

Trajectory optimization and control of a free-floating two arms humanoid robot

Jose Luis Ramon¹, Ramon Calvo², Adrian Trujillo³ and Jorge Pomares⁴
University of Alicante, San Vicente del Raspeig, Alicante, 03690, Spain

and
Leonard Felicetti⁵
Cranfield University, Cranfield, MK43 0AL, United Kingdom

An optimization algorithm for planning the motion of a humanoid robot during extravehicular activities is presented in this paper. The algorithm can schedule and plan the movements of the two robotic arms to move the humanoid robot by using the handrails present outside the international space station. The optimization algorithm considers the eventual constraints imposed by the topology of the handrails and calculates the sequence of grasping and non-grasping phases needed to push and pull the robot along the handrails. A low-level controller is also developed and used to track the planned arms and end-effectors trajectories. Numerical simulations assess the applicability of the proposed strategy in three different typical operations that potentially can be performed in an extravehicular activity scenario.

I. Introduction

Extravehicular activities represent one of the most dangerous and challenging activities currently performed on the International Space Station (ISS) [1]. Astronauts are periodically called to venture outside the ISS to perform maintenance and upgrade tasks. To accomplish such operations, astronauts require specific training sessions before the mission, utilize complex safety and life support equipment during the activities and work for several hours in an unfriendly environment, eventually subjected to space debris and space radiation [2][3].

¹ Lecturer in Robotics, Department of Physics, Systems Engineering and Signal Theory.

² MSc Student, Department of Physics, Systems Engineering and Signal Theory.

³ MSc Student, Department of Physics, Systems Engineering and Signal Theory.

⁴ Professor in Robotics, Department of Physics, Systems Engineering and Signal Theory.

⁵ Lecturer in Space Engineering, School of Aerospace, Transport and Manufacturing, AIAA Member.

Future and under-development missions will require extensive use of on-orbit assembling and manufacturing to build, for example, new human exploration facilities, such as the Lunar Gateway [4][5] and large commercial infrastructures, such as the Vast Satcom Antennas [6]. The on-orbit assembling of such infrastructures will involve complex tasks with strict reliability, efficiency, and safety requirements. The utilization of astronauts performing some extravehicular activities is still considered a viable option, but it is challenged by the numerous technical and technological limitations [5]. The utilization of autonomous robots is indeed a preferable option, especially for tasks where operations are repetitive, structured and standardized. On the other hand, it is also evident that robots will operate with tools and in environments strongly characterized by human presence: screwdrivers, brackets and pliers, as well as handles and handrails, are made to be easily used by human operators or astronauts. For this reason, humanoid robots appear to be preferable over other kinds of robotic systems in such kinds of scenarios. During its long lifetime, the ISS hosted numerous humanoid robots. Robonaut 2, developed by NASA and General Motors, was the first humanoid robot on the ISS in 2011 [7]. The Jaxa's Kirobo was launched to the ISS in 2013, followed by the European Crew Interactive Mobile CompanioN (CIMON), developed by the German Aerospace Center (DLR) and Airbus and launched in 2018 [8], and the Russian Skybot F-850 "Fyodor", developed by Roscosmos and tested on the ISS in 2019 [9]. Moreover, other robotic astronaut assistants, such as Spheres and Astrobee, were or are currently utilized onboard the ISS, respectively [10][11][12]. None of these robots was tested in an extravehicular activity scenario, but future robotic missions, such NASA's Valkyrie, will have improved autonomy and will also be tested outside the ISS [13].

This paper assumes that extra-vehicle activities can and will be performed by humanoid robots soon and, for this reason, an investigation over possible control and path planning strategies appear justified. This paper proposes and investigates strategies that can be used to plan the motion and control humanoid robots in some elementary tasks that characterize extravehicular activities. The humanoid robot taken into account is a torso (body) with two arms and two grippers at their extremities. Specifically, this study addresses the problem of robot's motion on the complex system of handrails and handles that characterize the ISS. It is assumed that the humanoid robot is not equipped with any auxiliary thruster system, and therefore, it is supposed to crawl over the handrails and handles outside the ISS. Such a complex task has been divided into two elementary sub-tasks: motion planning and tracking. First, an optimization procedure is presented to plan and coordinate the robot's arms motions and graspers to achieve the

ISS's desired location using handrails. Then, a controller is proposed to guarantee that the robots' actuators can follow these previously generated trajectories.

The path planning problem is solved by developing a new approach that considers both the kinematics and dynamics of the robot moving among the handrails. Such an approach considers the robot's free-floating and frictionless conditions in space, especially when the handrails are not grasped, and the contact dynamics when the robot manipulates the handrails. The problem of motion planning has been previously addressed in other applications, e.g. locomotion of biped robots. However, the application to the space scenario seems to offer specific challenges that make the problem attractive. Within this field, we should mention previous works that use an optimization-based trajectory planning method for free-flying spacecrafts [14] and for free-flying space manipulators [15][16]. In these previous works, sequential convex programming is used to find a local kinematically feasible path. Other previous works about locomotion used simplified robot dynamics like the linear inverted pendulum model to generate robot motion plans, mainly for biped robot walking trajectory generation (see e.g. [17] and [18]). In order to avoid the limitations of this kind of model, centroidal dynamics or rigid body dynamics models were used within the optimization problem to generate dynamically feasible paths. For example, an optimization algorithm is proposed in [19] to generate the optimal sequence of movements of a biped robot to climb stairs, also using handrails. In [19] a method is proposed to find the feasible center of mass trajectories and contact forces for a humanoid robot. The grasping sequence is previously defined, and the trajectory and contact forces are generated from a predefined contact set. On the other hand, the algorithm proposed in this paper automatically generates the handrail's grasping sequences and calculates the eventual contact forces and trajectory followed by the center of mass. These approaches based on centroidal dynamics have been proved to be valid to generate complex motions in humanoid robots generation impressive and realistic motions [20] [21]. Other approaches use two-step approach: first, to solve the kinematic path planning problem and then to compensate for the eventual robot's unbalancement by correcting the pose of the robot by considering the centroid and contact dynamics (see e.g. [21] and [22]). The path planning problem for a climbing application of a two-arms robot is presented and solved in [23]. The adopted solution uses a Lagrangian formulation of the dynamics for the motion scheduling phase. Thanks to this approach, the robot can be maneuvered to accomplish vertical-up and tilted or horizontal climbing tasks. Another kind of methodologies uses optimization tools for generating goal-directed robot motions. Differential dynamic programming [24] tools or Linear-quadratic regulator (LQR) based methods such as Iterative or Sequential Linear

Quadratic Programming [25] have been used to directly solve robot path planning problems. In [26], the linear quadratic regulator and rapidly-exploring randomized trees (LQR-RRT*) are proposed for the motion planning and control trajectories for a robotic Astrobbee free-flyer. This method provides the on-orbit free-flyer with the ability for on-orbit manipulation, avoiding collisions for the on-orbit assembly of space structures.

In order to define the adequate approach for robot planning, it is necessary to determine the required behaviour and optimization criteria to accomplish a specific proposed task. First, the arm motions shall be automatically generated by the planner. Within the field of space robotics, we should mention works such as [27], where the motion planning of a dual-arm is performed by using deep reinforcement learning. A trigonometric spline function is used to plan the ascending spiral trajectory of a space manipulator in [28]. In [29] and [30], an optimal motion planning and control scheme is presented for a space manipulator to achieve momentum dissipation of a grasped tumbling satellite in minimum time or minimum energy. The motion planning problem of dual-arm space robot manipulators in the presence of external forces and moments acting on the space robot system is also addressed in [31]. A quantum genetic algorithm is used in [32] to solve the optimization problem to attain the optimized joints' trajectories in a similar scenario. A particle swarm optimization algorithm has been developed in [33] to address the motion planning problem and the minimization of base disturbances for six degrees of freedom space manipulator. In the presented paper, a trajectory optimization formulation is also presented but applied to determine a two-arm robotic system's motions that automatically determine the grasping positions, arms trajectories, grasping times, and body motion. A non-linear programming solver is used to generate complex trajectories to achieve a given desired location for the robot body using the arms motion grasping trajectories.

Optimization strategies have also been used to solve specific issues that characterize humanoid robots. A covariance matrix adaptation evolution strategy is proposed in [34] to plan the motions of a quadruped robot. In [35], a sequential linear quadratic programming method is applied into a model predictive control loop for actuating a humanoid robot. The approach described in this paper is based on [36]. Such a strategy allows for the definition of the motion with no previous knowledge of the eventual favorable contact points. The planner automatically defines the grasping points on the handrails of the ISS and the timings, the arm motions (avoiding collisions and taking the robot kinematics and dynamics into account), the 6 degrees of freedom torso motion and the required contact forces over handrails. To do this a new nonlinear programming formulation is proposed that takes into account the robot kinematics and dynamics. In addition, this paper proposes a controller for tracking the planned trajectories obtained

from the solution of the non-linear programming problem. The undesirable disturbances to the robot's body due the manipulator motion must be considered in the definition of these controllers [37]. Some authors propose the use of reaction wheels and to study the dynamic equations with the contribution of reaction wheels to the angular momentum [38][39]. In this paper, an optimal control approach is proposed for tracking the previously generated trajectories. This approach allows for tracking trajectories considering the optimization of the motor commands with respect to a specified metric.

The remaining part of the paper is divided into the following sections. Section 2 describes the system dynamics and the proposed simulation system. The motion planning problem and the nonlinear programming formulation used to solve the problem are described in Section 3. Section 4 develops and demonstrates the stability of the direct visual servoing controller for the tracking of the previously generated trajectories. Simulation results, showing the applicability and robustness of the proposed approach in selected test case scenarios, are described and commented in Section 5. Concluding remarks and future areas of development are presented in Section 6.

II. System architecture and dynamics

Figure 1 shows a three-dimensional representation of the proposed scenario. A humanoid robot is supposed to perform extravehicular activities (EVA) outside the ISS. The humanoid robot has two arms with seven degrees of freedom each and grippers at their end-effectors; the joint coordinates of both arms are denoted as $\mathbf{q}_1, \mathbf{q}_2 \in \mathbb{R}^n$, where $n = 7$. The head of the robot hosts a range camera, which can be used to determine the positions of the handrails and grasping points. Different handrails are located on the exterior of the ISS, as shown in Fig. 1. This paper presents the planning and control of the trajectories of the robot's arms to alternative grasp the handrails using both hands to achieve a desired location on the ISS. Figure 1.b also shows the robot's body coordinate frame $\mathbf{B}$, located at the torso center. The robot center of mass position and attitude coordinates (Euler angles) with respect to the inertial coordinate frame are denoted as $\mathbf{r}$ and $\boldsymbol{\theta}$. It is assumed that the robot is close enough to the ISS so that the arms can achieve the handrails, and there is no need of using any other form of locomotion.

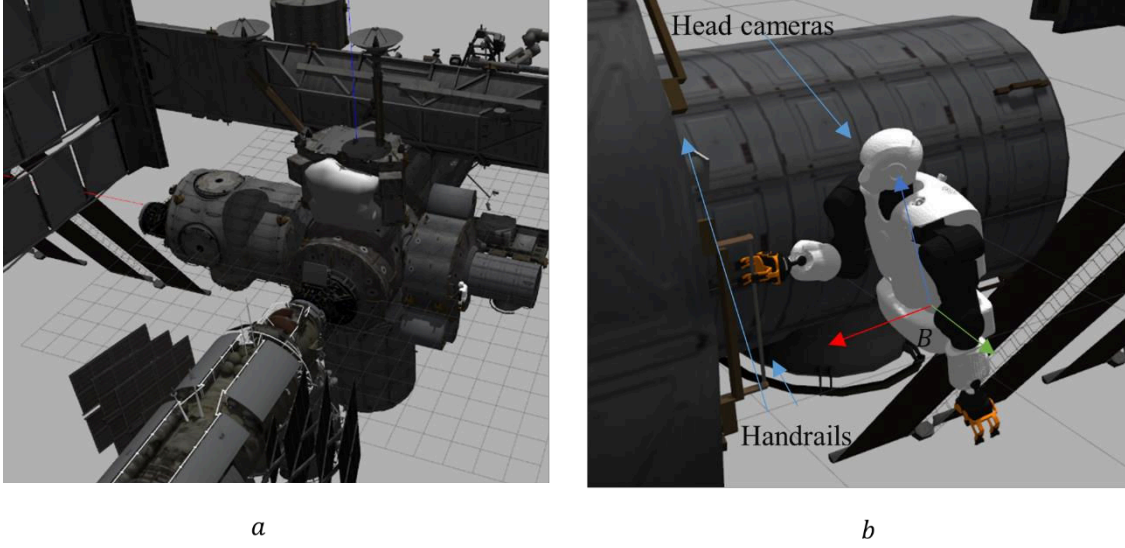


Fig. 1 Simulation system. a) ISS representation. b) Humanoid robot grasping an ISS handrail.

The full kinematics of the robotic system is defined by the vector $\epsilon = [r^T, \theta^T, q_1^T, q_2^T]^T$, where θ contains the yaw, pitch, roll Euler angles representing the orientation of frame B with respect the inertial coordinate frame. Additionally, $v_i^b = [\dot{r}_i^b, \omega_i^b]^T \in \mathbb{R}^6$ denotes the twist of the manipulator i ($i = 1, 2$) end-effector coordinate frame with respect the robot's body coordinate frame, B ; $\dot{r}_i^b \in \mathbb{R}^3$ and $\omega_i^b \in \mathbb{R}^3$ are its linear and angular velocity components respectively. The mapping between v_i^b and the time derivative $\dot{q}_i$ of the joint positions of the manipulator i is given by:

$$v_i^b = J_i^b(q_i) \dot{q}_i \quad (1)$$

where $J_i^b \in \mathbb{R}^{6 \times n}$ is the so-called geometric Jacobian of the robot manipulator. In order to control both robot arms end-effectors, the following vector is considered $v = [v_b^T, v_1^T, v_2^T]^T$, where v_i are the corresponding twist collecting the absolute linear and angular velocities of each end effector frame with respect the inertial frame, and v_b is the linear and angular velocities of the robot body. The corresponding differential equation is:

$$v = J(\epsilon) \dot{\epsilon} \quad (2)$$

where $J = \text{diag}(I_3, T_b, \bar{R}_b J_1^b, \bar{R}_b J_2^b) \in \mathbb{R}^{18 \times (6+2n)}$ with $\bar{R}_b = \text{diag}(R_b, R_b)$, R_b is the rotation matrix between the base and inertial frame, I_α denotes the $\alpha \times \alpha$ identity matrix, and T_b is the 3×3 transformation matrix between the time derivative of θ and the correspondent ω :

$$\boldsymbol{\omega} = \mathbf{T}_b(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} = \begin{bmatrix} \cos(\theta_y)\cos(\theta_z) & -\sin(\theta_z) & 0 \\ \cos(\theta_y)\sin(\theta_z) & \cos(\theta_z) & 0 \\ -\sin(\phi_y) & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} \quad (3)$$

$$\dot{\boldsymbol{\omega}} = \dot{\mathbf{T}}_b(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{T}_b(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} \quad (4)$$

With respect the system dynamics, this information provides a relationship between the acceleration and forces and torques in both the robot's body and manipulators. More specifically, the system dynamics relates the linear and angular accelerations of the robot body $\dot{\mathbf{v}}_b = [\ddot{\mathbf{r}}^T, \dot{\boldsymbol{\omega}}^T]^T \in \mathbb{R}^6$ expressed in the Inertial coordinate frame, the joint accelerations of each manipulator, $\ddot{\mathbf{q}}_i$, with the forces and torques exerted on the robot body expressed in the Inertial frame, and the torques applied on each robot manipulator joints, $\boldsymbol{\tau}_i \in \mathbb{R}^n$. When only a free-floating robot manipulator is considered, the free-floating dynamics is described in detail in [42]. In our case, two robot manipulators are considered and both manipulators' dynamics can be combined in the following expression:

$$\mathbf{u} - \mathbf{u}_e = \begin{bmatrix} \mathbf{M}_{rr} & \mathbf{M}_{r1} & \mathbf{M}_{r2} \\ \mathbf{M}_{r1}^T & \mathbf{M}_{11} & 0 \\ \mathbf{M}_{r2}^T & 0 & \mathbf{M}_{22} \end{bmatrix} \ddot{\mathbf{c}} + \begin{bmatrix} \mathbf{c}_r \\ \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix} \quad (5)$$

where $\mathbf{u}$ is the input vector and $\mathbf{u}_e = \mathbf{J}^T \mathbf{h}_e$ shapes the effects of generalized external wrench $\mathbf{h}_e = [\mathbf{f}_b^T, \mathbf{f}_1^T, \mathbf{f}_2^T]^T$ at joint level, with $\mathbf{f}_i$ the external wrench action of each arm (see Fig. 2), and $\mathbf{f}_b$ the ones applied to the robot base. Additionally, $\mathbf{M}_{rr} \in \mathbb{R}^{6 \times 6}$ is the inertia matrix of the spacecraft, $\mathbf{M}_{r1} \in \mathbb{R}^{6 \times n}$ is the coupled inertia matrix of the robot body and the left robot arm, $\mathbf{M}_{11} \in \mathbb{R}^{n \times n}$ is the inertia matrix of the left arm, $\mathbf{M}_{r2} \in \mathbb{R}^{6 \times n}$ is the coupled inertia matrix of the robot body and the right arm, $\mathbf{M}_{22} \in \mathbb{R}^{n \times n}$ is the inertia matrix of the right arm; $\mathbf{c}_r$, $\mathbf{c}_1$, and $\mathbf{c}_2 \in \mathbb{R}^6$ are a velocity/displacement-dependent, nonlinear terms for the body, left and right arm, respectively.

Equation (5) represents the free-floating dynamics of the proposed robotic system. A simplified robot model is used to consider the effects of the contact forces on the linear and angular motion of the system. Specifically, rigid body dynamics are used to model the robot's torso motion due to the end-effector contact forces:

$$m\ddot{\mathbf{r}} = \sum_{i=1}^2 \mathbf{f}_i(t) \quad (6)$$

$$\mathbf{M}_I \dot{\boldsymbol{\omega}}(t) + \boldsymbol{\omega}(t) \times \mathbf{M}_I \boldsymbol{\omega}(t) = \sum_{i=1}^2 \mathbf{f}_i(t) \times (\mathbf{r}(t) - \mathbf{p}_i(t))$$

The 6 degrees of freedom base motion is represented by the linear center of mass position $\mathbf{r}(t) \in \mathbb{R}^3$, and $\boldsymbol{\omega}(t)$ represents the angular velocity that can be computed from the Euler angles $\boldsymbol{\theta}(t)$ and the corresponding time

derivatives $\dot{\theta}(t)$ (see Eq. (3) and (4)). The mass of the entire robot is given by m and M_I is a constant rotational moment of inertia calculated from the nominal robot configuration. Figure 2.a shows the reaction forces $f_i(t)$ acting on the end-effectors. The position of each of the end-effectors is denoted as $p_i(t)$ and the distance between the torso centre of mass and end-effectors $r(t) - p_i(t)$ is used for the calculus of the torque due to the reaction forces.

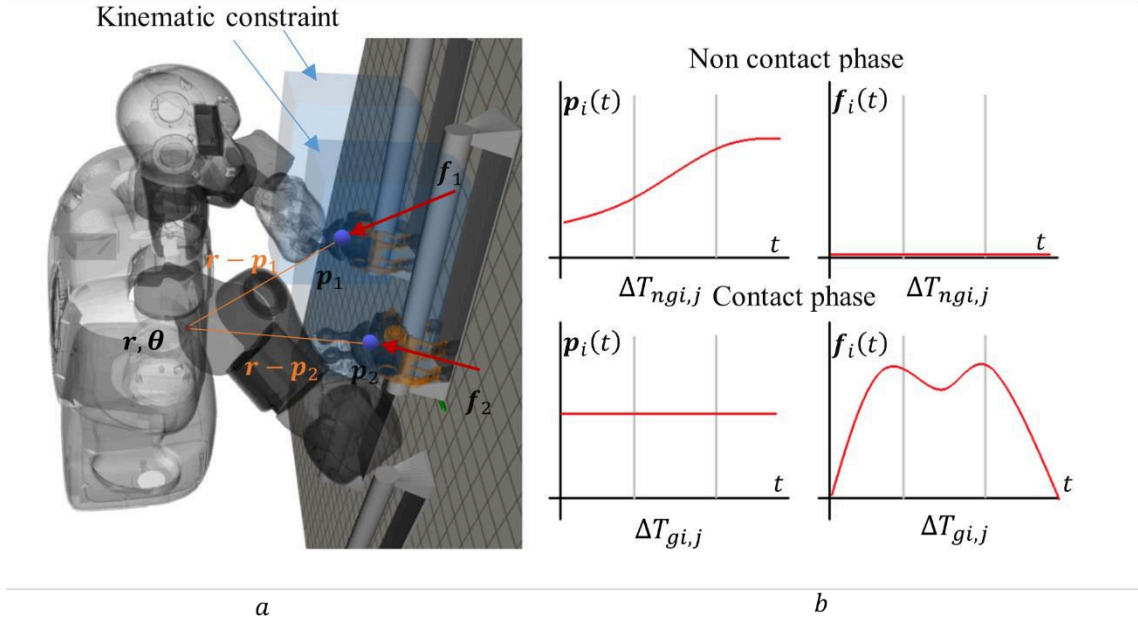


Fig. 2 a) 3D representation of the robot, contact forces and kinematic constraint. b) Example of end-effector position and force during non-contact and contact phases.

III. Trajectory optimization

This section describes the optimization problem for the generation of the arms movements on the handrails of the ISS to move the humanoid robot from an initial position r_0 to a final desired one r_d .

A. Optimization objective.

The trajectory optimization aims to generate trajectories for the linear position of the center of mass, $r(t)$, its attitude, $\theta(t)$ and the position and forces exerted by each arm $p_i(t)$ and $f_i(t)$, respectively. This trajectory is generated given only the desired goal position for the robot's body, the number of grasping positions and the duration of the maneuver, T . The number of grasping positions that each of the hands, i , will use during the trajectory are denoted with N . During the trajectory optimization, each hand, i , can perform two kinds of operations:

to grasp or not to grasp the handrails. The duration of the grasping and non-grasping phases are defined as $\Delta T_{gi,j}$ and $\Delta T_{ngi,j}$ respectively, where $j = 1 \dots N$. Thus:

$$T = \Delta T_{gi,1} + \Delta T_{ngi,1} + \dots + \Delta T_{gi,N} + \Delta T_{ngi,N}. \quad (7)$$

The task of achieving a given robot body location, $\mathbf{r}_d$, can be easily and intuitively predefined if the number of phases of grasping positions to achieve the desired position is low. However, when the number of possible grasping points increases and the final position is not close with respect to the initial one, it becomes difficult to determine the grasping positions required to complete the task. In such cases, a trajectory optimization problem is proposed that can be defined by using the following formulation:

Find $\mathbf{r}(t)$ $\boldsymbol{\theta}(t)$

and for every arm i :

$$\Delta T_{gi,1}, \Delta T_{ngi,1}, \dots, \Delta T_{gi,N}, \Delta T_{ngi,N} \quad (8)$$

$$\mathbf{p}_i(t, \Delta T_{gi,j}, \Delta T_{ngi,j}), j = 1 \dots N$$

$$\mathbf{f}_i(t, \Delta T_{gi,j}, \Delta T_{ngi,j}), j = 1 \dots N$$

so that the following cost function is minimized:

$$\Phi = \int_0^T \sum_1^2 \left[\sigma_{i1} (f_i^x(t))^2 + \sigma_{i2} (f_i^y(t))^2 + \sigma_{i3} (f_i^z(t))^2 \right] + \sigma_4 (\dot{r}^x(t))^2 + \sigma_5 (\dot{r}^y(t))^2 + \sigma_6 (\dot{r}^z(t))^2 dt \quad (9)$$

where $f_i^x(t)$, $f_i^y(t)$, and $f_i^z(t)$ are the end-effector contact forces in x, y, and z directions respectively; $\dot{r}^x(t)$, $\dot{r}^y(t)$, and $\dot{r}^z(t)$ are the robot torso velocities in each direction. Additionally, σ_i are weights that can be adjusted.

This optimization problem is subjected to the constraints described in Section 3.2.

The functions mapping the behavior of the components of the end-effector position $\mathbf{p}_i(t)$ and contact forces $\mathbf{f}_i(t)$ are assumed to be cubic polynomials. More specifically, three cubic polynomials are used to arbitrary generate end-effector trajectories in the non-contact phase. In the same way, we use three cubic polynomials to represent the contact forces during a contact phase. From this cubic representation the optimizer modifies the shape of these functions to represent the end-effector trajectories and the contact forces. Additionally, we should guarantee continuous derivatives at the end points while capturing typical force and position profiles. Therefore, these cubic polynomials present the following form:

$$\mathbf{p}_i(t) = \begin{bmatrix} p_i^x(t) \\ p_i^y(t) \\ p_i^z(t) \end{bmatrix} = \begin{bmatrix} a_0^x + a_1^x t + a_2^x t^2 + a_3^x t^3 \\ a_0^y + a_1^y t + a_2^y t^2 + a_3^y t^3 \\ a_0^z + a_1^z t + a_2^z t^2 + a_3^z t^3 \end{bmatrix} \quad (10)$$

$$\mathbf{f}_i(t) = \begin{bmatrix} f_i^x(t) \\ f_i^y(t) \\ f_i^z(t) \end{bmatrix} = \begin{bmatrix} b_0^x + b_1^x t + b_2^x t^2 + b_3^x t^3 \\ b_0^y + b_1^y t + b_2^y t^2 + b_3^y t^3 \\ b_0^z + b_1^z t + b_2^z t^2 + b_3^z t^3 \end{bmatrix} \quad (11)$$

where the coefficients a_k^x, a_k^y and a_k^z in Eq.(10), as well as the coefficients b_k^x, b_k^y and b_k^z in Eq.(11), with $k = 0, 1, \dots, 3$, are selected in such way to map the behaviours shown in Fig. 2.b, for each of the three time intervals of $\Delta T_{gi,j}/3$ or $\Delta T_{ngi,j}/3$ duration, whose the grasping and non-grasping phase can be divided. Specifically, if the arm is in a non-grasping phase, from Fig. 2.b it can be noted that the position of the end-effector $\mathbf{p}_i(t)$ will move from the initial handrail's position to the destination position of the selected handrail in $\Delta T_{ngi,j}$ time. Obviously, the contact forces will be null, $\mathbf{f}_i(t) = \mathbf{0}$, during this phase. On the other hand, when the robot is grasping a handrail, the contact force will assume a non-null value, $\mathbf{f}_i(t) \neq \mathbf{0}$, while the position $\mathbf{p}_i(t)$ will be maintained constant for the entire time interval $\Delta T_{gi,j}$.

The selection of the coefficients in Eqs. (10) and (11), for each of the time intervals $\Delta T_{gi,j}$ and $\Delta T_{ngi,j}$, will be obtained from the solution of the optimization problem in Eqs. (8) and (9), also including the constraints defined in the following subsection.

B. Trajectory optimization constraints

1. Kinematic constraints

The kinematic constraints guarantee that arm motions comply with the workspace and joints' physical limitations. The joints' limitations are accounted by using the following conditions:

$$\mathbf{q}_{i,min} < \mathbf{q}_i(t) < \mathbf{q}_{i,max} \quad (12)$$

where $\mathbf{q}_{i,min}$ and $\mathbf{q}_{i,max}$ are the lower and upper bounds of the joints variable working ranges of each of the two arms. In order to keep the formulation in the cartesian space, Eq. (12) needs to be substituted by equivalent conditions on each end-effectors' reachable workspaces. For example, such conditions are illustrated by the blue boxes in Fig. 2.a: if the arms end-effectors are inside these blue boxes, the joint limits are not violated. Thus, the bounds on the end-effectors' positions $\mathbf{p}_i(t)$ are set as follows:

$$|\mathbf{R}(\boldsymbol{\theta})[\mathbf{p}_i(t) - \mathbf{r}(t)] - \bar{\mathbf{p}}_i| < \varepsilon \quad (13)$$

where $\mathbf{R}(\boldsymbol{\theta})$ is the rotation matrix representing the attitude of the humanoid robot with respect the inertial reference frame, and $\bar{\mathbf{p}}_i$ is the nominal position of the arm i -th end-effector relative to the body frame. Therefore, the arm's workspace is approximated by a cube of edge length 2ε centered, for each hand, at position $\bar{\mathbf{p}}_i$.

2. Force and position constraints

As indicated in Eq. (8), the optimization algorithm generates the trajectories for the robot end-effectors and the time duration for the contact and non-contact phases: $\Delta T_{gi,1}, \Delta T_{ngi,1}, \dots, \Delta T_{gi,N}, \Delta T_{ngi,N}$. By defining different values of $\Delta T_{gi,j}$ and $\Delta T_{ngi,j}$ for each arm, different situations can be obtained when both arms are grasping, only one arm is in contact or even no hand is grasping any handrail.

- a) Arm is grasping. When the arm i is in a grasping phase, several constraints should be included to guarantee the robot-end position $\mathbf{p}_i(t)$ is maintained constant. It is assumed that the hand does not slip in this phase, and the velocity of the hand motion during the grasping phase is:

$$\dot{\mathbf{p}}_i(t) = 0, t \in \Delta T_{gi,j} \quad (14)$$

Moreover, the value of the contact forces $\mathbf{f}_i(t)$ should be estimated from Eq. (11), obtaining the three polynomials for duration $\Delta T_{gi,j}/3$ as indicated in Section 3.1.

In this paper, we supposed that a height map function, $h_w(x, y)$, is known about the workspace. This function provides information about the depth or z coordinate of the workspace from the x and y coordinates. This height map is supposed to be known in advance; however, in a real application it can be obtained by using Time-of-Flight 3D cameras or stereo cameras.

Another constraint is due to the position of the handrails. In order to formulate the handrail constraints in terms of mathematical equations, additional information is needed. The height of each handrail can be retrieved from a pre-loaded height map of the scenario $h_w(x, y)$ alongside with the normal $\mathbf{n}(x, y)$, and tangent $\mathbf{t}(x, y)$ directions of the surfaces composing the scenario.

The optimizer selects the best grasping positions, $\mathbf{p}_i = (p_i^x, p_i^y, p_i^z)$ based on the reachable points of the handrails. The position and orientation of the handrails are defined by its endpoints $(x_i, y_i)_k, (x_f, y_f)_k$ where

$k = 1 \dots \mathcal{r}$ being $\mathcal{r}$ the number of handrails. From these points, an algebraic constraint $\mathcal{R}(p_i^x, p_i^y) = 0$ is generated to guarantee that the coordinates p_i^x, p_i^y of the grasping position are lying on the handrail. Consequently, in order to guarantee a safe grasping, the desired p_i^z component of the end-effector position is selected based on the height-map by using the following relation:

$$p_i^z = h_w(p_i^x, p_i^y) + d, \quad (15)$$

where d is a fixed safety distance to be added to the height map to allow a safe grasping by using the gripper mounted on the end-effector.

Another constraint is given by the friction exerted during the contact of the end effector with the handrail. The gripper can exert a force along the normal to the surface but also along the tangent plane. Different friction models could be used, but Coulomb's law is the most commonly used [41]. Coulomb's law states that no relative motion between the robot end-effector and the handrail is allowed as long as the tangential component of the force, f_t , satisfies the following inequality:

$$f_t < \mu f_n \quad (16)$$

where f_n is the normal component and μ is the static friction coefficient characterising the coupling between the gripper and the handrail. According to this model, Eq. (16) defines the largest tangential force component that can be applied. Often, the tangential component f_t is expressed in terms of its two projections f_{t1} and f_{t2} along the axis defined on the tangent plane, as shown in Fig. 3. In this case, Eq. (16) is becomes:

$$\sqrt{f_{t1}^2 + f_{t2}^2} < \mu f_n \quad (17)$$

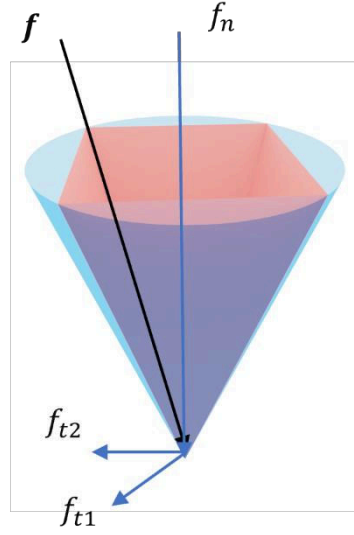


Fig. 3 Friction cone and inscribed pyramid

Thus, if the map of the normal directions of the scenario $\mathbf{n}(p_i^x, p_i^y)$ is known, the product $\mathbf{f}_i^T \mathbf{n}(p_i^x, p_i^y)$ gives the orthogonal component of the reaction forces f_n exerted by the end-effector. Thus, the following constraints are obtained by applying Eq. (17) [41]:

$$f_{t1} < \mu \mathbf{f}_i^T \mathbf{n}(p_i^x, p_i^y), \quad f_{t2} < \mu \mathbf{f}_i^T \mathbf{n}(p_i^x, p_i^y) \quad (18)$$

- b) Arm is not grasping. In this case, constraints need to be imposed to guarantee the contact forces $\mathbf{f}_i(t)$ are set to zero:

$$\mathbf{f}_i(t) = 0, t \in \Delta T_{ngi,j} \quad (19)$$

so that the optimizer can calculate the optimal coefficients a_k^x, a_k^y and a_k^z , with $k = 0, 1, \dots, 3$ in Eq.(10).

The constraints defined in this section are summarized in Table 1.

Additional constraints can be included, for example, to avoid a given obstacle in the workspace or to maintain the robot body at a constant height. Keeping the robot's body at constant height restricts the range of achievable motions, especially on some handrails configuration where the desired location can only be reached when also tilting the robot's body. In any case, a safety distance can be guaranteed by introducing the following restriction:

$$r^z - h_w(r^x, r^y) > sd \quad (20)$$

where sd is a safety distance that guarantees that no collisions with the surface will occur.

Table 1 Summary of the constraints of the optimization problem

Description	Constraint
Initial body position and attitude	$\mathbf{r}(t = 0) = \mathbf{r}_0, \boldsymbol{\theta}(t = 0) = \boldsymbol{\theta}_0$
Final body position and attitude	$\mathbf{r}(t = T) = \mathbf{r}_t, \boldsymbol{\theta}(t = T) = \boldsymbol{\theta}_t$
Duration of the trajectory	For each manipulator i: $\sum_{k=1}^N \Delta T_{gi,k} + \Delta T_{ngi,k} = T$
System dynamics constraint	$[\dot{\mathbf{r}}(t), \ddot{\boldsymbol{\theta}}(t)]^T = \mathbf{F}(\mathbf{r}(t), \mathbf{p}_i(t), \mathbf{f}_i(t))$
System kinematics constraint	For each manipulator i: $ \mathbf{R}(\boldsymbol{\theta})[\mathbf{p}_i(t) - \mathbf{r}(t)] - \bar{\mathbf{p}}_i < \varepsilon$
The robot-end does not slip	If gripper i is in contact with the workspace: $\dot{\mathbf{p}}_i(t) = \mathbf{0}$
Robot-end is in the grasping position	$\mathcal{R}(p_i^x, p_i^y) = 0, p_i^z = h_w(p_i^x, p_i^y) + d$
Friction pyramid constraint	$\mathbf{f}_{t1} < \mu \mathbf{f}_i^T \mathbf{n}(p_i^x, p_i^y), \mathbf{f}_{t2} < \mu \mathbf{f}_i^T \mathbf{n}(p_i^x, p_i^y)$
The robot does not exert forces	If gripper i is not in contact with the workspace: $\mathbf{f}_i(t) = \mathbf{0}$

IV. Tracking trajectories

In this section, the controller proposed for tracking the trajectories obtained in Section 3 is derived. Therefore, the controller allows the tracking of the planned trajectories for the phases the robot is not attached to the handrails. In these phases, the robot trajectory is given, for each arm i , as $\mathbf{p}_i(t)$, as shown in Table 1. When the robot does not interact with the workspace, Eq. (5) can be rewritten in the following form:

$$\mathbf{M}_{11}^* \ddot{\mathbf{q}}_1 + \mathbf{C}_1^* = \boldsymbol{\tau}_1 \quad (21)$$

$$\mathbf{M}_{22}^* \ddot{\mathbf{q}}_2 + \mathbf{C}_2^* = \boldsymbol{\tau}_2 \quad (22)$$

Where $\boldsymbol{\tau}_i \in \mathbb{R}^n$ is the applied joint torque on the corresponding arm ($i = 1, 2$). $\mathbf{M}_{ii}^* \in \mathbb{R}^{n \times n}$ is the generalized inertia matrix and $\mathbf{C}_i^* \in \mathbb{R}^n$ is the generalized Coriolis and centrifugal vectors for each arm i , defined explicitly as:

$$\mathbf{M}_{ii}^* = \mathbf{M}_{ii} - \mathbf{M}_{ri}^T \mathbf{M}_{rr}^{-1} \mathbf{M}_{ri} \quad (23)$$

$$\mathbf{C}_i^* = \mathbf{c}_i - \mathbf{M}_{ri}^T \mathbf{M}_{rr}^{-1} \mathbf{c}_r \quad (24)$$

The linear and angular momenta of the system $[\boldsymbol{\ell}^T, \Psi^T]^T \in \mathbb{R}^6$ are defined as:

$$\begin{bmatrix} \boldsymbol{\ell} \\ \Psi \end{bmatrix} = \mathbf{M}_{rr} \mathbf{v}_b + \mathbf{M}_{ri} \dot{\mathbf{q}}_i \quad (25)$$

$\mathbf{v}_b = [\dot{\mathbf{r}}^T, \dot{\boldsymbol{\omega}}^T]^T \in \mathbb{R}^6$ denotes the linear and angular velocities of the robot body expressed in the inertial coordinate frame, and $\dot{\mathbf{q}}_i \in \mathbb{R}^n$ represents joint speeds of the i -th arm. The relationship between the joint speeds and the corresponding end-effector's absolute linear and angular velocities can be expressed through differential kinematics:

$$\dot{\mathbf{p}}_i = \mathbf{J}_i \dot{\mathbf{q}}_i + \mathbf{J}_b \mathbf{v}_b \quad (26)$$

where $\dot{\mathbf{p}}_i \in \mathbb{R}^6$ is the linear and angular velocity of the manipulator end-effector in the inertial frame, $\mathbf{J}_i \in \mathbb{R}^{6 \times n}$ is the manipulator Jacobian matrix, and $\mathbf{J}_b \in \mathbb{R}^{6 \times 6}$ is the Jacobian matrix for the robot body. Combining Eq.(26) with Eq.(25) yields an equation that directly relates the joint speeds and end-effector motion of the robot manipulator:

$$\dot{\mathbf{p}}_i = \mathbf{J}_{gi} \dot{\mathbf{q}}_i + \mathbf{J}_b \mathbf{M}_{rr}^{-1} \begin{bmatrix} \boldsymbol{\ell} \\ \boldsymbol{\psi} \end{bmatrix} \quad (27)$$

$$\mathbf{J}_{gi} = \mathbf{J}_i - \mathbf{J}_b \mathbf{M}_{rr}^{-1} \mathbf{M}_{ri} \quad (28)$$

where $\mathbf{J}_{gi}$ is the Generalised Jacobian Matrix for the manipulator i . The second derivative of Eq. (27) can be expressed as:

$$\ddot{\mathbf{p}}_i = \mathbf{J}_{gi} \ddot{\mathbf{q}}_i + \dot{\mathbf{J}}_{gi} \dot{\mathbf{q}}_i + \dot{\mathbf{v}}_{ge} \quad (29)$$

where $\mathbf{v}_{ge} = \mathbf{J}_b \mathbf{M}_{rr}^{-1} \begin{bmatrix} \boldsymbol{\ell} \\ \boldsymbol{\psi} \end{bmatrix}$. More details about this last term and his computation can be seen in [42].

The task constraint is defined from the planned trajectory of both arms ($\mathbf{p}_{di}, \dot{\mathbf{p}}_{di}, \ddot{\mathbf{p}}_{di}$):

$$(\ddot{\mathbf{p}}_{di} - \ddot{\mathbf{p}}_i) + \mathbf{K}_d(\dot{\mathbf{p}}_{di} - \dot{\mathbf{p}}_i) + \mathbf{K}_p(\mathbf{p}_{di} - \mathbf{p}_i) = 0 \quad (30)$$

where $\mathbf{K}_p$ and $\mathbf{K}_d$ are proportional and derivative gain matrices, respectively. Equation (30) can be reformulated as follows:

$$\ddot{\mathbf{p}}_i = \ddot{\mathbf{p}}_{di} + \mathbf{K}_d(\dot{\mathbf{p}}_{di} - \dot{\mathbf{p}}_i) + \mathbf{K}_p(\mathbf{p}_{di} - \mathbf{p}_i) \quad (31)$$

and, by substituting Eq.(29) into Eq.(31), we can obtain:

$$\ddot{\mathbf{p}}_{di} + \mathbf{K}_d(\dot{\mathbf{p}}_{di} - \dot{\mathbf{p}}_i) + \mathbf{K}_p(\mathbf{p}_{di} - \mathbf{p}_i) = \mathbf{J}_{gi} \ddot{\mathbf{q}}_i + \dot{\mathbf{J}}_{gi} \dot{\mathbf{q}}_i + \dot{\mathbf{v}}_{ge} \quad (32)$$

Equation (32) allows for representing the tracking task in the form $\mathbf{A}(\mathbf{q}_i, \dot{\mathbf{q}}_i, t) \ddot{\mathbf{q}}_i = \mathbf{b}(\mathbf{q}_i, \dot{\mathbf{q}}_i, t)$, where:

$$\mathbf{A} = \mathbf{J}_{gi} \quad (33)$$

$$\mathbf{b} = \ddot{\mathbf{p}}_{di} + \mathbf{K}_d(\dot{\mathbf{p}}_{di} - \dot{\mathbf{p}}_i) + \mathbf{K}_p(\mathbf{p}_{di} - \mathbf{p}_i) - \dot{\mathbf{J}}_{gi} \dot{\mathbf{q}}_i - \dot{\mathbf{v}}_{ge} \quad (34)$$

Therefore, the tracking task can be achieved using a computed torque-based controller. To do this, we consider the following expression for the joint acceleration obtained from (29):

$$\ddot{\mathbf{q}}_i = \mathbf{J}_{gi}^+(\ddot{\mathbf{p}}_i - \dot{\mathbf{J}}_{gi}\dot{\mathbf{q}}_i - \dot{\mathbf{v}}_{ge}) \quad (35)$$

being $\ddot{\mathbf{p}}_i$ the reference Cartesian acceleration which is given by (31):

$$\ddot{\mathbf{p}}_i = \ddot{\mathbf{p}}_{di} + \mathbf{K}_d\dot{\mathbf{e}}_i + \mathbf{K}_p\mathbf{e}_i \quad (36)$$

where $\mathbf{e}_i = \mathbf{p}_{di} - \mathbf{p}_i$. The final control law with null-space stabilization can be obtained by replacing in (21) ($i = 1$) or (22) ($i = 2$) the joint acceleration obtained in (35):

$$\boldsymbol{\tau}_i = \mathbf{M}_{ii}^*\mathbf{J}_{gi}^+(\ddot{\mathbf{p}}_{di} + \mathbf{K}_d\dot{\mathbf{e}}_i + \mathbf{K}_p\mathbf{e}_i - \dot{\mathbf{J}}_{gi}\dot{\mathbf{q}}_i - \dot{\mathbf{v}}_{ge}) + \mathbf{M}_{ii}^*(\mathbf{I} - \mathbf{J}_{gi}^+\mathbf{J}_{gi})\mathbf{M}_{ii}^{*-1}\mathbf{v}_i + \mathbf{C}_i^* \quad (37)$$

where $\mathbf{v}_i$ is considered as a proportional derivative controller in joint space which pulls the joints, $\mathbf{q}_i$, toward a given configuration, $\mathbf{q}_{si}$:

$$\mathbf{v}_i = \mathbf{M}_{ii}^*(\mathbf{K}_{ps}(\mathbf{q}_{si} - \mathbf{q}_i) - \mathbf{K}_{ds}\dot{\mathbf{q}}_i) \quad (38)$$

Additionally, we can apply the optimal framework developed in [40] to derive a set of optimized controllers. This framework minimizes a cost function $\boldsymbol{\tau}_i^T \mathbf{W} \boldsymbol{\tau}_i$, where $\mathbf{W}$ is a weight matrix, for a robot with the dynamics as in Eq. (21) or Eq. (22). This optimal framework also considers a set of constraints that represent the task to be developed. The time derivate of these constraints is represented by the equation $\mathbf{A}(\mathbf{q}_i, \dot{\mathbf{q}}_i, t)\ddot{\mathbf{q}}_i = \mathbf{b}(\mathbf{q}_i, \dot{\mathbf{q}}_i, t)$ as it is shown in Eq. (33) and Eq. (34). By defining $\mathbf{z} = \mathbf{W}^{1/2}\boldsymbol{\tau}_i = \mathbf{W}^{1/2}(\mathbf{M}_{ii}^*\ddot{\mathbf{q}}_i + \mathbf{C}_i^*)$, it is possible to derive the joint accelerations $\ddot{\mathbf{q}}_i = \mathbf{M}_{ii}^{*-1}(\mathbf{W}^{-1/2}\mathbf{z} - \mathbf{C}_i^*)$. Taking into account the system constraints:

$$\mathbf{A}\mathbf{M}_{ii}^{*-1}\mathbf{W}^{-1/2}\mathbf{z} = \mathbf{b} + \mathbf{A}\mathbf{M}_{ii}^{*-1}\mathbf{C}_i^* \quad (39)$$

The vector $\mathbf{z}$ which minimizes $\mathcal{Q}(t) = \mathbf{z}^T \mathbf{z}$ while fulfilling Eq. (39) is given by $\mathbf{z} = (\mathbf{A}\mathbf{M}_{ii}^{*-1}\mathbf{W}^{-1/2})^+(\mathbf{b} + \mathbf{A}\mathbf{M}_{ii}^{*-1}\mathbf{C}_i^*)$, and the joint torque is given by $\boldsymbol{\tau}_i = \mathbf{W}^{-1/2}\mathbf{z}$. To conclude, the control law that minimizes $\mathcal{Q}(t)$ based on the dynamics model expressed in in Eq. (21) or Eq. (22), while performing the task described in Eq. (33) and Eq. (34), is given by:

$$\boldsymbol{\tau}_i = \mathbf{W}^{-1/2}(\mathbf{A}(\mathbf{M}_{ii}^*)^{-1}\mathbf{W}^{-1/2})^+(\mathbf{b} + \mathbf{A}(\mathbf{M}_{ii}^*)^{-1}\mathbf{C}_i^*) \quad (40)$$

Please note that the matrix $\mathbf{W}$ is an important factor in the control law determining how the control efforts are distributed over the joints. Two controllers, with different dynamics properties, can be indeed derived:

$$\begin{aligned} \boldsymbol{\tau}_i = & \mathbf{J}_{gi}^T(\mathbf{J}_{gi}\mathbf{M}_{ii}^{*-1}\mathbf{J}_{gi}^T)^{-1}(\ddot{\mathbf{p}}_{di} + \mathbf{K}_d\dot{\mathbf{e}}_i + \mathbf{K}_p\mathbf{e}_i - \dot{\mathbf{J}}_{gi}\dot{\mathbf{q}}_i - \dot{\mathbf{v}}_{ge}) \\ & + \mathbf{M}_{ii}^*(\mathbf{I} - \mathbf{M}_{ii}^{*-1}\mathbf{J}_{gi}^T(\mathbf{J}_{gi}\mathbf{M}_{ii}^{*-1}\mathbf{J}_{gi}^T)^{-1}\mathbf{J}_{gi})\mathbf{M}_{ii}^{*-1}\mathbf{v}_i + \mathbf{C}_i^* \end{aligned} \quad (41)$$

$$\tau_i = (J_{gi}M_{ii}^{*-1})^+ (\ddot{p}_{di} + K_d \dot{e}_i + K_p e_i - \dot{J}_{gi} \dot{q}_i - \dot{v}_{ge} + J_{gi} M_{ii}^{*-1} C_i^*) (I - (J_{gi} M_{ii}^{*-1})^+ J_{gi} M_{ii}^{*-1}) v_i \quad (42)$$

These two previous controllers have been by using $W = M_{ii}^{*-1}$ and $W = I$, respectively and considering nullspace resolution and the values of the constraints given in Eq. (33) and Eq. (34).

V. Results

The performances of the proposed strategy have been assessed through numerical simulations reported in this section. The simulated scenario uses the humanoid robot represented in Fig. 1. Table 2 and Table 3 summarize the main dynamic parameters of the robot. These tables list the moment of inertia, the mass and the dimensions of the main body of the robot and the links of both arms. It is assumed that the two arms are symmetric and have the same dynamic parameters. The trajectories were obtained using C++ code interfaced with Interior Point Method solver IPOPT [43]. The tracking controllers have been implemented in Robot Operating System (ROS) and applied to a scenario simulated in Gazebo/ROS Melodic using an Intel Core i7/2.8 GHz Quad Core laptop. The Jacobians and moment of inertia have been computed using the Pinocchio library [44]. Given the robot's initial and desired position, the proposed method was able to select and plan for different maneuvers and motions. The robot could grasp the handrails with both hands, push itself ahead with a single hand, or move by alternatingly moving both arms on the handrails. The grasping sequence is obtained by taking into account the robot position and the actual configuration of the handrails. The optimizer selects the grasping points and consequently calculates the arm trajectories and the eventual force contacts. The proposed controller can track the trajectories, and the final location can be achieved using the grasping of both hands. To better understand and highlight the complex motions generated by the proposed algorithm an accompanying video with the different experiments described in this section can be seen here [45].

Table 2 Main dynamic parameters of the robot' body.

	Mass (kg)	Height (m)	Inertia (kg·m ²)					
			Ixx	Iyy	Izz	Ixy	Ixz	Iyz
Body Parameters	93	0.843	18.6	15.4	4.1	-0.008	-0.027	0.058

Table 3 Main dynamic parameters of the robot.

	Mass (kg)	Length (m)	Inertia (kg·m ²)					
			I _{xx}	I _{yy}	I _{zz}	I _{xy}	I _{xz}	I _{yz}
Link 1	2.741	0.28	0.0124	0.0042	0.0136	3.6e-05	7.1e-05	-0.0002
Link 2	2.425	0.144	0.013	0.0138	0.0049	1.2e-05	-0.0032	-0.0001
Link 3	2.209	0	0.007	0.0069	0.0039	-0.0001	0.0007	0.0004
Link 4	0.877	0.274	0.0025	0.0027	0.0012	0.0001	-0.0003	0.0004
Link 5	1.878	0.265	0.0035	0.0044	0.0023	1.3e-05	1.03e-05	-9.7e-05
Link 6	0.409	0	0.0001	0.00014	0.00015	-8.9e-08	-4.4e-08	4.2e-07
Link 7	0.308	0	0.0003	0.0002	0.00017	-1.6e-06	1.7e-06	-1.2e-05

A. Trajectory planning.

The suitability of the optimization strategy shown in Section 3 was assessed via two different simulations. The first one requires the humanoid robot to along the same handrails, and the second one shows that the same strategy can be used for moving from a handrail to another. However, a first study about the system dynamics employed in the planning phase is shown in the next subsection.

1. Study of the dynamic consistency

One important aspect to generate adequate motion plans for both arms is to guarantee that these motions are physically feasible. Different constraints are introduced in the trajectory optimization problem to guarantee the feasibility of these motions (see Table 1). Although the joint motions of the arms can significantly influence the motion of the robot torso, in this specific application, the center of mass acceleration is mainly affected by the end-effector contact forces. The use of Eq. (6) dictate how these contact forces affect the center of mass acceleration. This dynamic equation allows generating feasible motions while the formulation is fully in Cartesian space. This equation allows a simpler formulation that can be solved in a reasonable time, avoiding the nonlinearities introduced by the joint angles (see the experiments presented in this section where the times required by the nonlinear programming solver are indicated). This simplification has been proved to be sufficient for planning feasible trajectories like the ones proposed in this paper. However, the dynamic equation presented in Eq. (5) is considered in the proposed controllers for tracking the planned trajectories.

This section shows an experiment to verify the dynamic consistency of the trajectories planed with the proposed method. To do this, a motion planning is presented where the robot's body is supposed to move 0.3 m. along the x direction. The linear accelerations of the robot's body are represented in Fig. 4.a. This last figure represents both, the robot's body linear accelerations obtained using Eq. (6), and the ones obtained taking also into account the effect on

the robot's body due the joint accelerations (in black). As it can be seen, although the linear accelerations obtained by Eq. (6) deviates slightly in several points with respect the full dynamics, it captures the main values and evolution. Therefore, enforcing the proposed equation, it allows generating feasible trajectories for both arms. The contact forces at the manipulator's end-effectors, and the robot's body position are represented in Fig. 4.b and Fig. 4.c respectively. As it can be seen in this last figure, the desired displacement is achieved for the robot's body.

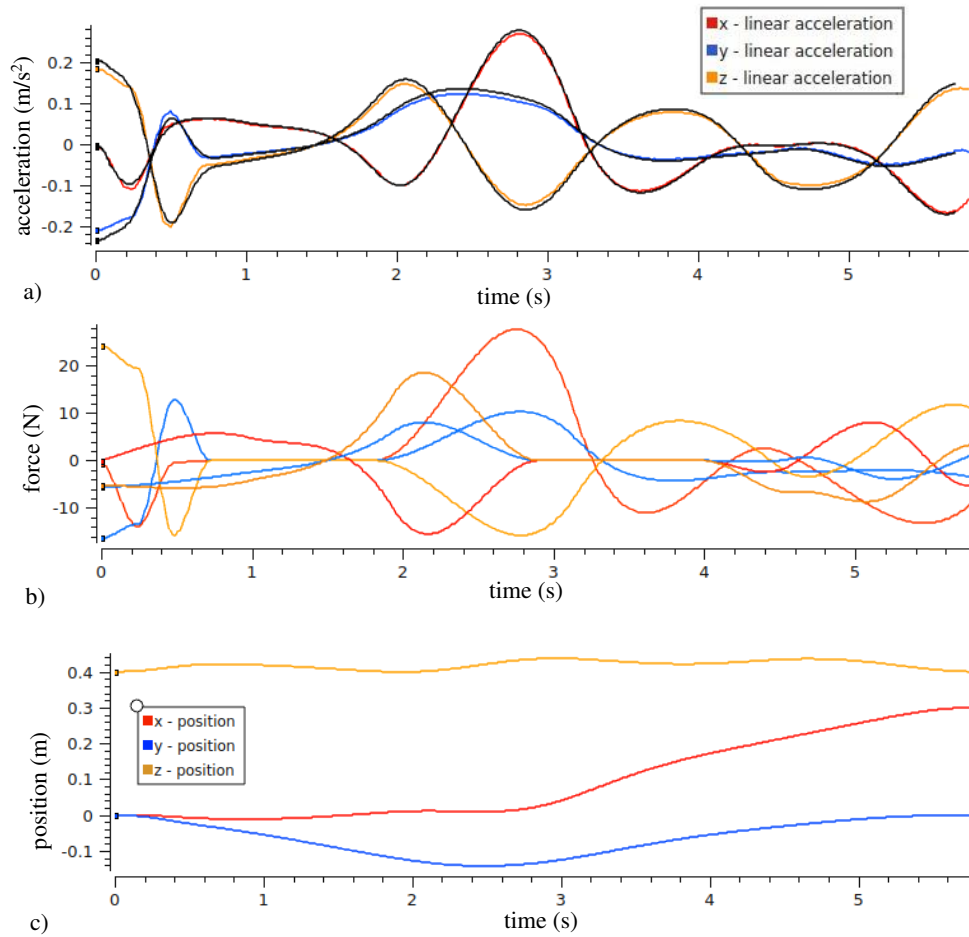


Fig. 4 a) Linear acceleration of the robot base only considering Eq. (2). b) Contact forces generated by both arms during the experiment. c) Position of the robot base.

2. Motion on handrails

The first scenario reproduces a handrail configuration similar to the 3D representation of the ISS in Fig. 1. The initial configuration of the robot grasping two parallel handrails are represented in Fig. 5.a. The initial and desired positions of the robot are represented as $\mathbf{r}_0$ and $\mathbf{r}_d$, respectively. Specifically, the robot is supposed to move along the x direction of 0.6 m and perform a rotation of -0.25 rad around z direction. The final configuration of the robot during this experiment is represented in Fig. 5.b. Figure 5.c shows the upper view of the initial and final robot positions in this experiment.

The algorithm returned a schedule of the robot operations in only 12.4 s. The linear and angular acceleration of the robot body during the experiment are shown in Fig. 6. The optimizer checks the compliance with all the dynamic constraints every 0.1 s and this is reflected in the segmented behaviour that is evident from the figure. The check could be enforced with a higher frequency, increasing the computational effort required to run the algorithm. However, the results obtained with this sample time show that an update frequency of 10 Hz is enough to generate feasible and dynamically consistent trajectories.

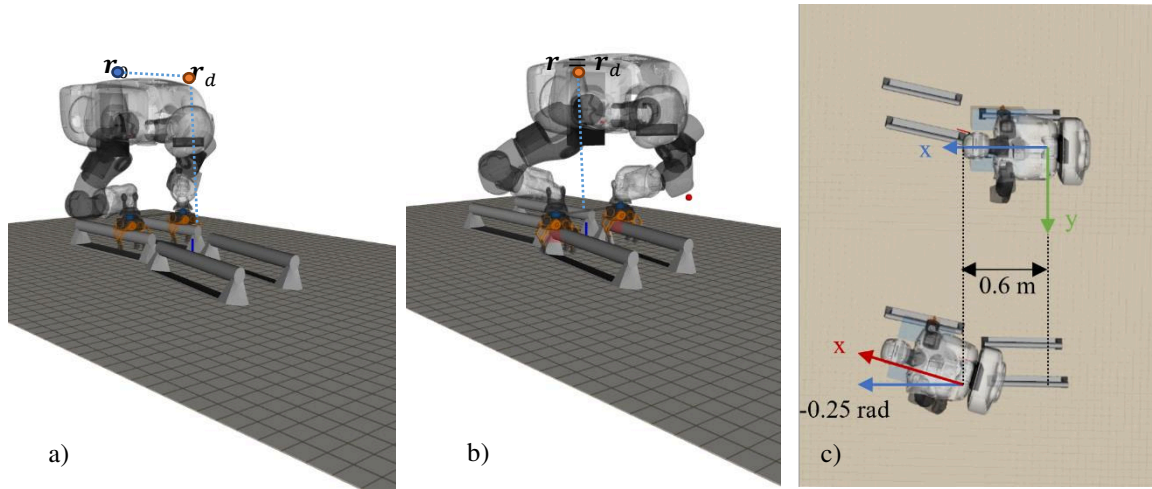


Fig. 5 Motion on handrails a) Initial position of the robot. b) Final robot position. c) Upper view of the initial and final robot position.

The position and attitude of the robot body during the maneuver are shown in Fig. 7. A smoother evolution for these variables is obtained because the kinematic constraint (section 3.2.1) is enforced every 0.05 s. The weights considered in Eq (9) are $[\sigma_{i1}, \sigma_{i1}, \sigma_{i1}, \sigma_4, \sigma_5, \sigma_6] = [1, 1, 0, 1, 1, 1]$. Figure 7.a represents the 3D position of the robot during the maneuver. The body of the robot moves 0.6 m along the direction of the handrails, while the distance between the torso and the handrail at the end of the trajectory is the same that the one obtained at the beginning of

the trajectory. Figure 7.b shows the contextual change of the attitude of the robot in Euler angles. The orientation of the torso with respect to the x and y directions remains constant, meaning that there is any roll and pitch of the torso, but a rotation of -0.25 rad around z-direction shows that there is a yaw change during the maneuver. Finally, Fig. 8 and Fig. 9 show the end-effector position and force of both left and right arms during the operations, respectively. In these figures, vertical blue bands are included to represent the time when the robot is in contact with the handrails. The left and right arm end-effector trajectories are represented in Fig. 8.a and Fig. 9.a. These figures show that the desired grasping positions are enforced by imposing the motion constraints in Eq. (15). These grasping positions are maintained during the grasping thanks to the constraint given by Eq. (14). The forces that appear during the contact are shown in Fig. 8.b and Fig. 9.b. As expected, due to the constraint given by Eq. (19), the robot does not exert forces when it is not in contact with the handrails.

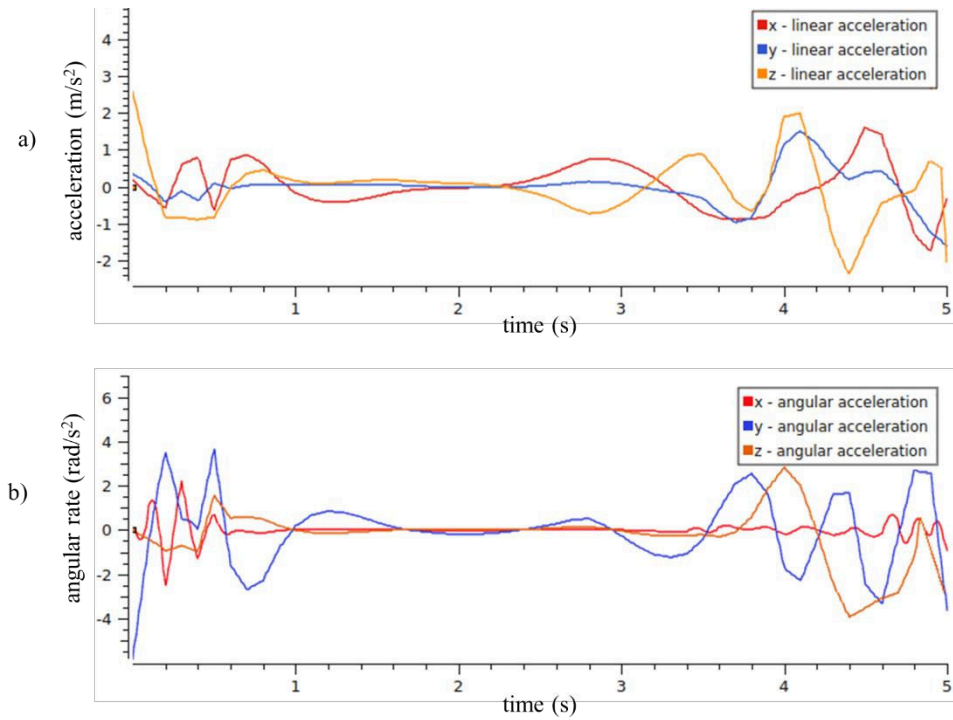


Fig. 6 Linear and angular acceleration of the robot body during the motion on handrails.

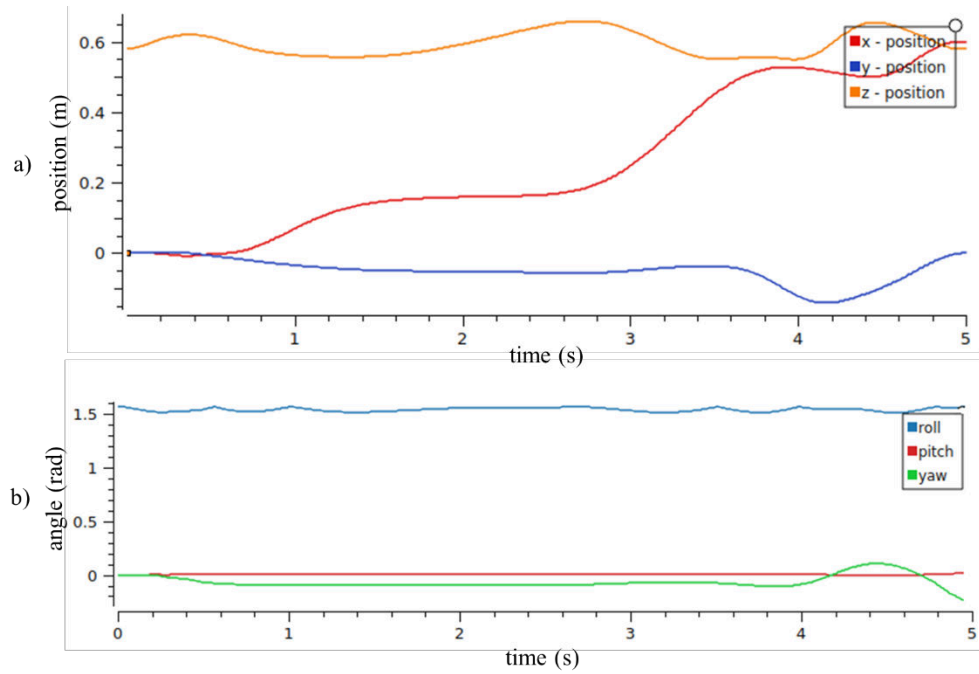


Fig. 7 3D Position and attitude of the robot body during the motion on handrails.

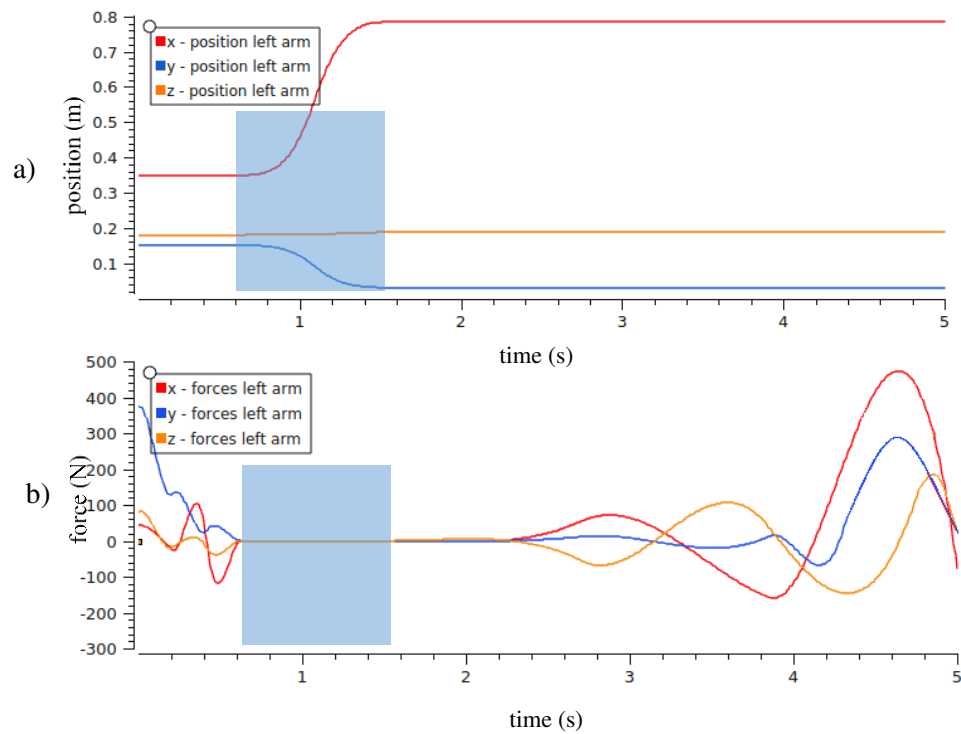


Fig. 8 3D end-effector position and forces of the left arm during the motion on handrails.

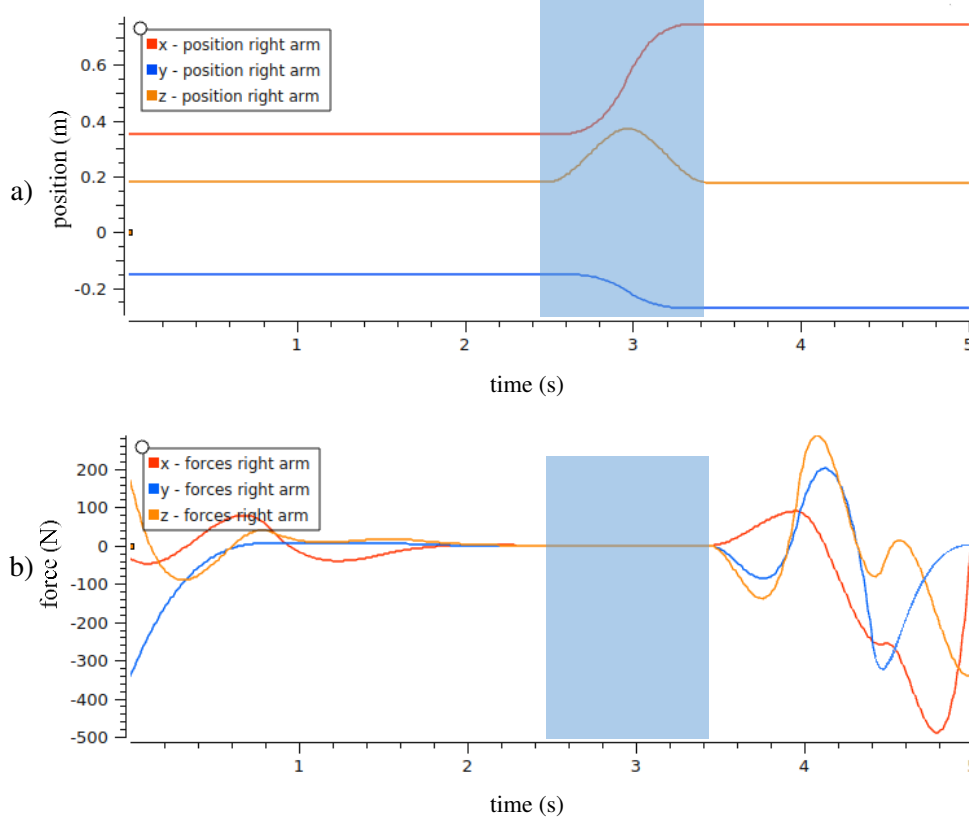


Fig. 9 3D end-effector position and forces of the right arm during the motion on the handrails.

3. Motion among multiple handrails

Figure 10 represents the scenario that assesses the optimizer's ability to plan motions involving multiple handrails. In Fig. 10.a the initial position ($\mathbf{r}_0$), and desired position ($\mathbf{r}_d$) of the robot are represented, respectively. The robot body is supposed to move 1 m along the x-direction while the desired displacements along y and z directions are zero. The robot needs to grasp the handrails alternately with the two arms.

In this case, the algorithm took 20.3 s to generate the optimal robot path. The weights considered in Eq (9) are $[\sigma_{i1}, \sigma_{i1}, \sigma_{i1}, \sigma_4, \sigma_5, \sigma_6] = [1, 1, 0, 1, 1, 1]$. The 3D position of the torso during the trajectory is represented in Fig. 11. Such a figure shows that the torso moved about 1 m ahead along the x axis while the other two components are approximately maintained constant. The end-effector position and force of both left and right arms during the maneuver are represented in Fig. 12 and Fig. 13. In these figures, vertical bands are included to represent the time when the robot is in contact with the handrails. The left and right arm end-effector trajectories are represented in Fig.

12.a and Fig. 13.a respectively. These figures show that the grasping positions are reached by imposing the constraints in Table 1. Such grasping positions are held for the time intervals necessary to let the robot's body move ahead on its path. This task is accomplished by imposing the constraint given in Eq. (14). The forces provided by the end-effectors when these are grasping the handrails are shown in Fig. 12.b and Fig. 13.b. It is also worth noting that the robot does not exert forces when it is not in contact with the handrails as expected due to the imposition of the constraint in Eq. (19).

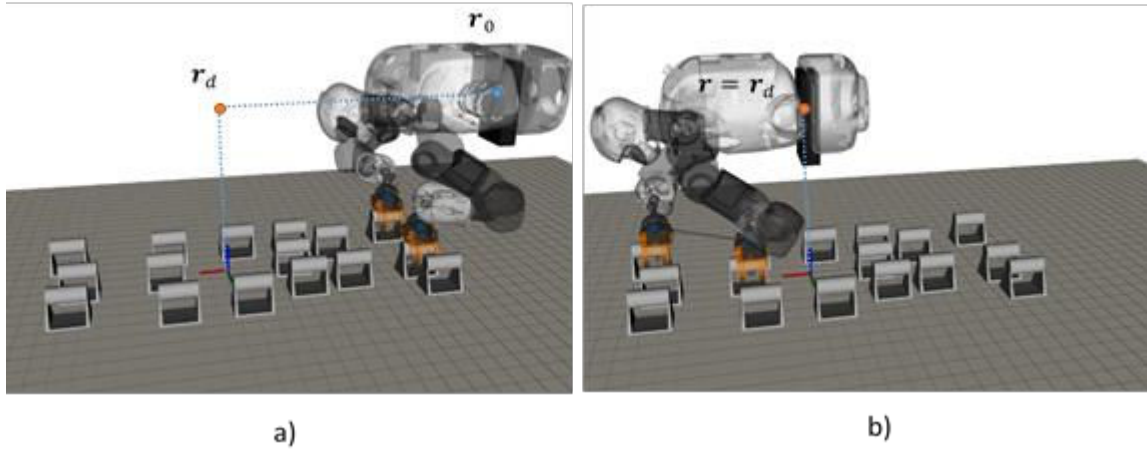


Fig. 10 a) Initial position of the robot. b) Final robot position.

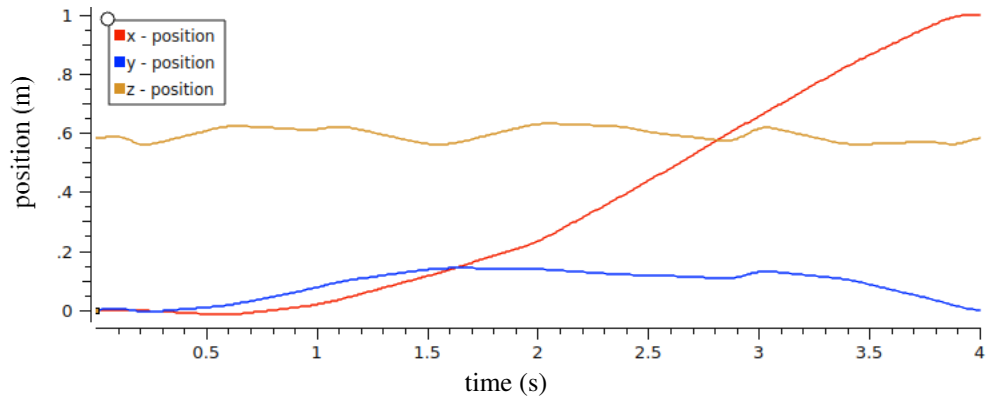


Fig. 11 3D Position of the torso during the motion among multiple handrails.

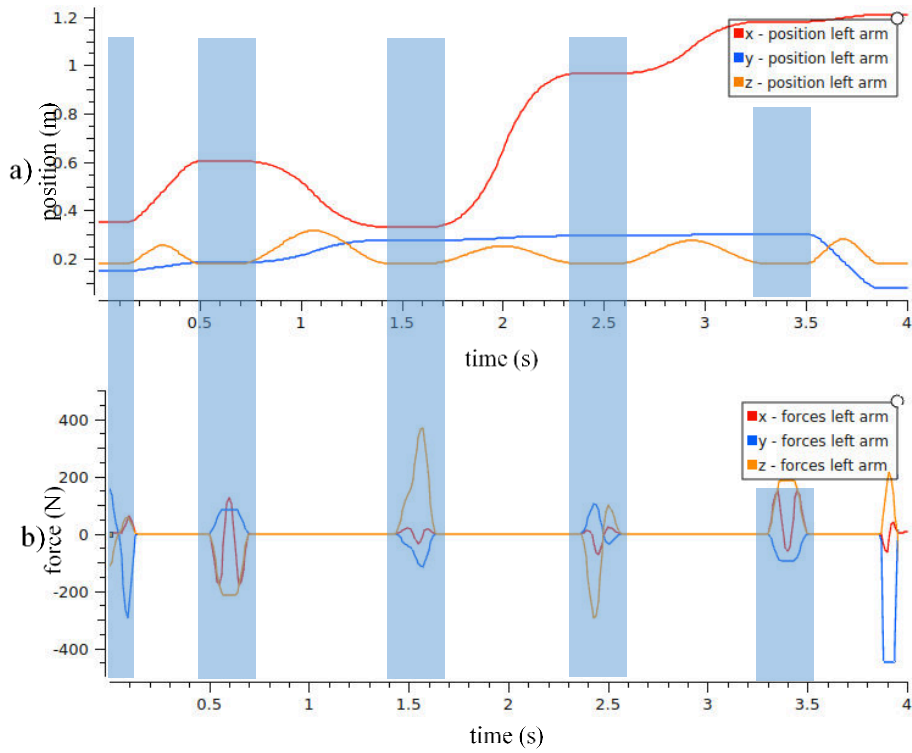


Fig. 12 3D end-effector position and forces of the left arm during the motion among multiple handrails.

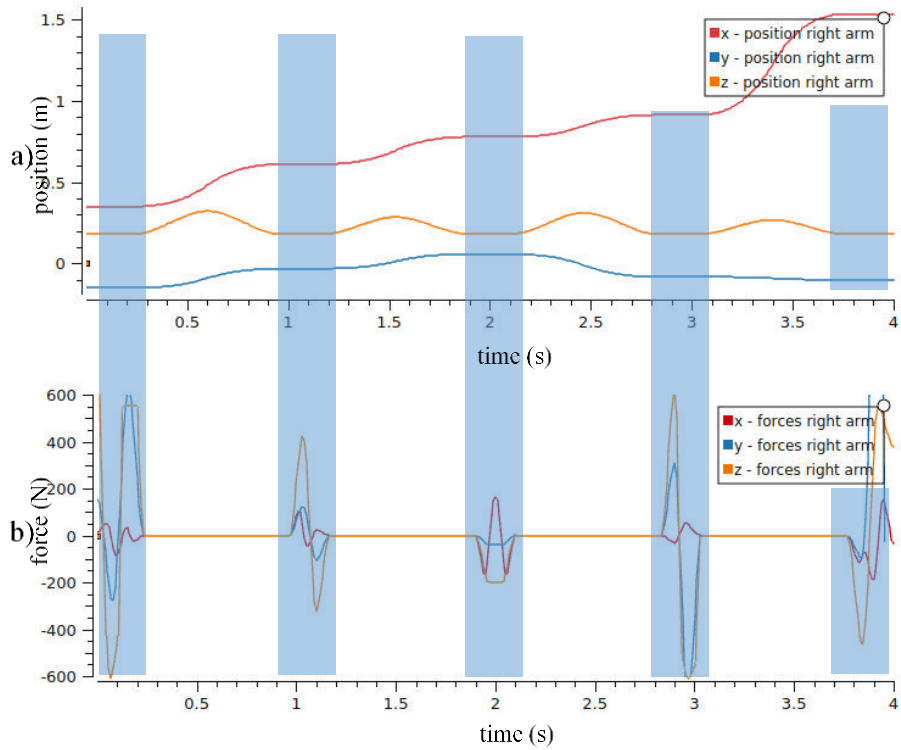


Fig. 13 3D end-effector position and forces of the right arm during the motion among multiple handrails.

4. Arms motion planning considering different number of grasping positions

The effect of considering different number of grasping positions for the same maneuver is considered in this initial experiment. To better show the arms motion planning with different grasping positions, the handrails are not used in this experiment. The robot's body is supposed to move 0.6 m along the x direction. The weights considered in Eq (9) are $[\sigma_{i1}, \sigma_{i1}, \sigma_{i1}, \sigma_4, \sigma_5, \sigma_6] = [1, 1, 1, 0.5, 0.5, 0.5]$. With these weights the system will try to find the desired arm and body trajectories while minimizing the contact forces. The results obtained considering $N = 2, 4$ and 6 are shown in Fig. 14.a, Fig. 14.b, and Fig. 14.c respectively. For each experiment, these figures show the contact forces at the end effectors, the linear accelerations at the robot's body due to the contact forces, the end-effectors positions, and the robot's body trajectory. In all the three experiments, the desired location for the robot's body is achieved in the desired duration $T = 5.4$ secs. However, the number of grasping positions is increasing as shown in the end-effector positions figures and the corresponding end-effector forces (third and first columns in Fig. 14).

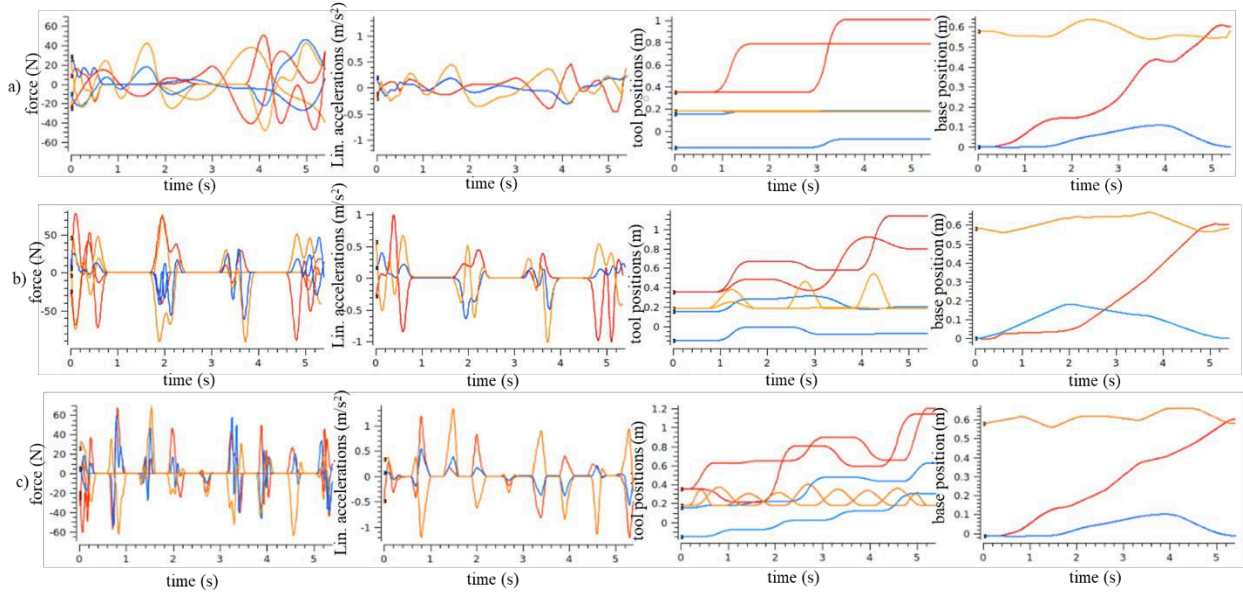


Fig. 14 Experiments considering different grasping positions. In red, blue and yellow are represented the x, y and z components

B. Performance Evaluation of the Tracking Control Strategy

This section evaluates the performance of the three controllers proposed in Eqs. (37), (41), and (42) during the tracking of a grasping trajectory generated by the planning algorithm. Figure 15 represents the robot at the initial, intermediate, and final configurations of the grasping maneuver performed by the robot. The grasping operation is performed in 12 s. At each iteration, a time-step of 1 *msec* is considered for the computation of the control law. Figure 16 represents, for each of the controllers, the desired trajectory (black) and the one obtained during the tracking (red). All three controllers reach the desired grasping pose by tracking the desired trajectory. In order to highlight more clearly the differences between each controller, Fig. 17 represents the control error, $\mathbf{e}_t$, for each controller during the tracking and Fig. 18 represents the control actions (joint torques) during the tracking. Although the torques remain low during the tracking, some differences can be observed in the behaviours of the three proposed controllers. The tracking errors of the controllers given by Eq. (37), (41) have similar behaviour. On the other hand, the behaviour produced by the controller in Eq. (42) is way more oscillative. This difference can also be observed from the behaviour of the control actions in Fig. 18. By analysing the behaviour in the joint space, it can be deduced that both the controllers obtained with Eq. (37) and (41) are the best candidate for this kind of task. However, higher torque peaks can be appreciated during the tracking when the controller following Eq. (41) is applied.

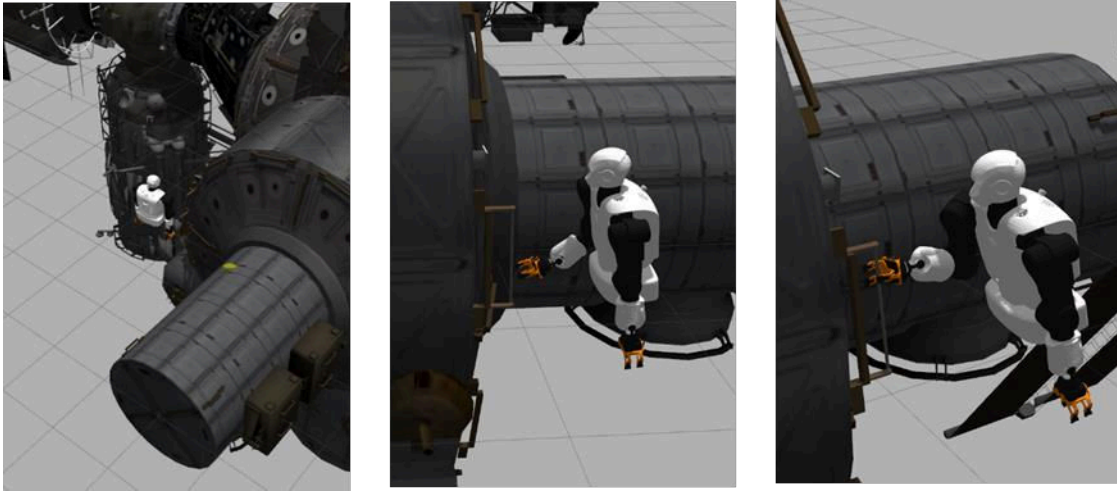


Fig. 15 3D representation of the robot at the initial, intermediate, and final grasping pose.

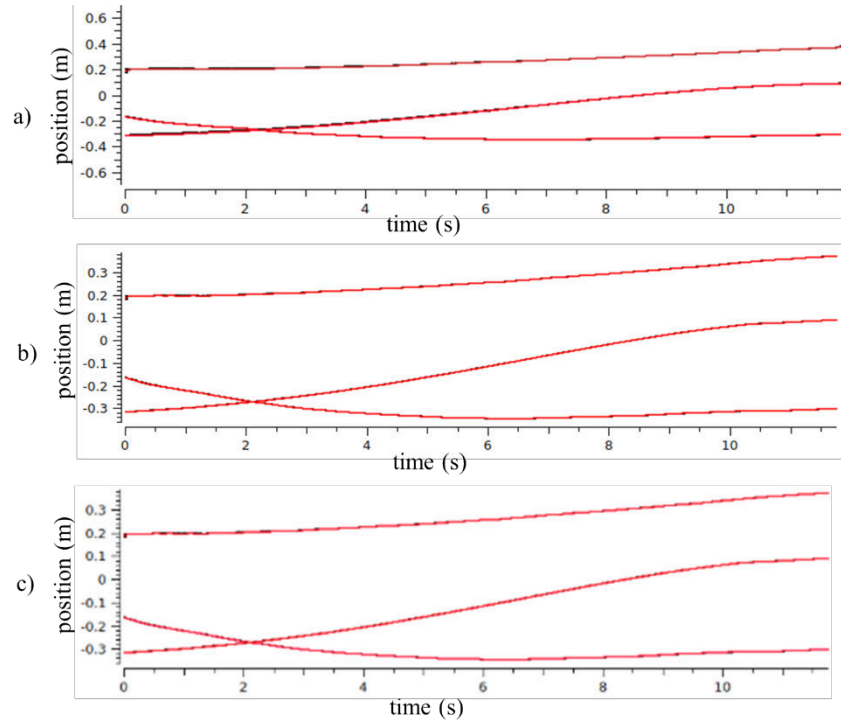


Fig. 16 Desired (black) and obtained 3D trajectory (red) during the tracking. a) Equation (37). b) Equation (41). c) Equation (42).

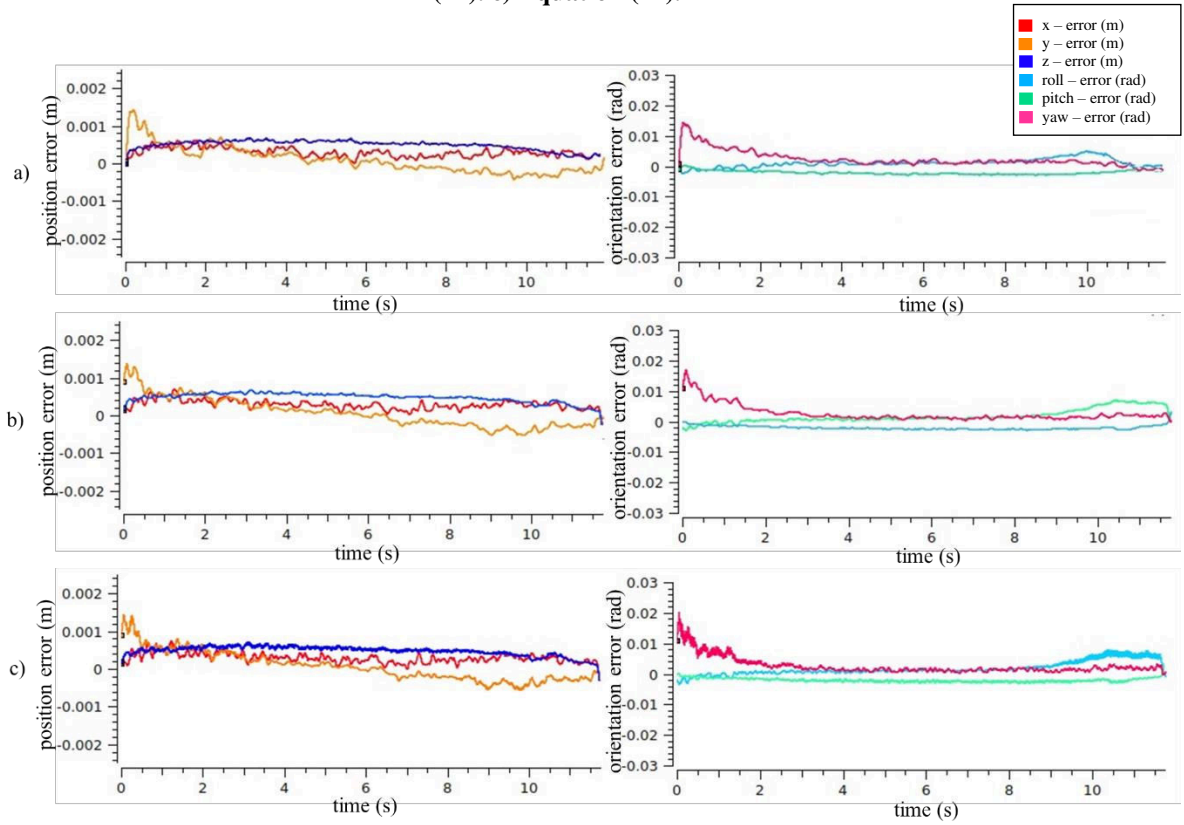


Fig. 17 Control error during the tracking. a) Eq. (37). b) Eq. (41). c) Eq. (42).

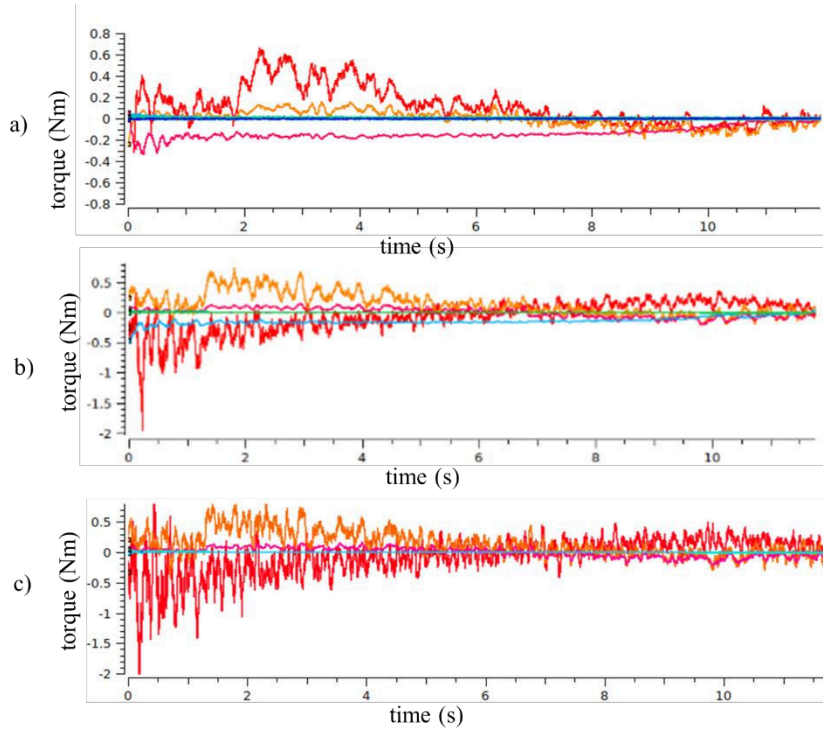


Fig. 18 Control actions during the tracking. a) Equation (37). b) Equation (41). c) Equation (42).

VI. Conclusions

An optimal strategy for path and motion planning of a humanoid robot in free-floating conditions was presented in this paper. The developed algorithm found optimal sequences that command the arms motions to push the handrails in a coordinated way to move forward and backwards the humanoid robot. Such a strategy considered the eventual map of the morphology handrails present in the simulated scenario and the eventual interactions of the grippers with them. As a result, a robust strategy for planning and selecting the grasping points and the articulated movements of the arms was developed and tested in a simulated environment. Specifically, the paper took as an example two representative operations that an eventual robot can perform in an extravehicular activity outside the ISS. If the handrails are sufficiently long, the robot could move by just alternatively grasping different points along them. Subsequently, the same algorithm was tested in a scenario when multiple handrails needed to be grasped to reach the desired position. In both cases, the algorithm could find the optimal sequence of operations and coordinate the grasping and non-grasping phases in a constructive and efficient way.

In addition, a controller that takes as input the planned trajectories of the end effector and finds the eventual control actions to be applied to the joints of both the arms was developed and tested in a representative simulated environment. Tracking errors were indeed always limited, and the arms could always follow the desired motion, even if in 1 of the 3 cases the oscillations were not sufficiently damped. The results showed that at least 2 out of 3 developed control strategies could be proficiently applied to track such kinds of trajectories.

In any case, the numerical results demonstrate that the overall scheme (trajectory optimizer + controller) is sufficiently robust and allows for complex and articulated motions of the humanoid robot in an eventual extravehicular activity outside the ISS.

References

- [1] Gary Eugene Musgrave, Axel (Skip) M. Larsen, Tommaso, *Safety Design for Space Systems*, Butterworth-Heinemann, 2009, Chapter 22, pp. 705-723.
<https://doi.org/10.1016/B978-0-7506-8580-1.00022-1S>.
- [2] N. C. Jordan, J. H. Saleh, D. J. Newman, "The extravehicular mobility unit: A review of environment, requirements, and design changes in the US spacesuit," *Acta Astronautica*, Vol. 59, No. 12, 2006, pp. 1135-1145.
<https://doi.org/10.1016/j.actaastro.2006.04.014>.
- [3] A.S. Johnson, M.J. Golightly, M.D. Weyland, T. Lin, E.N. Zapp, "Minimizing space radiation exposure during extravehicular activity," *Advances in Space Research*, Vol. 36, No. 12, 2005, pp. 2524-2529.
<https://doi.org/10.1016/j.asr.2004.05.008>.
- [4] T. D. Haws, J. S. Zimmerman and M. E. Fuller, "SLS, the Gateway, and a Lunar Outpost in the Early 2030s," *2019 IEEE Aerospace Conference*, 2019, pp. 1-15.
<https://doi.org/10.1109/AERO.2019.8741598>.
- [5] D. Richey, T. Cichan and D. Sabolish. "Gateway Mission Operations and Crew Activities," *2018 AIAA SPACE and Astronautics Forum and Exposition*. AIAA 2018-5246, 2018.
- [6] M. Stuttard, S. Amos, P. Aspden, M. Bowyer, C. Bridges and S. McLaren. "Vast Satcom Antennas (VASANTs) as an enabling technology for commercial satellite communications." *Proceedings of the 16th Reinventing Space Conference (RISpace 2018)*, 2018 , pp.1-3.
- [7] M. A. Diftler et al., "Robonaut 2 - The first humanoid robot in space," *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2178-2183.

<https://doi.org/10.1109/ICRA.2011.5979830>.

- [8] DLR, “Cimon - The Intelligent Robot Assistant. European Space Agency” from https://www.dlr.de/content/en/articles/news/2018/1/20180302_cimon-the-intelligent-astronaut-assistant_26307.html, accessed 18/07/2021.
- [9] E. Howell, space.com, “Russia Launches Humanoid Robot ‘Fyodor’ to Space Station on Rare Soyuz Test Flight”, <https://www.space.com/russia-launches-humanoid-robot-on-soyuz-spacecraft.html>, 22 Aug. 2019.
- [10] S. Nolet. "The SPHERES Navigation System: from Early Development to On-Orbit Testing," AIAA 2007-6354. AIAA Guidance, Navigation and Control Conference and Exhibit. August 2007. <https://doi.org/10.2514/6.2007-6354>
- [11] M.G. Bualat, T. Smith, E.E. Smith, T. Fong, D.W. Wheeler. "Astrobee: A New Tool for ISS Operations," AIAA 2018-2517. 2018 SpaceOps Conference. May 2018, <https://doi.org/10.2514/6.2018-2517>
- [12] J.L. Ramirez-Riberos, M. Pavone, E. Frazzoli, D.W. Miller, “Distributed Control of Spacecraft Formations via Cyclic Pursuit: Theory and Experiments”, *Journal of Guidance, Control, and Dynamics* 2010 33:5, 1655-1669, <https://doi.org/10.2514/1.46511>
- [13] N. Radford et al., “Valkyrie: NASA’s First bipedal humanoid robot”, *Journal of Field Robotics* Vol. 32, No. 3, 2015, pp. 397-419. <https://doi.org/10.1002/rob.21560>.
- [14] M. Mote, M. Egerstedt, E. Feron, A. Bylard, M. Pavone, Collision-Inclusive Trajectory Optimization for Free-Flying Spacecraft, *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 7, 2020, pp. 1247–1258. <https://doi.org/10.2514/1.G004788>
- [15] G. Misra, X. Ba, Optimal Path Planning for Free-Flying Space Manipulators via Sequential Convex Programming, *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 11, 2017, pp. 3019–3026. <https://doi.org/10.2514/1.G002487>
- [16] G. Misra, X. Ba, Task-Constrained Trajectory Planning of Free-Floating Space-Robotic Systems Using Convex Optimization, *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 11, 2017, pp. 2857–2870. <https://doi.org/10.2514/1.G002405>.
- [17] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, “Biped walking pattern generation by using preview control of zero-moment point”, *International Conference on Robotics and Automation*, 2003, pp. 1620–1626.
- [18] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, “Capture point: A step toward humanoid push recovery”, in *IEEE-RAS International Conference on Humanoid Robots*, 2006, pp. 200–207. <https://doi.org/10.1109/ICHR.2006.321385> (cit. on pp. 14, 19, 37, 41, 75).

- [19] M. Kudruss, M. Naveau, O. Stasse, N. Mansard, C. Kirches, P. Soueres, and K. Mombaur, "Optimal Control for Multi-Contact, Whole-Body Motion Generation using Center-of-Mass Dynamics for Multi-Contact Situations," *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 684–689.
<https://doi.org/10.1109/HUMANOIDS.2015.7363428> (cit. on pp. 19, 56).
- [20] B. Ponton, A. Herzog, S. Schaal, and L. Righetti, "A convex model of humanoid momentum dynamics for multi-contact motion generation," *IEEE-RAS International Conference on Humanoid Robots*, 2016, pp. 842–849.
- [21] A. Herzog, S. Schaal, and L. Righetti, "Structured contact force optimisation for kino-dynamic motion generation", in *IEEE International Conference on Intelligent Robots and Systems*, 2016, pp. 2703–2710.
<https://doi.org/10.1109/IROS.2016.7759420> (cit. on pp. 5, 19, 56).
- [22] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, "Trajectory generation for multicontact momentum control", in *IEEE-RAS International Conference on Humanoid Robots, IEEE*, 2015, pp. 874–880.
<https://doi.org/10.1109/HUMANOIDS.2015.7363464> (cit. on p. 19).
- [23] W. Ko, W. Chiang, Y. Hsu, M. Yu, H. Lin, P. Lin, "A Model-Based Two-Arm Robot With Dynamic Vertical and Lateral Climbing Behaviors." *ASME. J. Mechanisms Robotics*, Vol. 8, No. 4, 2016, 044503.
<https://doi.org/10.1115/1.4032777>.
- [24] I. Chatzinikolaidis, Z. Li, "Trajectory Optimization of Contact-Rich Motions Using Implicit Differential Dynamic Programming," *IEEE Robotics and Automation Letters*, Vol. 6, No. 2, 2021, pp. 2626-2633, doi: 10.1109/LRA.2021.3061341.
- [25] G. Rigatos, K. Busawon, J. Pomares, M. Abbaszadeh, "Non-linear optimal control for the wheeled inverted pendulum system," *Robotica*, Vol. 38, No. 1, 2019, pp. 29-47.
<https://doi.org/10.1017/S0263574719000456>.
- [26] B. Doerr, R. Linares, "Motion Planning and Control for On-Orbit Assembly using LQR-RRT* and Nonlinear MPC", arXiv preprint arXiv:2008.02846, 2020.
- [27] Y. Li, X. Hao, Y. She, S. Li, M. Yu, "Constrained motion planning of free-float dual-arm space manipulator via deep reinforcement learning," *Aerospace Science and Technology*, Vol. 109, 2021.
<https://doi.org/10.1016/j.ast.2020.106446>.
- [28] P. Huang, Y. Xu, B. Liang, "Tracking Trajectory Planning of Space Manipulator for Capturing Operation," *International Journal of Advanced Robotic Systems*, 2006.
<https://doi.org/10.5772/5735>.
- [29] F. Aghili, "Optimal Trajectories and Robot Control for Detumbling a Non-Cooperative Satellite", *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 5, 2020, pp. 981-988, <https://doi.org/10.2514/1.G004758>

- [30] E. Papadopoulos, F. Aghili, O. Ma, R. Lampariello, “Robotic Manipulation and Capture in Space: A Survey”, *Frontiers in Robotics AI*, 2021, <https://doi.org/10.3389/frobt.2021.686723>.
- [31] F.L. Basmadji, K. Seweryn, J.Z. Sasiadek, “Space robot motion planning in the presence of nonconserved linear and angular momenta,” *Multibody System Dynamics*, Vol.50, 2020, pp. 71–96.
<https://doi.org/10.1007/s11044-020-09753-x>
- [32] Zhengcang Chen, Weijia Zhou, "Path Planning for a Space-Based Manipulator System Based on Quantum Genetic Algorithm," *Journal of Robotics*, Vol. 2017, Article ID 3207950, 2017, 10 pages.
<https://doi.org/10.1155/2017/3207950>.
- [33] Q. Zhang, L.Ji, D. Zhou, X. Wei, “Nonholonomic motion planning for minimising base disturbances of space manipulators based on multi-swarm PSO,” *Robotica*, Vol. 35, No. 4, 2017, pp. 861-875.
<https://doi.org/10.1017/S026357471500085>.
- [34] C. Gehring, S. Coros, M. Hutter, C. Dario Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger, and R. Siegwart, “Practice Makes Perfect: An Optimisation-Based Approach to Controlling Agile Motions for a Quadruped Robot”, *IEEE Robotics and Automation Magazine*, Vol. 23, No. 1, 2016, pp. 34– 43.
<https://doi.org/10.1109/MRA.2015.2505910> (cit. on pp. 20, 24).
- [35] M. Neunert, F. Farshidian, A. W. Winkler, J. Buchli, “Trajectory optimisation through contacts and automatic gait discovery for quadrupeds”, *IEEE Robotics and Automation Letters*, Vol. 2, 2017, pp. 1502–1509.
<https://doi.org/10.1109/LRA.2017.2665685> (cit. on pp. 20, 24, 37, 57).
- [36] A. W. Winkler, C. D. Bellicoso, M. Hutter, J. Buchli, “Gait and Trajectory Optimization for Legged Systems Through Phase-Based End-Effector Parameterisation,” *IEEE Robotics and Automation Letters*, Vol. 3, No. 3, 2018, pp. 1560-1567.
<https://doi.org/10.1109/LRA.2018.2798285>.
- [37] S. Cocuzza, I. Pretto, S. Debei, “Least-Squares-Based Reaction Control of Space Manipulators,” *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 3, 2012, pp. 976–986. doi:<https://doi.org/10.2514/1.45874>.
- [38] H. S. Jayakody, L. Shi, J. Katupitiya, N. Kinkaid, “Robust Adaptive Coordination Controller for a Spacecraft Equipped with a Robotic Manipulator” *Journal of Guidance, Control, and Dynamics*, Vol. 69, No. 12, 2016, pp. 2699-2711.
<https://doi.org/10.2514/1.G002145>.
- [39] F. Giulietti, A. A. Quarta, P. Tortora P., “Optimal Control Laws for Momentum-Wheel Desaturation Using Magnetorquers,” *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 6, 2006, pp. 1464–1468.
<https://doi.org/10.2514/1.23396>

- [40] J. Pomares, C. A. Jara, J. Pérez, F. Torres. "Direct visual servoing framework based on optimal control for redundant joint structures," *International Journal of Precision Engineering and Manufacturing*, Vol. 16, No. 2, 2015, pp 267–274. doi:10.1007/s12541-015-0035-z
- [41] X Chen, K. Watanabe, K. Kiguchi, K. Izumi. "Optimal Force Distribution for the Legs of a Quadruped Robot," *Machine Intelligence & Robotic Control*, Vol. 1, No. 2, 1999, pp. 87–94.
- [42] N. Kostas, E. G. Papadopoulos, "On the Dynamics and Control of Free-floating Space Manipulator Systems in the Presence of Angular Momentum" *Frontiers in Robotics and AI*, Vol. 4, 2017. <https://doi.org/10.3389/frobt.2017.00026>.
- [43] A. Waechter, L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, Vol. 106, No. 1, 2006, pp. 25–57, <https://doi.org/10.1007/s10107-004-0559-y>.
- [44] J. Carpentier et al., "The Pinocchio C++ library : A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," *2019 IEEE/SICE International Symposium on System Integration (SII)*, 2019, pp. 614-619, <https://doi.org/10.1109/SII.2019.8700380>.
- [45] J. L. Ramón , R. Calvo , A. Trujillo, J. Pomares, L. Felicetti. <https://youtu.be/TDBZSGXR7VY>



DSPACE

<https://dspace.org/>

Trajectory optimization and control of a free-floating two arms humanoid robot

Ramón, José Luis; Calvo, Ramon; Trujillo, Adrian; Pomares, Jorge; Felicetti, Leonard

2022-05-02

AIAA

<https://doi.org/10.2514/1.G006828>

Ramon JL, Calvo R, Trujillo A, et al., (2022) Trajectory optimization and control of a free-floating two arms humanoid robot. Journal of Guidance, Control, and Dynamics, Volume 45, Number 9, September 2022, pp. 1661–1675