MAE 6780: Multivariable Control Theory

Recreation of Simple Adaptive Control (SAC) for
Spacecraft Proximity Operations

Eric Xue[1]

[1]Cornell University, Masters of Engineering Student

May 20, 2024

1. Introduction

Spacecraft rendezvous and proximity operations (RPO) are associated with the ability of two or more independent space objects to maneuver in close proximity to one another. This encompasses a wide field of important missions that include the supply and repair of the ISS, docking maneuvers of SpaceX's Dragon capsule, and on-orbit servicing of satellites from startups such as Starfish & Turion Space. With RPO, one spacecraft is designated as the Target, while the primary and often controllable spacecraft is referred to as the Chaser. There exist many other naming conventions including Chief/Deputy, Leader/Follower, but Chaser/Target will be the one used here.

Despite many past successful missions, RPO is still an active area of research as most existing control techniques for RPO involve a model-based approach, requiring substantial knowledge of the plant and system parameters for an accurate translation onboard. But RPO is inherently characterized by both uncertainties regarding the Target and Chaser. With the Target, the risks of unknown parameters are amplified when considering uncooperative targets such as decommissioned satellites, debris removal missions, and asteroid mining. Unfortunately, these are harder to mitigate with an adaptive control scheme and generally done with accurate prior knowledge. The Chaser on the other hand generally faces variability in mass properties due to fuel consumption, solar array deployment, sloshing, and more, which in the past has seen different successful robust control schemes.

A potential robust solution is one seen in this project, featuring Simple Adaptive Control (SAC) which involves changing the controller gain based on comparing the output of the system to a model [1]. It was seen that despite the model not being tuned to follow the reference directly, this setup is extremely robust to both mass property uncertainties ranging from 2x the nominal mass to 0.2x. The setup was also successful for small perturbations. All this was compared to a PD controller tuned to follow the reference, which still had some robustness to a lower mass from the nominal case, but showed oscillation when increasing the mass. The PD controller also failed to stabilize properly when faced with the same small perturbation.

2. Adaptive Control Theory

2.1 Model Reference Adaptive Control (MRAC)

Adaptive control focuses on using a "good" / ideal model as a reference such that the output of the "bad" / true plant can follow the output of the model, and obtain the desired closed-loop characteristics by driving the error between the two to zero. An adaptive controller takes in this error, as well as any other parameters within the model (reference input, control, state, etc), and adaptively changes the gains according to a set learning rate. This method is often referred to as direct Model Reference Adaptive Control (MRAC) since the gains are calculated directly from the error without any estimations of the parameters.

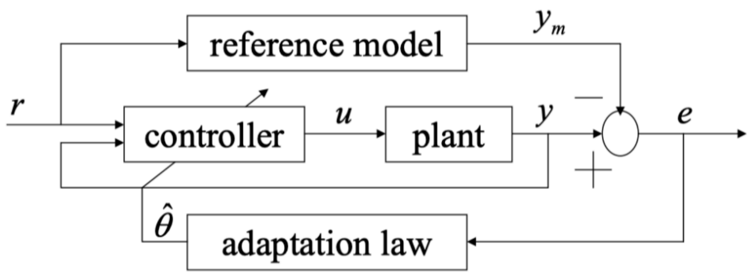

Figure 1: MRAC Block Diagram [2]

A big drawback regarding adaptive control is the lack of a general method in designing/selecting an adaptation law, as each one must be tailored toward a specific use case. This limitation comes from having to prove the stability of the law selected for the system. Proving the stability of the law is often the greatest challenge that also varies depending on the system of choice. Despite this, many different methods have been explored for more restricted systems, with a popular method being the MIT rule [3]. This rule is derived from the minimization of the integral squared error between the model and the plant and requires the system to be Linearly Time-Invariant (LTI) of the form:

$$\dot{x}_p = A_p x_p + B_p u_p, \qquad y_p = C_p x_p \tag{Eq1}$$

Where x_p represents the state vector, u_p represents the control input, and A_p, B_p, C_p are the system matrices. The proposed gains for the control input are then tuned via a gradient descent method as a way to minimize the cost function. Methods such as these result in an adaptation law that requires the order of the model to be matched to the plant such that the output of the model closely reproduces the output of the plant. This also forces a trade-off between the learning rate of the law versus the stability of the overall system.

Due to MRAC essentially forcing the output of the system to follow an ideal model, it often forces the model to match the order of the plant, further limiting the number of permissible plants to choose from as models. As an alternative, work has been done on Simple Adaptive control laws as a way to relax some of these constraints.

2.2 Simple Adaptive Control (SAC)

The first benefit that comes from SAC as opposed to traditional MRAC is the relaxation of the matching of the orders between the systems. With SAC, the model does not have to completely reproduce the output of the model aside from the desired system behavior of the plant. Recall that the goal of MRAC is to get the desired closed-loop response by tracking the exact output of the ideal model as closely as possible, thus leading to the restrictions mentioned above. Instead, SAC forces the plant to obtain just the desired system response of a chosen model, which will also indirectly track the reference model to a satisfactory degree, whether its a different order or not. The subtle difference between the two involves the relaxation of matching the plant with the model. With MRAC, things tend to fall apart if the model isn't accurate or has uncertainties associated with it due to it being a dynamic system with online parameter estimation [1]. However, SAC enjoys the same benefits provided without having to have an accurate model at all, instead focusing on other more general system responses such as rise time, settling time, damping, etc. This is desirable since the reference model can be something as simple as a first-order pole that satisfies some system requirement (Overshoot, settling time, rise time, etc) to a user input.

2.2.1: ASP / ASPR Systems

Prior to being able to implement a SAC, the plant will still have to be LTI of the form shown above (Eq1). In addition to this, the plant will also have to be almost strictly passive (ASP) and almost strictly positive real (ASPR) to stabilize with SAC. Mathematically, this is satisfied if there is an imaginary gain K such that the closed loop A matrix satisfies the Kalman-Yakubovich-Popov (KYP) condition, which is a generalization of the Lyapunov equation. This is essentially checking if a Lyapunov function exists for a given system, which fulfills one of the baselines for proving stability [1]. This means that two positive definite symmetric matrices, P and Q, exist for this system such that the following algebraic Lyapunov equation is satisfied:

$$PA_p + A_P{}^T P = -Q \tag{Eq2}$$

It is also noticed that with strictly positive real (SPR) systems, the Lyapunov equation can be shown to be asymptotically stable. Since SPR systems are rare, this definition can be further generalized to include ASPR which are much more common. However, for an ASPR system to have its Lyapunov function be asymptotically stable, its closed-loop dynamics (A_{CL}) for an imaginary feedback gain must satisfy the KYP conditions instead. I.E, a gain K must exist such that:

$$A_{CL} = A_p - B_p K C_p \tag{Eq3}$$

$$PA_{CL} + A_{CL}{}^T P = -Q \tag{Eq4}$$

A system then is considered ASPR if there exists a constant feedback that will make it SPR. Other definition of ASP / ASPR systems are also given in the following source [4]. Furthermore, any strictly proper strictly minimum phase LTI system with the system matrices C_p and B_p being positive definite symmetric is ASP and its transfer function ASPR, giving a faster check of this being true than deriving the Lyapunov function [1].

2.2.2: Adaptive Control Law

For a direct adaptive control law, the reference model must also be LTI of the form:

$$\dot{x}_m = A_m x_m + B_m u_m, \qquad y_m = C_m x_m \tag{Eq5}$$

where the input into this model, u_m, is simply the reference input that the plant is tracking as shown in **Fig 1**. As previously stated, adaptive control attempts to decrease the error between the outputs of the two systems, meaning the control objective will be the output tracking error defined as:

$$e_y = y_m - y_p \tag{Eq6}$$

The proposed SAC method uses all available information about the model as part of the control law, which includes its state x_m, control/reference to the system u_m, as well as the overall error e_y:

$$u_p = K_e(t)e_y + K_x(t)x_m + K_u(t)u_m \tag{Eq7}$$

Where the gains are changing according to an adaptation mechanism that guarantees stability. This control law is general to the basic SAC method, with slight alterations being made to the adaptation mechanism depending on the system.

2.2.3: Proof of Stability

A method to prove stability via the Lyunpunov function is described in detail by Ulrich, but the method will be summarized here [1]. The proof begins by defining an ideal LTI plant of the following (Eq.1):

$$\dot{x}_p^* = A_p x_p^* + B_p u_p^*, \qquad y_p^* = C_p x_p^* \tag{Eq8}$$

This ideal plant assumes that it is able to track the model perfectly along a bounded state trajectory, setting the output tracking error from Eq.3 to zero. This lends itself to an ideal control law by setting e_y it to zero.

$$u_p^* = \overline{K_x} x_m + \overline{K}_u u_m \tag{Eq9}$$

Where the gains are the constant steady-state values. From there, a new state error e_x is defined as the error between the ideal state vector (x_p^*) and the true state of the plant (x_p).

$$e_y = C_p x_p^* - C_p x_p = C_p e_x \tag{Eq10}$$

This state error is differentiable, allowing for an equation to be obtained in terms of the state matrices of the plant, the gain matrices, and the reference.

$$\frac{d}{dt} e_x = (A_p - B_p \overline{K}_e C_p) e_x - B_p (K_I(t) - \overline{K}) r - B_p K_p(t) r \tag{Eq11}$$

Where $\overline{K}$ represents a concatenated vector of the steady state values of the gains $[\overline{K}_e \overline{K}_x \overline{K}_u]$ Lastly, to guarantee stability, a quadratic positive definite Lyapunov function is considered to be of the form:

$$V(t) = e_x^T P e_x + tr[(K_I(t) - K) \Gamma_I^{-1} (K_I(t) - K)^T] \tag{Eq12}$$

Taking the derivative of this and plugging in the derivative of the state error, allows for the following:

$$\frac{d}{dt} V(t) = - e_x^T Q e_x - 2 e_y^T e_y r^T \Gamma_P r < 0 \tag{Eq13}$$

This shows the time derivative of this Lyapunov function is negative definite in e_x, meaning that the Lyapunov function will tend toward an equilibrium point for small perturbations in the state error, thus proving stability.

2.2.4: Parallel Feedforward for non-ASPR systems

The proof introduced above only applies to systems that can take advantage of the ASPR properties of the system itself, however, not every system satisfies the ASPR condition necessary. For this control scheme, the author uses a parallel feedforward configuration (PFC) can be used to satisfy the ASPR condition. This envisions a controller H(s) that will stabilize the original plant G(s). By assuming a high gain for H, a modified plant can be found via:

$$G_{modified}(s) = G(s) + H^{-1}(s) \tag{Eq14}$$

The goal of this is to introduce and modify the plant such that $G_{modified}(s)$ is now minimum phase and relative degree one, which makes the modified transfer function ASPR and thus allows access to the proof from 2.2.3. This strategy is akin to assuming an imaginary stabilizing controller and feeding forward the inverse to cancel its output in order to obtain the desired properties without changing the original output of the plant. This requires that the gain of the imaginary controller H is high, such that the inverse will be small, making the modified plant as close to the true plant as possible.

3. Simulation Setup

3.1 Dynamics

When the relative motion of two spacecraft is assumed to be relatively close to each other and in a Keplerian circular orbit, the non-linear dynamics can be linearized via a first-order Taylor Expansion & generalized binomial expansion to obtain the Clohessy-Wiltshire (CW) equations (Eq.15).

$$\begin{gathered} a_x - 3n^2x - 2nv_y = \frac{F_x}{m} \\ a_y + 2nv_x = \frac{F_y}{m} \\ a_z + n^2z = \frac{F_z}{m} \end{gathered} \tag{Eq.15}$$

Where a_x, a_y, a_z are the double derivatives of the states x y z, and F_x, F_y, F_z are the control input into the system via thrusters.

The CW equations are powerful as they describe the motions of the non-linear case to an accurate degree as long as the relative distance is small while presenting a linear model that allows for classical control techniques. Furthermore, when the spacecraft is moving orders of magnitude faster than the orbital motion (such as a spacecraft in LEO) such that the other terms dominate, the terms involving the mean motion (n) can be ignored, resulting in decoupled double integrator models along each axis (Eq.16).

$$G_{axis} = \frac{1/m}{s^2} \qquad \text{(Eq.16)}$$

Due to this decoupling, only one of the axes has to be simulated, with the others showing the same results as there will be no thruster dynamics nor any differences between them.

3.2 Adaption Mechanism

Referring back to (Eq.4), each of the gains can vary with time, as is the nature of adaptive control, and are defined as follows:

$$\begin{aligned} K_e(t) &= K_{Pe}(t) + K_{Ie}(t) \\ K_x(t) &= K_{Px}(t) + K_{Ix}(t) \\ K_u(t) &= K_{Pu}(t) + K_{Iu}(t) \end{aligned} \qquad \text{(Eq.10)}$$

Where the $K_e(t)$ represents the stabilizing control gain matrix $K_x(t)$ and $K_u(t)$ represents the feedforward control gain. Each of these has a proportional component and an integral component defined that are updated as follows:

$$\begin{aligned} K_{Pe}(t) &= e_y e_y^T \Gamma_{Pe} \\ K_{Px}(t) &= e_y x_m^T \Gamma_{Px} \\ K_{Pu}(t) &= e_y u_m^T \Gamma_{Pu} \\ \frac{d}{dt} K_{Ie}(t) &= e_y e_y^T \Gamma_{Ie} \\ \frac{d}{dt} K_{Ix}(t) &= e_y x_m^T \Gamma_{Ix} \\ \frac{d}{dt} K_{Iu}(t) &= e_y u_m^T \Gamma_{Iu} \end{aligned} \qquad \text{(Eq.11)}$$

Where Γ represents the associated learning rates and are held constant.

Going back to the control law (Eq.4), it can be seen that the terms associated with the model (x_m and u_m) are being feedforward to the control in an attempt to predict the amount of control necessary which can improve system responses. Meanwhile, the term associated with the error represents the feedback component and is the regular stabilizing term. As the error between the model and plant decreases and thus reaches a steady state, the e_y term will drop out, which then leaves the x_m and u_m terms. Notice that the gains are all dependent upon e_y, where if there is no error, then the proportional components will reach zero, and the derivative of the integral terms will reach 0, representing a constant integral term to maintain zero error at the desired steady-state value.

3.3 Modified Plant and Ideal Model

This proximity operation model is also not ASPR due to it having a relative degree of two, thus failing the simplified condition of ASPR. This then requires an additional implementation of the parallel feedforward mentioned above. A proposed PD controller is selected as the feedforward term such that [1]:

$$H(s) = K_p + K_d s = K_p(1 + \frac{s}{w}) \qquad \text{(Eq.12)}$$
$$H^{-1}(s) = (\frac{K_p^{-1}}{1+s/w})$$

This modifies the original plant from Eq.9 such that:

$$G_{modified}(s) = G_{axis} + H^{-1}(s) = \frac{K_p^{-1}(s^2+(K_d/m)s+K_p/m)}{s^2(1+s/w)} \qquad \text{(Eq.13)}$$

This modified plant can be observed to be a relative degree of one due to it having three poles and 2 zeros, as well as having a minimum phase (no extra poles/zeros are used to represent this same system).

Lastly, the proposed Ideal model for the plant to follow will be a simple second-order transfer function that represents a spring-mass-damper system:

$$G_{modified}(s) = \frac{w_n^2}{s^2+2\zeta w_n s+w_n^2} \qquad \text{(Eq.14)}$$

Where w_n and ζ represents the natural frequency and damping ratio of the ideal closed-loop system.

3.4 Simulink Model

Attached below is a block diagram of the proposed SAC architecture:

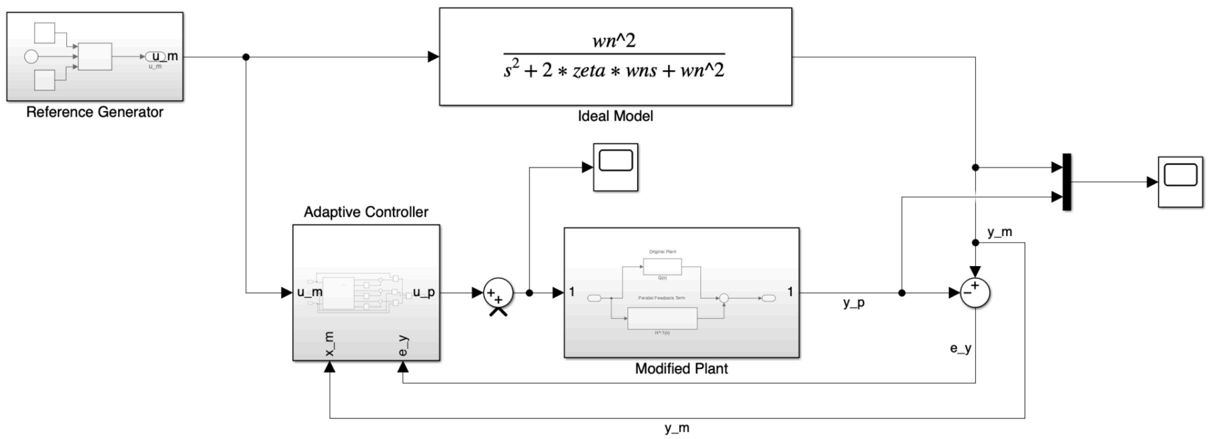

Figure 2: Block Diagram of SAC Architecture

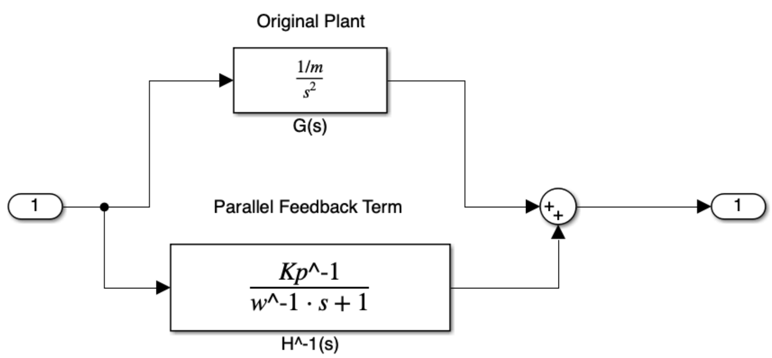

Figure 3: Modified Plant Block

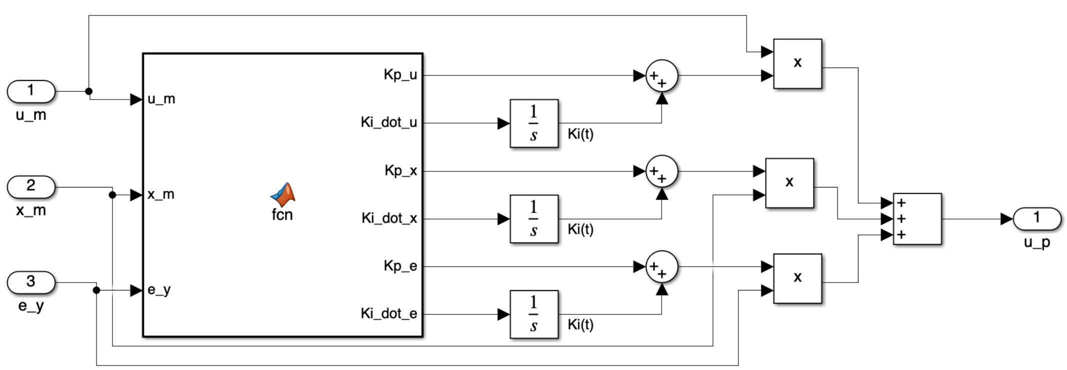

Figure 4: Adaptive Controller Block

The user-defined MATLAB function block in **Fig 4** returns the proportional gains and the derivative of the integral gains of the adaptive control law as described by Eq. 11.

4. Simulation Results

4.1. Simulation Constants

The simulation follows that of Barkana, who models nanosatellite maneuvering in close proximity to another spacecraft. The control parameters chosen for the adaptive control law are as follows:

Table 1: Selected Learning Rates

Γ_{Pe}	Γ_{Px}	Γ_{Pu}	Γ_{Ie}	Γ_{Ix}	Γ_{Iu}
10e3	10e2	25e3	100	10	10

Table 2: Other Parameters Selected

Ideal Model Param		Parallel Feedback		Mass of spacecraft	PD Controller	
ξ	w_n	K_p	w	m	Prop. Gain	Derv. Gain
0.9	0.13	200	2.5	4.3	0.1	20

The initial states were all started at 0. The adaptive algorithm was also initialized such that the integral gains were started at 0. Lastly, a basic PD controller with the parameters described in Table 2 was used as a comparison.

4.2. Baseline Simulation

Without changing the parameters, a baseline simulation was run to test the efficacy of the proposed plant. The reference was a sinewave with a step-off at ~50 seconds to simulate a gradual climb to a step input of 1 to reduce the actuator output from an instant change in the inputs. This isn't a large concern simulation-wise but would be a design consideration when having to choose between controller types. Thus, to make this process more realistic, a more gradual reference was given as an effort to give this controller a better comparison. The adaptive controller was then compared to a baseline PD controller tuned to the reference input. Note: Plots of the error between the reference and model output were ignored due to this being constant and deterministic based on the selected reference and model. Thus, analysis based on this is not insightful unless the tuning of the ideal model is desired. Instead, only error plots between the model and the plant were shown to evaluate the controller's ability to track the model.

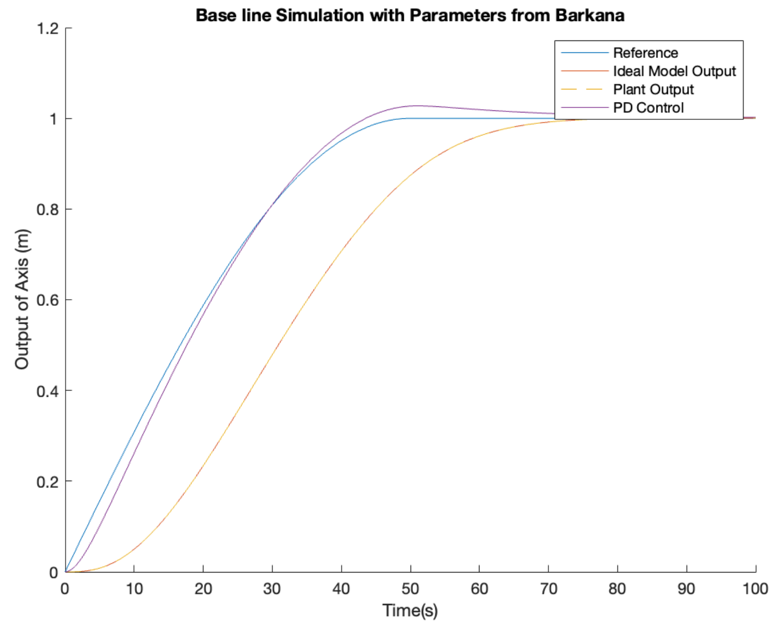

Figure 5: Overall Baseline Simulation

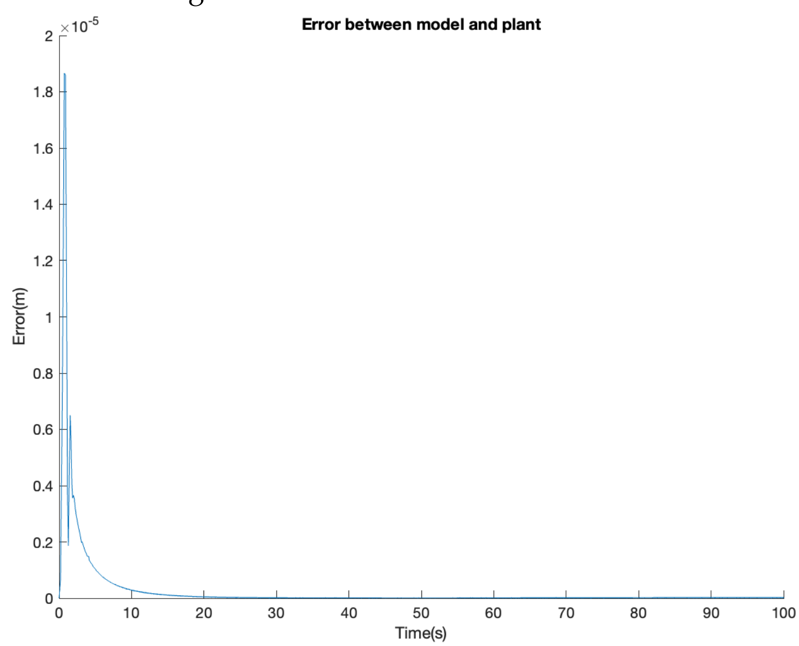

Figure 6: Error between Ideal Model and Plant model

In this base case, it is observed that with tuning, the PD controller works extremely well for tracking the desired reference. From the graphs, it can also be seen that the adaptive control law presented is able to track the ideal model with extremely low tracking error, and after ~20 seconds, the plant perfectly matches the plant. While the adaptive controller does not track the reference directly / as well as the PD controller, this is not an issue considering the overall goal of having the plant model track the ideal model output closely was achieved. If this were to be implemented and the response requirements were extremely stringent, then more work could be done on the ideal model such that it is able to track the reference a lot more closely than the PD controller was.

4.3. Adaptability Under Uncertainty

While tracking error should be a goal with the adaptive controller, it is more interesting to compare the PD controller with the adaptive controller for uncertainties in the plant, especially regarding differences in mass and random disturbances.

4.3.1. Uncertainty in Mass

With an adaptive control scheme, it is interesting to check the response of a system using the parameters in 4.2 to a plant with a different mass and observe if the response will change drastically. This would represent a scenario in which either fuel is depleted and the mass is lower, or if the spacecraft docks onto another and they now move together, in which the mass is higher. Since these parameters were made for a 4.3 kg satellite, a change in mass of +4.3 kg for the docking case with a similar SV and a change in mass of -3.44kg (representing a mass fuel ratio of 90%, where 80% of total fuel used) was observed.

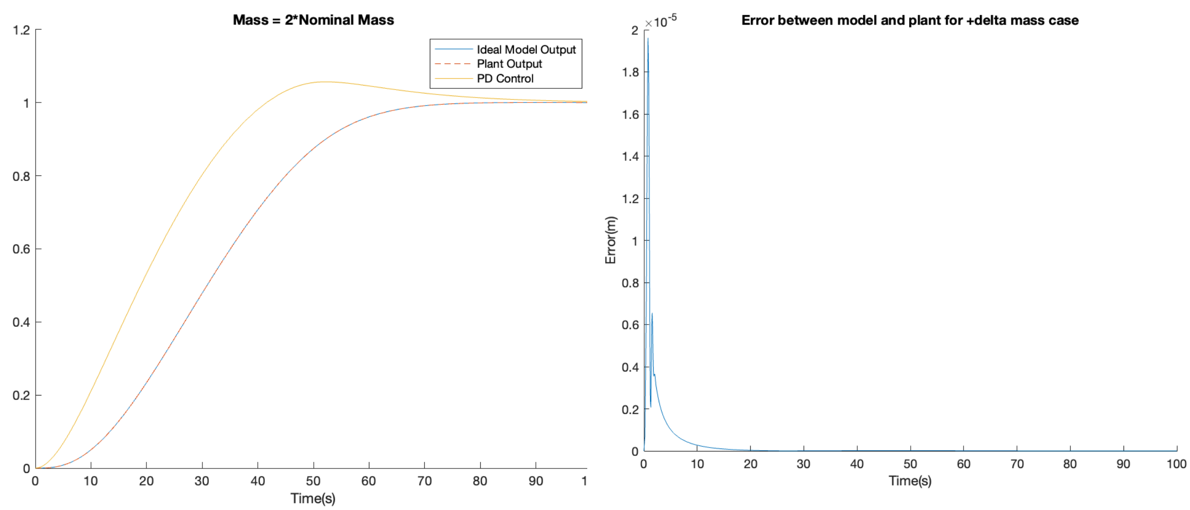

Figure 7: Simulation of Increase in mass case

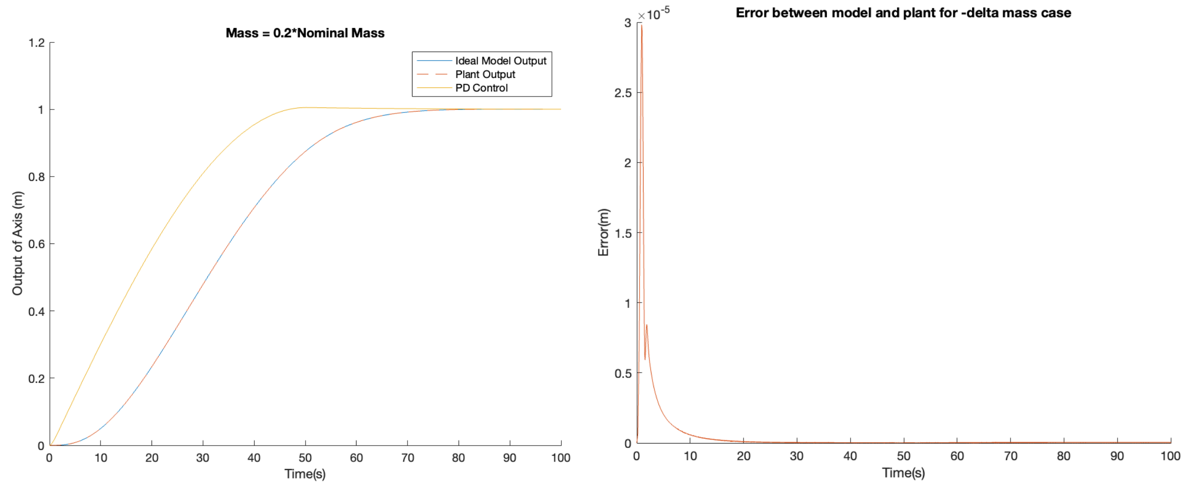

Figure 8: Simulation of Decrease in mass case

Comparing the nominal cases to these changes in the mass, it can be seen that there are little to no differences from the adaptive controller between the cases. However, there is a difference from the PD controller once the mass is increased, in which there is now an overshoot in the response, which for a given mission may not be desirable and even fail certain requirements. The decrease in the mass case did not see many changes, most likely due to the derivative gain being high enough to cancel out the change in this mass.

From these simulations, it is seen that while the error between the model and plant for the error does increase as the mass changes, these are an extremely small order of magnitude. Strangely enough, it can be seen by observation that the error plot of the increased mass case has a faster convergence to 0 than both the decreased mass and even the nominal mass. This isn't a cause for concern but indicates a further tradeoff between the parameters selected for the nominal mass being optimized for minimal error, but not necessarily minimal response time.

As a note, this doesn't account for any saturations of the actuator, so doesn't simulate the true cases of the same spacecraft with increased/decreased masses. Instead, this simulates two different spacecraft with different mass properties from the nominal case. This is because changing the mass without a limit on the actuator effort would just result in a change in the actuator input to make up for the differences. If further work were to be done into this to make it more realistic, a saturation on the actuator output between the three simulation would be done to observe whether the system response would still remain as accurate and observe any further changes.

4.3.2. Randon Disturances

Another evaluation of the adaptability of this control scheme would be to observe any random disturbances to the input of the system. A perturbation in the form of a sine wave was added to the control input:

$$u_{perb}(t) = 0.1sin(0.1t) \quad \text{(Eq.15)}$$

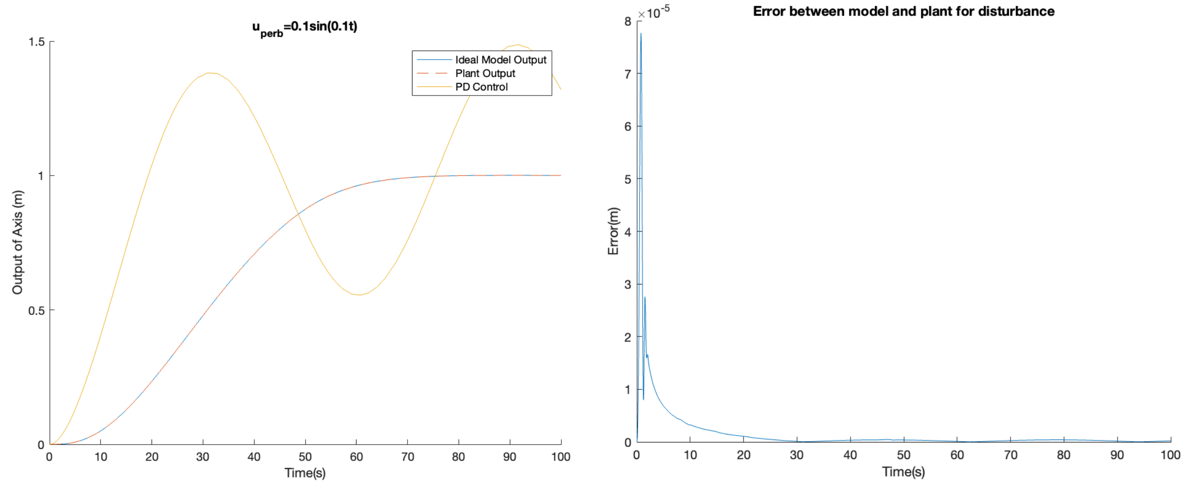

Figure 9: Simulation of Disturbance

This shows the benefit of the adaptive controller over a non-adaptive one. With a small disturbance, the PD controller is quickly destabilized due to the undesirable behavior early on being aggravated, which compounds into more oscillations. However, the adaptive controller is able to handle the same perturbations without any large deviation from the reference model. As seen from the error plot, while the error between the model and plant did increase slightly, this is still on a very small order of magnitude. There is also no longer a steady state of 0 error as the perturbation constantly acts on the system, forcing the controller to correct for it in an attempt to match the model. This could be a potential drawback in scenarios where fuel usage is extremely limited and may require some logic that prevents any correction maneuvers once both the position error and velocity are close to 0. Barakana also included further modification to the integral gain adaptive law for further disturbance rejection, which wasn't implemented here, but, it can be seen that even without it, the system is able to respond extremely well [1].

5. Conclusion:

In conclusion, SAC offers an attractive alternative to getting the benefits of MRAC without having the restrictions of an ideal model that normally comes with model-based adaptive control. When comparing the algorithm presented by Barkana to a traditional PD controller, the tuned PD controller performed better than the SAC, but once any form of uncertainty/disturbances to the system, this no longer becomes the case.

Further work could be done to validate the results of this system to the full CW equations or even the full non-linear dynamics. This was not done in this case since the assumptions that allow the non-linear dynamics to collapse onto the double integrator model hold extremely well on small time scales (such as the 100 seconds used here) as well as when the magnitude of the desired translational movements is orders of magnitude smaller than the orbit. Instead, it could be interesting to see whether the controller is able to adapt to cases where these assumptions no longer hold, forcing the plant to be non-linear.

Bibliography:

[1] Steve Ulrich, Dustin L. Hayhurst, Alvar Saenz Otero, David Miller, and Itzhak Barkana, "Simple Adaptive Control for Spacecraft Proximity Operations"
URL:
https://www.researchgate.net/publication/259702217_Simple_Adaptive_Control_for_Spacecraft_Proximity_Operations

[2] Ding Zhao, "Module 2-4: Introduction to Adaptive Control"
URL: https://safeai-lab.github.io/LCS/2020/M2-4.pdf

[3] Iven M.Y. Mareels, Brian D.O. Anderson, Robert R. Bitmead, Marc Bodson *, Shankar S. Sastry, "Revisiting the Mit Rule for Adaptive Control"
URL: https://www.sciencedirect.com/science/article/pii/S1474667017559546

[4] R. Livneh and G.L. Slatert, "SOME CONDITIONS FOR STRICTLY POSITIVE REAL AND ALMOST STRICTLY POSITIVE REAL STRUCTURES"
URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5725635