Rendezvous & Proximity Operations

ME 601 (Autonomous Feedback) Final Project

Ian Ruh
University of Wisconsin - Madison

***Abstract*—Using the material covered in class, this project analyzes the control and relative dynamics of two spacecraft in orbit around Earth by designing, implementing, and testing several linear-quadratic-regulator (LQR) and model predictive control (MPC) controllers. This is a subject of interest due to its current usage in vehicle docking to space stations, as well as its future applications to in-orbit debris collection, in-orbit assembly, and in-orbit refueling.**

We designed and implemented three infinite LQR feedback controllers: an infinite LQR controller using the linearized system dynamics that stabilizes the system at the origin; a trajectory tracking infinite LQR controller using the linearized system's time invariant error dynamics; and a time varying infinite LQR feedback controller that uses a nonlinear, but simplified, version of the system's error dynamics. We also designed and implemented an MPC controller using the time invariant linearized system dynamics, while incorporating state and control constraints.

To test our controllers, we wrote a restricted two-body simulator to simulate the dynamics of both the target and chaser vehicles. For simplicity, we modeled the Earth as a perfect sphere and only considered the idealized dynamics due to the gravitational interaction between the Earth and the satellites.

In order to further simplify the problem, several assumptions on the orbital parameters of the two spacecrafts, the configuration of actuators, the ideal behavior of the actuators, the control of the spacecraft attitude, and knowledge of the spacecraft state are made.

I. Introduction

The rendezvous of two spacecraft, or more generally the relative positioning of two spacecraft in orbit, is a complex problem due to the dynamics involved, and has applications to a wide variety of missions and customers. At present, one of the most important applications of this problem is in docking to space stations, such as the ISS, by Dragon, Starliner, Cygnus, and other vehicles. In addition, there are companies and organizations working on demonstrating the ability to collect debris from orbit, how to extend vehicle lifetimes through in orbit refueling, and how to assemble structures that are too large to be launched in a single piece (such as space telescopes or space stations).

A. Coordinate Systems

In the rest of this document, we identify two spacecraft: the target and the chaser (also referred to as the chief and deputy). The target is assumed to have attitude control such that it maintains a consistent orientation with respect to Earth's horizon but is otherwise inert. The chaser is the vehicle that we focus on controlling the position of.

The first step in approaching the problem is to select a coordinate system in which the relative dynamics will be studied. We identify two reference frames. The Earth Centered Inertial (ECI) frame [2] is used as our reference inertial frame and is used for numerical propagation of the system dynamics. As Earth orbits the sun, the ECI frame remains in the same orientation relative to the stars. Similarly, as Earth rotates around its axis, the ECI frame remains fixed relative to the stars. In the context of rendezvous and proximity operations, it is convenient to also identify the Radial-Transverse-Normal (RTN) coordinate frame centered with the origin at the center of mass of the target vehicle, as shown in figure 1. The three axes are aligned in the radial direction, along the vector from the center of the Earth to the target; the transverse direction, parallel to the target's velocity vector, and thus along the orbital path; and the normal direction, parallel to the target's orbital angular momentum vector.

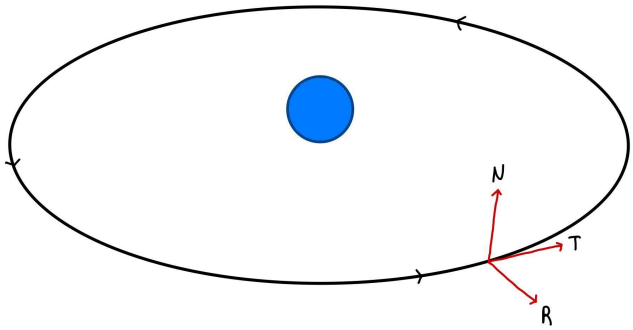

Fig. 1. Radial-Transverse-Normal (RTN) Coordinate Frame

Analyzing the chaser's dynamics and control in the RTN frame has the advantage of much smaller oscillations in the position-velocity coordinates as well as conveniently placing our target state at the origin.

Notice that as the target orbits the central body, the RTN frame rotates relative to the inertial frame of the central body, so the relative position and dynamics of the chaser is not just a translation from the inertial frame (the actual dynamics are shown in the next section.).

We also define our position-velocity (PV) state vector within both the ECI frame and the RTN frame as $[\vec{x}|\vec{v}]^T = [x, y, z, v_x, v_y, v_z]^T$ with respect to each frame. Though not the state representation primarily used in this project, several of the states from the classical orbital elements (COE) are

important. Note, however, that the COE states are only defined in the ECI frame, not the RTN frame. The full COE state vector is

$$\begin{aligned} e &= \text{Eccentricity} \\ a &= \text{Semi-major axis (SMA)} \\ i &= \text{Inclination} \\ \omega &= \text{Argument of perigee} \\ \Omega &= \text{Right ascension of the ascending node (RAAN)} \\ f &= \text{True anomaly} \end{aligned} \tag{1}$$

For this project, the three important orbital elements are: the eccentricity, which is the eccentricity of our orbit's ellipse; SMA, which (for our purposes) describes the orbital radius; and the true anomaly, which describes, unlike the other states, where within the orbit we are, rather than the shape or orientation of the orbit.

B. Relative Dynamics

Within the RTN coordinate frame the following relative dynamics can be derived from the equations for motion around a central body [6]:

$$\begin{aligned} \ddot{x} - 2\dot{f}_c\dot{y} - \ddot{f}_c y - \dot{f}_c^2 x &= \frac{-\mu(r_c + x)}{\left[(r_c + x)^2 + y^2 + z^2\right]^{3/2}} + \frac{\mu}{r_c^2} + d_R \\ \ddot{y} + 2\dot{f}_c\dot{x} + \ddot{f}_c x - \dot{f}_c^2 y &= \frac{-\mu y}{\left[(r_c + x)^2 + y^2 + z^2\right]^{3/2}} + d_T \\ \ddot{z} &= \frac{-\mu z}{\left[(r_c + x)^2 + y^2 + z^2\right]^{3/2}} + d_N \end{aligned} \tag{2}$$

Where f_c is the target's true anomaly, r_c is the target's radius, and $\vec{d} = [d_R, d_T, d_N]^T$ represents the relative disturbing accelerations between the target and the chaser in the target's RTN frame.

The first simplifying assumption we make is on the eccentricity of the target's orbit. If the target's orbit is near circular (the eccentricity is near 0), then $\dot{f}_c$ = mean motion = $\sqrt{\frac{\mu}{a^3}}$, where μ is the Earth's gravitational parameter ($\approx 3.986 \cdot 10^{14}$ m^2/s), and a is the orbit's semi-major axis (SMA). Thus, $\dot{f}_c$ is constant and $\ddot{f}_c = 0$. In addition, the target's radius is constant and $r_c = a_c$, where a_c is the target's SMA. This leads to the following simplified dynamics:

$$\begin{aligned} \ddot{x} &= -\frac{\mu(r_c + x)}{\left[(r_c + x)^2 + y^2 + z^2\right]^{3/2}} + \frac{\mu}{r_c^2} + 2n\dot{y} + n^2 x + d_R \\ \ddot{y} &= -\frac{\mu y}{\left[(r_c + x)^2 + y^2 + z^2\right]^{3/2}} + 2n\dot{x} + n^2 y + d_T \\ \ddot{z} &= -\frac{\mu z}{\left[(r_c + x)^2 + y^2 + z^2\right]^{3/2}} + d_N \end{aligned} \tag{3}$$

Linearizing this around the target's position, we arrive at the following linear dynamics known as the Hill-Clohessy-Wiltshire equations [5]:

$$\begin{aligned} \ddot{x} &= 3n^2 x + 2n\dot{y} + d_R \\ \ddot{y} &= -2n\dot{x} + d_T \\ \ddot{z} &= -n^2 z + d_N \end{aligned} \tag{4}$$

Which we can express in the standard state space form

$$\begin{aligned} \dot{x}_0 &= x_3 \\ \dot{x}_1 &= x_4 \\ \dot{x}_2 &= x_5 \\ \dot{x}_3 &= 3n^2 x_0 + 2n x_4 + d_R \\ \dot{x}_4 &= -2n x_3 + d_T \\ \dot{x}_5 &= -n^2 x_2 + d_N \end{aligned} \tag{5}$$

C. Other Assumptions

The reality of controlling a spacecraft involves complex actuators, possibly including electric propulsion, monopropellant attitude control systems, reaction wheels, control moment gyros, magnetic torquers, and others, each with their own dynamics and constraints (such as burn times, thrust profiles, discrete or continuous thrust levels, etc) that need to be accounted for in practice. For this project, only the dynamics due to gravity and accelerations in the RTN frame are considered to simplify the analysis. Therefore, we are neglecting the attitude control of the chaser and the arrangement (and therefore selection) of thrusters such that we have have an idealized force in the RTN frame as our only control input. However, in reality there may be significant efficiency gains that can be realized by considering the arrangement of thrusters when planning maneuvers.

In addition to assumptions on the actuators, we also assume that we have perfect knowledge of both the target's and chaser's state vectors. However, state estimation for spacecraft is a significant challenge and, depending on the location (e.g. close enough for relative positioning systems), the uncertainty in our knowledge can be quite significant.

II. Simulation

In order to test the controllers implemented, we created a simulation environment and several test scenarios to evaluate their performance [3]. The source code is available at https://github.com/ianruh/Rendezvous-Proximity.

We wrote a restricted two body simulator to propagate the states of the target and chaser in the ECI frame around Earth, where only the gravitational interaction between the Earth and the single satellite is considered. All nonidealities of the orbits, including oblateness perturbations, solar radiation pressure, lunar perturbations, etc., were neglected. Due to the relatively small timescales, distances, and parameters of the scenarios, these effects would be small, though should not be ignored in practice.

At every control time step, the chaser's state vector was converted into a PV vector in the RTN coordinate frame relative to the target. The simulator saturated the control

at 0.01 or 0.1 $\frac{m}{s^2}$, depending on the simulation, which is reasonable for the monoprop thrusters on a relatively small satellite.

The trajectory tracking controllers were provided a target state at each control time step, where the target state was generated analytically or read (and interpolated) from a CSV file depending on the scenario.

III. Controllers

We implemented four controllers. An infinite LQR controller that is able to stabilize the chaser within the vicinity of the target, two trajectory tracking infinite LQR controllers, and an MPC controller.

To evaluate the performance of the trajectory tracking controllers, we calculate the integral of absolute error (IAE) for the actual trajectory vs the tracked trajectory:

$$\text{IAE} = \int_0^{t_f} e(t)dt \tag{6}$$

A. Infinite LQR Linear Controller

Using the Hill-Clohessy-Wiltshire (4) expressed as matrices in state space form, we implemented an infinite LQR controller that solves the following optimal control problem:

$$\begin{aligned} \min_{x,u} \quad & \frac{1}{2}\int_{t_0}^{\inf} \left(x(t)^T\mathbf{Q}x(t) + u(t)^T\mathbf{R}u(t)\right)dt \\ \text{s.t. } & \dot{x} = \mathbf{A}x + \mathbf{B}u \\ & x(t_0) = x_0 \end{aligned} \tag{7}$$

Where $\mathbf{A}$ and $\mathbf{B}$ are defined as

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 0 & 2n & 0 \\ 0 & 0 & 0 & -2n & 0 & 0 \\ 0 & 0 & -n^2 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{B} &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{8}$$

and the $\mathbf{Q}$ & $\mathbf{R}$ weight matrices were selected by starting from the heuristic weighting $\frac{1}{(\text{error bound})^2}$, then modified by trial and error. Then we solve the Algebraic Ricatti Equation (ARE) to find the static state feedback controller

$$\vec{u}(t) = -R^{-1}B^T P\vec{x} \tag{9}$$

This controller successfully stabilizes the chaser in the vicinity of the target, though the size of the region of stability appears to significantly depend on the weight matrices Q and R chosen. Two scenarios are shown in figures 2 and 3.

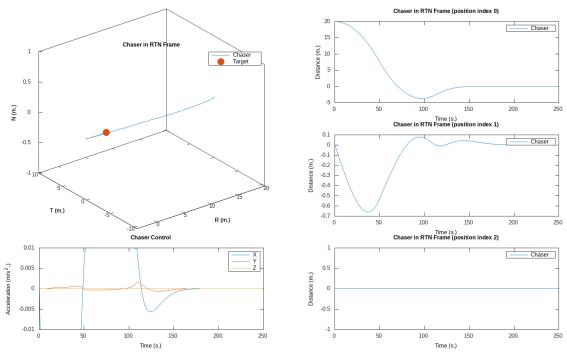

Fig. 2. Orbit stabilization starting from 20 meter larger radius.

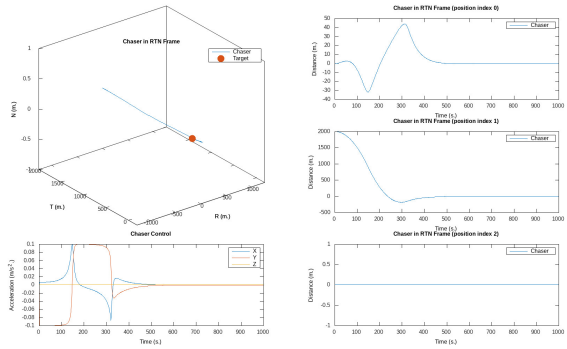

Fig. 3. Orbit stabilization starting from 2000 meters ahead.

Aside from showing that our dynamics and basic approach work, these controllers would have little practical use for rendezvous and docking, where safety constraints on the spacecraft state (both position and velocity) are very important. Therefore, we also implemented and tested two trajectory tracking controllers, as discussed in the next section.

B. Infinite LQR Time Invariant Trajectory Tracking Controller

We implemented an LQR trajectory tracking controller based on the error dynamics of the linearized system in (4). We define the state error $e(t) = x(t) - x_d(t)$, where x_d is our desired trajectory, and the control error $v(t) = u(t) - u_d(t)$, where u_d is our desired control. Then the error dynamics are

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{x}_d \\ &= f(x,u) - f(x_d,u_d) \\ &= f(x_d + e, u_d + v) - f(x_d,u_d) \\ &= (\mathbf{A}(x_d+e) + \mathbf{B}(u_d+v)) - (\mathbf{A}x_d + \mathbf{B}u_d) \\ &= \mathbf{A}e + \mathbf{B}v \end{aligned} \tag{10}$$

Where $\mathbf{A}$ and $\mathbf{B}$ are the same as in (8). Conveniently, the error dynamics of the linearized system are identical to the dynamics of the linearized system.

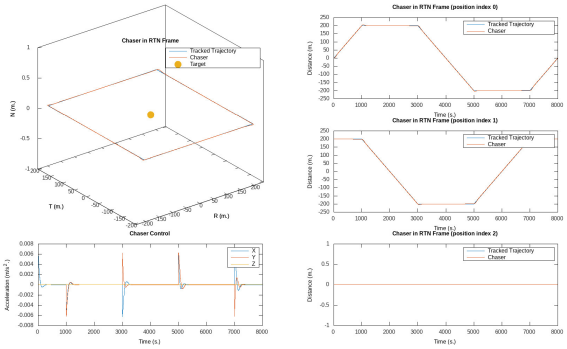

Fig. 4. Box trajectory tracking at GEO ($a \approx 42000$ km). This simulation has an IAE of 10265.5.

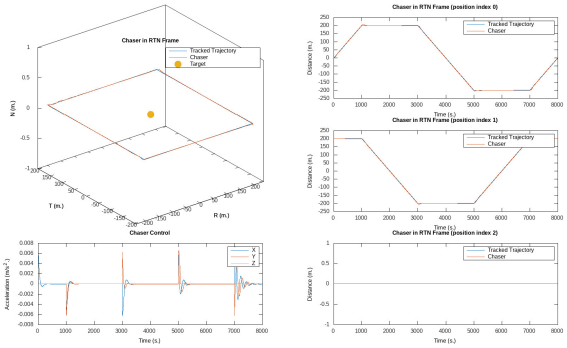

Fig. 5. Box trajectory tracking at an SMA of 30000 km SMA. This simulation has an IAE of 15581.3

$$\begin{aligned} \min_{e,v} \quad & \frac{1}{2}\int_{t_0}^{\inf} \left(e(t)^T \mathbf{Q} e(t) + v(t)^T \mathbf{R} v(t)\right) dt \\ \text{s.t. } & \dot{e} = \mathbf{A}e + \mathbf{B}v \\ & e(t_0) = e_0 \end{aligned} \tag{11}$$

Being provided a trajectory to track, one which can be guaranteed not to violate any safety constraints, is a more realistic scenario than just an LQR controller attempting to stabilize the system at the origin.

We tested the trajectory tracking controllers on several 'box' trajectories at varying orbital radii from Geo Synchronous (42,000 km) to LEO (8,000 km).

As the orbit gets smaller, the disturbing acceleration required to cancel out the dynamics of the system begins to saturate the inputs, causing a loss of control. This behavior is expected, as the 'box' trajectories ignore the natural dynamics of the system, which become more significant the smaller the orbit gets, and therefore needing more control authority to track.

C. Infinite LQR Time Varying Trajectory Tracking Controller

We also implemented a time varying LQR trajectory tracking controller based off of the nonlinear dynamics shown in eq. 3.

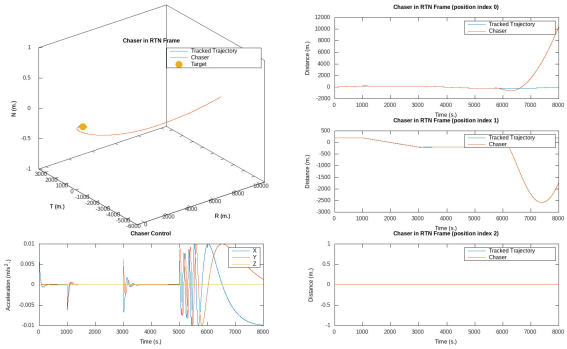

Fig. 6. Box trajectory tracking at an SMA of 20000 km SMA. This simulation has an IAE of $7.83 \cdot 10^6$.

$$\begin{aligned} \dot{e} &= \dot{x} - \dot{x}_d \\ &= f(x,u) - f(x_d,u_d) \\ &= f(x_d + e, u_d + v) - f(x_d,u_d) \\ &= F(e, v, x_d(t), u_d(t)) \end{aligned} \tag{12}$$

The time varying linearized error dynamics are then given by

$$\dot{e} = \mathbf{A}(t)e + \mathbf{B}(t)v \tag{13}$$

where

$$\mathbf{A}(t) = \frac{\partial F}{\partial e}|_{x_d(t),u_d(t)} \tag{14}$$

$$\mathbf{B}(t) = \frac{\partial F}{\partial v}|_{x_d(t),u_d(t)} \tag{15}$$

Rather than performing the derivation by hand, we used the symbolic mathematics library SymEngine [4] to derive the symbolic matrices for $\mathbf{A}(t)$ and $\mathbf{B}(t)$. From the symbolic matrices, we performed code generation at runtime to create a function for each symbolic matrix that, given a vector of the desired state and control, returned $\mathbf{A}$ and $\mathbf{B}$. The generated code was compiled to a shared object and then loaded into memory and the function pointers retrieved.

$$\begin{aligned} \min_{e,v} \quad & \frac{1}{2}\int_{t_0}^{\inf} \left(e(t)^T \mathbf{Q} e(t) + v(t)^T \mathbf{R} v(t)\right) dt \\ \text{s.t. } & \dot{e} = \mathbf{A}(t)e + \mathbf{B}(t)v \\ & e(t_0) = e_0 \end{aligned} \tag{16}$$

Figure 7 shows a simulation run using the time varying trajectory tracking controller and the associated IAE. Notice that the IAE improved marginally (0.02%) from the time invariant controller. This indicates that, at least in the near circular case at GEO, the nonlinear error dynamics that the invariant controller neglects play a very small role. For practical purposes, it may not be worth the extra computational resources needed to use the time varying controller.

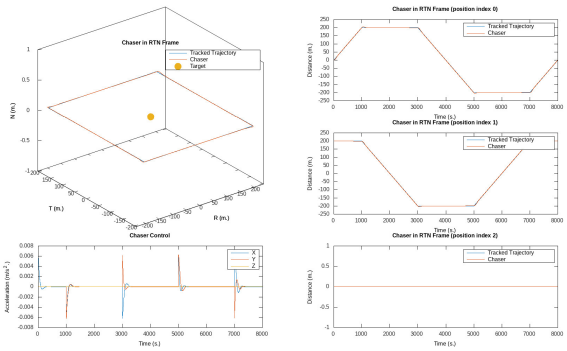

Fig. 7. Box trajectory tracking at GEO ($a \approx 42000$ km) using the time varying trajectory tracking controller. This simulation has an IAE of 10263.3.

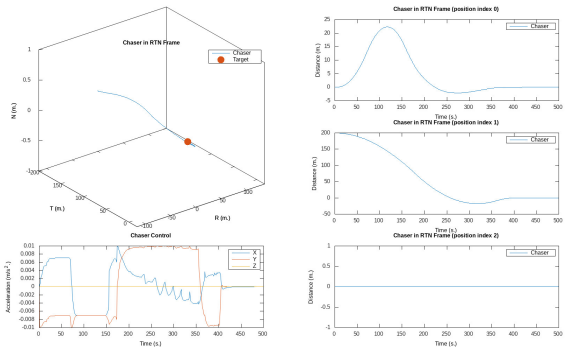

Fig. 8. MPC controller bringing the chaser to the target when starting 200 meters away. We finish at the target, but I would expect the optimal control to be smoother, so some additional debugging may be needed.

D. MPC Controller

We implemented a discrete time MPC controller based off of Hill-Clohessy-Wiltshire equations (4), which constrained the maximum control, velocity, and state. The MPC controller was implemented using CPPMPC [1], and formulated as:

$$\begin{aligned}
\min_{\vec{x}_n, \vec{u}_n, n=1,\ldots,N} \quad & L_f(\vec{x}_N) + \sum_{n=0}^{N} \left(\vec{x}_n^T \mathbf{Q} \vec{x}_n + \vec{u}_n^T \mathbf{R} \vec{u}_n\right) dt \\
\text{s.t. } & \vec{x}_{n+1} = \mathbf{A}_d \vec{x}_n + \mathbf{B}_d \vec{u}_n, \quad n = 1, ..., N \\
& \vec{u}_n \leq \vec{u}_{max}, \quad n = 1, ..., N \\
& \vec{u}_n \geq \vec{u}_{min}, \quad n = 1, ..., N \\
& \vec{x}(3:)_n \leq \vec{v}_{max}, \quad n = 1, ..., N \\
& \vec{x}(3:)_n \geq \vec{v}_{min}, \quad n = 1, ..., N \\
& \vec{x}_0 = x_0
\end{aligned} \tag{17}$$

Where

$$\begin{aligned}
\mathbf{A}_d &= I_{6\times 6} + \mathbf{A}\Delta t \\
\mathbf{B}_d &= \mathbf{B}\Delta t
\end{aligned} \tag{18}$$

As this was not formulated as a tracking controller (though it could be easily adapted to one by parameterizing goal state in the objective for every time step), we tested this controller on just bringing the chaser to the target when started within the vicinity of the target.

The CPPMPC library solves it using an interior-point primal barrier method. It is capable of solving it in real time, though this depends on the time horizon selected. In the simulations run, it ran at $\approx$ 30Hz on average with a 20 step horizon (using fairly aggressive hyper-parameters and warm starting each iteration).

IV. Extensions

There are many extensions and variations on the system and the controllers that could be explored. Of interest to myself is extending the LTV controller to the noncircular case, where the first and second time derivatives of the true anomaly come into the system. This would enable the controllers to function in a larger vicinity of the target, as well in elliptical orbits.

Similarly, it would be interesting to extend the simulation and controllers to the restricted three body problem involving the Earth, Moon, and spacecraft. Of particular interest, would be rendezvous and proximity operations in Near Rectilinear Halo Orbits about the Moon (these are extremely elliptical orbits that blur the line between lunar orbits and L2 halo orbits) where the planned NASA Gateway manned space station will be.

The assumptions made about the configuration and actuators could also be removed, specifically the arrangement of thrusters and its impact on the efficiency of maneuvers would be interesting.

Code

All of the code and raw outputs from this project can be found at https://github.com/ianruh/Rendezvous-Proximity. The `master` branch head may not be in the same state as at the time of writing this report, but the branch `archive-me601` will be preserved with the original code.

References

[1] Cppmpc github repo. https://github.com/ianruh/cppmpc.

[2] Earth-centered inertial frame. https://en.wikipedia.org/wiki/Earth-centered_inertial.

[3] Rendezvous-proximity github repo. https://github.com/ianruh/Rendezvous-Proximity.

[4] Symengine. https://github.com/symengine/symengine.

[5] Eugene M Cliff. Clohessy - Wiltshire Analysis. *N/A*, page 4, N/A.

[6] Joshua Sullivan, Sebastian Grimberg, and Simone D'Amico. Comprehensive Survey and Assessment of Spacecraft Relative Motion Dynamics Models. *Journal of Guidance, Control, and Dynamics*, 40(8):1837–1859, August 2017.