Orbital and Attitude Thrust Vectoring Control for Low Thrust Spacecraft Rendezvous

Brian Wodetzki

***Abstract*—This report proposes a light-weight controller to drive spacecraft docking for under-actuated satellites with only one thruster and three reaction wheels. The control methodology consists of running two independent controllers, first an attitude controller which tracks the desired thrust vector output by the second controller, an orbital controller implemented with Model Predictive Control (MPC). This control structure is found to perform well in most cases with or without applied moment made to simulate realistic perturbations. However, monte carlo analysis reveals clear cases in which the controller fails showing a lack of robustness. Simulations are also done with low thrust satellites, where it is shown the controller is unable to succeed due to the thrust restrictions.**

I. Introduction and Motivation

Spacecraft Rendezvous and Proximity Operations (RPO) are essential parts of the mission profiles for many satellites. Perhaps the vehicle that performers rendezvous most frequently is the International Space Station (ISS), which requires many rendezvous to supply the space station with necessities for crew members, science experiments, and delta V to keep the station in orbit. Indeed NASA did not only pioneer RPOs through the ISS, but they also started them with the first spacecraft docking of Gemini VIII [1]. During and after these accomplishments the field of spacecraft proximity operations blossomed. Many of the major concepts for RPOs resulting from this are presented in the book [2]. From this book there are multiple phases of spacecraft proximity operations. The first phase is phasing and transfer where the chaser spacecraft aims to reach a set area near to the target spacecraft to prepare for the second phase, typically this phase is done by preplanned impulsive maneuvers. The second phase is far-field rendezvous in which the chaser approaches the target and prepares for the fourth phase which is near-field rendezvous. In this phase the chaser has ideally navigated about all obstacles in the way of the docking port of the target spacecraft and close the distance between the spacecraft where it will will enter the fifth and final phase, the physical docking. This paper is focused on the third and fourth phases of this process. This is when preplanned maneuvers are not used and either manned, remotely operated, or autonomous guidance is used. Works will be pulled from many mission phases, however, as similar methods are used.

Much research has been done regarding the near-field rendezvous phase. Quite possibly one of the more difficult problem in the field of near-field rendezvous, and other phases, is docking with a tumbling satellite. Keenan et al. showed a full pipeline for docking with a tumbling target satelite by estimating the target's state, trajectory planning and trajectory tracking [3]. This, as with a large amount of other work in the field [4], is meant to be tested by a platform NASA has developed on the ISS called the astrobee and others. This platform is fully actuated and is the ideal vehicle in space. With the commercialization of space, the assumption of these algorithms being applied on fully actuated spacecraft begins to breaks down. The presence of vehicles that are not fully actuated (i.e. contains one thruster) have risen due to lower weight, complexity, and other factors. Indeed many companies intend on performing docking with under-actuated spacecraft where thrust vectoring is necessary [5], [6], [7].

One work by Gurfill [8] looks at tackling this near-field rendezvous problem with an under-actuated spacecraft using low-thrust. The method involves a Lyapanov-based feedback control approach which causes instability when near the target. Zhang et al. published a work [9] which aims to track a trajectory for rendezvous using only thrust vectoring, however, this approach fails to account for feedback control of the thrust magnitude which is required to reject disturbances. The state of the art algorithms in this field couple both the attitude and the orbital control of the target [10], [11]. This method neglects the knowledge that both orbital position and attitude control have very different dynamics and as a result may have different controller requirements. As has been shown to be the case [12], linear controllers perform quite well at the task of orbital position control. However attitude controllers require non-linear controllers such as the optimal formulation presented in [13]. A concurrent control design may also miss out on the breadth of research done in the individual fields of position control and attitude control [14].

Little research has been done on combining separate controllers for position and attitude on thrust vectoring spacecraft, and no research known to the author that implements separate position and attitude controllers for low thrust spacecraft in near-field rendezvous scenarios, like what may be seen in companies like Starfish Space [5] and Argo Space [7].

II. Problem Formulation

Given two satellites, a chaser satellite and a target satellite, the problem consists of developing a control system that will perform a soft docking of the two satellites using only control of one satellite, the chaser. The chaser satellite will have a singular thruster along with 3 control moment gyros. The thruster will be oriented in the correct direction with the use of the control moment gyros will only be capable of low to medium thrust. The target will be in a circular orbit in Low Earth Orbit (LEO) while the chaser will start

Purdue University

near the target, similarly in a circular orbit but but lagged behind the target spacecraft by a distance on the order of 1000s of meters to simulate near-field rendezvous. The chaser spacecraft will then perform a soft docking on the target with only the use of the available actuators. Soft docking will not include collision constraints and only consider reaching the final position with zero final velocity. The control algorithm for the spacecraft will be constructed in a separate manner as previously stated with an attitude controller and a orbital controller. The attitude controller should have the ability to reject the moment disturbances experienced at the altitude of LEO.

The coordinate system used for the analysis will be based off of what is shown in [2]. There are 3 reference frames that are used.

1) Orbital Reference Frames: Earth centered frames that parameterize the spacecraft orbit.
   a) F_{eq} is the earth equatorial frame and is defined by a_1, a_2, a_3. a_1 points to the direction of the mean of the vernal equinox. a_3 points normal to the equatorial plane. $a_3 = a_1 \times a_2$.
2) Spacecraft Local Orbit Reference Frames: Spacecraft centered frames that characterize other objects' motion around the spacecraft
   a) F_{lo} is determined again by three axes. This reference frame shares the same axis as F_{eq} with the difference of being centered at the center of mass of the satellite. This frame is used to determine the base orientation of the satellite.
3) Spacecraft Attitude and Body Frames: Spacecraft centered frames that characterize the attitude of the spacecraft with respect to it's orbital frame.
   a) Likewise, three axes are used to determine F_a. a_1 points through the smallest principal axis as defined by the inertia matrix I, a_2, and a_3 point along the remaining principal axis, and $a_3 = a_1 \times a_2$. The rotation matrices transition the spacecraft from F_{lo} to F_a

For simplicity the chaser spacecraft is assumed to have a mass $m_c = 1$kg, inertia values of $I = diag([1, 1, 1])$, a thruster with a variable thrust topping out at $T_0 = 1.0$N. Since this thrust level is about one order of magnitude higher than what can be expected when using a Hall-effect thruster but lower than what is expected using chemical propulsion, it is said that this is a medium thrust application. The thruster is assumed to be pointing in a_1 in F_a, simulating a single thruster pointing in this direction. This single thruster is what is said to make the spacecraft under-actuated.

To model the spacecrafts the same model for both the chaser and target are used. This model is a simple model with only the simplest interactions with the earth considered. The dynamics are given below:

$$\ddot{\boldsymbol{r}}_a = \frac{\mu}{|r|^3}\boldsymbol{r}_{lo} + \frac{1}{m}\boldsymbol{\Gamma}_s \tag{1}$$

Here $\boldsymbol{r}_a$ represents the vector in F_{lo} that points from the center of the earth to the center of the spacecraft. μ is the gravitational constant. $\boldsymbol{\Gamma}_s$ is the thrust of the spacecraft in the attitude frame. Excess force perturbations such as drag, solar radiation pressure, and the J2 perturbation are not considered because timescales are assumed to be small. Although this assumption often does not hold for low and medium thrust spacecraft, the near-field rendezvous phase that is aimed at modeling implies the chaser is a distance on the order of 1000s of meters from the target spacecraft. This allows the spacecraft to complete the near-field rendezvous without experiencing a large disturbance from perturbations.

The attitude of the spacecraft can be determined using two sets of equations of motion, the kinematic and the dynamic equations of motion. The dynamic equations of motion that govern the angular rates of the spacecraft are given below:

$$\dot{\boldsymbol{\Omega}} = \boldsymbol{I}^{-1}(-\boldsymbol{\Omega}^{\times}\boldsymbol{I}\boldsymbol{\Omega} + \boldsymbol{T}_s) + \boldsymbol{M} \tag{2}$$

Where $\boldsymbol{\Omega}$ is the current angular velocity in F_a. $\boldsymbol{I}$ is the diagonal inertia matrix of the spacecraft. $\boldsymbol{\Omega}^{\times}$ represents the skew symmetric matrix of $\boldsymbol{\Omega}$. $\boldsymbol{T}_s$ represents the torque on the spacecraft generated from the reaction wheels, whose speed is said to be controllable. $\boldsymbol{M}$ is a moment from perturbations outside the spacecraft, these are considered as they directly impact the effectiveness of the attitude controller. These typically come from three sources main sources: gravitational moment, solar radiation pressure, drag. From [15] we can see that at the altitude of the ISS the primary moment effecting a spacecraft comes from drag and is liberally on the order of $1e^{-4}$. This order of magnitude is used to simulate the effects of moment perturbations on the spacecraft.

The attitude parameterization is chosen to be the rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$. The rotation matrix is a linear map from the spacecraft cenetered inertial frame to the spacecraft centered body frame $R : F_{lo} \rightarrow F_a$. This rotation matrix is a lie group in SO(3). The rotation matrix was chosen as the global attitude representation as opposed to quaternions as they have a redundant attitude parameterization which can cause issues when implemented in control systems. The dynamics for the rotation are given below.

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{\Omega}^{\times} \tag{3}$$

These dynamics are integrated for both the chaser and the target satellite using Runge-Kutta 45 method with maximum tolerances of up to $1e^{-12}$.

III. Control Methodology

The proposed controller will consist of two independent controllers. One to control the position and velocity of the satellite and one to control the attitude of the satellite. The position controller will be a receding horizon optimal controller, Model Predictive Control (MPC). The attitude controller will use classical control methodology to track the desire attitude of the spacecraft. It is assumed all states are fully observable for simplicity during the implementation of this controller.

A. Attitude Controller

The attitude controller used is showcased in [16] as an attitude controller that is globally asymptotically stable. The controller controls the torques on the spacecraft that are generated from reaction wheels. A simple reaction wheel model was used and is shown below:

$$\dot{\boldsymbol{\omega}}_w = -\frac{\boldsymbol{T_s}}{I_w} \tag{4}$$

Here $\dot{\boldsymbol{\omega_w}}$ represents the vector of angular acceleration experienced by the wheels in the body frame. It is assumed there are three reaction wheels, one aligned with each axis of the body frame, each placed at the center of mass of the spacecraft. I_w represents the moment of inertia of each of the reaction wheels.

The controller itself is based on a simple PD controller where the proportional error signal comes from the vector part of the error quaternion $\boldsymbol{q}_e$, where $q = [\boldsymbol{q}, q_4]$. The derivative error signal comes from the angular rates in the body frame $\boldsymbol{\omega}$. The vectorized version of this controller is given below:

$$\boldsymbol{T}_s = k_p \boldsymbol{q}_e - k_d \boldsymbol{\omega} \tag{5}$$

The error quaternion is found by calculating the intermediate rotation that is the difference between the target orientation and the current orientation. This calculation is done using rotation matrices and converting the intermediate rotation to a quaternion after the calculation, however, the equation below only shows the calculation using quaternions for simplicity.

$$\begin{aligned} q_{curr} \otimes q_e &= q_{des} \\ q_e &= q_{curr}^{-1} \otimes q_{des} \end{aligned} \tag{6}$$

Here q_{des} is the desired rotation, q_{curr} is the current rotation, and q_e is the intermediate rotation or the error quaternion used in the attitude controller.

Using this controller in [16] it is shown that for any gains above 0, $k_p > 0, k_d > 0$ the controller presented in eq.5 is Globally Asymptotically Stable. Practically, the controller performs well with the gains shown in table I.

k_p	50
k_d	30

TABLE I: Gains Used in Attitude Controller

It is important to note that the outputs of the controller do match torques that reaction wheels are capable of generating. This saturation was neglected for simplicity as the maximum torque for reaction wheels can be as low as on the order of mN in some cases. Saturation was implemented to keep the commands bounded, but a large value was used. All torques are saturated at 10N.

$$T_s \leftarrow \begin{cases} T_s & \text{if} \quad T_s < 10 \quad \text{else} \quad 10 \\ T_s & \text{if} \quad T_s > -10 \quad \text{else} \quad -10 \end{cases} \tag{7}$$

B. Orbital Controller

Since the problem to be solved is one of low to medium thrust satellite control it was decided to use optimal control methodologies to design the target trajectory of the satellite. Optimal control methodologies have a long history in trajectory optimization to ensure optimal trajectories of satellites from one position to another. One popular method of combining the problem of optimal trajectory formulation with that of control is the Model Predictive Controller (MPC). MPC calculates the optimal trajectory up to a certain horizon at each time step and uses the first control input as the input to the satellite. MPC, however, is known to be extremely computationally costly. One method of dealing with this problem is to approximate the dynamics of the system in question with a linearized system that can be solved faster than the true non-linear dynamics that characterize the true system. Approximating the dynamics, however, come at the expense of MPC finding inaccurate optimal trajectories especially when far from the point of linearization and when large timescales exist which allow non-linearities to effect the system for extended periods. Despite these drawbacks, this remains one of the best methods to increase the computation speed of the controller and simplify the problem so this method is used.

In 1960 W. H. Clohessy and R. S. Wiltshire linearized [17] the non-linear dynamics that represent the relative orbital positions and velocities of two satellites in orbit. The relative positions and velocities are the states that are fed into the controller.

$$\boldsymbol{x} = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$$

The linearization is done around the target satellites orbit, as such one assumption used in this project is that the target satellite has a fixed circular orbit as this makes the linear system time invariant. The exact linearized model is given below:

$$\begin{aligned}
\Phi_{rr} &= \begin{bmatrix} 4 - 3\cos(nt) & 0 & 0 \\ 6(\sin(nt) - nt) & 1 & 0 \\ 0 & 0 & \cos(nt) \end{bmatrix} \\
\Phi_{rv} &= \begin{bmatrix} \frac{1}{n}\sin(nt) & \frac{2}{n}(1 - \cos(nt) & 0 \\ -\frac{2}{n}(1 - \cos(nt) & \frac{1}{n}(2\sin(nt) - 3nt) & 0 \\ 0 & 0 & \frac{1}{n}\sin(nt) \end{bmatrix} \\
\Phi_{vr} &= \begin{bmatrix} 3n\sin(nt) & 0 & 0 \\ -6n(1 - \cos(nt) & 0 & 0 \\ 0 & 0 & -n\sin(nt) \end{bmatrix} \\
\Phi_{vv} &= \begin{bmatrix} \cos(nt) & 2\sin(nt) & 0 \\ -2\sin(nt) & 4\cos(nt) - 3 & 0 \\ 0 & 0 & \cos(nt) \end{bmatrix} \\
\Phi &= \begin{bmatrix} \Phi_{rr} & \Phi_{rv} \\ \Phi_{vr} & \Phi_{vv} \end{bmatrix}
\end{aligned} \tag{8}$$

Here Φ represents the state transition matrix of the system where t is the length of time of the transition. $n = \sqrt{(\frac{\mu}{a_t^3}}$ where μ is the gravitational constant of the earth and a_t is the length of the semi-major axis. Since the orbit is assumed to be circular this is the radius of the orbit. The system dynamics using this linearized state transition matrix can be approximated below.

$$\boldsymbol{x}(k+1) = \Phi(a_t, 1)\boldsymbol{x} + B\boldsymbol{u} \tag{9}$$

In this equation $u \in \mathcal{R}^3$ is the control found using the MPC algorithm and B is the matrix that maps the control inputs to the states.

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{m_2}\Delta t & 0 & 0 \\ 0 & \frac{1}{m_2}\Delta t & 0 \\ 0 & 0 & \frac{1}{m_2}\Delta t \end{bmatrix}$$

Above Δt represents the timestep of the system which is chosen to be 0.1 seconds. Using this linearized system a quadratic programming (qp) solver is used for MPC. The qp problem is solved using the pyMPC library [18]. The qp optimization problem posed to be solved by the pyMPC is given below.

$$\begin{aligned} \min_{u} \quad & Q\boldsymbol{x} + R\boldsymbol{u} + R_d\Delta\boldsymbol{u} \\ \text{s.t.} \quad & \boldsymbol{x}(k+1) = \Phi(a_t, 1)\boldsymbol{x} + B\boldsymbol{u} \\ & |\boldsymbol{u}| \leq \boldsymbol{u}_{max} \end{aligned} \tag{10}$$

Above, Q is a positive definite matrix with the diagonal entries representing the penalties on each state in $\boldsymbol{x}$. The matrix R is positive definite and represents the penalties on the controls. R_d is also positive definite and represents the penalties on the change of the control values, this prevents rapid changes in the control law which in turn helps the attitude controller track the desired attitude.

$$\begin{aligned} Q &= 1 \times I_{3\times3} \\ R &= 1e^4 \times I_{3\times3} \\ R_d &= 1e^4 \times I_{3\times3} \end{aligned}$$

Other parameters that effect the controller are u_{max} which represents the maximum value the thrust can attain in one direction. This value constrains the control input to simulate low and medium thrust spacecraft. The control horizon is set to 2,000 so the optimization will solve the problem for 2000 time-steps in advance.

IV. RESULTS

The tests in this section were ran assuming a medium thrust spacecraft with one thruster. parameters are given in table II.

TABLE II: Your caption.

Parameters	Values
m_{chaser}	1.0 Kg
I_{chaser}	$1.0 \times I_{3\times3}$ Kg $\cdot m^2$
$I_{\text{reaction wheels}}$	$0.1 \times I_{3\times3}$ Kg $\cdot m^2$
$\Gamma_{\text{max,min}}$	± 1.0 N

With these parameters defining our chaser satellite, the target satellite is said to be in a circular low earth orbit, specifically the orbit of the ISS which is said to have an altitude of 408 kilometers. The chaser satellite is said to start in the same orbit, but delayed by a certain angle. In this case the chaser starts roughly 1800 meters away from the target satellite. This distance is chosen to emulate close and medium range rendezvous operations. At the altitude of the orbit of the target satellite the primary attitude perturbation is atmospheric effects causing the satellite to rotate. The effect perturbations of that order of magnitude and without are tested and their effect on the controller is seen by selecting a random moment applied to the satellite for an entire run defined by $M \sim \mathcal{N}(0, \Sigma)$, where Σ is the order of magnitude of the perturbation.

The baseline performance is tested with no moment applied to the spacecraft other than what is caused by the reaction wheels. Fig. 1 shows the relative trajectory of the two satellites. Here it is clear to see that a non-optimal route is taken, this is due to the MPC controller having a finite time horizon to plan control actions. Nevertheless, the chased satellite is able to reach the target. An easier way to visualize the trajectory is shown in figure 2. This is a visualization of the magnitude of the relative distance. This plot will be used from here on to analyze the results.

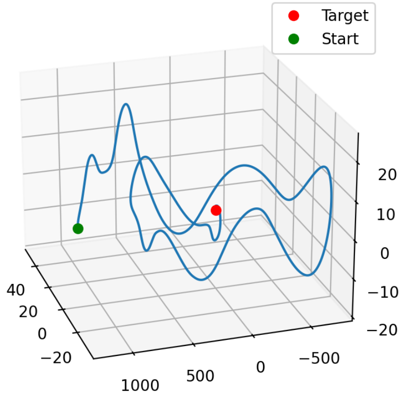

Fig. 1: Relative Trajectory of Baseline Test

The behavior shown in fig 2 is expected as the relative position seems to consistently decrease over time, however, it is clear there are large overshoots. This is due to two reasons.

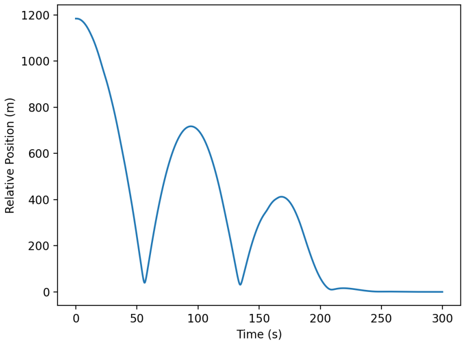

Fig. 2: Relative Position Error of the Target and Chaser

First, the limited prediction horizon by the MPC. Second, the attitude controller which is enable to track fast changes in control. To see this behavior fig. 3 shows the desired control inputs overlayed with the actual control inputs.

Fig. 3 shows one axis of the desired control and how it is tracked over time. The overall shape is tracked but with a massive amount of noise. This is due to a multitude of reasons, one being the original control law has been discretized when it was originally continuous, and finding a timestep that prevents jitter and allows the program to run in a timely manner proved a challenge. Overall, what the controller lacks in accuracy, MPC makes up for in adaptability. The jitter present in the controls is clearly shown by looking at the wheel speeds over time shown in figure 4.

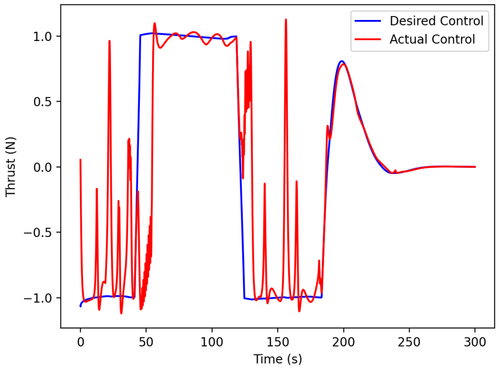

Fig. 3: Actual Control vs Desired Control Signals for the y axis Overlayed

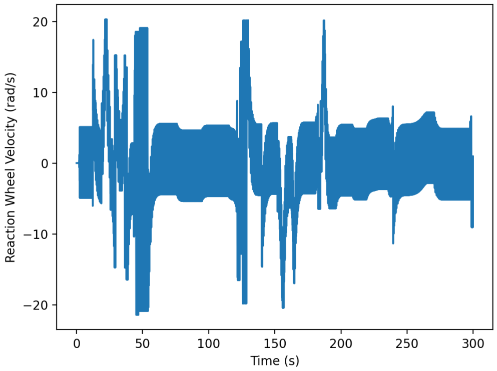

Fig. 4: Reaction Wheel Velocities Over Time

Fig. 4 shows that the reaction wheel velocities are constantly changing and jittery, however, they are bounded. This is expected since there is no constant moment applied on the spacecraft, even with a constant moment, there is nearly no change due to the short timescales. To quantify the effectiveness of the controller monte carlo trials were run with a sample sizes of 100. Fig. 5 shows the monte carlo trials for the baseline case.

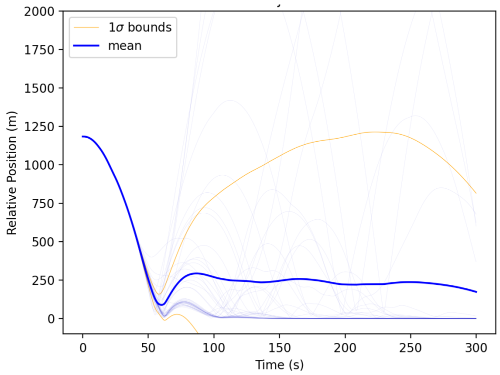

Fig. 5: monte carlo Trial With 100 Samples, Constant Random Moment $M \sim \mathcal{N}(0, 1e^{-16})$ Applied to the Spacecraft

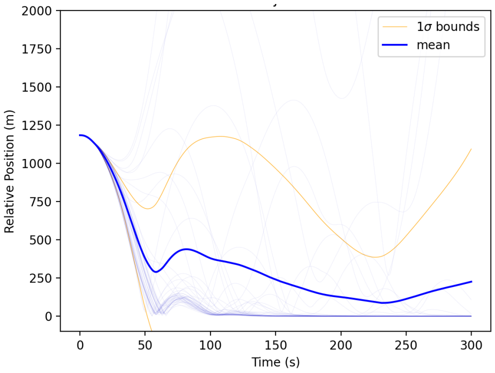

Fig. 6: monte carlo Trial With 100 Samples, Constant Random Moment $M \sim \mathcal{N}(0, 1e^{-4})$ Applied to the Spacecraft

Fig. 5 shows a monte carlo trial, with the mean trajectory plotted, with the 1σ bounds of the trajectory, and 50 sample trajectories from the run in light-blue. This run was done by applying an undetectable small moment on the order of $1e^{-16}$ (machine epsilon) to the spacecraft. As seen from the sample trajectories, not all are bounded and a small amount diverge, however, the vast majority converge to 0 error. This divergence may be a result of a combination of factors that arise from breaking assumptions in our controllers. For one, there exists no guarantees on the attitude controller when tracking is involved. Also, although MPC is robust, is does not expect actual control inputs to be different than desired control inputs. The lack of guarantees align to make it so behavior like this is possible, and as seen in the monte carlo trials, it seems to happen often with this controller. Fig. 6 shows the effect of random moments on the controller.

Surprisingly the effect of a moment on the chaser spacecraft results in a similar monte carlo trial to the baseline. Fig. 6 demonstrates this, and shows that a vast majority of the runs converge while a few diverge on bring the mean and the standard deviation upwards. This again can be seen as a result of what is believed to have caused the baseline case, however, with a significant applied moment the attitude controller is not said to converge even for a regulation problem. This adds even more uncertainty into our controller, which may be why the trajectories tend to diverge sooner than in the baseline case.

Finally, this controller was tested on a low thrust case in fig. 7. The low thrust controller had bounded inputs in between $[-0.01N, 0.01N]$, about the thrust of a hall thruster. As demonstrated by fig. 7 this controller structure failed to dock the satellites and instead the error increases without bound within the timescale of the simulation. This is believed to be

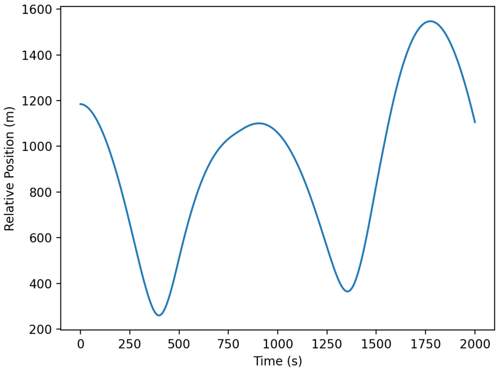

Fig. 7: Relative Error of a Low Thrust Trajectory

the result of two factors. First, the non-linearities in the system may have a greater effect on the spacecraft in this case as it has less control authority to counteract the effects, as the spacecraft is far away from the linearization point (the target spacecraft) for extended periods the non-linearities progressively overload the capabilities of the controller. Second, the MPC controller with a finite time horizon is unable to plan for long enough periods into the future to keep the spacecraft bounded near the target spacecraft.

V. Conclusion

The attitude controller and orbit controller pair successfully caused the target spacecraft and the chaser spacecraft to dock for the majority of trajectories despite applied moments. However, through monte carlo analysis there are a significant number of outliers whose relative position error increases unbounded nature, again irregardless of moment applied. This is thought to be the result of violating assumptions made in the attitude controller and the orbital controller. This control scheme was tested with a spacecraft only capable of low-thrust and the result was a failure to cause docking. This is believed to be caused by a shortcomings of the linearization used for the orbit controller as well as a limited planning horizon for the orbit controller.

VI. Future Work

The first extension on this preliminary work is with respect to the performance of the attitude controller. As seen in fig. 3 the actual thrust vector and the desired thrust vector are quite different. As stated in the analysis this is mainly due to constraints in the timestep, as to keep the program running within a reasonable amount of time a large timestep had to be used. Ideally a lower timestep would be needed, however, this is prohibitive because of the orbital controller. One fix is to allow for controllers that operate on different timesteps with the attitude controller reacting much faster than the MPC. This is indeed how many control systems work with a slow loop and a fast loop. Beyond this, more effective controller tuning could be considered. One way to accomplish this is to devise a loss function over which to tune the controller and optimize the parameters based on this. An effective way of doing this would be to build this simulation to be differentiable, therefore gradients of the loss function with respect to the controller parameters could be used to improve the performance.

Of course neither pf these fixes address the underlying issue, that this controller has no guarantees for tracking or when outside moments are introduced. Using a controller specially built to bound the worst case error in the presence of these situations would be most desirable. One example of a controller that adapts to drag forces is shown in [19]. A similar methodology could be used to design a controller that limits the error for the use case of satellite docking. This combined with the suggestions above would likely result in a massive performance increase compared to the previous attitude controller.

- Non-linear MPC As for the orbital controller, two main problems exist. First, a linearization is used which inhibits the best performance of the controller. Non-linear MPC would naturally fix this, however, this comes at the expense of speed. The second issue relates to the problem of speed, the MPC implementation had a relatively short horizon. To fix this a longer horizon could be used, again at the expense of speed. To fix both of these problems a linearization about a desired trajectory found via optimal non-linear trajectory optimization may be used to control the satellite on this trajectory. This would in essence be a single MPC run that calculates the full optimal trajectory from point to point, followed by a much quicker controller that aims to keep the satellite on that trajectory using a linearization about the trajectory. A single optimization run is not required, and these runs could be computed at larger time intervals, as often is allowable by the MPC controller.

Now individually, both of these fixes would drastically improve the performance of the controller, however, together they still lack robust guarantees for the shortcomings of either controller schemes. To overcome there seems to be two avenues. First, a probabilistic approach could be taken for the orbital controller where, if certain guarantees on the boundedness of the attitude controller are present, worst case scenario of where the satellite is pointing the satellite will still make it to its target with some accuracy. Second, a more classical robust control approach could be taken where methods like tube MPC are implemented to keep the satellite within a certain tube of uncertainty despite what happens. Other robust control methods could be used to build a more reliable attitude controller that will still control the satellite despite uncertainty.

Of course, a non-linear MPC controller that contains both the attitude states and the orbital states would be an effective way of solving this problem, however, in order to create a lightwight controller that involves solving as small of an optimization problem as possible is desirable. Nevertheless, it would be prudent in the future to compare the controller with this, possibly more costly, approach to truly see the best controller to implement on a true satellite.

Note: The code is not attached for brevity, for a detailed look at this code please see the github repository at [20]

References

[1] March 16, 1966: Gemini's first docking of two spacecraft in earth orbit. https://www.nasa.gov/image-article/march-16-1966-geminis-first-docking-of-two-spacecraft-earth-orbit/. Accessed: 2024-03-19.

[2] Wigbert Fehse. *Automated Rendezvous and Docking of Spacecraft*. Cambridge Aerospace Series. Cambridge University Press, 2003.

[3] Keenan Albee, Charles Oestreich, Caroline Specht, Antonio Espinoza, Jessica Todd, Ian Hokaj, Roberto Lampariello, and Richard Linares. A robust observation, planning, and control pipeline for autonomous rendezvous with tumbling targets. *Frontiers in Robotics and AI*, 8, 09 2021.

[4] Yazhong Luo, Jin Zhang, and Guojin Tang. Survey of orbital dynamics and control of space rendezvous. *Chinese Journal of Aeronautics*, 27(1):1–11, 2014.

[5] Starfish space. https://www.starfishspace.com/the-otter/. Accessed: 2024-03-19.

[6] Impulse space. https://www.impulsespace.com/helios. Accessed: 2024-03-19.

[7] Argo space. https://www.argospace.com/. Accessed: 2024-03-19.

[8] Pini Gurfil. Spacecraft rendezvous using constant-magnitude low thrust. *Journal of Guidance, Control, and Dynamics*, 46(11):2183–2191, 2023.

[9] Jianqiao Zhang, James Biggs, Dong Ye, and Zhaowei Sun. Finite-time attitude set-point tracking for thrust-vectoring spacecraft rendezvous. *Aerospace Science and Technology*, 96, 11 2019.

[10] Vijay Muralidharan and Reza Emami. Concurrent rendezvous control of underactuated spacecraft. *Acta Astronautica*, 138, 05 2017.

[11] Yaguang Yang. Coupled orbital and attitude control in spacecraft rendezvous and soft docking. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 233(9):3109–3119, 2019.

[12] Annie Jose, Divina D V, Dona Thomas, Malavika S, Venkateswaran J, Rajeev U P, and Imthias Ahmed T P. Guidance for spacecraft docking incorporating model uncertainties: A linear quadratic tracking approach. In *2021 Seventh Indian Control Conference (ICC)*, pages 383–388, 2021.

[13] R. Sharma and A. Tewari. Optimal nonlinear tracking of spacecraft attitude maneuvers. *IEEE Transactions on Control Systems Technology*, 12(5):677–682, 2004.

[14] H. B. Hassrizal and J. A. Rossiter. A survey of control strategies for spacecraft attitude and orientation. In *2016 UKACC 11th International Conference on Control (CONTROL)*, pages 1–6, 2016.

[15] D.M. Schrello. Passive aerodynamic attitude stabilization of near earth satellites. volume i. librations due to combined aerodynamic and gravitational torques.

[16] Anton H De Ruiter, Christopher Damaren, and James R Forbes. *Spacecraft dynamics and control: an introduction*. John Wiley & Sons, 2012.

[17] W. H. CLOHESSY and R. S. WILTSHIRE. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960.

[18] Marco Forgione, Dario Piga, and Alberto Bemporad. Efficient calibration of embedded MPC. In *Proc. of the 21st IFAC World Congress 2020, Berlin, Germany, July 12-17 2020*, 2020.

[19] Runhan Sun, Camilo Riano-Rios, Riccardo Bevilacqua, Norman G. Fitz-Coy, and Warren E. Dixon. Cubesat adaptive attitude control with uncertain drag coefficient and atmospheric density. *Journal of Guidance, Control, and Dynamics*, 44(2):379–388, 2021.

[20] Github repository. https://github.com/Bwodetzki/final_project.