POLITECNICO DI TORINO

Dipartimento di Ingegneria Meccanica e Aerospaziale

Course of *"Orbital Robotics and Distributed Space Systems"*

Final Project Report

Students:

Cocuzza Santuccio Federico

Bellinazzi Christian

Bisceglia Matteo

Cavallo Federico

Professors:

Romano Marcello

Apa Riccardo

Aliberti Stefano

Matonti Catello Leonardo

A.A. 2022/2023

Contents

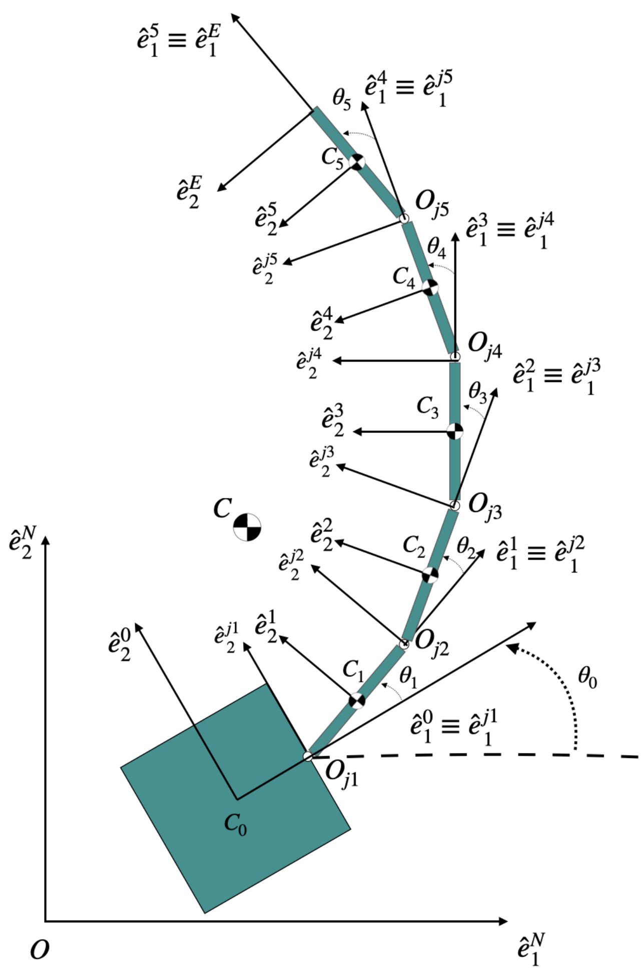


Figure 1: Free-Flyer

1 Organization & Requirements

The objective of this Report is to report the results obtained in the analysis of the following problem: Analysis S/W Tool for the planar orbital and attitude/configuration motion of a Satellite-Robotic Manipulator System orbiting Earth. To get the following objective we arranged according to the time available, as shown in **Table 1**.

Partecipants		start	end
Federico C. & Christian B.	**Part 1.1/1.2/1.3**	23/5/2023	27/5/2023
Christian B. & Federico S.	**Part 1.4**	26/5/2023	3/6/2023
All students	**Part 2.1/2.2**	28/6/2023	8/6/2023
Christian B & Matteo B.	**Report Writing**	6/6/2023	9/6/2023

Table 1: Schedule timeline

First of all, let's begin defining the most important procedural steps accomplished for this document:

- An ECI reference frame schematized in a 3-D orbital plane, whose motion has been studied assuming a circular CHIEF's altitude of $h = 500\,Km$, according to the *KR2BP*, a spherical Earth defined by a mean radius $R_T = 6371$ Km. Consequently, the eccentricity will be equal to $e_c = 0$;

- The mass of the entire satellite has been imposed to $m_{SC} = 200$ Kg, that of the base-link equal to 100 Kg and each of manipulators' links to $m_i = 15$ Kg;

- The semi-lengths relative to each of these links just described have been considered equal to $l_i = 0.3$ m, while the entire manipulator's width to $s = 0.02$ m, considering always a squared & ungrounded base body;

- The base-link has been dimensioned as a cube of half-side $l_0 = 0.45\,m$;

- We have considered the third keplerian law to define Deputy's orbital period relative to that of the CHIEF, assuming to have for a circular orbit $a = r = p$:

$$T = 2\pi\sqrt{\frac{a^3}{\mu_T}} = 2\pi\sqrt{\frac{r^3}{\mu_T}} = 5668.1\,s \quad \text{and} \quad \omega = \frac{2\pi}{T} = \sqrt{\frac{\mu}{r^3}} = 0.0011\,\frac{rad}{s}$$

  This relationships have been applied to the HCW EoMs letting us to determine relative positions & Velocities. It's needed to know the gravitational parameter $\mu = G \cdot (M_T + m_{SC}) = 3.9860 \cdot 10^{14}\,\frac{m^3}{s^2}$ and the orbital radius equal to $r = R_T + h = 6871$ Km;

- An other fundamental quantity to remember is the state-transition matrix A $[6x6]$, which has been associated to the column vector standing for all initial conditions inserted also by the user;

- Finally, it's been considered both the effects of torques and forces relative to the gravity gradient.

A 2D graphic representation was made with the actual measurements, and positioning in the link points between link-base and link-links of the Revolute Joint that allow the rotation of the arm (1 DOF), as seen in **Figure 1**. In pink you see the structure completely unfolded to observe the space occupied.

2 Analysis of the translational motion

As requested from the text, from several I.C.s imposed into the code and assigning them to the state-transition matrix A, we have been able to find the following results, assuming to fix the z-coordinate in order to obtain a 3-D chart.

2.1 Computation of the minimum $2 - \Delta V$ impulses to complete a RDV maneuver

The goal of this unit was to define the best, or optimized, path between the initial Deputy's configuration, in terms of positions and velocities, and the final desired approaching point, represented by the CHIEF itself.

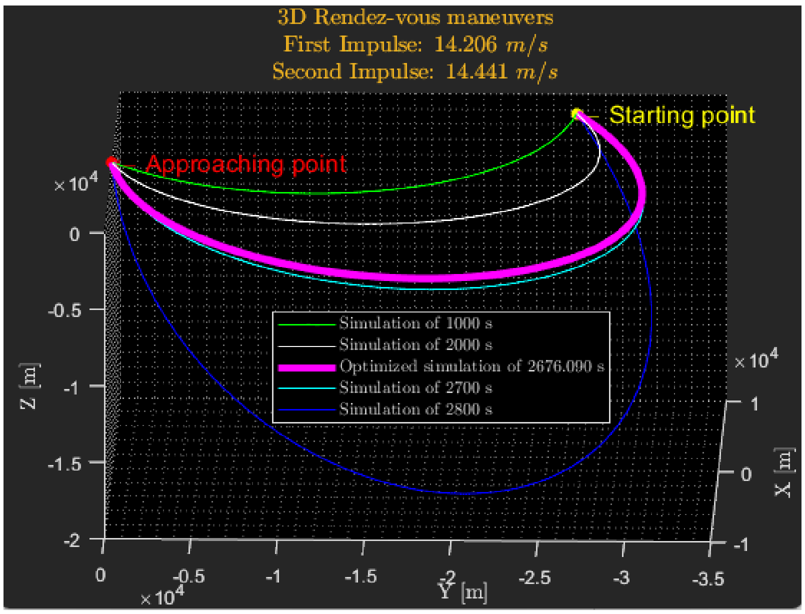


Figure 2: Double ΔV-optimized Impulse

The first passage was to apply a first impulse corresponding to the initial Deputy's position and, in the same type, looking for the final one connected to the second impulse, which is needed to block Deputy's position on that of the CHIEF.

- For each instant of time we have decided to consider both the impulses, summing them and finally assigning the final result to a pre-located cell belonging to an initial specified column vector. After that, the code looks for the minimum cell's value and then compare the previous ΔV to that just obtained, saving the index relative to that for cycle.
- It's important to underline that the cycle operates defining at each step the minimum value between these two impulses just described and, possibly, updates the newer value corresponding to that obtained by the current index.

The code in **Figure 2** could be defined optimized only through both the Impulses, so ΔV-optimized, and not considering what's happening to the required simulation's time. An other important operative choice was to represent the final plot considering also other generical paths in order to help the user identifying how the optimized one is running. The most difficulties were to learn how writing, computationally speaking, the function capable to calculate this optimized traiectory and in which way looking for the time associated to that condition. After that, the consequent challenge was been converting it as a function reclaimable as easier as possile from the main code, as described in the 'README.txt' file attached to the folder.

2.2 Game-like simulation and visualization

This animation has been made-up introducing the same I.C.s used for the ΔV-optimzed traiectory, whose final result will be: The first thing to notice is that these resulting traiectories change through specific ways, which depends directly on ΔV's components and on their relative final positions and velocities stopped until the last point touched by the Deputy on the older paths. The plan to develop this software was thincking about the following conceptual & procedural steps:

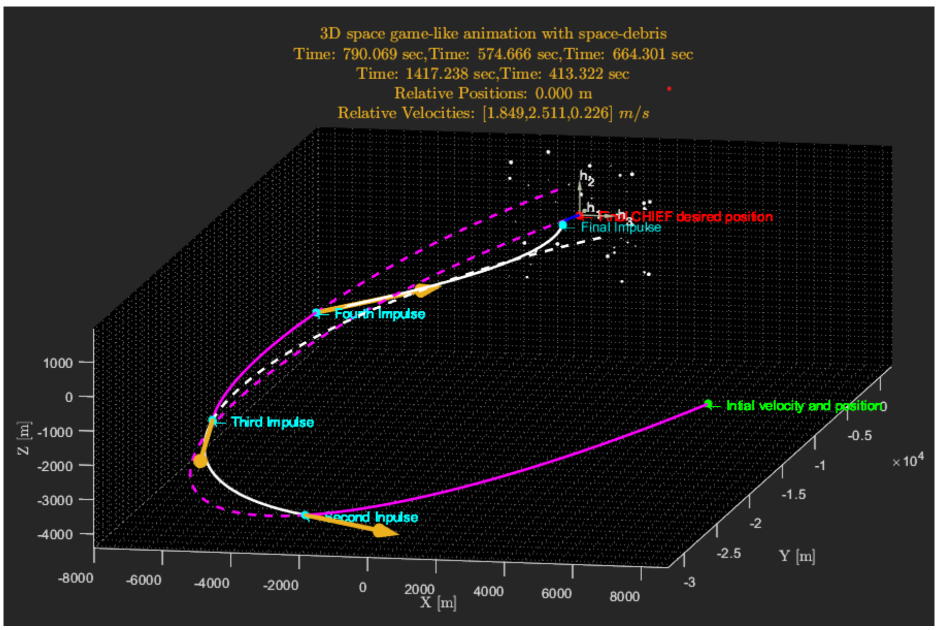


Figure 3: Game-like simulation

- In this case the software is capable to take into input newer values for each configuration desired path's change and establishing its consequence path. These I.C.s have been imposed regarding the initial optimized path (the violet-colored one starting from the green-text), so all consecutive traiectories will strictly depend on that initial curve. Following, the user will be capable to apply a maximum of 3 casual impulses and, as a self choice, evaluating each of these paths' changes in real-time;
- The game in **Figure 3** has been thought to look for a finish, so in a certain sense the user is capable to insert every desired value and, at the end, the code will run automatically run again a second ΔV's optimized appproaching maneuver to reach teh CHIEF;
- Each ΔV's triad of values imposed by the user will be showed as an arrow vector (that orange-colored) to visualize better the effects on the newer path described by the Deputy. Then also the

HCW CCS attached to a cloud of points belonging to the CHIEF has been shown, this to impress as better as possible the physical meaning of the entire maneuver;

- Other operative choices made for this game were to put current relative positions, velocities and the time required to follow that maneuver before stopping it and then applying a newer one;
- The back-ground grid helps the user a 3-D visualization of the problem and, operatively speaking, the colored shown texts represent tha most important motion-states through-out the animation. All this arrows and vectors have been scaled to specific input conditions, as a conceptual passage, in order to generilize them as more as possible to the user's input.

The highest difficulties were been to imagine in which way concatenating the previous path to that following, so looking for the correct I.C.s needed to be passed to the next for cycle along the run. This aspect is also correlated to that of deleting older paths and creating newer ones, discarding the 'hypothetical' curves not even more followed by the eputy, at ΔV's single impulse, and substituting to them a 'dotted' line with the same color. At the end it's been decided to insert also a background relaxing music to give a better sense of the attempts made and hours spent to implement it!

3 Analysis of the rotational-internal motion

The goal of this unit is to define which are the main effects of GG around our Free Flyer's COM. The initial conditions are referred to 5 robotic arms, where each of them is connected to the next by a single RJ, which is characterized by a single DOF.

- First of all, we have defined the radial unit vector by considering the ratio between COM positions and their relative norm;
- In this case the GG has been considered as an external disturbance and its relative effects have been included in wrenches' definition. It's been applied perpendicularly to the Free Flyer's orbital plane and it acts as a yaw torque-free external torque and force. Wrenches need to be defined inside the function *FFP6L5R-odefun-state2dotstate* and then it's been considered both torques and forces. In particular these last ones needs a vector named c_{base}, capable to consider the relative distances between the base-link COM and that belonging to the COM of the system itself. It's been made the same for each robotic arm, but now taking the difference between the i-th COM of each arm and obviously the COM components of the entire manipulator;
- At the end have been created the Wrench matrix containing the torque, as the first three components, and forces, as the last three. This passage has been made for each arm, concatenating each of the five columns to obtain a [6x5] dimension.
- It's been supposed to consider as initial time-derivative generalized coordinates equal quantities, in order to provide a sense of rotation for each manipulator's link, excluded that of the base-link.

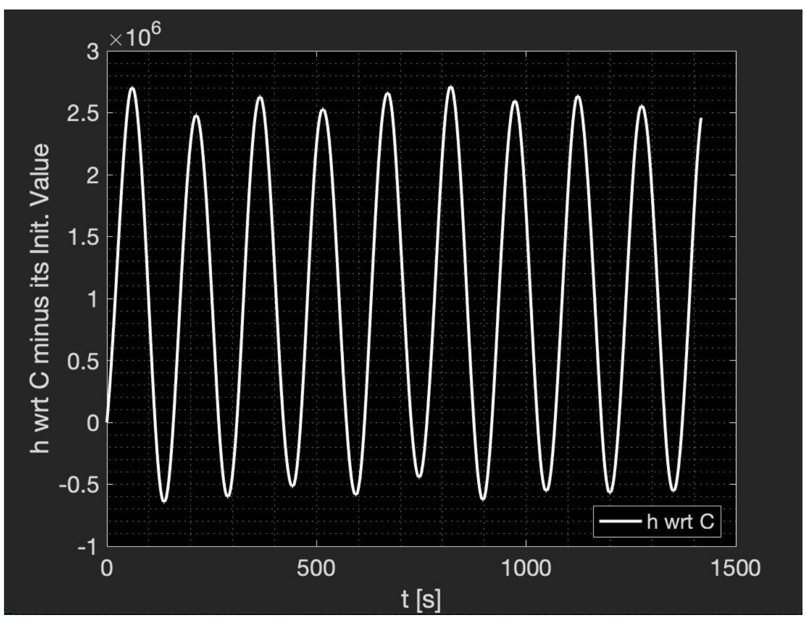
The main difficulties we have reached have been to learn exactly the position to which fix our forces and torques and how to save them into the final wrench matrix. Finally, imposing as the simulation's time an orbital quarter-period, we reached the final result and, re-arranging the rows in the code, we completed the task of this problem, improving the numerical computation and reducing general propagation errors.

3.1 Animation of the rotational-internal motion

Starting from the data obtained from the simulation, we have developed a script capable of creating a post-processing animation of the manipulator. We utilized the following data: the positions of the center of mass of the base and the 5 links. Essentially, the code is responsible for displaying on the screen, for each iteration of the for loop, the trajectory traveled up to that point in time by updating the position of the base and its respective links. The greatest challenge was to ensure that the entire structure moves coherently with the simulated data during the animation. To achieve this, we used the positions $O_{j1,j2,j3,j4,j5}(t)$ to plot the link in space. As for the base, we leveraged knowledge of the central point, the length of the side, and the position of the first vertex of link 1, which must always remain attached to the right side of the base.

4 Final Conclusions and Comments

From the implementation of $2 - \Delta V$'s optimization it's possible to conclude which is the best traiectory to bring the Deputy to the CHIEF's final position. The code requires a bit of time to elaborate all computations but the final result will be truly accurate. The same code has been used 2 times for the game-like animation in order to obtain a more fidelity program. Running it for different I.C.s, we may conclude that RDV maneuvers highly depend on initial Deputy's positions and velocities, but also on ΔV's components variations inserted by the user him/her self. During that simulation, but going through the rotational-motion analysis, it's possible to distinguish the sinusoidal and Angular Momentum, Axial Momentum and COM oscillations, always considering a time's evolution of a quarter orbital-period. COM



variations are very little to observe w.r.t. an ECI reference frame, but all this aspects will be clear once runt the final Animation, which focuses on the base-link's COM. Also final generalized time-derivative's coordinates will oscillate w.r.t. a mean value for O_{j1} and w.r.t. zero for all consequent joints.

References

[1] M. Romano, *Orbital Robotics and Distributed Space Systems, Class Notes*. Politecnico di Torino, 2023

[2] David A. Vallado, *Fundamentals of Astrodynamics and Applications, David A. Vallado, Space Technology Library*

[3] Website Reference, `https://it.mathworks.com/help/matlab/ref/getframe.html`

[4] Thesis, `https://webthesis.biblio.polito.it/10298/1/tesi.pdf`

[5] Drawnow function `https://it.mathworks.com/help/matlab/ref/drawnow.html`

[6] Roger R. Bate, Donald D. Mueller, Jerry E. White, **Fundamental of Astrodynamics**

[7] Michael J. Rycroft, Robert F. Stengel, Marcel J. Sidi, *Spacecraft Dynamics and Control, Cambridge Aerospace, series 7, Tel Aviv University*