Laboratory Testbed Development for Modeling Spacecraft Proximity Operations

Samantha Romano
Vehicle Optimization, Dynamics, Control and Autonomy Lab
Department of Aerospace Engineering, University of Michigan
1320 Beal Ave, Ann Arbor, MI 48109
saromano@umich.edu

ABSTRACT

Proximity operations, such as rendezvous, proximity maneuvering, and docking procedures are difficult to conduct between two spacecrafts in orbit. This is due to no longer understanding the dynamics of either system fully, as they may have changed in the time since leaving the surface of Earth. To combat these challenges, while ensuring mission success in the form of constraint enforcement and efficient procedures, machine learning via a Learning Reference Governor (LRG) is being investigated to perform these tasks without relying on a dynamic model of the spacecraft during the mission. For such developing technological advancements, testing and proof of concept must be completed. To do this a laboratory testbed is being developed to model spacecraft proximity operations on omni-directional robots by using their ground-based data to simulate docking and rendezvous in the orbital frame. This development has involved the construction of four omni-directional robots, and a systemic configuration that involves the repeated scaling of data from the lab frame to the orbital frame, implementing control and guidance algorithms, reverse scaling, and communicating between the robots and lab's VICON system for state tracking and updating. Initial testing involved an LQR controller and a circular orbit around Earth, however this testbed is being designed so it can test a variety of controllers and orbits in future work, specifically the LRG. This report is on the current state of this testbed development and its potential capabilities.

Figure 1: Example illustrations of proximity operations conducted by spacecraft in orbit.

1. Introduction

Spacecraft proximity operations is a topic that encompasses many common tasks for spacecraft in orbit such as on-orbit inspection, repair, upgrades, assembly or refueling. These tasks are critical for the success of many space missions, but oftentimes difficult to complete as one spacecraft may be damaged or disabled leading to uncertainty about the relative dynamics of the system, or even potential collisions during proximity maneuvers. This difficulty leads to the rationale of this research project which is that there is a need to develop algorithms for proximity operations that learn about the spacecraft that needs repair, while maintaining safe operations during learning. For such developing technological advancements, testing and proof of concept must be completed. In this project we support the investigation of data-driven learning for constraint enforcement of spacecraft proximity operations.

The way to support development of the Learning Reference Governor (LRG) is to create a testbed to perform hardware validation of the new control algorithm. Ideally testing would be completed on the vehicle in which the goal is to have the control method implemented, however spacecrafts are expensive and difficult to access for testing such as this. This obstacle for hardware verification is the leading motivator for the laboratory testbed developed in this project. The overall goal of this project is to create a ground-based testbed, performing proximity maneuvers with omni-directional robots, and using software to model robot maneuvers in the orbital frame as though they were spacecraft.

This project began with identifying the type of omni-directional robot that would be utilized for the performance of these maneuvers, and then developing a robot fleet for testing. It then transitioned into consulting previous research relevant to this topic, most notably *Hardware Implementation of Model Predictive Control for Relative Motion Maneuvering* [2], which discusses using omni-directional robots to model model-predictive control algorithms, as well as scaling techniques for translating ground-based data to an orbital scale. Inspiration drawn from this previous work along with developing software and methods to suit our laboratory capabilities and project needs has led to a full system of communicating loops, robots, and motion tracking cameras to perform proximity operations and complete successful docking as two spacecrafts would. This system, the hardware and software utilized, as well as the general methods and plans for future work are outlined in this report.

2. System Architecture

The system architecture and experimental setup can be seen in Figure 2 below. It is a closed loop system that will continue operating until rendezvous or docking procedure being conducted is completed successfully. The robots will be moving around in the tracking region – one designated as the target while the other is assigned as the chaser, who is responsible for “catching” the target at its location. The experimental setup loop begins with the VICON Motion Capture System which will monitor the position and velocity of the robots while they move in the testbed. It will then output this data into the orbital frame scaling loop. This loop will take the ground-based data and scale it larger so that it translates to the position and velocity of a spacecraft moving in orbit. This data will then be passed into the LRG, which will enforce constraints and pass its outputs into the nominal controller block, which will update the position of the spacecraft in that orbital frame. From here, the updated spacecraft position is passed into the lab frame scaling loop, which does the inverse of the orbital frame scaling loop, by shrinking the orbital positions and velocities to fit the laboratory testbed motion. Finally, the updated lab frame state values are given to the robot command which processes the data and makes the robot move to its next location. This process then repeats until completion.

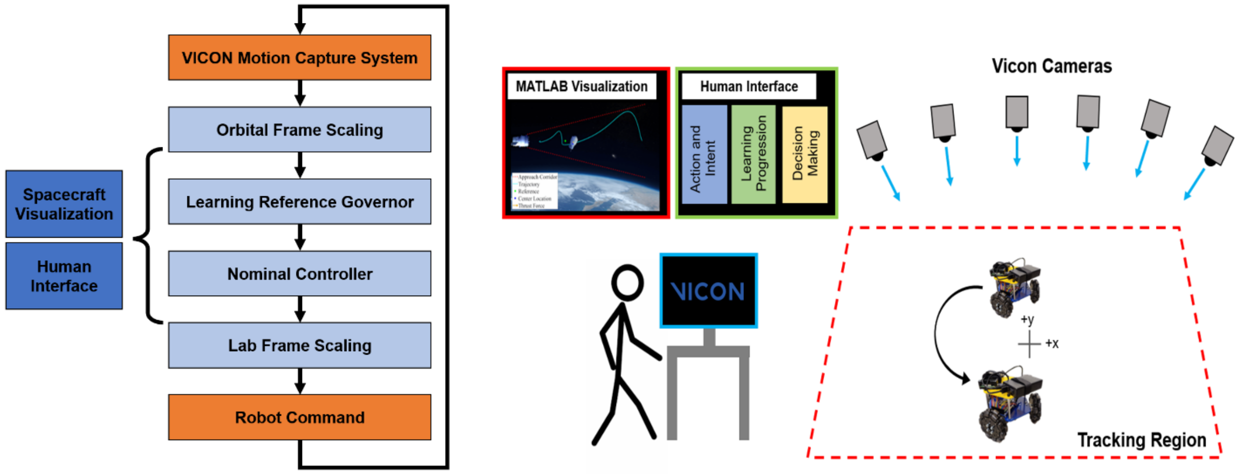

Figure 2: Experiment setup stack (left), system architecture (right).

Two more components of this system to highlight are the spacecraft visualization and the human interface blocks to the side of the scaling and orbital blocks. These two blocks are still in the early development stages, and it has not been finalized yet where they will be integrated into this system. The spacecraft visualization block will be responsible for demonstrating how the motion of the robots in the tracking region will correspond to the motion of a spacecraft performing a proximity operation. The specifics have not been fully determined yet; however, the general idea is for the visualization to be running as the robots are maneuvering so the user can compare them in real time. The human interface block will be responsible for putting a human user in the loop of this system to potentially help speed up the learning process of the LRG and make docking more efficient, while maintaining safe completion of the proximity maneuver. It is also going to be responsible for allowing human users access to the information as it is produced for this system, specifically in terms of the trajectories and state convergence data that will be generated by the nominal control block and the LRG. Current progress on this element of the experimental testbed includes the development of a GUI that displays lab and orbital frame data as it is updated in real laboratory time, allowing the user to visualize how the spacecraft are moving.

3. Hardware and Software

This section discusses the hardware and software that this testbed setup relies upon. The main components included are the omni-directional robots, the VICON motion tracking system, and the orbital control and visualization software. All of these pieces eventually come together to enable the modeling of spacecraft proximity operations on a ground-based testbed.

3.1 Omni-Directional Robots

The omni-direction robot model selected for this testbed and all affiliated experimentation is the Mbot-Omni. This robot is low cost, at $350 per robot, and features in-house University of Michigan open-source platforms for hardware and software, making it great for applications for education and research. The team developed a fleet of four Mbot-Omnis, building them from the ground up. These robots feature holonomic motion control and odometry with IMU and wheel encoders, as well as Kiwi drive with three degrees of freedom. A Raspberry Pi 4 is used as the primary compute, doing high level control with its open-source architecture. The embedded processor is a Raspberry Pi Pico. These robots also feature their own position tracking capabilities via their own odometry or using an external system. The four robots are named Copernicus, Kepler, Halley, and Hubble, and an image of what they look like can be seen in Figure 3 below.

Figure 3: Fully manufactured omni-directional robot side view (left), front view (center) and top view (right).

3.2 VICON Motion Tracking System

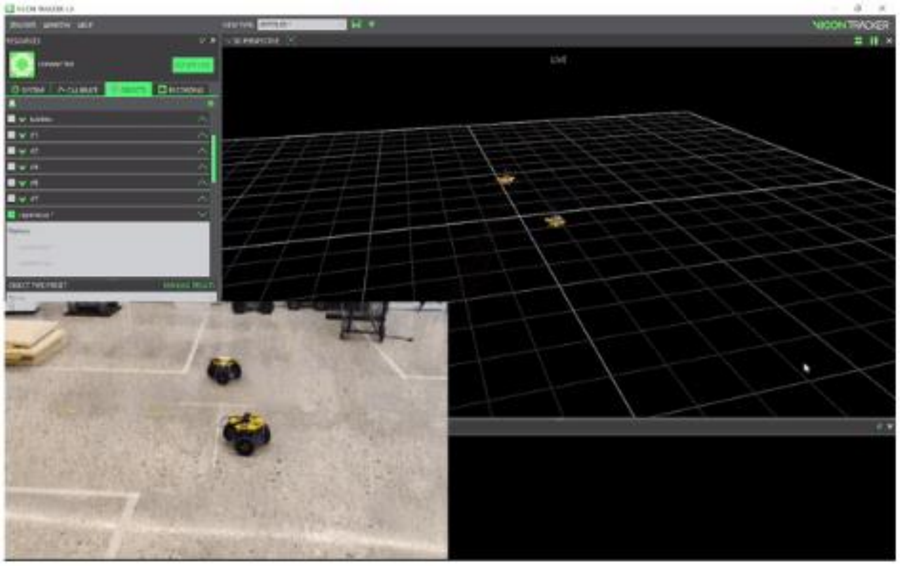

Figure 4: VICON motion tracking of two Mbot-Omnis in the tracking region.

The VICON Motion Tracking System had been previously incorporated into the Vehicle Optimization, Dynamics, Control and Autonomy Lab – home to this project. It is being utilized for tracking the omni-directional robots and reporting their position and velocity data to the orbital scaling loop as is described in earlier sections. The system utilized for this project consists of 17 cameras, with up to 1mm of precision, guaranteeing accurate and precise measurements being reported and utilized to inform state updates and hardware verification. Figure 4 above shows the Mbot-Omnis in the tracking region being detected and monitored by VICON.

3.3 Orbital Control and Visualization Software

The software that this section discusses is that required to perform scaling and visualization tasks, and that is currently being used for the nominal controller implementation. MATLAB has been the primary tool for conducting these tasks, due to its modeling and powerful built-in toolbox capabilities. Investigations are currently being conducted to see if other software platforms would be better suited to our project goals and future work, including making the full system run as efficiently as possible which means the robots being able to communicate with these blocks at top speeds, which MATLAB may hinder.

4. Methods

This section includes the methods for modeling the motion of the spacecraft performing proximity operations, scaling data from the lab frame to the orbital frame and vice versa, as well as the general guidance and control loop setup.

4.1 Relative Motion and Dynamics

The orbital frame simulated by the testbed is the relative motion Hill's frame, which is a rotating frame centered about the target spacecraft [1]. The spacecraft whose motion is simulated relative to the target is known as the chaser spacecraft. This reference frame can be seen in Figure 5 below, where the radial coordinate is x, the in-track coordinate is y, and the out-of-plane coordinate is z.

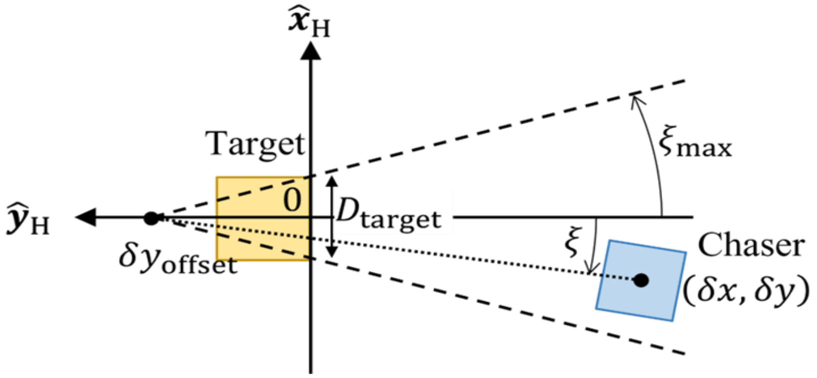

Figure 5: Illustration of the relative motion Hill's Frame of two spacecraft.

For all preliminary testing conducted, a circular orbit around Earth is utilized and the Clohessy-Wiltshire equations (equations 1-3 below) are used to determine the relative motion of the chaser with respect to the target [2]. In these equations n denotes the mean motion of the target spacecraft, and m_d denotes the mass of the chaser spacecraft. These are linear equations of motion that are discretized via ode45 and a time step of 20s.

$$\ddot{x} - 3n^2 x - 2n\dot{y} = \frac{F_x}{m_d} \tag{1}$$

$$\ddot{y} + 2n\dot{x} = \frac{F_y}{m_d} \tag{2}$$

$$\ddot{z} + n^2 z = \frac{F_z}{m_d} \tag{3}$$

4.2 Orbital and Lab Frame Scaling

To connect the lab frame, where the robot is operating, and the orbital frame where a spacecraft would be performing the same maneuvers with an additional dimension, orbital and lab frame scaling loops were developed to correlate the different magnitudes experienced in each setting. Two variables were specified: Length scaling parameter v, and time scaling parameter κ. For initial testing, the length scaling parameter had a value of 0.001, meaning that one meter in the orbital frame would correspond to one millimeter in the laboratory frame. The time constant was given an initial value of 0.01, meaning that one second in the orbital frame would correspond to 0.01 seconds in the laboratory frame. These values were chosen based on past literature that demonstrated successful results in their testing [2]. In the future, this testbed has been designed to be easily adjusted so other scaling parameters or initial conditions could be tested if so desired. Lab position is scaled by $1/v$, and velocity is scaled by κ/v. Orbital position and velocity is the inverse of those respectively.

4.3 Guidance and Control

The guidance and control block followed for this project can be seen in Figure 6 below. Important features to note about this block diagram is that the reference command going into the LRG is going to be the lab to orbital scaled VICON data, and that the nominal controller and plant system will be working together to update the state vector of the system.

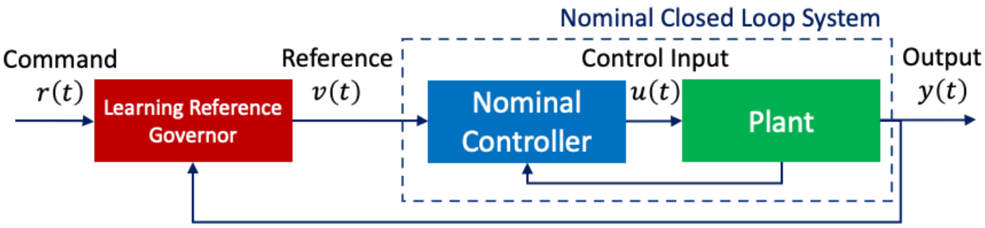

Figure 6: LRG and control loop system block diagram [3].

The state of the system is denoted by $X(k) = [x(k), y(k), z(k), \dot{x}(k), \dot{y}(k), \dot{z}(k)]$ and the update equation for this state is implemented as $X(k+1) = AX(k) + BU(k)$, where k denotes the time instant, and U denotes the control input which is thrust in this case. The nominal controller used for initial experimentation has been a linear quadratic regulator (LQR). The LQR controller was selected because it provides good stability to the system and is computationally efficient. The plant component of this system is the function that calculates the Clohessy-Wiltshire equations of motion and updates the velocity. This function is run within ode45, to discretize the continuous equations of motion.

5. Ongoing Work

This section will discuss the progress made by the team within the past year of pursuing this project. In the past year the team has been able to develop a fleet of four Mbot-Omnis. These robots are all fully functional and have proven to be able to follow trajectories successfully. The VICON system is also fully operational and has been proven to track our robots efficiently and accurately.

More on the software side, the scaling loops, LQR controller and software test scenarios have all been developed and run successfully. The test scenario being worked with is a circular orbit around Earth, where the chaser has a mass of 100kg, there is a time step of 20s, and the target mean motion is n=0.0012. These values were again taken from the successful case study from the MPC implementation into omni-directional robots [2]. The test scenario begins by feeding in a lab frame initial state for the chaser robot of arbitrarily $X(k) = [0, 2, 0, 0, 0,0]$, meaning that the chaser is located 2m away from the target in the lab frame along the in-track direction. This state vector then gets scaled to the orbital frame, and this new data is passed into the LQR loop, which calculates the gain applied to the control inputs and calls ode45 with the Clohessy-Wiltshire equations of motion to apply those gains and update the state variables. From there this data is scaled back down to the laboratory frame and, because this is just a simulated test, instead of being passed back out to the robot it instead reenters the loop that will continue running until rendezvous is accomplished within a specified tolerance. Figure 7 shows the control inputs of this initial test case.

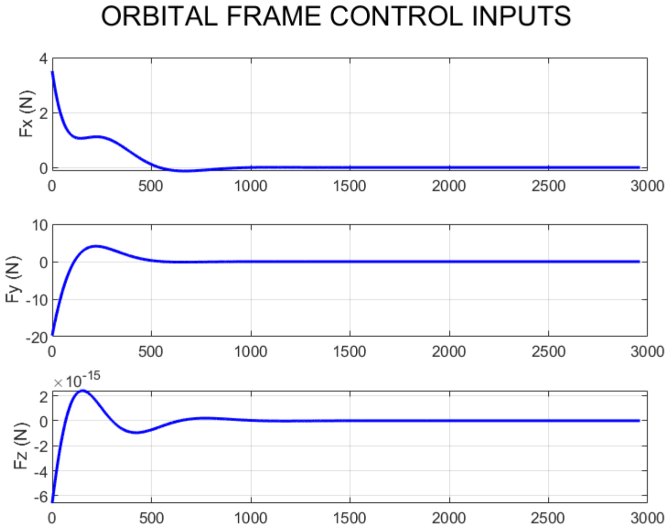

Figure 7: Control inputs of the initial test case using LQR.

The magnitudes of these control inputs are considered “reasonable” for spacecraft proximity operations at this point and time, meaning that they are on the order of tens of Newtons as opposed to hundreds or even thousands. However, these values may still be violating constraints even with added weights to the LQR matrices, which is why the addition of a reference governor, and eventually a learning reference governor will be so vital to this system’s success.

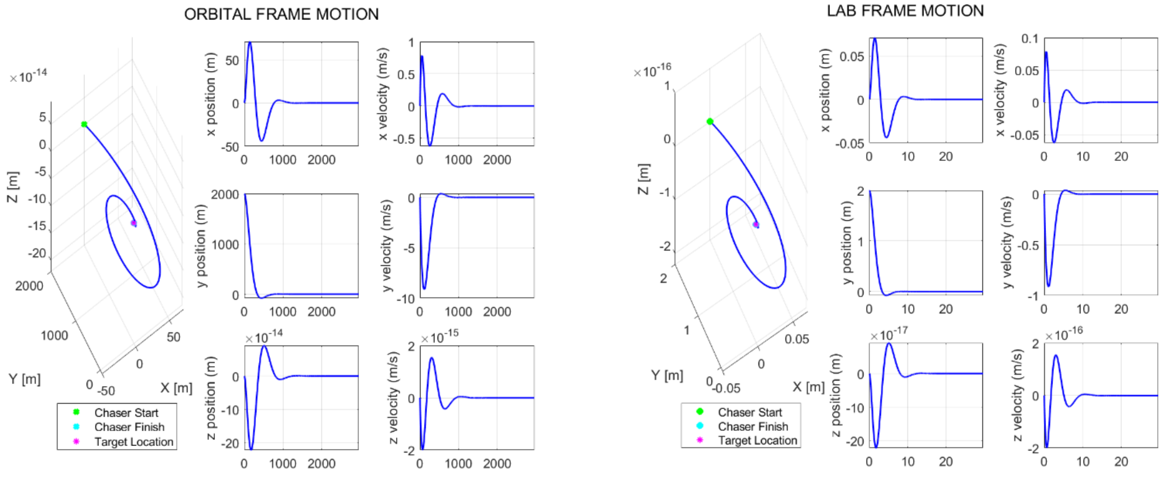

Figure 8: Trajectory of a rendezvous maneuver in the orbital (left) and lab (right) frames.

Figure 8 above shows the trajectories followed in this simulation in both the orbital and laboratory frames. The big difference to note between the two is the magnitudes of the position and velocity data. The laboratory frame state has smaller values, that are attainable for an Mbot-Omni to accomplish while the orbital frame shows realistic data for a spacecraft relative to another spacecraft in orbit. For ground-based movement of these robots, the trajectories will be flattened to two dimensions, x and y.

6. Future Work and Conclusions

Future work will consist of furthering the development of our system. We are aiming to incorporate and do real time verification of the learning reference governor within the next few months and continue refinement of software tools and processes. Progress will also continue on the spacecraft visualization and human interface components of these projects, eventually to use humans in the loop to augment training or execution phases for these operations. Lastly, future work will include the development of an outreach component of this project, to speak with younger students who may be interested in STEM, and particularly space. The goal will be to introduce them to a simplified version of controls and robotics by utilizing our testbed in a unique way.

References

[1] Curtis. (2020). Orbital Mechanics for Engineering Students. Elsevier.

[2] Goodyear, A., Petersen, C., Pierre, J., Zagaris, C., Baldwin, M., & Kolmanovsky, I. (2015). Hardware implementation of model predictive control for relative motion maneuvering. 2015 American Control Conference (ACC). https://doi.org/10.1109/acc.2015.7171077

[3] Ikeya, K., Liu, K., Girard, A., & Kolmanovsky, I. V. (2022). Learning to satisfy constraints in spacecraft rendezvous and proximity maneuvering: A learning reference governor approach. AIAA SCITECH 2022 Forum. https://doi.org/10.2514/6.2022-2514