly split in two subswarms $S_{s,s\in\{1,2\}}$ of equal size, which are initialized as follows:

- $\eta_j(m+1) = \eta_{gbest}(m) + \varepsilon_1 rand(0,1)$, if $\eta_j \in S_1$ to exploit the $\eta_{gbest}(m)$ information through performing a local search around $\eta_{gbest}(m)$.
- $\omega(m+1) = \omega(m) + |\varepsilon_2| rand(0,1)$ if $\eta_j \in S_2$ to escape from the stagnation through dispersing the subswarm.

ε_1 and ε_2 are two weighting scalar and $rand(0, 1)$ is a random number between 0 and 1.

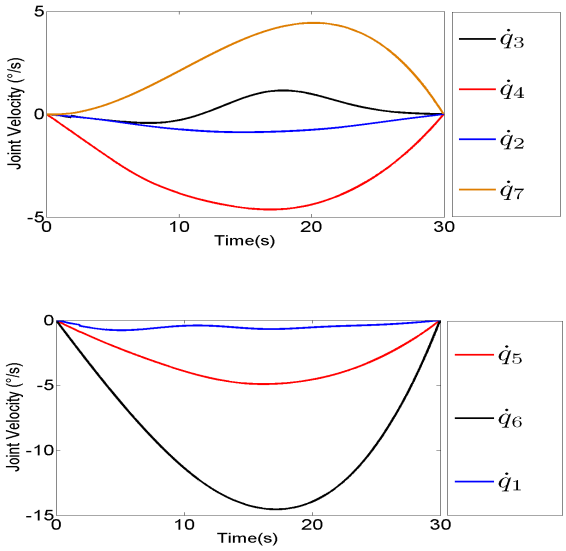

Figure 3. The joint velocity profiles.

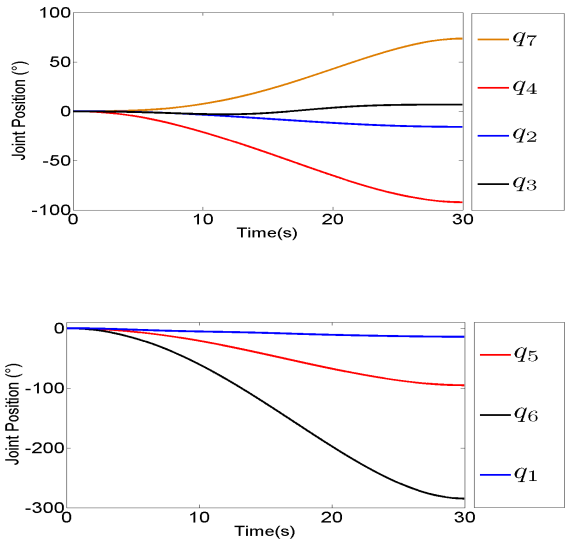

Figure 4. The joint position profiles.

5.3. The motion planning process

The chart of the motion planning algorithm is depicted in Fig. 2. Each optimization round begins with η_1 and ends with η_T. At the first optimization round, the whole swarm is uniformly randomly initialized while respecting (18). The joint trajectory corresponding to η_j is followed until $t = t_f$, where the discrepancy between the reached and the desired end-effector pose is compared to σ_e (see (16)). To compute this trajectory, the null space problem (2) is solved at each discrete simulation step $t = z \cdot dt$, with $z \in \{0, 1, \cdots, \text{int}(\frac{tf}{dt})\}$.

A solution for the motion planning problem is found if for an $\eta_j = \hat{\eta}$ the desired end-effector pose accuracy is satisfied. Otherwise, if $j = T$, the η_j are updated according to (20) and (21) for the next optimization round. There is no solution if $f > \sigma_e$, $j = T$ and $m = m_{max}$.

6. SIMULATION RESULTS

We have done numerical simulations to investigate the effectiveness of the proposed motion planning approach. As a physical model we used a seven Degree of Freedom chaser with parameters listed in Tab. 1. The manipulator is desired to move from the start end-effector pose $X_{init} = [-0.5186, 0.2536, -0.36291, 180.0°, 0.01°, 179.80°]^T$ to the final end-effector pose $X_{final} = [-0.6034, 0.8095, -0.27442, -94.78°, 5.10°, 100.23°]^T$. $n = 1$ for each joint. The joint limits are set to $|q_i| \leq 100°$ for $i \in \{2, 3, 4, 7\}$ and $-280° \leq q_i \leq 10°$ for $i \in \{1, 5, 6\}$. $|\dot{q}_i| \leq 15°/s$ is required for each joint. The PSO parameters are:

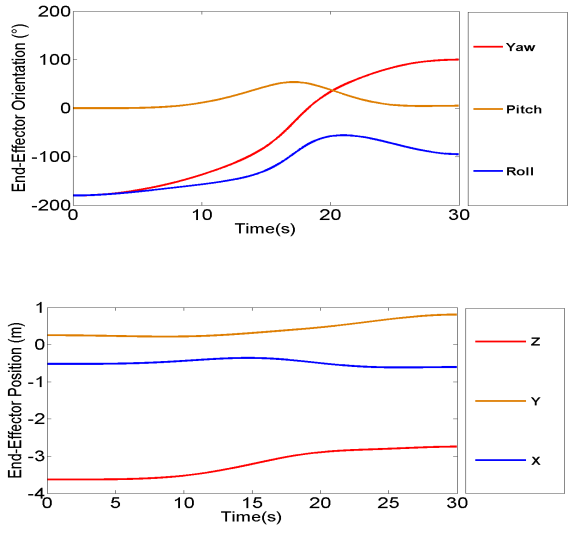

Figure 5. The end-effector trajectory.

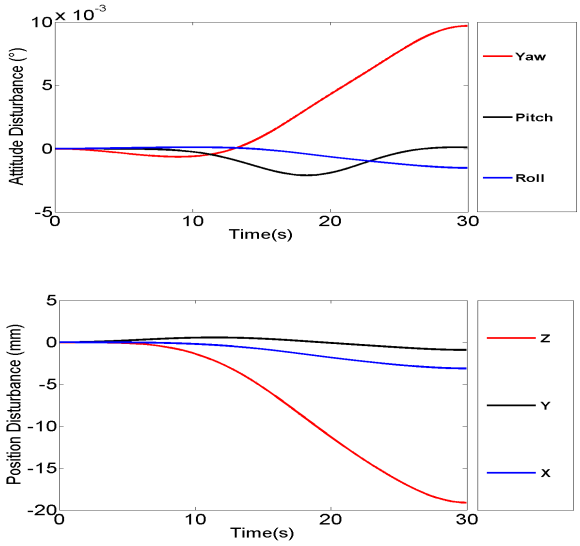

Figure 6. The base disturbance.

$T = 40$, $c_1 = c_2 = 1.49618$, $w_{max} = 0.9$, $w_{min} = 0.4$ and $\sigma_e = 0.05$.

The synthesized profiles of the joint position, the joint velocity and the TCP trajectory for the optimal vector $\hat{\eta} = [3.14771, 0.809316, 1.12249, 0.754006, 4.7352, 0.406031, -4.1572, 4.29895]^T$ are depicted in Fig. 3, Fig. 4 and Fig. 5, respectively. The trajectories are smooth and the manipulator starts its motion and reaches the target pose with zero velocity. The constraints are satisfied. Fig. 6 shows that the base disturbance during the motion of length $t_f = 30s$ remains lower than $0.01°$.

7. CONCLUSION

We presented and demonstrated by simulation an algorithm to plan the motion of a free floating space robot manipulator which minimizes the base disturbance and steers the end-effector to a desired pose. Polynomial time varying inputs were used to express the feasible velocity profile as linear combination of the RNS basis vectors. The trajectories were parameterized in such a way that the space robot is at rest at the beginning as well as at the motion end. The motion planning was formulated as an optimization problem with joint limits as constraints. We used the Constrained Particle Swarm Optimization algorithm to solve the optimization problem. A warning function and a split-and-disperse scheme to fix possible stagnation problems were introduced. The presented results illustrate that it is effectively possible to reactionless steer a space robot manipulator with polynomial time varying inputs.

Table 1. The model parameters. l (Length,m), m (Mass,kg), I (Inertia tensor,$kg.m^2$)

Links	l	m	I_{xx}	I_{yy}	I_{zz}
$Base$		1022	612.69	328.61	657.44
L_1	0.113	10	0.038	0.047	0.025
L_2	1.325	25	0.056	5.85	5.84
L_3	0.08	5	0.009	0.017	0.017
L_4	0.926	15	0.029	1.74	1.74
L_5	0.08	3	0.008	0.005	0.008
L_6	0.25	10	0.026	0.084	0.076
L_7	0.04	3	0.0036	0.0036	0.0034

REFERENCES

[1] W. de Peuter, G. Visentin, W. Fehse, A. Elfving, D.L.. Brown and E.W. Ashford, "Satellite Servicing in GEO by Robotic Service Vehicle," *ESA Bulletin* , No. 78, pp. 33-39, May 1984.

[2] C. Xavier, K. S.-Wook, I. Michel, M. Arun, J. Gilbert, "Post-impact dynamics of two multi-body systems attempting dockingberthing," *Acta Astronautica*,Vol. 40, Issue 11, pp. 759-769, June 1997.

[3] E. Papadopoulos, "On the Dynamics and Control of Space Manipulators," *Ph.D. Thesis*, Dep. of Mech. Engin., MIT, October 1990.

[4] Y. Nakamura, R. Mukherjee, "Nonlinear control for the nonholonomic motion of space robot systems," *Advanced Robot Control,Lecture Notes in Control and Information Sciences*, Vol. 1621991, pp. 83-105, 1991.

[5] I. Tortopidis, E. Papadopoulos, "On point-to-point motion planning for underactuated space manipulator systems," *Robotics and Autonomous Systems*, pp. 122131, 2007.

[6] Y. Sato, M. Hirata, F. Nagashima, T. Maruyama, T. Uchiyama, "Reducing attitude disturbances while teleoperating a space manipulator," *IEEE International Conference on Robotics and Automation*, Vol. 3, pp. 516-523, 1993.

[7] K. Yoshida, "Space Robot Dynamics and Control: To Orbit, From Orbit, and Future," *Robotics Research, The Ninth International Symposium*, pp. 449-456, 1999.

[8] D. Dimitrov, K. Yoshida, "Utilization of Holonomic Distribution Control for Reactionless Path Planning," *International Conference on Intelligent Robots and Systems, 2006 IEEE/RSJ* , pp. 3387-3392 , Oct. 2006.

[9] L.D. Li, Y. Xinghuo; L. Xiaodong; W. Guo, "A Modified PSO Algorithm for Constrained Multi-objective Optimization," *IEEE Third International Conference on Network and System Security*, pp. 462-467, 2009.

[10] V. Kadirkamanathan, K. Selvarajah and and P. J. Fleming "Stability Analysis of the Particle Dynamics in Particle Swarm Optimizer," *IEEE Transactions of Evolutionary Computation*, Vol. 10, No. 3, pp. 245-255, June 2006.

[11] S. M. Lavalle, "Planning Algorithm," *Cambridge University Press*, 2006.

[12] M. Clerc, "Particle Swarm Optimization," *Wiley-ISTE*, February 2006.