IAC-18,D1,6,7,x44372

Towards an Autonomous Free-flying Robot Fleet for Intra-vehicular Transportation of Loads in Unmanned Space Stations

Rodrigo Ventura[a*], **Pedro Roque**[c], **Monica Ekal**[b]

[a] *Institute for Systems and Robotics, Instituto Superior Tcnico, Universidade de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal,* rodrigo.ventura@isr.tecnico.ulisboa.pt

[b] Institute for Systems and Robotics, Instituto Superior Técnico, mekal@isr.tecnico.ulisboa.pt

[c] KTH Royal Institute of Technology in Stockholm, padr@kth.se

[*] Corresponding Author

Abstract

The prospect of the Deep Space Gateway space station deployment in late 2020s, designed to support deep space, as well as lunar missions, raises new challenges to the space robotics community. In particular, given the expectation for this space station not to be permanently crewed, logistics operations, such as inspection and handling of material to and from automated cargo spacecraft must be handled autonomously by robots. This paper starts by discussing the challenges for such robots to be able to perform loading and unloading operations to and from a cargo spacecraft docked to the station. The paper proceeds by presenting the current research being carried out at the Institute for Systems and Robotics (ISR-Lisbon) on this problem. In this context, we will first review the design of Space CoBot, a free-flyer robot for intra-vehicular operations, designed for collaborative tasks including mobile manipulation. Then, we will present initial work on the problem of object tracking and grasping using convergent controllers. And finally, our current work on load inertial parameter estimation using excitatory trajectories will be presented. Both object tracking and grasping, and the load inertial parameter estimation work are platform agnostic, meaning that they can be evaluated in current free-flyer platforms. We expect this work to produce concrete steps towards the autonomous operation of mobile service robots inside space stations.

Keywords: Space robotics, Free-flyer, Mobile service robots, Load transportation, Space station.

Introduction

The Evolvable Mars Campaign (EMC) is a quarter-century long term plan devised by NASA with the goal of roadmapping the human presence on Mars [2, 6]. An intermediate step in this plan consists in the return to the Moon, both in orbital space and with actual landers. A component instrumental for this endeavour is the design and construction of a new space station, called Deep Space Gateway (DSG) [3, 4], orbiting the Moon on a near rectilinear halo orbit [17]. This station will be an outpost fundamental for the validation of deep space technology, as well as to support a Moon lander mission. Later on, the DSG is expected to be part of the spacecraft that will do the journey to Mars.

The construction and operation of the DSG will require multiple launches, based on the SLS launcher currently being developed by NASA, including both crewed and uncrewed missions. Some of this missions include the deployment of cargo into the DSG, e.g., astronaut supplies, science equipment, etc. Currently, multiple options for the delivery of cargo are currently being studied, e.g., including both comercial expendable vehicles and specific logistic modules co-manifested with the Orion module on a SLS launch [3].

Once a logistics module (LM) is docked at the DSG, its cargo has to be transfered to the station. There are two kinds of cargo to transport: pressurized and unpressurized. Here we focus on the transfer of pressurized cargo into the DSG habitable area. However, since the station is not expected to be crewed all the time, the use of autonomous robots for this transfer is an highly likely option.

This paper is primarily focused on the problem of cargo transport from a LM into (as well as from) the DSG using autonomous service robots. Moreover, we propose that these robots should be based on free-flying robots. These are a well-studied platform class in microgravity, with operating cases such as the NASA SPHERES [10], the JAXA IntBall [11], the DLR/Airbus CIMON, and the NASA Astrobee [1]. However, these platforms are neither designed nor capable of moving cargo of medium/large mass.

The NASA SPHERES free-flyer robot [10], deployed on the ISS in 2006, was designed to be an experimental platform. It is propelled by 12 nozzles fed by a pressurized CO2 bottle inside the robot. It weighs 4Kg and its diameter is 25cm. Its locomotion is limited to a volume inside the Destiny ISS module, due to the pseudolite localizaton system based on combined infrared and ultrasound positioning. The maximum thrust of a SPHERES is 0.58N.

The JAXA IntBall is a smaller free-flyer [11], primarily designed to be an autonomous flying camera. Its diameter is 15cm, uses micro-propellers for locomotion, and its navigation is currently based on visual fiducials. The attitude control is based on miniaturized reaction wheels. It was deployed on station in 2017 and currently operates on the Japanese module on the ISS (JEM).

The DRL/Airbus is another recently deployed free-flyer robot[1]. It was designed to be an astronaut assistant in the form of a conversational robot. The conversation interface build upon the well-known Watson AI engine by IBM. It also includes an array of cameras for documenting activities on station and its navigation is also vision-based.

The NASA Astrobee free-flyer robot [1], expected to be deployed in late 2018, is seen as the successor of SPHERES on station. The propulsion is based on servo controlled nozzles expelling low pressure air, obtained from two counter rotating centrifugal pumps and stored in a plenum. The maximum thrust is 0.6N and the robot weighs 6Kg. It is equipped with multiple cameras, RGB and depth, one of them being used for vision-based navigation. This allows the Astrobee to move, in principle, along the entire ISS. This robot also includes two bays, one at the bottom to host scientific experiments, and one at the top where a 2 DoF perching arm will be mounted.

In this paper we describe the Space CoBot free-flyer robot, originally proposed in [13], initially designed to perform collaborative tasks with astronauts. Its design is focused on Human-Robot Interaction (HRI) and on mobile manipulation. However, being an autonomous robot, its design is applicable for uncrewed space stations, although pressurized.

This paper is structured as follows: section 2 discusses the challenges of load transportation by autonomous free-flyer robots, section 3 presents the free-flyer robot design of Space CoBot, followed by sections 4 detailing its guidance system, then, sections 5 and 6 present our current work on two of the above mentioned challenges, namely the object approach and its inertial properties estimation, finally, section 7 wraps up this paper by drawing a few conclusions.

Challenges of load transportation

We consider here the problem of transporting a load from one place to another, in microgravity, performed autonomously by one or more free-flyer robots. For instance, consider the scenario where a fleet of robots has to unload a cargo ship and transport the load to some designated area inside an orbiting space station, such as the DSG. Figure 1 depicts the interior of a cargo spacecraft together with an astronaut transporting a load inside the ISS.

In what follows, we will break down this problem into a sequence of phases, to systematize the discussion:

1. *Navigation to the pickup area*: one of more robots will have to navigate from their current positions to the area where the target load is expected to be. In the case of more than one robot, there is the need of coordinated navigation, so that a minimum safe distance is always kept among them.

2. *Detection, identification, and tracking*: once in the same area where the target load, the fleet has to detect and identify the target load. We assume here computer vision based method. This can be done either by a designated robot, or by multiple robots cooperatively. The detection and identification can be aided by visual markers, as current RFID technology is not sufficiently accurate for robot manipulation. These visual markers may convey both load identification and geometrical shape of the load (e.g., a chess painted covering cloth). Once identified, the target load has to be tracked along time.

3. *Approach to the target load*: once tracked by one (or more) robots, these have to approach the load with the goal of attaching physically to it. This requires an accurate motion control of the robot, in closed loop with the visual tracker (also known as visual servoing), so that the load can be approached safely while within the field of view of the robot(s) camera(s). This problem is further studied in section 5.

4. *Grasping the target load*: at a sufficiently close distance, one (or more) robot(s) need to physically attach to the load. The design of the grasp scheme is particularly challenging, from a mechanical point of view. One possibility is to wrap the load with a net-like fabric that allows for a large variety of grasp configurations. The grasping mechanism of the robot is also a challenging problem: on the one hand, a strong grip is needed to both overcome the inertia of the load and allow maneuvering its trajectory along the transport, and on the other, the unlocking of the mechanism must be sufficiently robust to prevent it from being stuck once unlock from the load.

5. *Load transportation*: after grasping the object, the robot(s) need to navigate inside the station while physically attached to the load. However, accurate navigation, specially inside the confined areas of a space station, requires a good dynamical model. In particular, an accurate knowledge of the load inertia

[1] `http://www.dlr.de/dlr/en/desktopdefault.aspx/tabid-10212/332_read-26307/#/gallery/29913` (Retrieved September 2018).

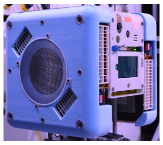
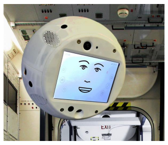
	 SPHERES	 Astrobee	 IntBall	 CIMON	 Space CoBot
Mass	4Kg	6Kg	1Kg	5Kg	6Kg
Size	25cm diameter	30cm cube	15cm diameter	32cm diameter	45cm x 21cm
Propulsion	press. CO2 gas nozzles	low press. air nozzles	ducted fans	ducted fans	propellers
Thrust	0.58N	0.6N	1mN	?	10N

Table 1: Design specifications of the free-flyer robots mentioned in the text.

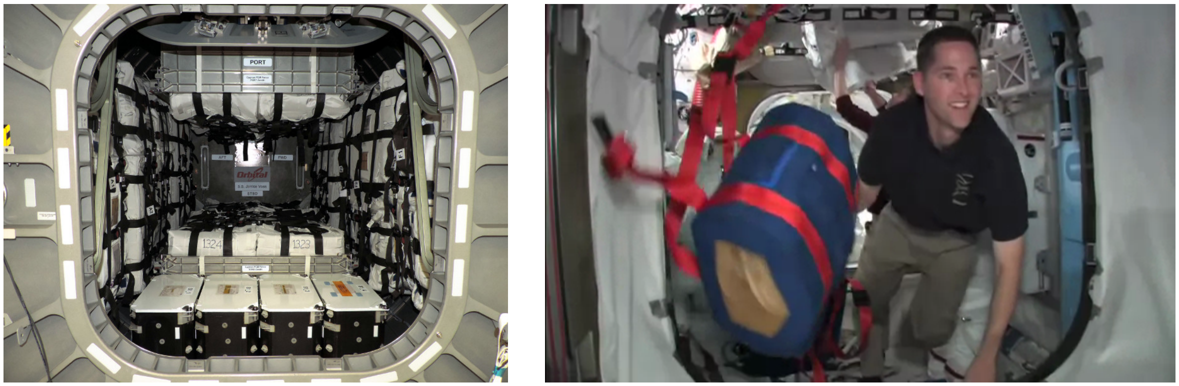


Fig. 1: Photographs of the interior of a Cygnus spacecraft loaded with cargo (left) and cargo being transported by astronauts inside the ISS (right).

properties enables the use of model-based control of the robot(s) motion. We further divide this challenge into two sub-challenges:

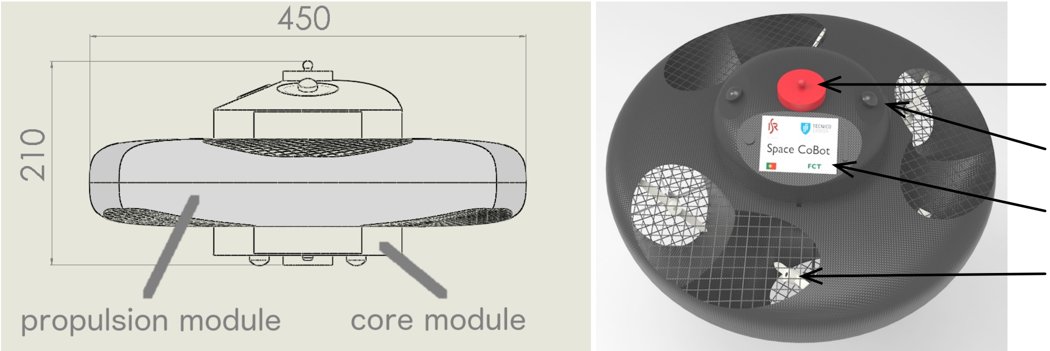
(a) *Inertial parameter estimation*: one effective way of estimating the inertia properties of a load, namely mass, center of mass, and moment of inertia, is for the robot to track a predefined trajectory, collect position, attitude, force, and torque data, and perform data fitting. Our study of this problem comprises of (a) the determination of good trajectories that maximize the information, (b) tracking of the obtained trajectory, and (c) estimation of the inertia parameters. Section 6 details our approach, validated in simulation.

(b) *Model-based control*: with a good estimation of the inertia properties of the load, the control of the robot(s) plus load can be approached in a principle way, using model-based control methods, such as the well-known Model Predictive Control (MPC). We have used this method to evaluate our above mentioned inertial parameter estimation method (see Section 6). The use of more than one robot requires a precise coordination among them. The use of decentralized MPC methods is an area of active research.

6. *Placement of the load*: given a drop-off location, the robot(s) need to locate and move the load to its target pose. Depth sensors are particularly useful to determine both a feasible pose and a trajectory that is collision free. Since the robot(s) are attached to the load, occlusions to such a depth camera pose constraints on the trajectories to approach the target pose. As in the previous item, an accurate motion control is paramount here.

7. *Detachment from the load*: once the load is placed on the target location, the mechanism grasping it needs to detach from the load. Without any astronaut to help overcoming mechanical issues, a high degree of robustness is required to prevent, e.g., entanglement of the robot with the load.

Some of these phases poses quite challenging problems that need to be addressed by the autonomous robotics community. We believe that the use a fleet of free-fliers robots are a strong candidate solution, given both their mobility and the redundancy that a multi-robot system inherently offers.



(a)

(b)

Fig. 2: Design of the Space CoBot: (a) side view with the core and propulsion modules indicated with dimensions in millimeters; (b) CAD rendering showing some of the main components.

Space CoBot design

The design of Space CoBot is modular, comprising an outer propulsion module and an inner core module, illustrated in Figure 2. The propulsion module comprises six propellers, arranged in such a way the motion is fully holonomic. We further detail this module below. The core module contains the power supply (battery), interface electronics, computers, a touchscreen, an array of video cameras for perception and telepresence, and an inertial measurement unit (IMU). This module can be easily extended with other components, such as a robotic arm. The user interface can either be based on the onboard touchscreen or on a remote device, such as a tablet or a wearable device. Current multitouch technology is very sensitive to touch, so we expect the touch pressure to be negligible when compared with the weight of the robot and its capability of using its propulsion system to compensate any motion provoked by the touching. We consider vision-based navigation, using the camera array. At the opposite sides of the core module we include two docking ports, for two purposes: (1) charging the batteries and (2) attachment to another Space CoBot (see Figure 3). The design of these docking ports follows previous work described in [8].

The propulsion module is based on a hexarotor design, where the rotation axes of the six propellers are not parallel. Rather, each propeller $i = 1,\ldots,6$ has an angular offset ϕ_i from the Z axis, as shown in Figure 4. Each such propeller contributes with a force F_i and a torque M_i on the robot center of mass (CoM), given by the following expression [13], following the notation of this figure:

$$\begin{aligned} \begin{pmatrix} F_i \\ M_i \end{pmatrix} &= a_i u_i \\ a_i &= \begin{pmatrix} K_1 \hat{u}_i \\ K_1 r_i \times \hat{u}_i - w_i K_2 \hat{u}_i \end{pmatrix} \end{aligned} \tag{1}$$

where × denotes vector cross product, u_i is the actuation signal, and w_i is a flag with value either -1 or 1

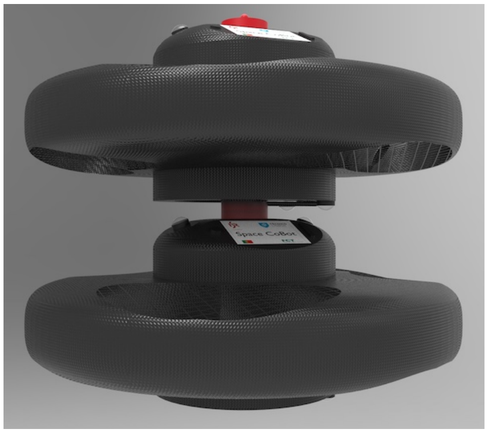


Fig. 3: Render of two Space CoBots rigidly attached through their docking ports.

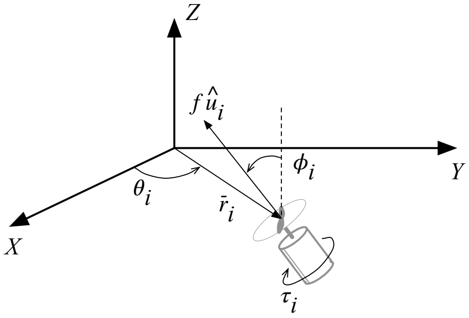


Fig. 4: Notation used for modeling a single propeller with respect to the body frame of the robot, centered on its center of mass.

depending on whether the propeller rotates clockwise or anti-clockwise for a positive thrust long $\hat{u}_i$. Constants K_1 and K_2 are given by

$$K_1 = \rho D^4 C_T \qquad K_2 = \frac{\rho D^5}{2\pi} C_P \tag{2}$$

where ρ is the air density, D is the propeller diameter, and the C_T and C_P are blade dependent dimensionless constants called thrust and power coefficients, following the momentum-blade element theory [9]. The actuation signal is $u_i = n_i^2$, where n_i is the blade revolutions per second of propeller i. We will adopt u_i as the input signal for the i-th motor controller, to maintain a linear relation between actuation and forces/torques.

Since the contributions of each propeller is expressed in the same reference frame, the net force and torque will be given by the sum of these contributions. This sum can be put in matrix form as

$$\begin{pmatrix} F \\ M \end{pmatrix} = \mathbf{A}\, u \tag{3}$$

where $\mathbf{A} = [a_1 \cdots a_N]$ is a square matrix, hereby called *actuation matrix*, and $u = [u_1 \cdots u_N]^T$ is the actuation input vector. The crucial observation is that, if the actuation matrix $\mathbf{A}$ is at least rank 6, the linear equation (3) can be solved for u for any given combination of F and M. A necessary (but not sufficient[2]) condition for this to be true is to have at least 6 propellers, thus justifying the hexarotor design. Note that this matrix only depends on these parameters: angles $\{\theta_i\}$ and $\{\phi_i\}$, distance d, the trust coefficients K_1, K_2, and $\{w_i\}$.

As mentioned before, if the axes of all propellers were parallel, $\{\phi_i = 0\}$, holonomy would be lost. The question is then which angle values to choose. Considering that the thrust is bounded, the value of these angles hold a direct relationship with the maximum force and torque attainable along an arbitrary direction. For instance, when all propeller axes are parallel, one gets maximum thrust along the Z axis, but zero along any orthogonal direction; as these angles depart from zero, we are able to tradeoff the maximum thrust along Z with non-zero maximum thrust along any given orthogonal directions. In [13] we addressed this problem by formulating it as a multi-criteria optimization problem. We summarize the main results below.

We consider each actuation signal to be bounded between -1 and 1, that is,

$$-1 \leq u_i \leq 1 \quad \text{for} \quad i = 1, \ldots, 6 \tag{4}$$

since (3) is linear, and thus the optimization problem is invariant to the scaling of these signals. According to (3), this hypercube will map onto a 6-dimensional convex polyhedron[3] in the (F, M) space. Our goal will be to find the configurations of angles $\{\phi_i\}$ and flags $\{w_i\}$ that maximize the range of forces (and torques) over all directions. Geometrically, this corresponds to changing $\{\phi_i\}$ and $\{w_i\}$ such that a ball of nonzero radius can fit inside the 3-dimensional convex polyhedron in the F space mapped by the actuation hypercube in (4), while keeping zero torque, $M = 0$. A similar reasoning applies to the torque space M, while keeping $F = 0$.

propeller (i)	1	2	3	4	5	6
θ_i	0	60	120	180	240	300
ϕ_i	55	-55	55	-55	55	-55
w_i	-1	1	-1	1	-1	1

Table 2: Design parameters of the selected solution. Both $\{\theta_i\}$ and $\{\phi_i\}$ are expressed in degrees.

Since we intend to both maximize force and torque, we make the trade-off between the two explicit by taking a multi-criteria optimization approach. The formulation of this problem is detailed in [13], having the following form:

$$\begin{aligned} &\text{minimize}\,(p, q) \\ &\text{subject to:} \\ &\quad p \geq \|b_i\|^2, i = 1, \ldots, 6 \\ &\quad q \geq \|c_i\|^2, i = 1, \ldots, 6 \end{aligned} \tag{5}$$

In this problem, the optimization variables are $\{p, q\} \cup \{\phi_i\} \cup \{w_i\}$ and the cost functions are p and q. For the sake of symmetry we kept the angles $\{\theta_i\}$ equally spaced in 60 deg intervals. Intuitively, p^{-1} and q^{-1} correspond to the radius of the above mentioned balls in the F and M spaces, respectively. Minimizing p or q, subject to the problem constraints, correspond to maximizing the radius of the balls that fit inside the polyhedrons mapped by the bounded actuation.

The solution of this multi-criteria optimization problem is the set $\mathcal{P}$ of non-dominated solutions, also known as the *Pareto optimal set* [15]. We obtained a numerical approximation to this set, from which we chose a solution maximizing the force component, since the corresponding maximum torque is not significantly lower than other non-dominated solutions [13]. This solution[4] is shown in Table 2. All of the following results shown in this paper use this selected configuration.

Guidance, Navigation, and Control

The dynamical model of the vehicle can be derived from the Newton and Euler equations of motion. Let us denote the position and velocity of the body frame $\mathcal{B}$,

[2] Sufficiency requires $\mathbf{A}$ to be full rank.

[3] A convex polyhedron is an intersection of a finite number of half-spaces.

[4] The angles are rounded off to the nearest degree unit for the sake of simplicity. However, the impact of this on the cost is minimal.

centered on the vehicle CoM and aligned according to Figure 4, with respect to the inertial frame $\mathcal{I}$ as x and v, the rotation matrix of frame $\mathcal{B}$ with respect to $\mathcal{I}$ as $\mathbf{R}$, and the angular velocity of the vehicle in the body frame $\mathcal{B}$ as ω. Then,

$$\begin{cases} \dot{x} = v \\ m\dot{v} = \mathbf{R}F \\ \dot{\mathbf{R}} = \mathbf{R}\mathbf{S}(\omega) \\ \mathbf{J}\dot{\omega} = M - \omega \times \mathbf{J}\omega \end{cases} \tag{6}$$

where the constants m and $\mathbf{J}$ are the vehicle's mass and moment of inertia, while $\mathbf{S}(\omega)$ is the skew-symmetric matrix defined by

$$\mathbf{S}(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{7}$$

for $\omega = [\omega_x \; \omega_y \; \omega_z]^T$. Note the absence of the gravity force in this model.

The approach used for the motion control of the vehicle exploits its holonomic design by decoupling the translational and rotational modes. To do so, we first apply feedback linearization [14] to the translational part of (6):

$$\begin{cases} \dot{x} = v \\ \dot{v} = p \\ F = m\mathbf{R}^T p \end{cases} \tag{8}$$

and then design a feedback controller for p. Since this dynamical system is diagonal and second order, a PD controller is enough to ensure exponential convergence:

$$\begin{aligned} e_x &= x - x_d \\ e_v &= v - v_d \\ p &= -k_x e_x - k_v e_v \end{aligned} \tag{9}$$

where x_d and v_d are the desired position and velocity vectors in the inertial frame $\mathcal{I}$, and k_x and k_v are the proportional and derivative gains of the PD controller.

For the attitude control we follow the exponentially convergent $SO(3)$ controller proposed in [7]:

$$\begin{aligned} e_R &= \frac{1}{2\sqrt{1+tr[\mathbf{R}_d^T\mathbf{R}]}} S^{-1}(\mathbf{R}_d^T\mathbf{R} - \mathbf{R}^T\mathbf{R}_d) \\ e_\omega &= \omega - \mathbf{R}^T\mathbf{R}_d\,\omega_d \\ M &= -k_R\, e_R - k_\omega\, e_\omega \\ &\quad + S^{-1}(\mathbf{R}^T\mathbf{R}_d\omega_d)\mathbf{J}\mathbf{R}^T\mathbf{R}_d\omega_d + \mathbf{J}\mathbf{R}^T\mathbf{R}_d\dot{\omega}_d \end{aligned} \tag{10}$$

where $\mathbf{R}_d$ is the rotation matrix of the desired attitude, ω_d is the desired angular velocity vector, with k_R and k_ω as controller gains. The S^{-1} function performs the inverse operation as the one defined in (7), that is, recovers the vector from a given skew-symmetric matrix.

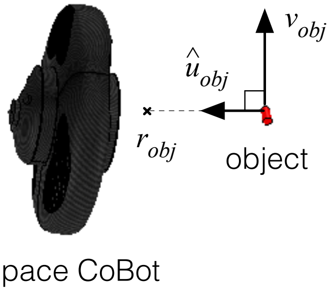


Fig. 5: Geometry of the object grasping problem.

Object tracking and grasping

In this section we address the problem of grasping a free floating object in microgravity by the Space CoBot. That is, we consider this robot to be able to detect and track free floating objects, e.g., by the use of its camera array, and to be equipped with a device capable of grasping a tracked object.

To provide a proof of concept for this task, we considered a small object flying freely in an uncluttered environment. Since it is free, the linear and angular velocities are constant. To grasp it we consider a three phase process: first, the Space CoBot matches its trajectory at a distance D_1 of the object; second, it approaches the object at a constant velocity with respect to it until a threshold distance D_2 is achieved, where we consider it to be within reach; finally, the object is grasped by the vehicle.

Approach

The geometry of the problem is shown in Figure 5. Let x_{obj} and v_{obj} be the object current position and velocity. We define $\hat{u}_{obj}$ to be the unit vector orthogonal to v_{obj} such that $x_{obj} + \hat{u}_{obj}$ is closest to the Space CoBot position: letting $\hat{v}_{obj} = v_{obj}/\|v_{obj}\|$ be the object velocity normalized to a unit vector and $s = x - r_{obj}$ be the displacement from the object to the robot, $u = s - \hat{v}_{obj}^T s\,\hat{v}_{obj}$ is the projection of this displacement on the plane orthogonal to v_{obj}, from which $\hat{u}_{obj} = u_{obj}/\|u_{obj}\|$ results from normalization. The position reference for tracking the object, r_{obj}, is defined along $\hat{u}_{obj}$, that is:

$$r_{obj} = x_{obj} + d\,\hat{u}_{obj} \tag{11}$$

In the first phase of the process, the Space CoBot matches its trajectory by tracking $x_d = r_{obj}$ and $v_d = v_{obj}$ with $d = D_1$, with an attitude of $R_d = [\hat{v}_{obj}; \hat{u}_{obj}; \hat{v}_{obj} \times \hat{u}_{obj}]$ and $\omega_d = [0;0;0]^T$. In the second, it approaches the object at a constant velocity along $\hat{u}_{obj}$ by decreasing d at a constant rate from D_1 to D_2. Finally, when the object is sufficiently close to the reference at $d = D_2$, it is grasped by the vehicle.

Simulation Results

The realistic simulator V-REP [12] was used to validate the approach. Results showing convergence of

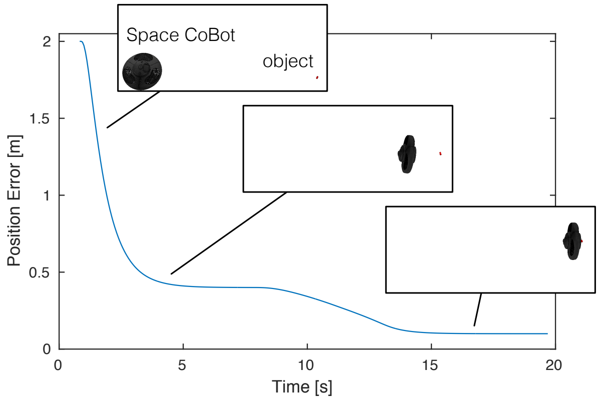


Fig. 6: Simulation results for object grasping: distance from Space CoBot to the object position, along the three phases explained in the text.

the controller to arbitrary setpoints have been previously shown in [13], including sensitivity to localization noise and to unmodeled dynamics, e.g., an heavy load attached to the robot.

Figure 6 shows a simulation result of the object grasping task. The target object is moving at a constant velocity of 0.03m/s along the Y axis, while Space CoBot goes through the three phases described above, also indicated in the figure. The distance between the Space CoBot and the object is plotted in this figure. Two plateaus are visible: at $D_1 = 0.4m$, where Space CoBot matches the object velocity, and at $D_2 = 0.1m$, when we consider the object to be reachable by the robot.

Inertial parameter estimation

Often, model-based controllers are used to carry out precise robotic missions like docking or repair in microgravity environments. In these cases, the tracking performance of the system depends on the knowledge of its parameters. Our proposed method is an extension of [16], where it was used for parameter identification of ground robots. It involves making the robot track a Fourier Series based excitation trajectory, which has been optimized for maximum accuracy of the estimated parameters. Motion data gathered during this time is then used for estimation. We adapt this algorithm for free-flying space robots, where we provide excitation in the inertial space, instead of the joint space. If the robot has a manipulator arm, we propose to keep that arm in a fixed position during excitation. Further, we consider the possibility that more harmonics might be present in the executed trajectory than the excitatory reference, thus accounting for imperfect trajectory tracking. More details about this method can be found in our previous work [5].

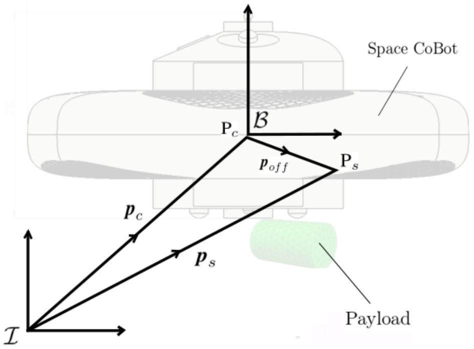


Fig. 7: Geometry of a free-flying robot (here, the Space CoBot) transporting an arbitrary load

Dynamics of a free-fying robot grasping an object

Fig.7 shows the co-ordinate system of a free-flying robot. For simplicity, the payload is considered to be directly attached to the robot. The origin of the body frame of the robot, $\mathcal{B}$, is its center of mass, P_c. The new center of mass (CoM) after grasping of the payload is P_s. The position vectors of these CoMs with respect to the inertial frame, $\mathcal{I}$, are $\boldsymbol{p}_c$ and $\boldsymbol{p}_s$. The vector denoting the offset between them, $\boldsymbol{p}_{off}$, is expressed in the body frame, $\mathcal{B}$. The relation between them is:

$$\boldsymbol{p}_s = \boldsymbol{p}_c + \mathbf{R}\,\boldsymbol{p}_{off} \tag{12}$$

$$\dot{\boldsymbol{p}}_s = \dot{\boldsymbol{p}}_c + \mathbf{R}\left(\boldsymbol{\omega} \times \boldsymbol{p}_{off}\right) \tag{13}$$

where $\mathbf{R}$ is the rotation matrix of frame $\mathcal{B}$ with respect to $\mathcal{I}$.

Newton-Euler equations for the acceleration of the vehicle's center of mass P_c with respect to the inertial frame, and the rotation of the system in its body frame are:

$$m\left\{\ddot{\boldsymbol{p}}_c + \mathbf{R}\left(\dot{\boldsymbol{\omega}} \times \boldsymbol{p}_{off} + \left(\boldsymbol{\omega} \times \left(\boldsymbol{\omega} \times \boldsymbol{p}_{off}\right)\right)\right)\right\} = \mathbf{RF} \tag{14}$$

$$\mathbf{J}_s\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}_s\boldsymbol{\omega} + \boldsymbol{p}_{off} \times \mathbf{F} = \mathbf{M} \tag{15}$$

where m stands for mass of the system, $\mathbf{J}$ for its moment of inertia and the operator × denotes cross product. $\boldsymbol{\omega}$ is the angular velocity of the vehicle expressed in the body frame $\mathcal{B}$.

Formulation of a Least Squares estimation problem

In this section, we describe the method for estimation of inertial parameters using measured data. The parallel axis theorem can be used to express $\mathbf{J}$ about the origin of body frame P_c instead of the robot-object system's center of mass P_s as:

$$\mathbf{J}_c = \mathbf{J}_s + m\left[(\boldsymbol{p}_{off}^T\boldsymbol{p}_{off})I - (\boldsymbol{p}_{off}\boldsymbol{p}_{off}^T)\right] \tag{16}$$

Using (14)-(16), the Newton-Euler equations can now be expressed linearly in terms of the inertial parameters to be estimated as:

$$\boldsymbol{\gamma}(\mathbf{X},\dot{\mathbf{X}},\ddot{\mathbf{X}})\boldsymbol{\pi} = \boldsymbol{\tau} \tag{17}$$

where $\boldsymbol{\gamma}(\mathbf{X},\dot{\mathbf{X}},\ddot{\mathbf{X}})$ is the regressor matrix with $\mathbf{X} = [x,y,z,\phi,\theta,\psi]$, and ψ,θ,ϕ denote the Z-Y-X Euler Angles used to express orientation relative to frame $\mathcal{I}$,

$$\boldsymbol{\gamma} = \begin{bmatrix} \mathbf{R}^{-1}\ddot{p}_c & \mathbf{S}(\dot{\boldsymbol{\omega}})+\mathbf{S}(\boldsymbol{\omega})\mathbf{S}(\boldsymbol{\omega}) & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{3\times 1} & -\mathbf{S}(\mathbf{R}^{-1}\ddot{p}_c) & [*\dot{\boldsymbol{\omega}}]+\mathbf{S}(\boldsymbol{\omega})[*\boldsymbol{\omega}] \end{bmatrix} \tag{18}$$

with $\mathbf{S}(\boldsymbol{\omega})$ representing the skew-symmetric matrix. The 10x1 vector of the parameters to be estimated, $\boldsymbol{\pi}$, is

$$\boldsymbol{\pi} = \begin{bmatrix} m & m\mathbf{p}_{off}^T & J_{xx} & J_{xy} & J_{xz} & J_{yy} & J_{yz} & J_{zz} \end{bmatrix}^T \tag{19}$$

and $\boldsymbol{\tau}$ is the vector of the applied forces and moments

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{F} \\ \mathbf{M} \end{bmatrix} \tag{20}$$

When N measurements of the states $\mathbf{X}$ (x,y,z and Euler angles, ϕ,θ,ψ) are available, a least square problem can be set up as

$$\mathbf{W}\boldsymbol{\pi} = \mathbf{b} \tag{21}$$

where

$$\mathbf{W} = \begin{bmatrix} \boldsymbol{\gamma}(X(1),\dot{X}(1),\ddot{X}(1)) \\ \vdots \\ \boldsymbol{\gamma}(X(N),\dot{X}(N),\ddot{X}(N)) \end{bmatrix} \tag{22}$$

and

$$\mathbf{b} = \begin{bmatrix} \tau(1) \\ \vdots \\ \tau(N) \end{bmatrix} \tag{23}$$

Then the solution vector $\hat{\boldsymbol{\pi}}$ is given by the linear least squares solution as:

$$\hat{\boldsymbol{\pi}} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{b} \tag{24}$$

where the regressor matrix, $\mathbf{W}$, must be full-rank. This means that the result of parameter identification depends on $\mathbf{W}$, which is a function of $X,\dot{X},\ddot{X}$. Therefore, appropriate excitation trajectories are necessary for finding good estimates.

The excitation trajectory - Generation and tracking

Excitation trajectories are generated using finite Fourier Series. They are periodic, and have limited bandwidth. These properties facilitate noise-filtering, as the measured data can be averaged over cycles due to its periodic nature. This improves the signal to noise ratio. Further, knowing the bandwidth of the reference makes it straightforward to filter out the unwanted frequencies appearing in the measured data. Each of the constituents of the pose, x,y,z and ϕ,θ,ψ, are represented as a Fourier Series with a known angular frequency ω_f, where $\omega_f = 2\pi/\mathbf{T}_f$ and $\mathbf{T}_f$ is the period. The number of harmonics is n. The trajectories X_i, where $\boldsymbol{X}$ = $[x,y,z,\phi,\theta,\psi]$, and i = 1 to 6 are written as

$$\begin{aligned} X_i(t) &= a_{io} + \sum_{k=1}^{n} \frac{a_{ik}}{\omega_f k} sin(\omega_f kt) - \frac{b_{ik}}{\omega_f k} cos(\omega_f kt) \\ \dot{X}_i(t) &= \sum_{k=1}^{n} a_{ik} cos(\omega_f kt) + b_{ik} sin(\omega_f kt) \\ \ddot{X}_i(t) &= \sum_{k=1}^{n} -a_{ik}\omega_f k sin(\omega_f kt) + b_{ik}\omega_f k cos(\omega_f kt) \end{aligned} \tag{25}$$

The Fourier coefficients, a_{io}, a_{ik}, b_{ik} for k = 1 to n, and i from 1 to 6, are denoted by vector $\boldsymbol{\delta}$. The value of these co-efficients is obtained by minimizing the criterion J_1

$$J_1 = cond(\mathbf{W}) - \sum_{i=1}^{6} q_i \sqrt{\frac{1}{N}\sum_{\eta=1}^{N} |X_i(\eta)|^2} \tag{26}$$

J_1 minimizes the condition number, which is a ratio of the largest singular value of the regressor matrix to the smallest one. Minimizing it would bring this ratio close to 1, rendering all parameters as equally observable as possible. The minimum condition number criterion is modified by adding a term that maximizes the root mean square (RMS) level of the Fourier Series, or the square root of the wave's power. This is done in order to avoid minuscule and infeasible trajectories. In the above equation, N are the number of elements present in each series X_i from time 0 to T_f. The vector of weights given to each of the six RMS terms is q. The constraints of this minimization problem are spatial limits, as well as those on the velocities and accelerations of the robot. Additionally, the robot is confined to rest state at the beginning and end of each period. The optimization problem for calculating the excitation trajectory can be represented as:

$$\begin{aligned} \min_{\delta} \quad & J(\mathbf{X},\dot{\mathbf{X}},\ddot{\mathbf{X}}) \\ \text{subject to:} \quad & X(1) = 0, \dot{X}(1) = 0, \ddot{X}(1) = 0 \\ & X(N) = 0, \dot{X}(N) = 0, \ddot{X}(N) = 0 \\ & X_{min} \leq X(t) \leq X_{max} \\ & \dot{X}_{min} \leq \dot{X}(t) \leq \dot{X}_{max} \\ & \ddot{X}_{min} \leq \ddot{X}(t) \leq \ddot{X}_{max} \end{aligned} \tag{27}$$

The robot with the grasped object is now made to track C number cycles of the excitation trajectory using a convergent controller. Clearly, the robot will require more thrust to transport the object, as compared to transporting its own weight. We use Non-linear Model Predictive

Control (NMPC) in simulations, so that the controller takes the robot's actuation limits into account. NMPC uses the model of the robot that was known before it attached to the load.

Treating the measured data

Estimating velocity and acceleration from measured pose data is non-trivial due to sensor noise. Numerically differentiating these noisy measurements will result in discontinuities and huge errors in the obtained velocities and accelerations. To avoid this, we resort to analytical differentiation of the filtered measured data. Using the properties of the Fourier Series, the operations of noise-filtering and fitting a basis function to the waveform can be done in the same step.

First, the values of the periodic position and attitude measurements are averaged over C cycles. Next, the Fast Fourier Transform (FFT) is used. As the excitation is limited in bandwidth, all harmonics in the reference trajectory are expected to be present in the executed one. If the tracking is perfect, all other frequencies can be filtered out, claiming that they are high frequency noise. So, the Fourier Series co-efficients corresponding to the unwanted frequencies are set to zero, and the remaining co-efficients give the Fourier Series fit of the filtered wave. They are related to the positions, velocities and accelerations by (25). The matrix of regressors, $\hat{\mathbf{W}}$ is estimated from eqs. (18) and (22). Finally, the linear least squares solution, (24), is used to obtain the parameter estimates.

Algorithm 1: Parameter estimation algorithm

```
Find excitation trajectory, $\mathbf{X}_{ref}$, with $n_{ref}$ harmonics
Make robot track $C$ cycles of excitation with payload
Calculate average of the motion data ($\tilde{\boldsymbol{X}}$) over cycles
for $n \in \{n_{ref}, ..., n_{end}\}$ do
    Find FFT of $X_i$, i = 1 to 6
    Coefficients of harmonics $\geq n$ are set to 0
    Find $X_i$ by inverse FFT, & $\dot{X}_i$ and $\ddot{X}_i$ analytically
    $\mathbf{W} \leftarrow$ CalculateRegressorMatrix($\mathbf{X}$,$\dot{\mathbf{X}}$,$\ddot{\mathbf{X}}$)
    $\hat{\pi} \leftarrow$ LeastSquaresSolution($\mathbf{W}, \tilde{\mathbf{b}}$)
    $||\mathbf{e}_n|| \leftarrow$ LeastSquaresResidual($\hat{\pi}, \mathbf{W}, \tilde{\mathbf{b}}$)
end
Determine $\hat{\pi}*$ as the solution with least $||\mathbf{e}_n||$
```

Harmonic selection based on the least square residual

However, our controller does not place limits on bandwidth at the time of execution, like other works using this approach [16]. Further, the robot might not track the reference perfectly or the actuation might be saturated. As a result, more frequencies than the reference could appear in the executed trajectory, in addition to high frequency measurement noise. The number of harmonics that is sought, n_{filt}, should model the measured data and exclude measurement noise the best, leading us to the goal of most accurate parameter estimates. This value is determined autonomously by using n_{filt} that results in the lowest Least Squares Residual error.

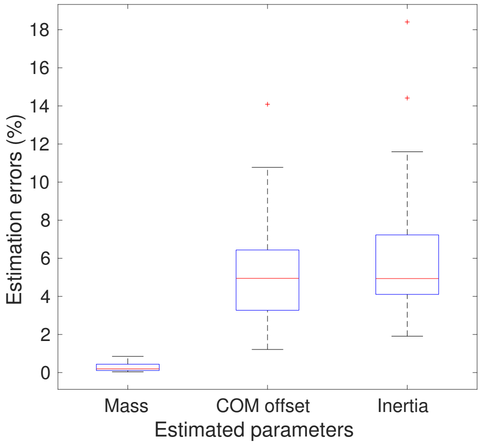


Fig. 8: Percent estimation errors for mass, inertia and centre of mass offset parameters

The following method is used:

1. The number of harmonics are increased incrementally from n_{ref} (the number of harmonics present in the reference trajectory), to n_{end}, a chosen value, $n_{end} > n_{ref}$. For each n, the algorithm is followed till the end, where the parameter estimates are found.

2. The residual error is then calculated as

$$\mathbf{e} = \tilde{\mathbf{b}} - \mathbf{W}\hat{\boldsymbol{\pi}} \tag{28}$$

where $\tilde{\mathbf{b}}$ is the vector of the measured forces and torques. We consider equal standard deviations for these measurements.

3. The value of n at which the norm of the residual error, $||\mathbf{e}||$ is the least, is finally selected as the best number of harmonics, n_{filt}.

Simulation Results and final remarks

Fig.8 presents a compilation of estimation results for five normally generated reference loads. A mean (μ) of 8.8 Kg for mass, with standard deviation (σ) of 2 Kg was used. A μ of 0.0m for the centre of mass offset, with σ = 0.08m was used. For the off-diagonal values of the inertia tensor, μ = 0.06 and σ = 0.01 kg-m^2 and for the diagonal values, a mean value of 0.15 kg-m^2, with σ = 0.05. Five excitation trajectories were computed using the optimization of criterion J. Each of the five unmodeled loads was estimated using one of the five excitation trajectories, resulting in a total of twenty five estimates. The simulations were carried out in Matlab, considering the Space Cobot. Measurements were corrupted with zero mean Gaussian noise with a standard deviation of 0.05 for the position values and 0.03 for the attitude values.

As a measure of the accuracy between the parameter estimates and their true values, and for ease of illustration, the percent *norm* estimation error was used. Estimates of mass, inertia and CoM offset were considered separately, and the norm of the errors was calculated. E.g., the CoM offset error was found as

$$p_{off,error} = \sqrt{\sum_{i=1}^{3} (\hat{p}_{off,i} - p_{off,i})^2} \tag{29}$$

The norm inertia error was found in a similar manner, whereas for mass, simply the absolute value of the error was used. The percentages that these errors form of the true values of mass, or norm of inertia, or norm of offset, are the percent estimation errors shown in the figure. Considering the example of the CoM offset,

$$\%\ p_{off,error} = \frac{p_{off,error} \times 100}{\sqrt{\sum_{i=1}^{3} (p_{off,i})}} \tag{30}$$

Fig. 8 shows that the combined robot-object mass is estimated the more accurately than the inertia and center of mass offset in the simulation runs. In our case, the mass parameters are more than ten times the value of the inertia and centre of mass parameters. Future investigation will be aimed at improving this. Moreover, as periods of the measured data are averaged for better signal-to-noise ratio, the cycles containing transient data have to be discarded. Further, if the robot encounters a high amount of input saturation while tracking the excitation trajectory, a lot of high frequencies might appear in the measured data. In those cases, the task of separating the high frequency harmonics belonging to measurement noise from those belonging to the measured data would be complicated. Thus, robustness of this method to transient and input saturation remains to be improved.

Conclusions

This paper addressed the problem of cargo transport aboard an orbiting space station by a fleet of autonomous free-flyer robots. This problem is particularly relevant in the context of the DSG, namely the need to unload cargo ships during the periods the station is uncrewed. This problem was then broken down into several challenges, and for each one we discussed possible approaches to them. Then, we presented the design of the Space CoBot, a free-flyer robot for microgravity environments, which we claim to address the requirements for autonomous load transport. Then, two of the above mentioned challenges in autonomous cargo transport are addressed, namely the approach to a target load and the inertial parameter estimation of that load, once attached to a free-flyer robot. Simulation results on both of these challenges complement the presentation.

Future work includes extending our study to other challenges of the load transportation problem, as well as the validation of the proposed methods on current (and future) experimental free-flyer robots aboard the ISS, namely the SPHERES and the Astrobee free-flyers from NASA.

Acknowledgements

This work was supported by the FCT project [UID/EEA/50009/2013].

References

[1] Maria G Bualat, Trey Smith, Ernest E Smith, Terrence Fong, and DW Wheeler. Astrobee: A new tool for ISS operations. In *Proceedings of the SpaceOps Conference*, 2018.

[2] Douglas A Craig, Patrick Troutman, and Nicole Herrmann. Pioneering space through the evolvable mars campaign. In *Proceedings of the AIAA Space 2015 Conference and Exposition*, 2015.

[3] Jason C Crusan, Douglas A Craig, and Nicole B Herrmann. NASA's deep space habitation strategy. In *Proceedings of the IEEE Aerospace Conference*, 2017.

[4] Jason C Crusan, R Marshall Smith, Douglas A Craig, Jose M Caram, John Guidi, Michele Gates, Jonathan M Krezel, and Nicole B Herrmann. Deep space gateway concept: Extending human presence into cislunar space. In *Proceedings of the IEEE Aerospace Conference*, 2018.

[5] Monica Ekal and Rodrigo Ventura. On inertial parameter estimation of a free-flying robot grasping an unknown object. In *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2018.

[6] Terry Haws, Joshua Zimmerman, and Michael Fuller. SLS, the gateway, and a lunar outpost in the early 2030s. In *Proceedings of the AIAA Propulsion and Energy Forum*, 2018.

[7] Taeyoung Lee. Exponential stability of an attitude tracking control system on SO(3) for large-angle rotational maneuvers. *Systems & Control Letters, Elsevier*, 61(1):231–237, 2012.

[8] Carlos Marques, João Cristovão, Paulo Alvito, Pedro Lima, João Frazão, Maria Isabel Ribeiro, and

Rodrigo Ventura. A search and rescue robot with tele-operated tether docking system. *Industrial Robot*, 34(4):332–338, 2007.

[9] Barnes Warnock McCormick. *Aerodynamics, aeronautics, and flight mechanics*. John Wiley & Sons, 2nd edition, 1995.

[10] D. Miller, A. Saenz-Otero, J. Wertz, A. Chen, G. Berkowski, C. Brodel, S. Carlson, D. Carpenter, S. Chen, S. Cheng, D. Feller, S. Jackson, B. Pitts, F. Perez, J. Szuminski, and S. Sell. SPHERES: a testbed for long duration satellite formation flying in micro-gravity conditions. In *Proceedings of the AAS/AIAA Space Flight Mechanics Meeting*, 2000.

[11] Shinji Mitani, Masayuki Goto, Ryo Konomura, Yasushi Shoji, Keiji Hagiwara, Shuhei Shigeto, and Nobutaka Tanishima. Crew-supportive autonomous mobile camera robot on ISS / JEM. In *Proceedings of i-SAIRAS*, 2018.

[12] E. Rohmer, S. P. N. Singh, and M. Freese. V-REP: a versatile and scalable robot simulation framework. In *International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[13] Pedro Roque and Rodrigo Ventura. Space CoBot: modular design of an holonomic aerial robot for indoor microgravity environments. In *IEEE/RSJ International Conference On Intelligent Robots And Systems (IROS)*, 2016.

[14] Shankar Sastry. *Nonlinear Systems: Analysis, Stability, and Control*. Springer, 1999.

[15] Roman Statnikov and J. B. Matusov. *Multicriteria Optimization and Engineering*. Chapman & Hall, 1995.

[16] J. Swevers, C. Ganseman, D. B. Tukel, J. de Schutter, and H. Van Brussel. Optimal robot excitation and identification. *IEEE Transactions on Robotics and Automation*, 13(5):730–740, 1997.

[17] Emily M Zimovan, Kathleen C Howell, and Diane C Davis. Near rectilinear halo orbits and their application in cis-lunar space. In *Proceedings of the 3rd IAA Conference on Dynamics and Control of Space Systems*, 2017.