

Research paper

Collision-free optimal trajectory generation for a space robot using genetic algorithm

Asma Seddaoui^{a,1,*}, Chakravarthini M. Saaj^{b,2}^a Surrey Space Centre, University of Surrey, GU2 7XH, UK^b University of Lincoln, Lincoln, LN6 7TS, UK

ARTICLE INFO

Keywords:

Trajectory planning

Optimisation

Genetic algorithm

Controlled-floating space robot

ABSTRACT

Future on-orbit servicing and assembly missions will require space robots capable of manoeuvring safely around their target. Several challenges arise when modelling, controlling and planning the motion of such systems, therefore, new methodologies are required. A safe approach towards the grasping point implies that the space robot must be able to use the additional degrees of freedom offered by the spacecraft base to aid the arm attain the target and avoid collisions and singularities. The controlled-floating space robot possesses this particularity of motion and will be utilised in this paper to design an optimal path generator. The path generator, based on a Genetic Algorithm, takes advantage of the dynamic coupling effect and the controlled motion of the spacecraft base to safely attain the target. It aims to minimise several objectives whilst satisfying multiple constraints. The key feature of this new path generator is that it requires only the Cartesian position of the point to grasp as an input, without prior knowledge a desired path. The results presented originate from the trajectory tracking using a nonlinear adaptive H_∞ controller for the motion of the arm and its base simultaneously.

1. Introduction

Advancements in the field of Robotics and Autonomous systems are paving the way for realising complex orbital space missions. Space robotic arms on-board the International Space Station (ISS) have been extensively operated by astronauts to deploy satellites and for assistance during ISS construction as well as during Extra Vehicular Activities (EVA) [1,2]. The ISS is a large attitude stabilised spacecraft and hence, the robotic manipulators mounted on-board the ISS are similar to terrestrial fixed base manipulators. However, these manipulators are tele-operated by astronauts and have a workspace constrained within the ISS environment, which makes them unsuitable for other missions in different orbits. For instance, the refuelling and maintenance of satellites [3] or the assembly of large telescopes [4] will require smaller space robots with the capability to perform several tasks whilst operating in an extreme environment. The next generation of space robots will be consisting of one or more dexterous arms mounted on smaller-sized platforms. However, the modelling, design and operation of these space robots is highly intricate. This is because space robots suffer from the large dynamic coupling effect, that is primarily because of the smaller size of their spacecraft base when compared to ISS.

Planning the motion of space robots is paramount regardless of the size and mass of their spacecraft base. Path planning for ISS manipulators can easily be achieved using classical methods due to their similarity with fixed-base manipulators. However, planning the motion of small-sized space robots is more complex as the end-effector's inertial pose is highly affected by the dynamic coupling effect i.e. it depends on the instantaneous pose of the spacecraft base as well as the additional spacecraft pose variations originating from the dynamic coupling. This latter category of space robots can operate in the free-floating mode, free-flying mode [5] or controlled-floating mode [6]. Free-floating space robots are under-actuated systems with an uncontrolled base in response to the motion of the arm, which produces undesired changes in the position and attitude of the spacecraft base due to the dynamic coupling effect. On the other hand, free-flying space robots have an actively controlled base to compensate for the dynamic coupling effect and retain a fixed pose. Free-flyers can also have a final desired pose, for their base, that is different from the initial one. Notably, the Controlled Floating Space Robot (CFSR) takes advantage of the benefits of the free-flying and free-floating modes [6]. In the CFSR case, the extra Degrees of Freedom (DoF) offered by the spacecraft base are utilised, in addition to the DoF of the arm, to

* Corresponding author.

E-mail address: a.seddaoui@gmail.com (A. Seddaoui).¹ PhD researcher in space robotics.² Prof. C. M. Saaj was previously with the University of Surrey, UK and she is currently with the University of Lincoln, UK.

reach the target in an optimal way, whilst avoiding collisions and singularities.

Researchers in [7–12] have been investigating ways to reduce the effect of dynamic coupling, but their focus was only related to free-floating space robots. For instance, the Enhanced Disturbance Map (EDM) [7], Reaction Null Space (RNS) [8,9], optimal planning using redundant manipulators [10,11] and the utilisation of several arms for balance [12]. Free-flying and controlled-floating space robots on the other hand, are known to have a controlled spacecraft base to suppress the dynamic coupling effect. Nonetheless, it is vital to minimise this effect, as it would help to conserve the limited on-board fuel. This can be achieved through careful planning and scheduling of the approach trajectory towards its target.

When traversing a path, the space robot could encounter singularities that can affect the motion of the robotic arm. Depending on the nature of motion of space robots, singularities can be kinematic or dynamic singularities. The former category of singularities are equivalent to kinematic singularities affecting fixed-base manipulators [13], are related to the kinematics of the space robot that cause the manipulator to lose one or more DoFs i.e. even a large joint displacement does not change the position and orientation of the end-effector. This occurs when the determinant of the Jacobian matrix is null. Whereas dynamic singularities, are commonly observed in free-floating space robots and they restrain the end-effector from attaining some inertial direction [14]. It is associated to the dynamics of the space robot through a Jacobian matrix that involves the inertia parameters of the system. Moreover, when the space robot is operated in the controlled-floating mode, it is only affected by kinematic singularities, as mentioned in [6]. The space robot becomes redundant with six degrees of redundancy offered by the spacecraft base, which makes avoiding singularities a simpler task, when using the redundancy of the CFSR.

Another challenge that space robots face when approaching a target is the potential collision with obstacles in their workspace as well as with the target itself. Avoiding collisions is crucial for a safe approach and capture of the target. For this reason, many algorithms, based on well-known methods, were developed for avoiding obstacles [15], such as the Artificial Potential Field (APF), the Rapidly exploring Rapid Tree (RRT), Model Predictive Control (MPC). However, due to the many challenges that space robots have to overcome, research in collision-free trajectory planning remains limited.

As previously mentioned, when planning the motion of a space robot, the main issues is the dynamic coupling effect and collision avoidance. In that context, optimal path planning has been known for finding a solution to the dynamic coupling problem by finding the best solution for the path planning that minimises a desired objective i.e. the dynamic coupling. For free-floating space robots, several studies utilised the Particle Swarm Optimisation (PSO), as part of a one-objective or a multi-objective optimisation, to reduce the dynamic coupling effect [16–21]. Other research involved the Genetic Algorithm (GA) [22,23] and Differential Evolution (DE) [24], which are evolutionary approaches that solve a problem iteratively by improving a population of candidate solutions, to optimise trajectories for free-floating space robots. On the other hand, for free-flying space robots, designing a path can be simpler because of their controlled base to maintain one desired pose or to track a pre-defined final pose. Nevertheless, it remains crucial to find optimal trajectories, for free-flying space robots, in order to save on-board fuel. Researchers presented novel path planning algorithms based on Sequential Quadratic Programming (SQP) [25], convex programming [26,27] and neural networks [28] for path optimisation.

It is common practice to rely on an initial desired path when using optimisation techniques for path planning, the algorithm then finds the best joint trajectories corresponding to end-effector's positions close to the desired initial path. In other words, the literature review suggests that the optimisation algorithm always has *a priori* knowledge

of an initial desired end-effector path or a final desired configuration. However, the pre-selected path itself may not be the best, as there is a potential existence of several optimal paths.

In a practical mission scenario, where the grasping point on the target is out of reach of the end-effector or when the arm suffers from singular configurations with a risk of collision, it will be difficult to operate the space robot if its motion is limited to the n DoF of the arm only. Hence, the CFSR is more suitable for such critical scenarios as it takes advantage of the redundancy offered by the additional DoFs of the spacecraft base to aid the arm safely approach the target and avoid singularities and obstacles. In that context, this paper presents a new optimal collision-free and singularity-free trajectory generator for a CFSR. A more primal version of this algorithm was presented in [29], which is improved and advanced in this paper. The key highlight of this new intelligent algorithm is that it enables the selection between two methods of tracking the generated trajectories depending on the mission requirements; namely, the direct tracking (DT) and the dynamic coupling tracking (DCT). The former involves a compensation of the effect of the dynamic coupling through active control and the latter takes advantage of that effect to help the motion of the spacecraft base whilst saving on-board fuel. Additionally, a unique feature of this algorithm, introduced in [29], is that it only requires the Cartesian location of the grasping point without *a priori* information of any desired joint trajectory, end-effector Cartesian path or predefined final pose for the spacecraft base. The new optimal path planning algorithm is solved using a Non-dominated Sorting Genetic Algorithm, developed in [30], that achieves a constrained multi-objective optimisation to find the best configuration trajectories i.e joint and spacecraft base trajectories. Furthermore, the mathematical model for the kinematics and dynamics of the CFSR used in this paper is the model presented in [6]. Also, the nonlinear adaptive H_∞ controller, introduced in [31], is utilised to track the designed trajectories whilst compensating for internal parametric uncertainties and external disturbances.

The remainder of this paper will introduce a detailed description of the optimal trajectory planning algorithm in Section 2. The efficacy of the new proposed algorithm is assessed through extensive simulations and the results are presented and discussed in Section 3. Lastly, Section 4 concludes this paper and suggests directions for future research on the number of obstacles to avoid.

2. Path optimisation using Genetic algorithm

Optimisation is the minimisation of one or more objectives whilst keeping the solution within one or several imposed constraints. Designing an optimal path planning algorithm for space robots usually involves solving an optimisation problem using the following kinematic equation [32]:

$$\begin{bmatrix} \dot{r}_e & \omega_e \end{bmatrix}' = \mathbf{J}(q)\dot{q}, \quad (1)$$

where vectors $\dot{r}_e \in \mathbb{R}^3$ and $\omega_e \in \mathbb{R}^3$ represent the linear and angular velocities on the end-effector, $\mathbf{J} \in \mathbb{R}^{6 \times N}$ is the Jacobian matrix of the system, with $N = 6 + n$ is the total DoFs of a space robot comprising an n DoF arm and 6 DoF spacecraft base and vector q represents configuration variables of the space robot involving: the position and attitude of the spacecraft base and the joints of the arm.

According to (1), finding the velocities in the configuration space \dot{q} involves the inverse of the Jacobian matrix \mathbf{J}^{-1} . It is known that the inverse of a matrix exists if the matrix is square with a non-zero determinant it has full rank; an example is the case of non-redundant n DoF manipulators. However, for redundant manipulators, the Jacobian matrix is non-square and its inverse does not exist. One can utilise the pseudo-inverse method, but in the case of the redundant CFSR, presented in this paper, an alternate method is preferred. In this method, (1) is embedded into an optimisation problem to find the velocities in the configuration space without needing the inverse or the pseudo-inverse of the Jacobian matrix.

For a CFSR, finding optimal configuration trajectories by minimising multiple functions within well defined regions set by several constraints, is key to designing optimal end-effector paths. This procedure is the so-called constrained multi-objective optimisation problem. In this paper, the Non-dominated Sorting Genetic Algorithm II (NS-GAII), developed in [30], is used to solve the path planning problem. There are not many algorithms available in the literature that deal with constrained multi-objective optimisations. The Constrained Multi-Objective PSO (CMOPSO), presented in [33], can also be a good candidate to solve the path planning problem, but little research has been conducted in its regards when compared to NSGAII.

The velocity vector $\dot{q} = [\dot{r}_T \ \omega_{sc} \ \dot{\theta}]'$ at time $(t + 1)$ represents the optimisation variable, when the algorithm tries to finds the following step for the configuration trajectories, where $\dot{r}_T \in \mathbb{R}^3$ and $\omega_{sc} \in \mathbb{R}^3$ are vectors of the linear and angular velocities of the spacecraft base and $\dot{\theta} \in \mathbb{R}^n$ is the vector of joint velocities of the arm. All the vectors involved in the optimisation are illustrated in Fig. 1.

2.1. The objective functions

There is a total of six functions to minimise in order to ensure a safe and optimal approach path. Each function has a specific role to play in the optimisation to help the end-effector converge towards the grasping point. The algorithm stops when this relative distance is null i.e. end-effector is at the grasping point.

2.1.1. First objective function

This is the principal function that defines the final end-effector path. It is a function of $\dot{q}(t + 1)$ and is defined as follows:

$$f_1 = \left\| r_{ef} - r_e(t + 1) \right\|_2, \quad (2)$$

using $\begin{bmatrix} r_e(t + 1) & \phi_e(t + 1) \end{bmatrix}' = \begin{bmatrix} r_e(t) & \phi_e(t) \end{bmatrix}' + \int_t^{t+1} J(q)\dot{q}(t)dt.$

It is related to the distance separating the desired final point r_{ef} and the next Cartesian step travelled by the end-effector $r_e(t + 1)$. As previously mentioned, the desired final Cartesian position is the only known input for the algorithm which is defined by the user for simulation purposes. The next Cartesian point of the end-effector is found through a sequence of operations as follows:

- Calculate the next linear position $r_T(t + 1)$ and attitude $\phi(t + 1)$ of the spacecraft base and the joints displacement $\theta(t + 1)$ from $\dot{q}(t + 1)$
- Use $r_T(t + 1)$, $\phi(t + 1)$ and $\theta(t + 1)$ to compute the Jacobian matrix of the space robot $J \in \mathbb{R}^{6 \times N}$
- Use (1) with the computed Jacobian J and the optimisation variable $\dot{q}(t + 1)$ to compute the velocity of the end effector $\dot{r}_e(t + 1)$
- Compute the position of the end-effector $r_e(t + 1)$ from $\dot{r}_e(t + 1)$

2.1.2. Second objective function (singularity avoidance objective)

This function maximises the manipulability of the CFSR as follows:

$$f_2 = \frac{1}{1 + M}, \quad \text{with} \quad M = \sqrt{\det(J_{rot}J_{rot}')} \quad (3)$$

where $J_{rot} \in \mathbb{R}^{6 \times (3+n)}$ is the rotational part of the analytical Jacobian matrix J as presented in [32].

The algorithm presented in this paper finds configuration trajectories without needing the inverse or the pseudo-inverse of the Jacobian matrix, it is therefore not necessary to use classical methods, such as those presented in [34,35], for singularity avoidance. Nevertheless, the manipulability of the redundant CFSR has to be maximised. These configurations are sets of joints angles and spacecraft attitude that result in an unreachable end-effector Cartesian position. When computing the manipulability of CFSR, as first developed in [36] for a terrestrial redundant manipulator, only the joints angles and the attitude of the spacecraft are involved. This is because of the structure

of the Jacobian matrix that only depends on ϕ and θ excluding r_T . Hence, the rotational part of the Jacobian $J_{rot} \in \mathbb{R}^{(6 \times 3+n)}$ is used to compute the manipulability. Therefore, in order to maximise the manipulability, its inverse is selected as the second objective as follows:

2.1.3. Third objective function

This function penalises maximum displacements and angles linked to actuator limitations. It is defined as follows:

$$f_3 = \frac{1}{1 + \|q_m\|_2}, \quad \text{with} \quad q_m = \left| q(t) + \int_t^{t+1} \dot{q}(t)dt \right| - q_{max}. \quad (4)$$

For a vector q_{max} that represents the maximum allowed values of r_T , ϕ and θ , the difference between $q(t + 1)$ and q_{max} , denoted q_m , is to be maximised to keep the motion of the space robot within the imposed limits. Therefore, by minimising the inverse of q_m it is guaranteed that the saturation limits of the actuators are not attained.

2.1.4. Fourth objective function

This is the norm of the vector representing the angular velocities of the spacecraft base and the joints' velocities of the arm, defined as follows:

$$f_4 = \|\dot{\phi}(t + 1) \ \dot{\theta}(t + 1)\|_2 \quad \text{with} \quad \dot{\phi}, \dot{\theta} \subset \dot{q}. \quad (5)$$

It is used to minimise the motion involved in the maximisation of the manipulability as well as when the rotational motion (attitude of spacecraft base and arm's joints) is needed to reach the target. The fourth objective function is defined as follows:

2.1.5. Fifth objective function

This function represents the norm of the linear velocity of the spacecraft base in order to minimise its linear motion. It is defined as follows:

$$f_5 = \|\dot{r}_T(t + 1)\|_2 \quad \text{with} \quad \dot{r}_T \subset \dot{q}. \quad (6)$$

2.1.6. Sixth objective function (obstacle avoidance objective)

The CFSR will have the capability to avoid obstacles when minimising the following objective:

$$f_6 = \frac{1}{\| [d_{ob_0} \dots d_{ob_n}] \|_2} \quad \text{with} \quad \begin{cases} d_{ob_0} = \| r_T(t + 1) - r_{ob} \|_2 \\ d_{ob_i} = \| r_i(t + 1) - r_{ob} \|_2 \end{cases} \quad (7)$$

This function is enabled when an obstacle is in the vicinity of the spacer robot. It is defined as the inverse of the distance d_{ob_i} between the CoM of the links as well as the base of the space robot and the detected obstacle, as illustrated in Fig. 1. This function is used to maximise the distance d_{ob_i} whilst minimising its inverse by firstly computing the relative distance between the space robot and the potential obstacle. This goes through the following series of operations:

- Compute the Jacobian matrix of each link $J_i \in \mathbb{R}^{6 \times N}$ using r_T , ϕ and θ previously calculated.
- Use (1) with the full Jacobian matrices of the links J_i and the optimisation variable $\dot{q}(t + 1)$ to find the velocity of the i th link $\dot{r}_i(t + 1)$.
- Compute the position of each link $r_i(t + 1)$ from $\dot{r}_i(t + 1)$.

The resulting $r_i(t + 1)$ is used to compute the distance between the i th link and an obstacle located at $r_{ob} = [X_{ob} \ Y_{ob} \ Z_{ob}]$.

2.2. The constraints

Defining bounds for the optimisation variable is necessary to narrow down the search for the best solution. Nevertheless, sometimes these bounds are not sufficient and additional constraints are required. In the case of the CFSR, the bounds constitute the maximum allowed joints' and spacecraft base's velocities i.e. maximum negative and positive value for $\dot{q}(t + 1)$. When the GA is optimising the solution for the path

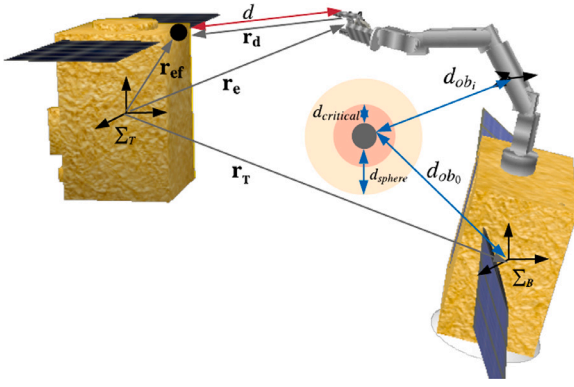


Fig. 1. Artistic illustration representing the relative position vectors and distances used in the optimal path generator.

planning problem, the difference between the best previous solution (best value for the velocity vector) and the current solution might be large enough to cause abrupt space robot motion. Therefore, in addition to the upper and lower bounds, there is a compelling need to further constrain $\dot{q}(t+1)$. In this paper, the constraints on the velocity vary depending on the trajectory segment where the links of the arm and the spacecraft base are located at time $(t+1)$. Three segments, introduced in [29], are defined as: the obstacle-free trajectory segment, the collision avoidance trajectory segment and the safety (final) trajectory segment.

2.2.1. The obstacle-free segment

In this segment, the algorithm finds a solution for $\dot{q}(t+1)$ without accounting for obstacles as the workspace is free of them. Several constraints are imposed on this segment:

Constraints on the velocity. This constraint is imposed on the velocity vector i.e. the optimisation variable vector $\dot{q}(t+1)$. As previously stated, the algorithm does not have any information about a predefined velocity. Hence, in order to prevent abrupt undesired motion, $\dot{q}(t+1)$ is constrained to minimally change with respect to the current velocities $(\dot{q}(t) \pm \epsilon_1)$ depending on the sign of $\dot{q}(t)$, with $\epsilon_1 \in \mathbb{R}^N$ is a vector containing two vectors $\epsilon_{1_{sc}} \in \mathbb{R}^6$ and $\epsilon_{1_m} \in \mathbb{R}^n$ of small positive scalars $\epsilon_{1_{sc}}$ and ϵ_{1_m} .

Furthermore, this paper introduces the following two tracking methods: the direct tracking (DT) and the dynamic coupling (DCT) tracking. Hence, depending on the requirements of the mission, one of these two methods can be selected through the following constraints:

$$|\dot{X}(t+1) - \dot{X}(t)| < \epsilon_{1_{sc}}, \text{ DT} \quad (8a)$$

$$|(\dot{X}(t+1) + \dot{X}_{coup}(t+1)) - \dot{X}(t)| < \epsilon_{1_{sc}}, \text{ DCT} \quad (8b)$$

$$\text{with } X = [r_T \ \phi]'$$

$$|\dot{\theta}(t+1) - \dot{\theta}(t)| < \epsilon_{1_m} \quad (8c)$$

where $\dot{X}_{coup}(t+1)$ is the vector representing the extra velocity generated from the motion of the arm due to the dynamic coupling. This velocity is computed by solving the dynamic coupling equation defined as: $\ddot{X} = D_{sc}^{-1} (f_{coup} - C_{sc} \dot{X})$, where f_{coup} is the vector representing the forces and torques transferred to the spacecraft base due to the motion of the arm.

Constraint on the approach angle. Although the NSGAII optimises all objectives and satisfies all constraints, sometimes the resulting Cartesian path may cause the end-effector to collide with the target. This case is illustrated by the dashed line shown in Fig. 2. To overcome this issue, another constraint has been introduced to generate a path that prevents the end-effector from colliding with the target; an example of an alternate trajectory is illustrated by the solid line in Fig. 2. This

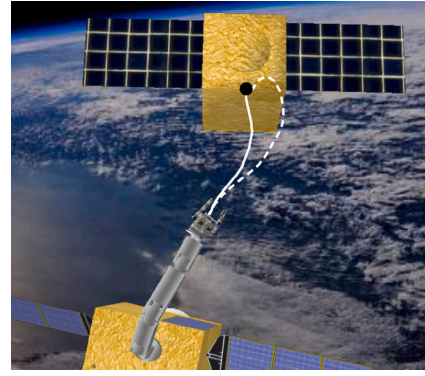


Fig. 2. A scenario illustrating the importance of the approach angle constraint to avoid collision with target.

constraint keeps the vector from the end-effector to the grasping point (r_d) within an angle σ to the plane containing the grasping point and parallel to the unit vector \hat{e}_p , as illustrated in Fig. 3. This constraint is defined, using the cross product equation, as follows:

$$(r_d \times \hat{e}_p) - (\|r_d\| \|\hat{e}_p\| \sin(\sigma)) \hat{e}_n \leq 0, \quad (9)$$

with $r_d = r_{ef} - r_e(t+1)$,

where \hat{e}_p is the unit vector in the direction of approach of the CFSR and \hat{e}_n is the unit vector normal to the plane containing r_d and \hat{e}_p .

Other constraints. To prevent the spacecraft base from colliding with the target, another constraint has to be imposed on r_T to keep it at a distance d_{impact} from the target, where d_{impact} is the user-defined safety distance between the CoM of the target and the CoM of the spacecraft base.

Sometimes the optimisation algorithm finds the best solution that satisfies all constraints but results in an increase in the distance d between the end-effector and the grasping point. Hence an additional constraint is imposed that prevents the end-effector from stepping back and enables it to always converge towards the grasping point. It is defined as follows:

$$d(t) - d(t+1) < 0.$$

This guarantees that the distance between the end-effector and the grasping point on the target decreases at every time step; thus, guaranteeing a safe capturing of the target.

2.2.2. The collision avoidance segment

In this segment, the trajectory optimisation algorithm minimises all objectives $f_1 \rightarrow f_6$, whilst satisfying two more constraints to avoid obstacles.

Constraint on obstacle avoidance. This trajectory segment is enabled when one or several links of the space robot are within the sphere around the obstacle; this sphere has a radius d_{sphere} that determines a zone known here as *the warning zone*, as illustrated in Fig. 1. The distance between the i th link and an obstacle is referred to as d_{ci} .

The constraint for obstacle avoidance is then defined as:

$$d_{obi} - d_{critical} > 0, \quad (10)$$

$$\text{with } \begin{cases} d_{obi} = \|r_i(t+1) - r_{ob}\|_2 \\ [r_i(t+1) \ \phi_i(t+1)]' = [r_i(t) \ \phi_i(t)]' + \int_t^{t+1} J(q) \dot{q}(t) dt \end{cases}$$

where $d_{critical}$ is the radius of the *keep-out zone* around the obstacle, with $d_{critical} < d_{sphere}$ as shown in Fig. 1. Vectors r_i and ϕ_i represent the position and orientation of the i th link.

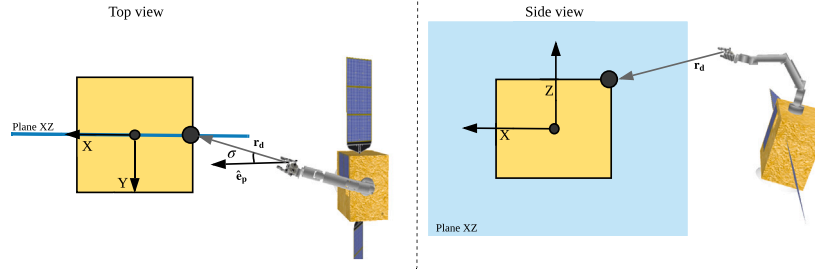


Fig. 3. Artistic illustration describing the approach angle constraint.

Constraint on the distance travelled by the end-effector. During the obstacle-free trajectory, the end-effector travels towards the grasping point by reducing the distance d whilst satisfying a constraint that this distance always decreases. However, in the presence of one or more obstacles, this constraint would prevent the obstacle avoidance process. This is because the arm may get stuck at a configuration where it cannot find an escape configuration that might increase d to avoid the obstacle. Hence, the constraint on the distance d is eliminated during the obstacle avoidance segment. In other words, during this segment, the priority is given to avoiding obstacles whilst minimising the distance d without an extra constraint.

2.2.3. The safety trajectory segment

Given the bounds and constraints imposed on the velocity vector \dot{q} (optimisation variable) to avoid abrupt changes, the algorithm selects potential solutions from the constrained values. This is sufficient when the distance d is large enough, but the risk of collision with target increases as this distance decreases when the end-effector is closer to the grasping point. This distance is referred to as the *safety distance* during which the configuration velocities must decrease gradually to zero when the end-effector reaches the grasping point. In other words, the algorithm must be able to reduce the configuration velocities depending on the distance d between the end-effector and the grasping point.

Constraint on decreasing velocity. Here, a function is introduced to satisfy the constraint on decreasing the configuration velocity as follows:

$$f(s) = \frac{1}{1 + \nu s^2}, \text{ with } s = \frac{d_{safe} - d(t+1)}{d_{safe}}, \quad (11)$$

where d_{safe} is the minimum user-defined safety distance and ν is a positive scalar that sets the final value of $f(s)$ which represents the precision of the end-effector.

The constraint on the velocity during the safety trajectory segment is then defined as an equality constraint, taking into account both direct and dynamic coupling tracking, as follows:

$$\dot{X}(t+1) = (X(t+1) - X(t)) f(s), \text{ DT} \quad (12a)$$

$$\dot{X}(t+1) + \dot{X}_{coup}(t+1) = (X(t+1) - X(t)) f(s), \text{ DCT} \quad (12b)$$

$$\dot{\theta}(t+1) = (\theta(t+1) - \theta(t)) f(s). \quad (12c)$$

Constraint on the distance travelled by the end-effector. This constraint is different from the one imposed during the obstacle-free segment and the obstacle avoidance segment. This constraint determines the gradient of the decreasing function (11) by imposing a user-defined step d_{step} on the distance travelled. It is defined as follows:

$$d(t) - d(t+1) = d_{step}. \quad (13)$$

2.3. Shortcomings of the equality constraints

The constraints described by Eqs (8)–(13) are either equality or inequality constraints. It is common knowledge that the equality constraints are computationally heavy as the optimisation algorithm is incapable to find the best solutions to the optimisation problem within these constraints. The literature review reveals that it is possible to transform the equality constraints to inequality constraints by adding a tolerance to loosen the optimisation process [37]. This is performed as follows:

$$f(x) = 0, \text{ becomes } |f(x)| - \epsilon \leq 0, \quad (14)$$

where ϵ is the user-defined tolerance.

In the path planning algorithm introduced in this research, the constraints described by Eqs. (12) and (13) are the only equality constraints presented. Transforming these constraints into inequality constraints using Eq. (14) gives:

$$\begin{aligned} |\dot{X}(t+1) - [(X(t+1) - X(t)) f(s)]| - \epsilon_{2_{sc}} &\leq 0, & \text{DT} \\ |[\dot{X}(t+1) + \dot{X}_{coup}(t+1)] - [(X(t+1) - X(t)) f(s)]| - \epsilon_{2_{sc}} &\leq 0, & \text{DCT} \\ |\dot{\theta}(t+1) - [(\theta(t+1) - \theta(t)) f(s)]| - \epsilon_{2_m} &\leq 0, \\ |[d(t) - d(t+1)] - d_{step}| - \epsilon_2 &\leq 0, \end{aligned}$$

with $\epsilon_{2_{sc}} \in \mathbb{R}^6$ and $\epsilon_{2_m} \in \mathbb{R}^n$ are column vectors with each row representing the tolerance $\epsilon_{2_{sc}}$ and ϵ_{2_m} respectively and ϵ_2 is the tolerance for the desired path steps.

2.4. The resultant trajectory optimisation algorithm

The overall collision-free and singularity-free optimal trajectory generator using NSGAII is described in Table 1. The constrained configuration velocity vector results in a motion free from abrupt changes. Nevertheless, it was observed that seeking the best solution to minimise all objectives is not always possible when satisfying the velocity constraints. As a result, undesired and small peaks in the velocity appear. In order to circumvent this issue, a smoothing function, using the widely used exponential smoothing technique [38], was introduced at the velocity level to ensure smoothness of motion. This is shown in Fig. 4 depicting the smoothing of the joints' velocities. Although, only the arm's joints are shown here, the same smoothing technique was applied on the spacecraft base's velocities.

3. Simulations and results

The simulation environment that the following result are based on is similar to the one utilised in [29]. The physical parameters of the space robot as well as the initial conditions on the position vectors are listed as follows:

- Mass: 200 kg for the spacecraft base with 40% of fuel mass, the arm has 6 DoF and 5 links with masses equal to: 7 kg, 5 kg, 4 kg, 2 kg, 1 kg respectively

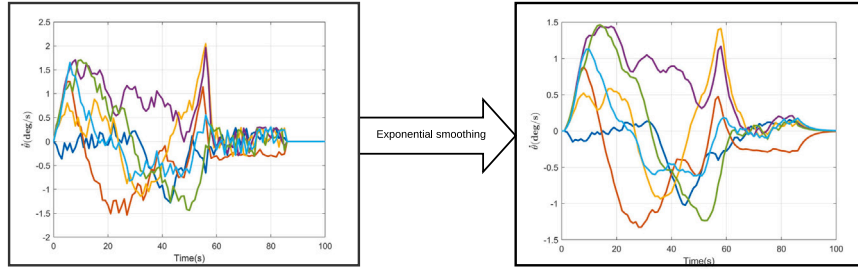


Fig. 4. Velocity smoothing to avoid undesired abrupt changes in the velocity.

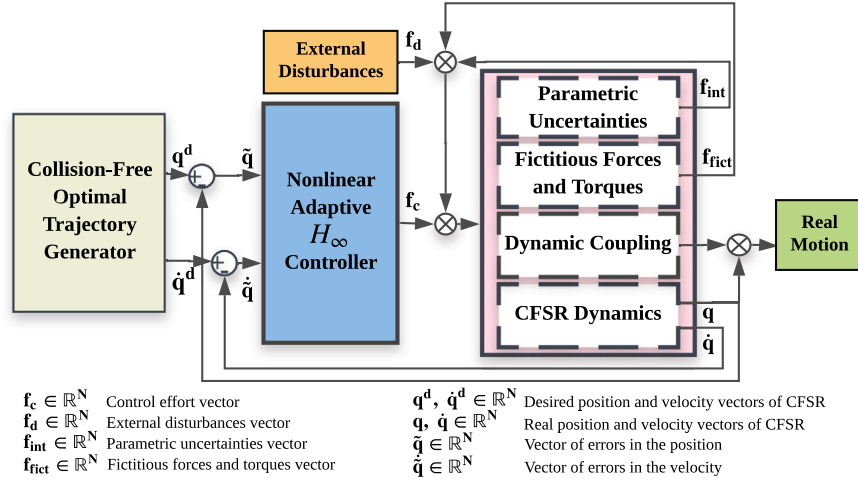


Fig. 5. Diagram representing the Closed-loop architecture of the feedback control to track the generated optimal trajectories using an adaptive H_∞ controller.

Table 1

Description of the optimal trajectory algorithm.

if obstacle-free segment

$$\begin{aligned}
 & \min f_i(\dot{q}), \quad i = 1, 2, \dots, 5 \\
 & s.t. \quad \dot{q}_{min} \leq \dot{q} \leq \dot{q}_{max} \\
 & \quad |\dot{X}(t+1) - \dot{X}(t)| < \epsilon_{1_{sc}} \quad \text{or} \quad |(\dot{X}(t+1) + \dot{X}_{coup}(t+1)) - \dot{X}(t)| < \epsilon_{1_{sc}} \\
 & \quad |\dot{\theta}(t+1) - \dot{\theta}(t)| < \epsilon_{1_m} \\
 & \quad \|r_T\|_2 > d_{impact} \\
 & \quad d(t) - d(t+1) < 0 \\
 & \quad (r_d \times \hat{e}_p) - (\|r_d\| \|\hat{e}_p\| \sin(\sigma)) \hat{e}_n \leq 0
 \end{aligned}$$

elseif obstacle avoidance segment

$$\begin{aligned}
 & \min f_i(\dot{q}), \quad i = 1, 2, \dots, 6 \\
 & s.t. \quad \dot{q}_{min} \leq \dot{q} \leq \dot{q}_{max} \\
 & \quad |\dot{X}(t+1) - \dot{X}(t)| < \epsilon_{1_{sc}} \quad \text{or} \quad |(\dot{X}(t+1) + \dot{X}_{coup}(t+1)) - \dot{X}(t)| < \epsilon_{1_{sc}} \\
 & \quad |\dot{\theta}(t+1) - \dot{\theta}(t)| < \epsilon_{1_m} \\
 & \quad d_{ob_i} - d_{critical} > 0 \\
 & \quad \|r_T\|_2 > d_{impact} \\
 & \quad (r_d \times \hat{e}_p) - (\|r_d\| \|\hat{e}_p\| \sin(\sigma)) \hat{e}_n \leq 0
 \end{aligned}$$

else safety segment

$$\begin{aligned}
 & \min f_i(\dot{q}), \quad i = 1, 2, \dots, 5 \\
 & s.t. \quad \dot{q}_{min} \leq \dot{q} \leq \dot{q}_{max} \\
 & \quad |\dot{X}(t+1) - [(X(t+1) - X(t)) f(s)]| - \epsilon_{sc} \leq 0 \\
 & \quad \text{or} \quad |[\dot{X}(t+1) + \dot{X}_{coup}(t+1)] - [(X(t+1) - X(t)) f(s)]| - \epsilon_{sc} \leq 0 \\
 & \quad |\dot{\theta}(t+1) - [(\theta(t+1) - \theta(t)) f(s)]| - \epsilon_{sm} \leq 0 \\
 & \quad \|r_T\|_2 > d_{impact} \\
 & \quad |d(t) - d(t+1) - d_{step}| - \epsilon_2 \leq 0, \\
 & \quad (r_d \times \hat{e}_p) - (\|r_d\| \|\hat{e}_p\| \sin(\sigma)) \hat{e}_n \leq 0
 \end{aligned}$$

- Size: a cube with side 0.5 m accounts for the spacecraft base and the lengths of the links are: 0.16 m, 0.25 m, 0.1 m, 0.16 m, 0.072 m
- Initial position for the CFSR: $r_T = [1.4 \text{ m } 0 \text{ m } -0.5 \text{ m}]'$ and $\phi = [0^\circ \ 0^\circ \ 0^\circ]'$ for the position and orientation of the spacecraft base and $\theta = [0^\circ \ 0^\circ \ 0^\circ \ 0^\circ \ 0^\circ]'$ for the arm's joints
- Grasping point position: $r_{ef} = [0.5 \ 0 \ 0.5]'$
- Other: $d_{safe} = d_{sphere} = 0.2 \text{ m}$, $d_{critical} = 0.1 \text{ m}$ and $v = 20$, $\epsilon_1 = 0.002$, $\epsilon_2 = 0.006$

Moreover, the adaptive H_∞ controller, shown in Fig. 5, is used to track the generated trajectories. The detailed design of the adaptive H_∞ controller for a CFSR can be found in [31]. This controller guarantees a small and precise control effort regardless of the errors in the trajectory or disturbances.

Weighted objectives. The six previously mentioned objectives can be assigned weights depending on the desired priority. In order to maintain a stable solution to the optimisation problem, it is desired that the highest weight is always assigned to the most contributing objectives in the different segments. Namely, the first objective related to the distance between the end-effector and the grasping point, or the second objective that maximises the manipulability of the CFSR. In this simulation, the following weights were selected: 0.3, 0.2, 0.1, 0.1, 0.1, 0.2 from the first to last objective respectively.

3.1. Motion of CFSR

As previously mentioned, the first objective to minimise is the distance between the end-effector and the grasping point. This determines the final path of the end-effector by taking into account all the introduced constraints and remaining objectives to find optimal configuration trajectories. In this context, Fig. 6 shows the results of the minimisation of the first objective.

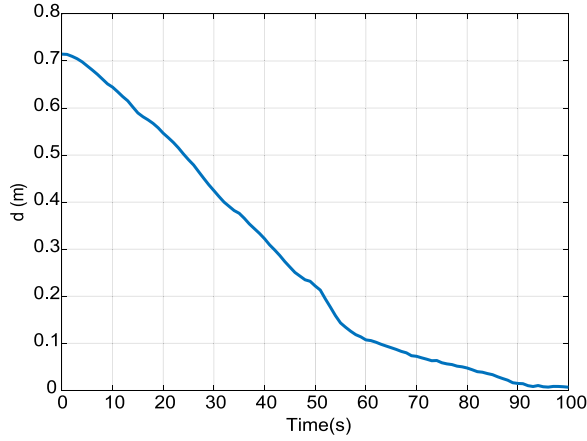


Fig. 6. The minimisation of the distance (d) between the end-effector and the grasping point throughout the execution time.

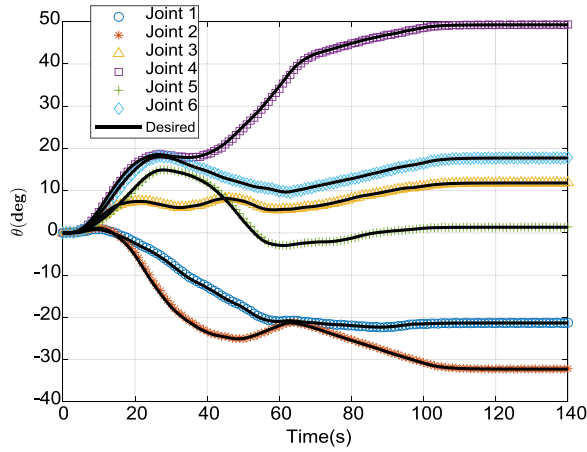


Fig. 7. Arm's joints trajectories ($\theta \in \mathbb{R}^6$).

3.1.1. The resultant motion of the CFSR

Fig. 7 shows the results of the tracking of the desired arm trajectories using the closed-loop nonlinear adaptive H_∞ controller shown in Fig. 5. The desired trajectories are generated using the algorithm detailed in Section 2. The respective torques responsible for the arm's motion are depicted in Fig. 8. Also, the tracking of the desired motion of the spacecraft base shown in Fig. 9 which originates from the small applied forces and torques shown in Fig. 10. The variations in the applied forces and torques for both the arm and its base are inherited from the counter-effort of the nonlinear adaptive H_∞ controller to suppress disturbances.

The corresponding velocities of the arm's joints and the linear and angular velocities of the spacecraft base are shown in Figs. 11 and 12 respectively. It is evident from the presented results that the new optimal trajectory planning algorithm can generate a smooth motion for the CFSR and decelerates until it comes to a stop when the final desired point is reached.

Moreover, considering the desired grasping point r_{ef} and the motion described by Figs. 7–12, the corresponding Cartesian end-effector path is depicted in Fig. 13. This figure shows that the arm is able, with the help of the spacecraft base, to reach the grasping point in an optimal way.

It is important to mention that the results presented, using the algorithm presented here, are improved when compared to the trajectories designed using its previous version introduced in [29]. In fact, given the same initial conditions and final end-effector location, the space robot

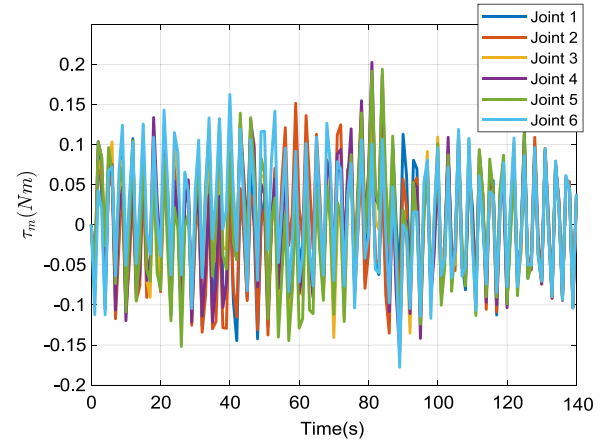


Fig. 8. Arm's joint torques ($\tau_m \in \mathbb{R}^6$).

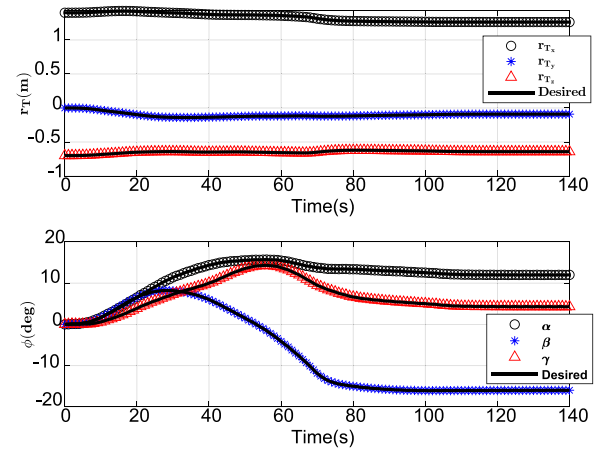


Fig. 9. Spacecraft position ($r_T \in \mathbb{R}^3$) and attitude ($\phi \in \mathbb{R}^3$).

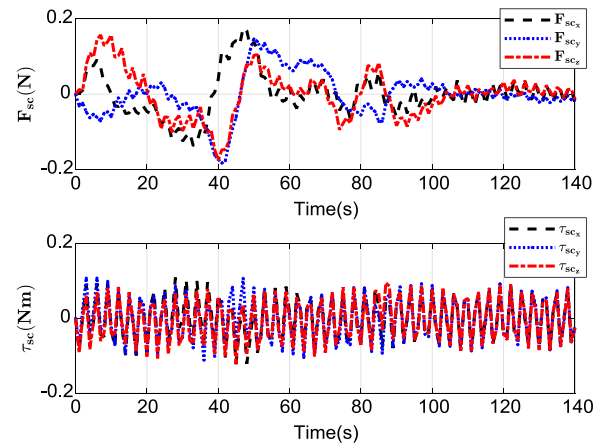


Fig. 10. Spacecraft forces ($F_{sc} \in \mathbb{R}^3$) and torques ($\tau_{sc} \in \mathbb{R}^3$).

in this paper, undertakes a smaller motion as seen in Figs. 7 and 9. The arm's joints all required 50% less displacement to reach the target, which means that less energy is required for the joints motors to reach the target. Also, the spacecraft base's linear and angular displacements were reduced by 60%, to help the arm reach the grasping point whilst saving on-board fuel required for pose control. In other words, the algorithm was capable of finding a better path that results in a reduced motion of the space robot. This proves that the optimal path generator

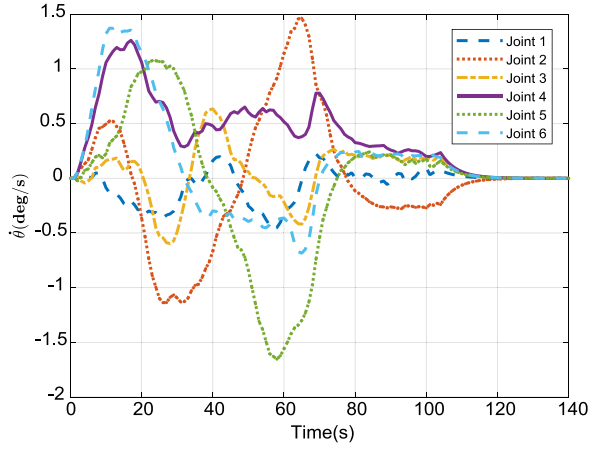
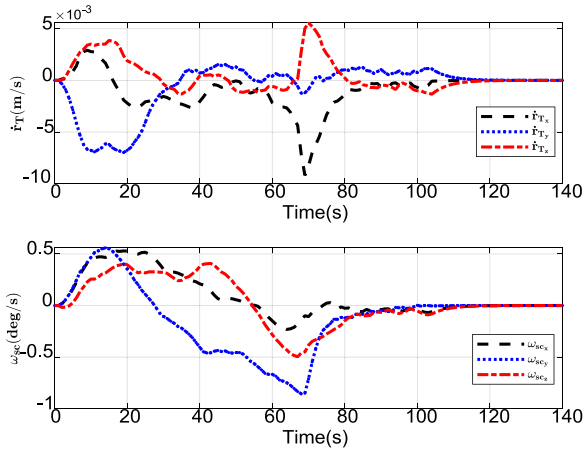
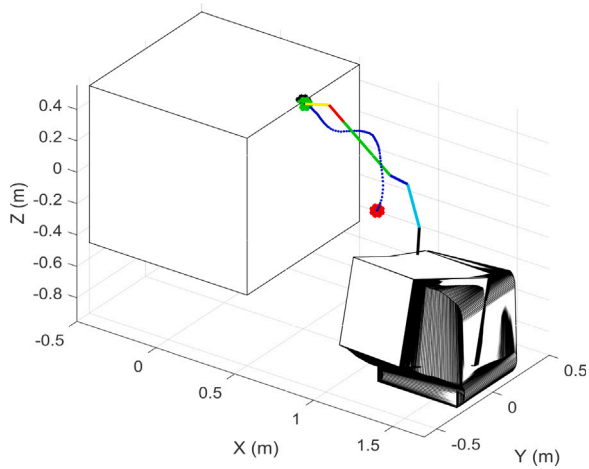
Fig. 11. Joint velocities of arm ($\theta \in \mathbb{R}^6$).Fig. 12. Spacecraft linear and angular velocities (\dot{r}_T and $\omega_{sc} \in \mathbb{R}^3$).

Fig. 13. End-effector Cartesian path using the optimal trajectory generator.

presented in this paper has a better performance than its predecessor introduced in [29].

3.1.2. Importance of using the dynamic coupling effect in saving on-board fuel

The optimisation problem introduced in this paper aims to minimise the motion of the space robot and the dynamic coupling effect. This

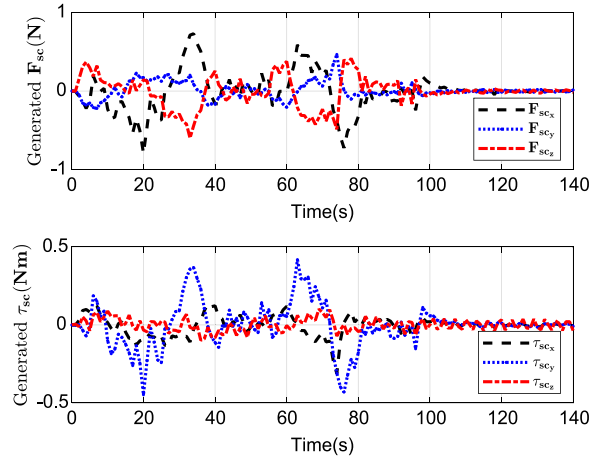


Fig. 14. Spacecraft base forces and torques generated from the dynamic coupling.

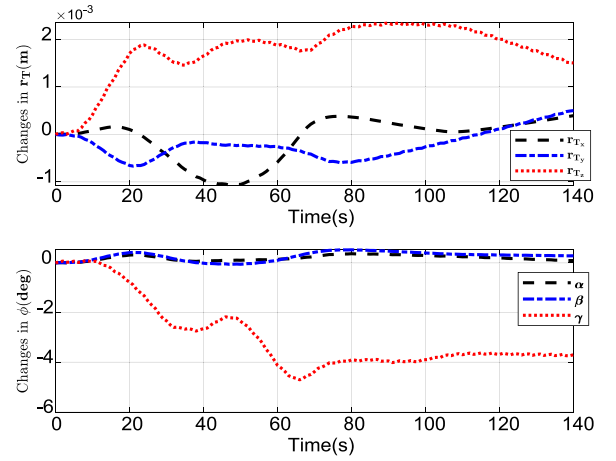


Fig. 15. Spacecraft base position and attitude generated from the dynamic coupling.

effect is reduced but not suppressed. Indeed, the magnitudes of the reaction forces and torques affecting the spacecraft base and generated by the motion of the arm are shown in Fig. 14 and the corresponding changes in the position and attitude are depicted in Fig. 15. Although the amplitudes of these changes are small compared to the actual motion of the spacecraft base, they can affect the resulting end-effector path. The algorithm presented in this paper can take advantage of the dynamic coupling effect to help the CFSR reach the grasping point whilst saving on-board fuel, by selecting the constraints described by (8b). Fig. 16 shows the effect that the dynamic coupling has on the Cartesian trajectory of the end-effector. The errors shown in Fig. 16 prove that the dynamic coupling has a significant effect on the end-effector's position (red line) compared to the motion of the end-effector when this effect is compensated (blue line). Compensating for this effect uses active feedback control as well as a feed-forward compensation as presented in [6], which requires the use of extra on-board fuel for pose correction. This typically undesired dynamic coupling effect can be exploited to save on-board fuel by utilising the resulting large end-effector displacement, as seen in Fig. 16. In such cases, the dynamic coupling tracking is preferred, using (8b).

3.1.3. Direct tracking versus dynamic coupling tracking

As previously mentioned, the path planning algorithm presented in this research can take advantage of the dynamic coupling effect to help the CFSR reach the grasping point whilst saving on-board fuel, by selecting the constraint described by Eq. (8b).

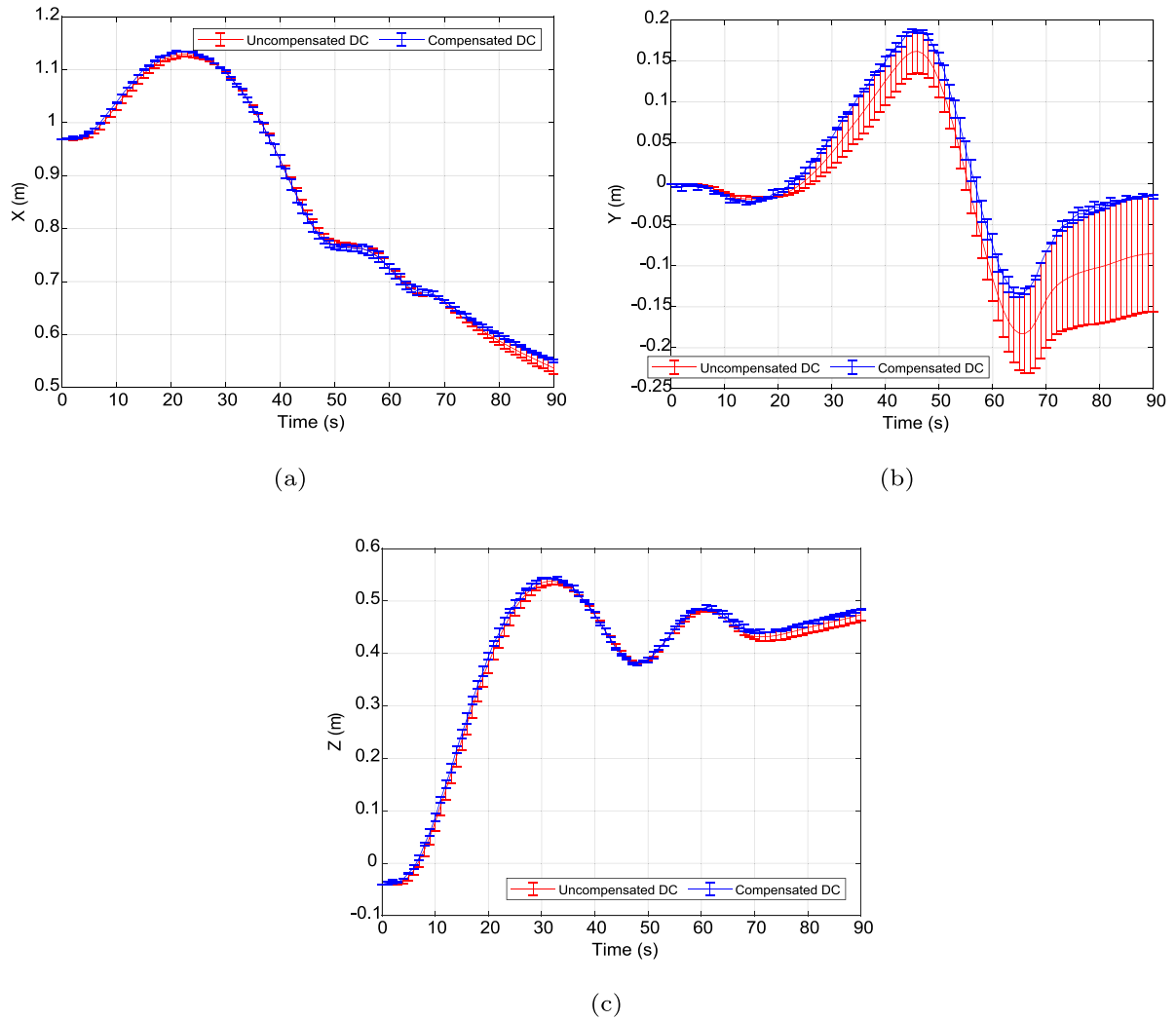


Fig. 16. Errors along the (a) X axis, (b) Y axis and (c) Z axis of the end-effector's Cartesian path due to the dynamic coupling (DC), compared to the errors in the Cartesian path when both feedback and feed-forward control are applied i.e. case where DC is suppressed and when it is not.

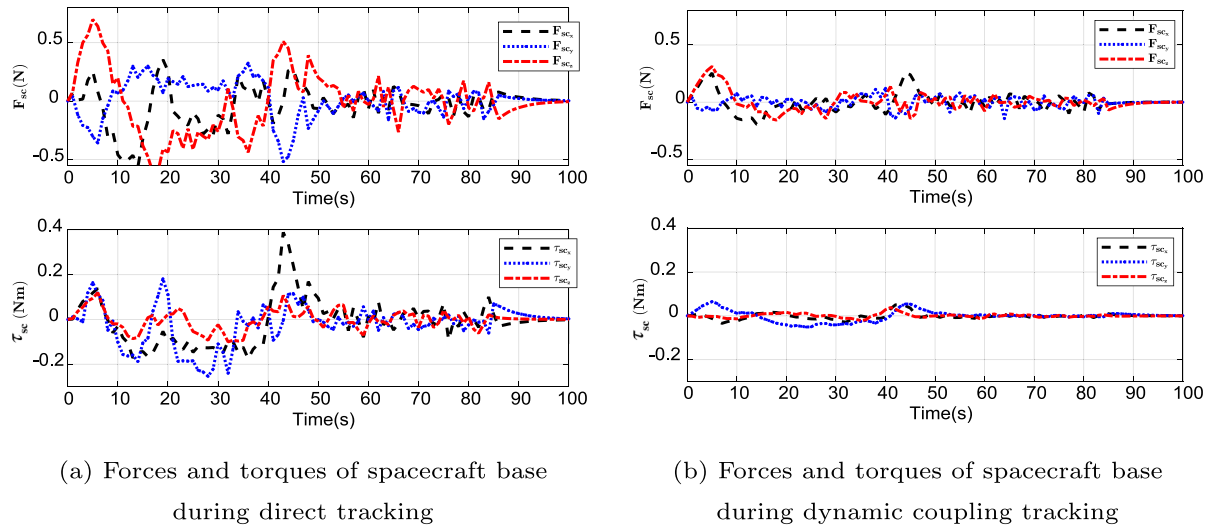
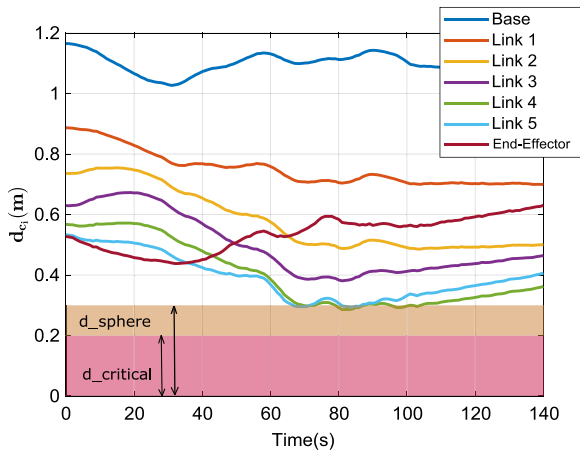


Fig. 17. Forces and torques of spacecraft base during (a) direct and (b) dynamic coupling tracking.

Table 2

Fuel consumption comparison between direct and dynamic coupling tracking.

Tests	Test 1	Test 2	Test 3	Test 4
DC	0.045 kg	0.076 kg	0.08 kg	0.13 kg
DCT	7×10^{-4} kg	7.6×10^{-4} kg	8.9×10^{-4} kg	11×10^{-4} kg

**Fig. 18.** Relative distance between the obstacle and the links when collision avoidance is taken into account during optimisation [29].

It is essential to perform a comparison between the direct and the dynamic coupling tracking to highlight the fuel saving feature of the latter. Fig. 17 shows the forces and torques necessary to generate the motion of the spacecraft base during direct and dynamic coupling tracking. Large forces and torques are applied when using feedback and feed-forward control during direct tracking, as seen in Fig. 17a. These values are small enough for close-proximity manoeuvres but larger than the effort applied during the dynamic coupling tracking, as shown in Fig. 17b.

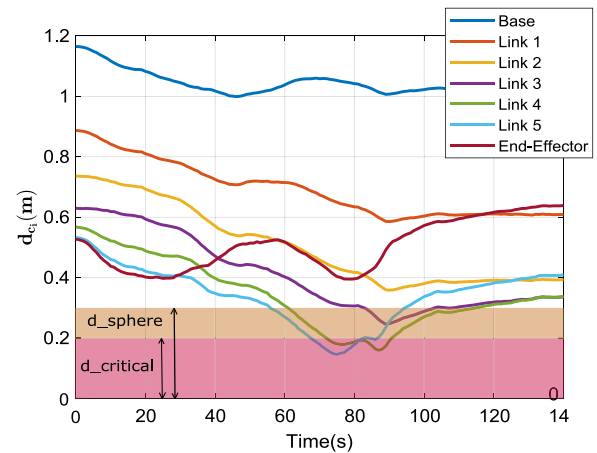
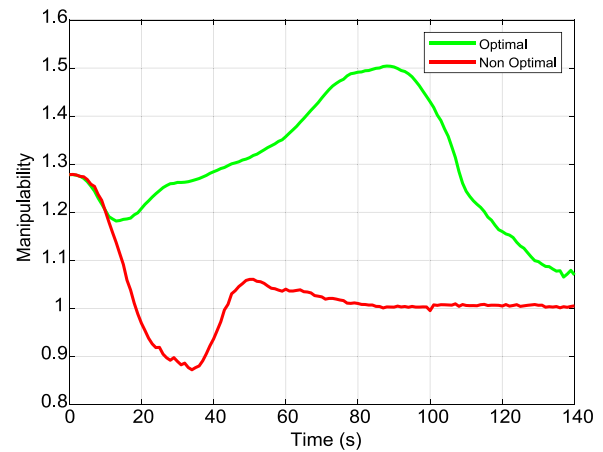
Due to the ideal environment of simulations, it is not possible to validate the accuracy for both methods of tracking. The simulated tests resulted in a 0.0025 m accuracy for both direct and dynamic coupling tracking, even though the desired accuracy set by v is 0.001 m. However, it is known from the literature that the extra motion of the spacecraft base produced from the dynamic coupling has to be suppressed in cases where higher accuracy is required. In such cases, the direct tracking is preferred to guarantee accuracy.

Furthermore, several simulation tests have been conducted to reveal the fuel saving trait of using the dynamic coupling tracking. Table 2 shows the variation in fuel consumption, throughout the motion of the CFSR, when using direct tracking and dynamic coupling tracking. This highlights the merit of using the motion of the spacecraft base, resulting from the motion of the arm, instead of compensating for it using active control.

3.2. Obstacle and singularity avoidance

The nature of the motion of the CFSR enables the exploitation of the extra DoFs offered by the spacecraft base. These extra DoFs are utilised to avoid obstacles and singularities as they make the space robot redundant with 6 degrees of redundancy originating from the 6 DoF of the spacecraft base.

Although the results were presented in [29], it is worth including them in this paper for completeness. Fig. 18 shows that the optimisation helps the space robot to keep away from the *keep-out zone* as soon as one or several links enter the *warning zone*. Fig. 19 on the other hand, shows how the space robot approached the risk of collision if the obstacle avoidance constraint (10) is not taken into account. Moreover, Fig. 20 shows that when objective f_3 is optimised, the manipulability

**Fig. 19.** Relative distance between the obstacle and the links when collision avoidance is not taken into account during optimisation [29].**Fig. 20.** Manipulability of the CFSR with and without optimisation [29].

is higher when compared to the case where f_3 is not optimised. When the manipulability decreases, the space robot could potentially reach singular configuration during its motion.

4. Conclusion

A novel optimal trajectory generator, capable of avoiding collisions and singularities using a NSGAIL, was presented. The key highlight of the path generator is the possibility to select between two modes of tracking, namely, the direct tracking and the dynamic coupling tracking. The former mode is to be utilised when the effect of the dynamic coupling is undesirable and has to be compensated using active control. The latter favours the fuel conservation by taking advantage of the dynamic coupling effect on the end-effector's inertial position to help reach the grasping point. The selection of the tracking mode is dependent on the mission. For instance, when handling mirrors during in-orbit telescope assembly, the direct tracking is preferred, where the extra precision is necessary to guarantee a safe and gentle capture at all times. Whereas when the space robot is required to perform multiple tasks in different orbits, fuel conservation becomes a priority and the dynamic coupling tracking can be selected. Additionally, this new algorithm only requires the Cartesian location of the grasping point without a prior knowledge of a desired path or final configuration. The effectiveness of this optimal trajectory generator was verified by operating the space robot in the controlled-floating mode. Although several paths exist, this novel algorithm ensures that the resulting path is optimal, through optimising the configuration trajectories.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to acknowledge the UK Space Agency, the Surrey Satellite Technology Ltd., UK and the Algerian Space Agency for funding this research.

References

- [1] G. Jorgensen, E. Bains, Srms history, evolution and lessons learned, in: AIAA SPACE 2011 Conference & Exposition, 2011, pp. 72–77, URL <http://dx.doi.org/10.2514/6.2011-7277>.
- [2] N.J. Currie, B. Peacock, International space station robotic systems operations-a human factors perspective, in: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, Vol. 46, 2002, pp. 26–30, URL <http://dx.doi.org/10.1177/154193120204600106>.
- [3] C.G. Henshaw, The darpa phoenix spacecraft servicing program: Overview and plans for risk reduction, in: Proc. International Symposium on Artificial Intelligence, Robotics and Automation in Space (I-SAIRAS), Montreal, 17 - 19 June, 2014.
- [4] S. Eckersley, C. Saunders, D. Gooding, M. Sweeting, C. Whiting, M. Ferris, J. Friend, L. Forward, G. Aglietti, A. Nanjangud, et al., In-orbit assembly of large spacecraft using small spacecraft and innovative technologies, in: Proc. 69th International Astronautical Congress (IAC), Bremen, Germany, 1-5 October, 2018.
- [5] A. Flores-Abad, O. Ma, K. Pham, S. Ulrich, A review of space robotics technologies for on-orbit servicing, Prog. Aerosp. Sci. 68 (2014) 1–26, URL <http://dx.doi.org/10.1016/j.paerosci.2014.03.002>.
- [6] A. Seddaoui, C. Saaj, Combined nonlinear h-infinity controller for a controlled floating space robot, J. Guid. Dyn. Control (JGDC) (2019) accessed April 21, 2019, URL <http://arc.aiaa.org/doi/abs/10.2514/1.G003811>.
- [7] S. Dubowsky, M.A. Torres, Path planning for space manipulators to minimize spacecraft attitude disturbances, in: Proc. IEEE International Conference on Robotics and Automation, Sacramento, 9-11 April, 1991, URL <http://dx.doi.org/10.1109/robot.1991.132005>.
- [8] P. Piersigilli, I. Sharf, A. Misra, Reactionless capture of a satellite by a two degree-of-freedom manipulator, Acta Astronaut. 66 (1–2) (2010) 183–192, URL <http://dx.doi.org/10.1016/j.actaastro.2009.05.015>.
- [9] K. Yoshida, K. Hashizume, S. Abiko, Zero reaction maneuver: Flight validation with ets-vii space robot and extension to kinematically redundant arm, in: Proc. IEEE International Conference on Robotics and Automation, Seoul, South Korea, 21-26 May, 2001, URL <http://dx.doi.org/10.1109/robot.2001.932590>.
- [10] T. Rybus, K. Seweryn, Manipulator trajectories during orbital servicing mission: numerical simulations and experiments on microgravity simulator, Prog. Flight Dyn. Guid. Navig. Control 10 (2018) 239–264, URL <http://dx.doi.org/10.1051/eucass/201810239>.
- [11] O.P. Agrawal, Y. Xu, On the global optimum path planning for redundant space manipulators, IEEE Trans. Syst. Man Cybern. 24 (9) (1994) 1306–1316, URL <http://dx.doi.org/10.1109/21.310507>.
- [12] W. Xu, D. Meng, H. Liu, X. Wang, B. Liang, Singularity-free trajectory planning of free-floating multiarm space robots for keeping the base inertially stabilized, IEEE Trans. Syst. Man Cybern.: Syst. (99) (2017) 1–14, URL <http://dx.doi.org/10.1109/tsmc.2017.2693232>.
- [13] M.W. Spong, S. Hutchinson, Robot modeling and control, 2006.
- [14] E. Papadopoulos, S. Dubowsky, Dynamic singularities in free-floating space manipulators, J. Dyn. Syst. Meas. Control 115 (1) (1993) 44–52, URL <http://dx.doi.org/10.1115/1.2897406>.
- [15] T. Rybus, Obstacle avoidance in space robotics: Review of major challenges and proposed solutions, Prog. Aerosp. Sci. 101 (2018) 31–48, URL <http://dx.doi.org/10.1016/j.paerosci.2018.07.001>.
- [16] X.-P. Wei, J.-X. Zhang, D.-S. Zhou, Q. Zhang, Optimal path planning for minimizing base disturbance of space robot, Int. J. Adv. Robot. Syst. 13 (2) (2016) 41, URL <http://dx.doi.org/10.5772/62126>.
- [17] M. Wang, J. Luo, U. Walter, Trajectory planning of free-floating space robot using particle swarm optimization (psa), Acta Astronaut. 112 (2015) 77–88, URL <http://dx.doi.org/10.1016/j.actaastro.2015.03.008>.
- [18] M. Wang, J. Luo, J. Yuan, U. Walter, Coordinated trajectory planning of dual-arm space robot using constrained particle swarm optimization, Acta Astronaut. 146 (2018) 259–272, URL <http://dx.doi.org/10.1016/j.actaastro.2018.03.012>.
- [19] E.G. Kaigom, T.J. Jung, J. Roßmann, Optimal motion planning of a space robot with base disturbance minimization, in: Proc. 11th Symposium on Advanced Space Technologies in Robotics and Automation, Noordwijk, 12-14 April, 2011.
- [20] L. Zhang, Q. Jia, G. Chen, H. Sun, Pre-impact trajectory planning for minimizing base attitude disturbance in space manipulator systems for a capture task, Chin. J. Aeronaut. 28 (4) (2015) 1199–1208, URL <http://dx.doi.org/10.1016/j.cja.2015.06.004>.
- [21] W. Xu, C. Li, B. Liang, Y. Liu, Y. Xu, The cartesian path planning of free-floating space robot using particle swarm optimization, Int. J. Adv. Robot. Syst. 5 (3) (2008) 27–37, URL <http://dx.doi.org/10.5772/5605>.
- [22] P. Huang, K. Chen, Y. Xu, Optimal path planning for minimizing disturbance of space robot, in: Proc. 9th International Conference on Control, Automation, Robotics and Vision, Singapore, 5-8 Dec, 2006, URL <http://dx.doi.org/10.1109/icarcv.2006.345055>.
- [23] Z. Chen, W. Zhou, Path planning for a space-based manipulator system based on quantum genetic algorithm, J. Robot. (2017) URL <http://dx.doi.org/10.1155/2017/3207950>.
- [24] M. Wang, J. Luo, J. Fang, J. Yuan, Optimal trajectory planning of free-floating space manipulator using differential evolution algorithm, Adv. Space Res. 61 (6) (2018) 1525–1536, URL <http://dx.doi.org/10.1016/j.asr.2018.01.011>.
- [25] R. Lampariello, S. Agrawal, G. Hirzinger, Optimal motion planning for free-flying robots, in: Proc. IEEE International Conference on Robotics and Automation, Taipei, 14-19 September, 2003, URL <http://dx.doi.org/10.1109/robot.2003.1242056>.
- [26] G. Misra, X. Bai, Optimal path planning for free-flying space manipulators via sequential convex programming, J. Guid. Control Dyn. 40 (11) (2017) 3019–3026, URL <http://dx.doi.org/10.2514/1.g002487>.
- [27] J. Virgili-Llop, C. Zagaris, R. Zappulla, A. Bradstreet, M. Romano, Laboratory experiments on the capture of a tumbling object by a spacecraft-manipulator system using a convex-programming-based guidance, in: Proc. AAS/AIAA Astrodynamics Specialist Conference, Washington, 20-24 August, 2017, URL <http://dx.doi.org/10.1016/j.ast.2017.09.043>.
- [28] E. Wilson, S.M. Rock, Neural-network control of a free-flying space robot, Trans. Soc. Model. Simul. Int. 65 (2) (1995) 103–115, URL <http://dx.doi.org/10.1177/003754979506500203>.
- [29] A. Seddaoui, C.M. Saaj, Collision-free optimal trajectory for a controlled floating space robot, in: Annual Conference Towards Autonomous Robotic Systems, Springer, 2019, pp. 248–260, URL http://dx.doi.org/10.1007/978-3-030-25332-5_22.
- [30] K. Deb, J. Sundar, Reference point based multi-objective optimization using evolutionary algorithms, in: Proc. 8th Annual Conference on Genetic and Evolutionary Computation, Washington, 08 - 12 July, 2006, URL <http://dx.doi.org/10.1145/1143997.1144112>.
- [31] A. Seddaoui, C.M. Saaj, S. Eckersley, Adaptive h infinity controller for precise manoeuvring of a space robot, in: Proc. IEEE International Conference on Robotics and Automation (ICRA), Montreal, 20-24 May, 2019, URL <http://dx.doi.org/10.1109/icra.2019.8794374>.
- [32] A. Seddaoui, C. Saaj, H-infinity control for a controlled floating robotic spacecraft, in: Proc. International Symposium on Artificial Intelligence, Robotics and Automation in Space (I-SAIRAS), Madrid, 4-6 June, 2018.
- [33] W. Zhang, X. Huang, X.-Z. Gao, H. Yin, A constrained multi-objective particle swarm optimization algorithm based on adaptive penalty and normalized non-dominated sorting, Int. J. Innovative Comput. Inf. Control 11 (6) (2015) 1835–1853.
- [34] Y. Huang, Y.S. Yong, R. Chiba, T. Arai, T. Ueyama, J. Ota, Kinematic control with singularity avoidance for teaching-playback robot manipulator system, IEEE Trans. Autom. Sci. Eng. 13 (2) (2016) 729–742, URL <http://dx.doi.org/10.1109/tase.2015.2392095>.
- [35] G. Chen, L. Zhang, Q. Jia, H. Sun, Singularity analysis of redundant space robot with the structure of canadarm2, Math. Probl. Eng. (2014) URL <http://dx.doi.org/10.1155/2014/735030>.
- [36] T. Yoshikawa, Dynamic manipulability of robot manipulators, Trans. Soc. Instrum. Control Eng. 21 (9) (1985) 970–975, URL <http://dx.doi.org/10.1109/robot.1985.1087277>.
- [37] A. Chehouri, R. Younes, J. Perron, A. Ilinca, A constraint-handling technique for genetic algorithms using a violation factor, J. Comput. Sci. Sci. Publ. 12 (2016) 350–362, URL <http://dx.doi.org/10.3844/jcsp.2016.350.362>.
- [38] R.G. Brown, Smoothing, Forecasting and Prediction of Discrete Time Series, Courier Corporation, 2004.