Final Project Report

FEDERICO MUSTICH
LORENZO PORPIGLIA
GAETANA GAIA SPANÓ
VINCENZO TRENTACAPILLI

Orbital Robotics and Distributed Space Systems (Prof. Marcello Romano)
Politecnico di Torino

Contents

1 Description of the software

1.1 Translational motion

The final project of the course required different groups of students to analyze and perform a simulation of a multi-body system, consisting of a spacecraft with a 5-revolute-joints robotic manipulator, which had to perform a two-impulses rendezvous maneuver. Both the translational and rotational-internal motion of the system should have been object of our analysis and simulation. To do so, our team had to code and optimize a bunch of scripts using the MatLab environment, integrating also the functions provided by Prof. Romano within his OROLAB software. Our group produced 4 scripts in total.
The first one (**HCW_IC_MakeFigs.m**) aimed to replicate some figures reported in the book *"Fundamentals of Astrodynamics and Applications"* (D. Vallado), which plot the evolution in time of the position of Deputy (called "Interceptor") with respect to Chief given different initial conditions on position and velocity. The script uses and resolves the Hill-Clohessy-Wiltshire (HCW) set of equations in order to perform the task given.

$$\ddot{x} - 2n\dot{y} - 3n^2x = u_x$$
$$\ddot{y} + 2n\dot{x} = u_y$$
$$\ddot{z} + n^2z = u_z$$

where n is the mean motion of the orbit, given by $n = \sqrt{\frac{\mu}{(R+h)^3}}$, where R is the radius of the Earth, μ is the Earth gravitational constant and h is the altitude of the spacecraft. Since the motion is unforced, u_x, u_y and u_z are equal to zero, and the solution of the system can be brought back to the solution of a homogeneous linear time invariant system:

$$x(t) = \Phi(t, t_0)x(t_0)$$

where Φ is called *state transition matrix* and is given by $e^{A(t-t_0)}$, where A is the *state matrix* of the system. A total of 5 plots are obtained, which can be visualized by simply running the software. One of them, as an example, has been attached below.

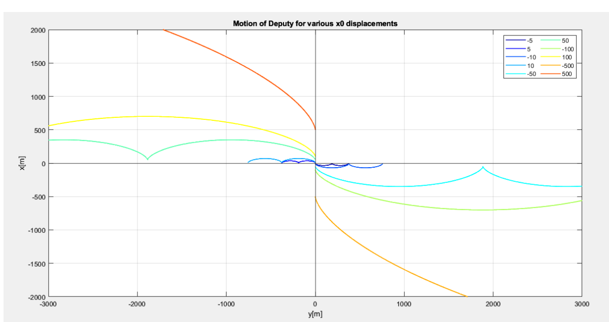

Figure 1: *Motion of Deputy w.r.t. Chief for various x_0 displacements, propagated in 200 minutes*

The second script (**TwoImp_min.m**) computes the minimum total V two-impulse maneuvers, knowing the initial conditions of position and velocity of Deputy with respect to the Chief. To do this, the function considers a maximum final time equal to half of the orbit period. Once again, the HCW equations are solved in order to obtain the minimum ΔV and the corresponding time for the Deputy to reach the Chief.

Furthermore, a game-like animation of the system has been programmed, **Rendezvous_Simulator.m** (the third script), with an integrated user-interface to simulate the Deputy approaching to the Chief. When this script is executed a full screen 3D plot is visualized. In the 3D space the Chief is fixed at the origin of the CCS, while the initial position and velocity of the Deputy are randomly generated. The goal of the game is to let the deputy reach the Chief (within a 0.5 meters tolerance range) at maximum relative velocity of 0.01 m/s. In order to do so, the user can control the Deputy by applying small Delta-V impulses along its axis through the keyboard with the following commands:

- **W** for Prograde Acceleration (positive x-axis).
- **S** for Retrograde Acceleration (negative x-axis).
- **A** for Normal Acceleration (positive y-axis).
- **D** for Anti-Normal Acceleration (negative y-axis).
- **Q** for Radial-Out Acceleration (positive z-axis).
- **E** for Radial-In Acceleration (negative z-axis).
- **T-G** Speed Up/Slow Down simulation time.
- **Space** Pause the simulation.
- **X** Interrupt the simulation and exit.

The Deputy's trajectory is updated in real time and data on its position and velocity are displayed on screen, together with the projected position and velocity w.r.t the Chief after a given time interval, set at 1500 s. Each impulse given to the Deputy is memorized and the total ΔV is visualized on the GUI. The simulation ends when the Chief is successfully reached or it is interrupted by the user.

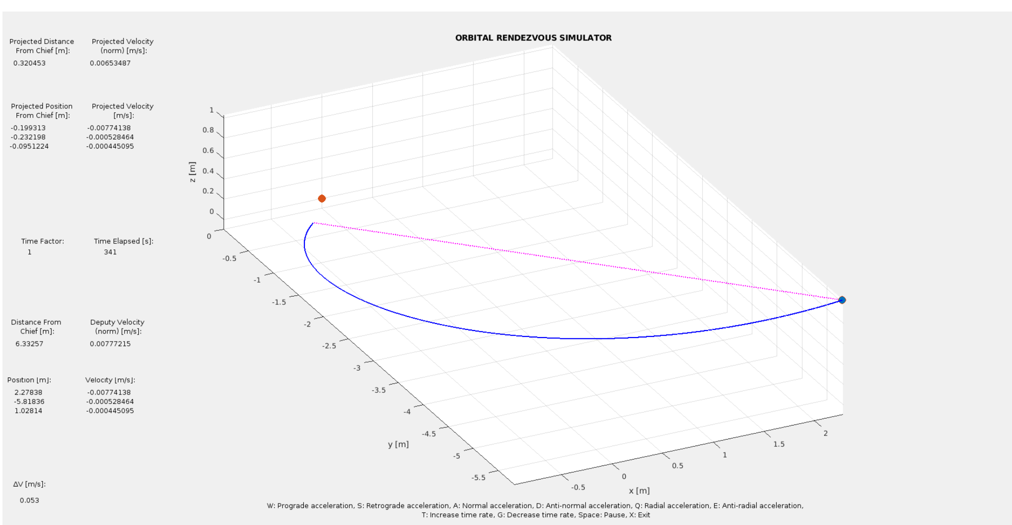

Figure 2: *Screenshot captured from the Rendezvous Simulation*

1.2 Rotational-internal motion

The fourth script, which actually consists of 4 different scripts, is aimed to study the rotational-internal motion of the system, and develops and integrates the free-flyer equations of motions using the recursive approach, accounting also the gravitational effect. In particular, **FFP6L5R_Robot_Description.m** allows to input the physical and geometric characteristics of the base and the links, as well as initiating the position of the joints and of the End Effector. **FFP6L5R_Robot_Simulate.m** is the core of the whole simulation: given the input parameters, it propagates position and velocity of the different links, set a specific delta time.
The function **FFP6L5R_odefun_state2dotstate.m** is called in Robot_Simulate.m to perform the above-mentioned actions. Furthermore, the script computes other quantities such as the momenta of the system and the General Inertia Matrix (GIM). **FFP6L5R_Robot_Figures.m** finally plots the results obtained, illustrating the motion of the manipulator.
The figure attached below shows the spacecraft-robotic manipulator system object of our study. It consists of a free-flyer with a squared-shape base, with a protruding manipulator composed by 5 bodies connected by revolute joints. The manipulator has a tree-architecture, meaning that exists a unique path from a body (also called Link) and its successor. Since the motion occurs in the plane of orbit, the system has 8 Degrees Of Freedom (DOF) in total, 3 of which associated to the base (2 translational and 1 rotational), and one associated to each of the joints (1 rotational DOF, being the joints revolute-type). The masses of the base body free-flyer and of the manipulator are similar in value. Links and joints are highlighted by the letters *"L"* and *"J"* respectively, while a *"C"* indicates the Centre of Mass of each body.

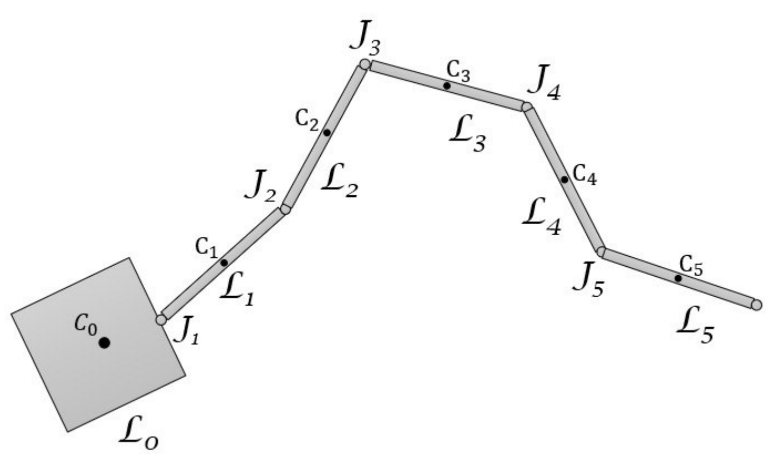

Figure 3: *Free-flyer with manipulator*

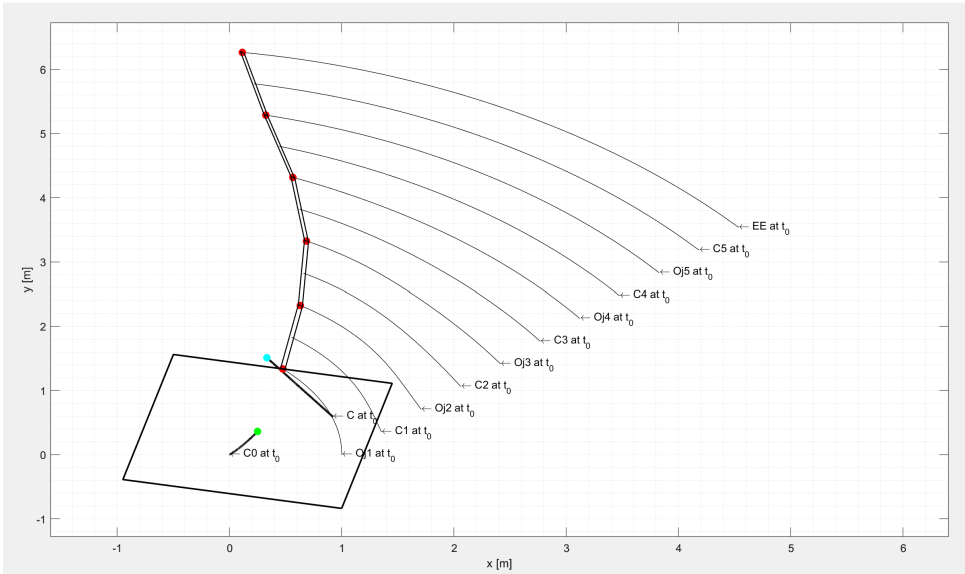

Figure 4: *Free-flyer with manipulator frame of the animation*

2 The development

The realization of this project was everything but an easy task, but at the same time it has been a challenging and thrilling experience useful to put in practice what we covered in class. There have been many difficulties to overcome since the beginning, but each member of our team has been dedicated and determined to build the software and complete the tasks with the best possible outcome. At first, the biggest challenges to face were mainly related to these factors: the limited amount of time until the deadline, since each of us had different things to bring on during the week, from regular classes to sports, student teams and personal life (and the exams session was approaching very fast, disallowing us to focus strictly on the project), and the fact that we didn't know each other very much or at all, forcing us to build the team spirit from scratch and to deal with each other strengths and weaknesses, which we weren't aware of prior to our first meeting. But the one thing that really drove us to the outcome we desired was the unconditioned determination we had as a team to complete the task assigned.

We decided to proceed by dividing the different tasks we had to do among the members of the team, making sure that even if some things would had been carried on individually, there would have been a constant and intensive exchange of opinions, ideas and knowledge, as well as an assiduous reporting of the status of each one's own work. On top of that, we decided to regularly meet in person to deal with the major challenges we were facing together and to discuss potential changes to the initial path we had decided to follow.

The first major decision we made was to deal with the software first, leaving the writing of the report at the end. Therefore, the tasks were assigned: Lorenzo dealt with Task 1.2, i.e. the realization of the HCW_IC_MakeFigs.m script. Vincenzo took care of Part 2, the analysis of the rotational-internal motion of the system: again, this part was carried out in very short time. Both Gaia and Vincenzo then focused on Task 1.3, the minimum Delta-V function: this was probably the most challenging assignment we had to do, as it required quite a long time to be completed and was the most discussed topic during our in-person meetings. Finally, Federico dealt with Task 1.4, i.e. the game-like animation of the system: this was the latest thing to be developed and it came with its own challenges of building an interactive and correct simulation of a 3D environment on MATLAB. The developed application is quite user-friendly but it may require a bit of practice from inexperienced users since matching the Chief's position and velocity is not a trivial task anyway.

Then, we started optimizing a little bit our software. Indeed, not only we wanted our scripts to run and perform the tasks, but we wanted them to run quickly, smoothly and with the least possible usage of memory. Therefore, after the completion of all of them, we took some time to optimize our code and to make the figures and the animations intuitive and pleasing to the eye. After this step, finally, we started writing the report: Gaia created the figure which illustrates the system, and together with Lorenzo took care of writing the report itself. In parallel to that, each of us took the time to write the *help* section of the script she or he produced, making sure to expose clearly and smoothly the purposes of the codes and, when needed, the instructions to run them. On top of that, Federico dealt with the realization of the README.md file, which briefly and synthetically illustrates the capabilities of the entire software.

3 Observations and comments

We were very thrilled and excited when we got assigned the project, since it was one of the first chances we had at Politecnico to produce something tangible with what we learned in class. And we have to say, the project really lived up to our expectations. It was not an easy task, as said before, since for all of us was the first time working on a project of this kind, and we spent a lot of time trying to figure out how to build the software, what to include and what to exclude, what to prioritize, using at their best the strong points of each one. We were all very confident about our knowledge pertaining the subject, since we are all very interested in the topic and have been following the classes assiduously: nonetheless, understanding the true power of theorems and equations, and the role they play in practical problems, is something one can't get enough before actually putting his own hands on it. Therefore we had to struggle a little, but in the end we are very happy of our final outcome: the plots of HCW_IC_MakeFigs.m match the ones we had to replicate; the TwoImp_min.m function correctly computes the minimum total-V of the two impulses maneuver; the scripts composing *Part2* precisely describe the manipulator and propagate position and velocity, giving as output a nice figure that reflects what we expected to obtain; finally, the game-like rendezvous simulation is an entertaining and intuitive way of visualizing how an approach maneuver is affected by the application of different impulses and what are the possible outcomes that different choices can lead to. Overall, the software provides a nice insight on some peculiar aspects of the space sector, by adopting a strong engineering approach, without on the other hand neglecting the clarity required to be understood by a wider and diverse audience. Furthermore, the requested report and *help* sections forced us to find an intuitive and comprehensible way to expose and explain our work, which is an aspect that is often neglected in our field.

On top of that, we are very happy with ourselves too: each one of us contributed at his or her best, making her or his skills and knowledge available to the team, stimulating the others to their best as well. One of the most useful implications of this project is without any doubt the chance it gave us to work in team with people we barely knew previously: this job-like situation was new to most of us, and the limited time given forced us to build quickly a spirit of collaboration among the team members, which normally would have required several weeks to happen. The fact that every decision has been discussed and approved "democratically", the intense exchange of updates as well as ideas, knowledge and help between us, the amount of time we spent working on the software together, in presence or occasionally from remote, is the proof of how well we got along with each other, regardless the final outcome.

We all wish we could have opportunities like this again, since each one of us truly understand the power of such work: in the end, this is what we are passionate about, and having the chance to deal with it directly is what we are always waiting for.

References

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