PID and LQR controller for orbital rendezvous of two spacecraft

Andres Pulido

December 2021

1 Problem Statement

The design will consist of an autopilot to achieve an orbital rendezvous for two spacecraft. These spacecraft are denoted as resident space object (RSO) and chaser (CHS) with δ_r representing the relative difference in their positions. The dynamics are formulated in the orbital frame of the RSO using an assumption of a circular orbit. The relative position, δr, is reduced to a 2-component vector since any out-of-plane motion is assumed to be negligible. The dynamics can thus be represented using 4 states of x as the radial separation in km, y as the tangential separation in km, $\dot{x}$ as the radial velocity in km/s, and $\dot{y}$ as the tangential velocity in km/s. The inputs are thrust resulting as u1 in the radial direction and u2 in the tangential direction. The vehicle has 2 sensors with z1 measuring radial separation and z2 measuring tangential velocity. The model also includes a pair of actuator with transfer function models. Check response from initial conditions of xo= [3.10,39.16,0.00002,0.0041]

1.1 Objectives

1. Regulate the states toward the origin 2. Limit the actuator motion: radial actuator has position limits of $\pm$5880 N and rate limits of $\pm$30 N/s and tangential actuator has position limits of $\pm$820 N and rate limits of $\pm$500 N/s. 3. Account for noise and disturbances: disturbances in the actuation of $\pm$3% and noise in the sensors of $\pm$8% 3. Minimize actuation:

2 Solution

To solve the regularization problem, a PID and a LQR controller were designed and tuned. Both controllers were implemented in Simulink (Figure 1 and 2) because it was more straightforward to implement the noise, disturbances, saturation and rate limiters than the scripting method. Additionally, the estimator was chosen to be a Kalman Filter because it was much easier to implement in Simulink than a state observer.

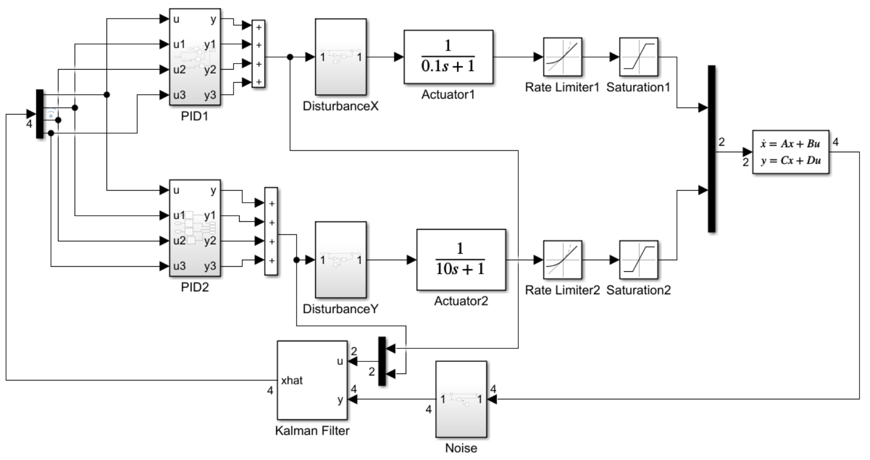

Figure 1: Block Diagram of PID controller in Simulink

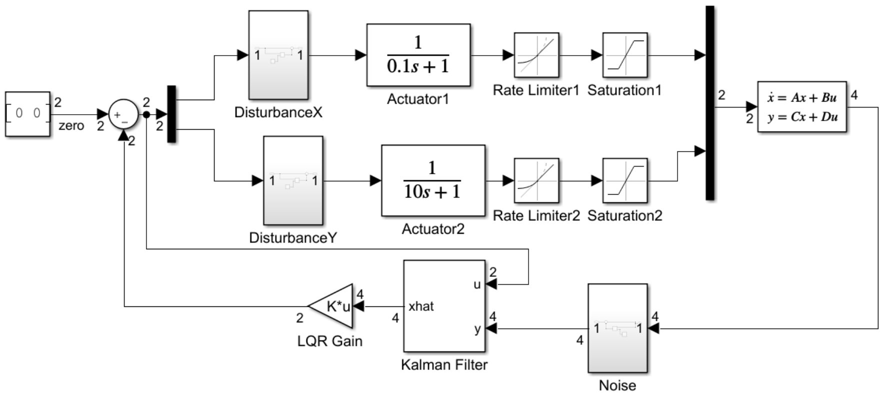

Figure 2: Block Diagram of LQR controller in Simulink

2.1 Derivation of each controller

Even in the faster development environment of Simulink, the PID controller took way more time to tune to just be stable. LQR the other hand, it is stable on the first pick of identity matrix for the Q matrix. Therefore, my approach with the PID was to feedback all the states to each actuator and apply a PID controller to each of the states Figure 3 and with gains that would provide a similar ratio of actuation per state error as the K matrix did in the LQR controller. The contents of the PID block seen in Figure 1 are shown in Figure 3. The four predicted state errors are feed through the PID which was later tuned to achieve appropriate performance. The anti-wind up effect was included with the Simulink PID built-in option to include back propagation anti-wind up.

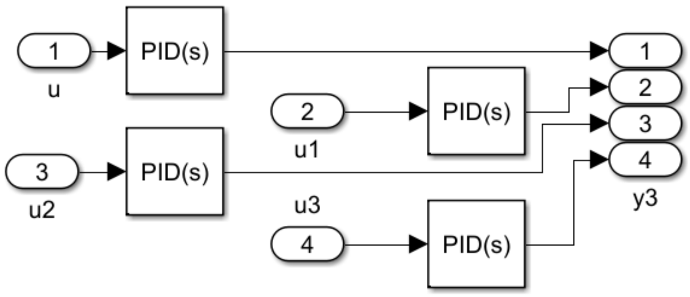

Figure 3: Implementation of PID for each actuator

2.2 Gain/phase margins

GM and PM for the LQR are zero and 90 deg as well as for PID because in Figure 4 there is no crossover point for 180deg in the phase diagram. The phase margin can be seen with the blue dot, since it is the point the gain diagram crosses 0dB.

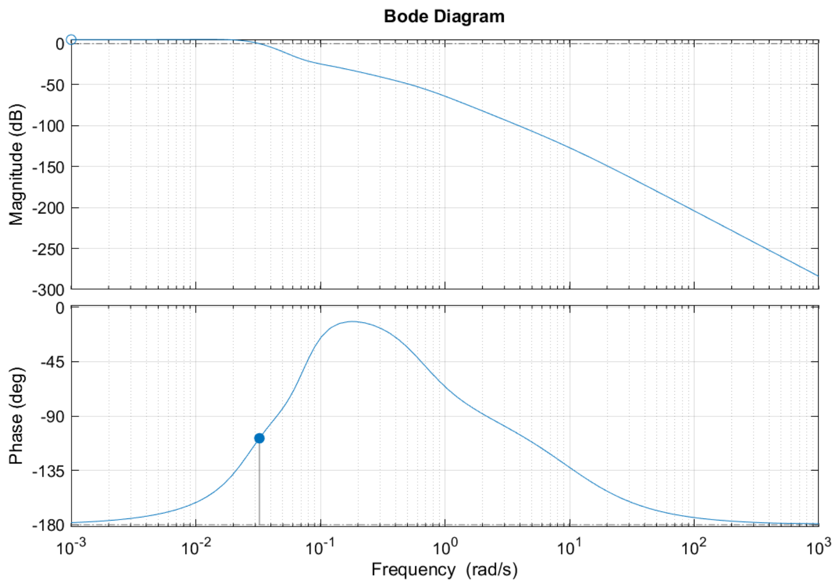

Figure 4: Bode plot of the PID controller in the x state (radial separation)

2.3 Effect of Noise

Noise was modeled as a random number with uniform distribution and disturbance was added as a sinusoid, both bounded by the 0.08% and 0.03% respectively and added to the signal. At first, I wanted to skip the observer and get my missing states by integrating and differentiating the ones that I could measure, which worked for the perfect system. However, when I included noise and disturbances, I was not able to integrate the y y signal to get clean readings, even with the addition of a low pass filter. When the estimator was used, there was not a noticeable difference between including and not including noise and even with high noise, which can be seen in Figure 5 where the state prediction with 20% noise in the sensor outputs is compared to the state prediction with only the 8% sensor noise instructed in the problem. The reason for this is because the Kalman Filter takes into account the covariance of the sensor noises and account for them based also on the covariance of the state prediction.

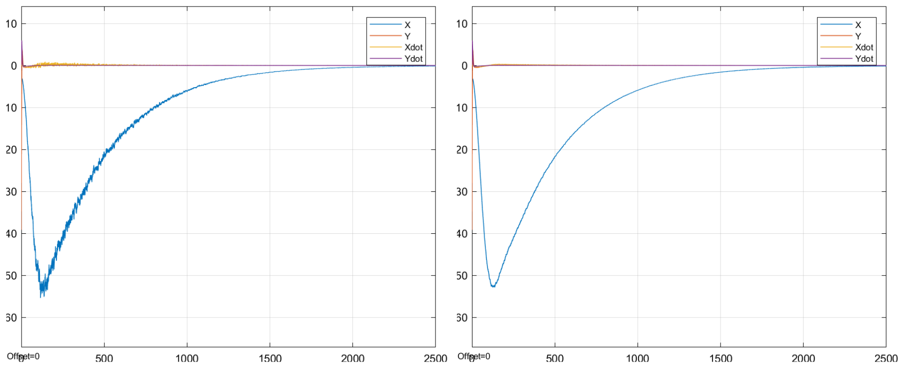

Figure 5: LQR response with noise (left) and without noise in the (right)

2.4 Position/rate limits on the actuators

Without actuators, as well as the position and rate limiter, the controller achieved the regulation faster(from settling time of 1000s to more than 2000s in Figure 6). However, besides that increase in performance, there was not significant stability effect. The benefit of including the limiters is that for some reason there is a noise that creates a large error in real hardware and the command is really high or changing too quickly, the actuator will not respond to those.

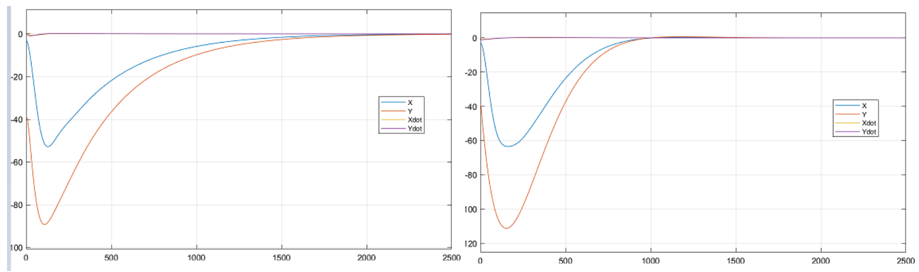

Figure 6: Higher rise time (left) for LQR controllers with Actuator dynamics and limiters

2.5 Design of Observer/estimator

As mentioned before, the estimator was chosen to be a Kalman Filter because the block diagram could be easily included in Simulink. In Figure 7, the left plot is the prediction from the Kalman Filter, and on the right is the actual states during the same simulation for the LQR controller (in Figure 8 is the same plot but zoomed in to $\dot{x}$ and $\dot{y}$). The same behavior can be seen in the response for the PID controller (Figure 9) As can be seen the x position is the same, but the estimator thinks that the y position goes to zero almost soon after the simulation starts. This could be due to the way that the Kalman filter combines the measured state to what the model outputs depending on their covariance, so if at the beginning it is just getting zeros of y and $\dot{x}$ then the prediction is closer to zero than reality which could then the dynamics could push the other states.

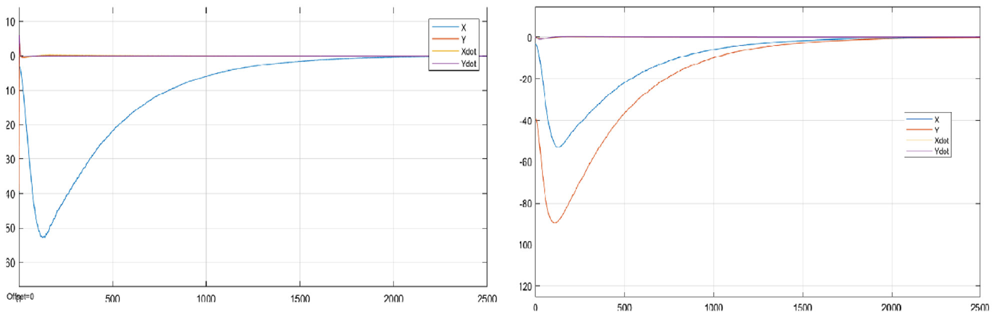

Figure 7: Kalman Filter Prediction response and Actual Response with LQR controller

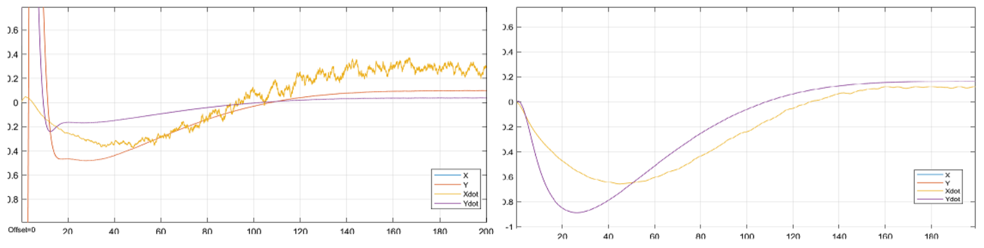

Figure 8: Kalman Filter Prediction of $\dot{x}$ and $\dot{y}$ (left), Actual Response (right)

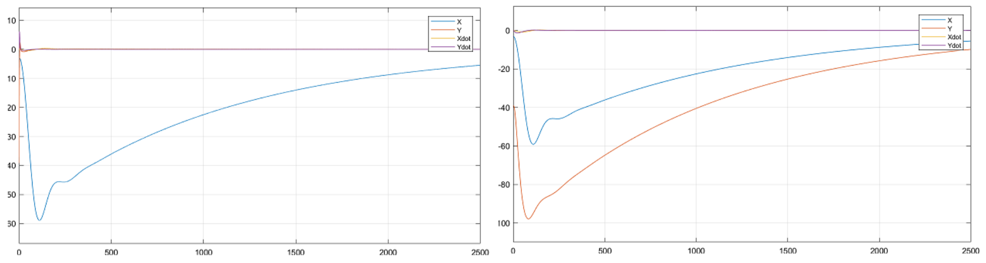

Figure 9: Kalman Filter Prediction response and Actual Response with PID controller

2.6 Sensitivity

The Figure 10shows the sensitivity of the radial separation, which shows the desired behavior of low sensitivity at low frequencies for good tracking (and good disturbance rejection) and low complementary sensitivity at high frequencies for noise attenuation.

$$Sensitivity = \frac{1}{1 + PK}$$

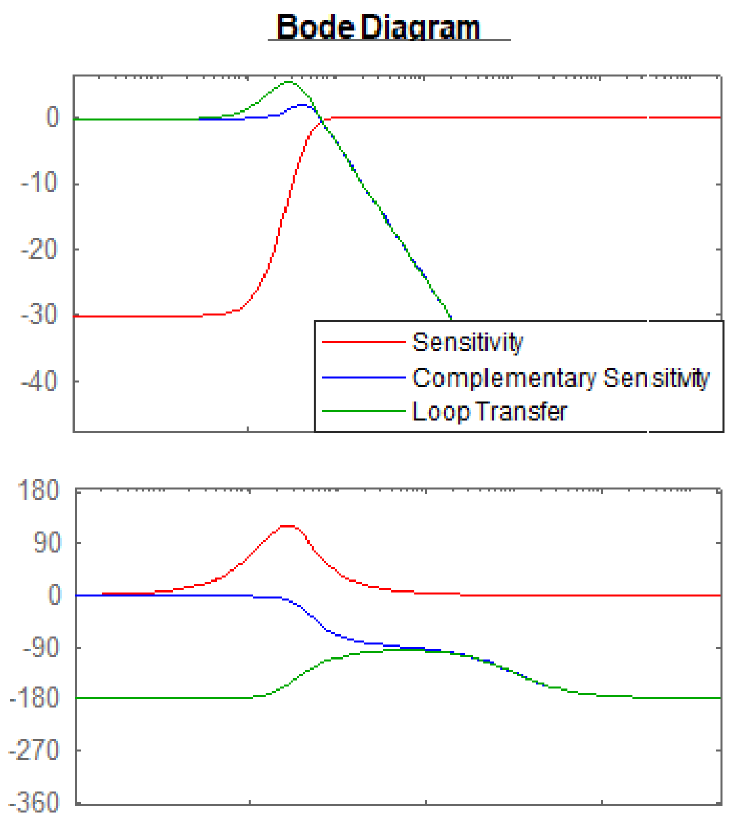

Figure 10: Sensitivity of the x state (radial separation)

2.7 Trade off between total actuation and response characteristics

The following Figure 11 and Figure 12 show the 2-norm of the actuation over a simulation run for the LQR and PID controller respectively. It can be seen that the PID uses more actuation than the LQR, but it is not a significant amount, it just increases the settling time because it is using less thrust to move the spacecrafts closer.

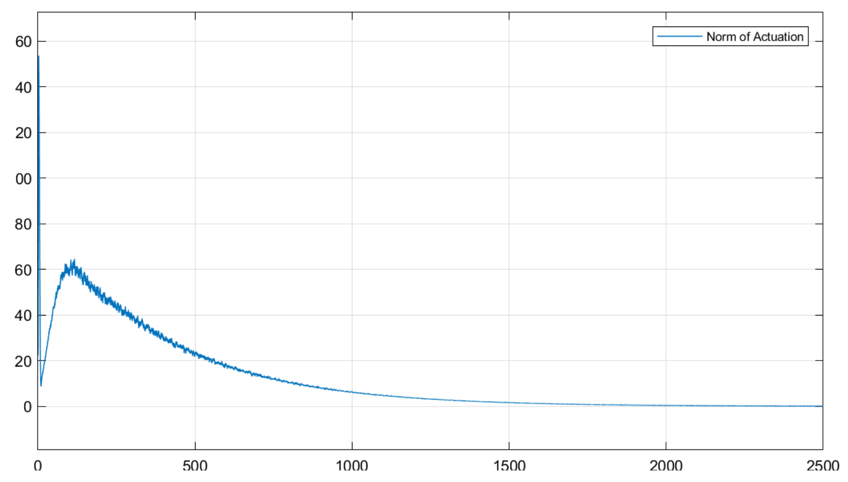

Figure 11: 2-Norm of Actuator response for LQR controller

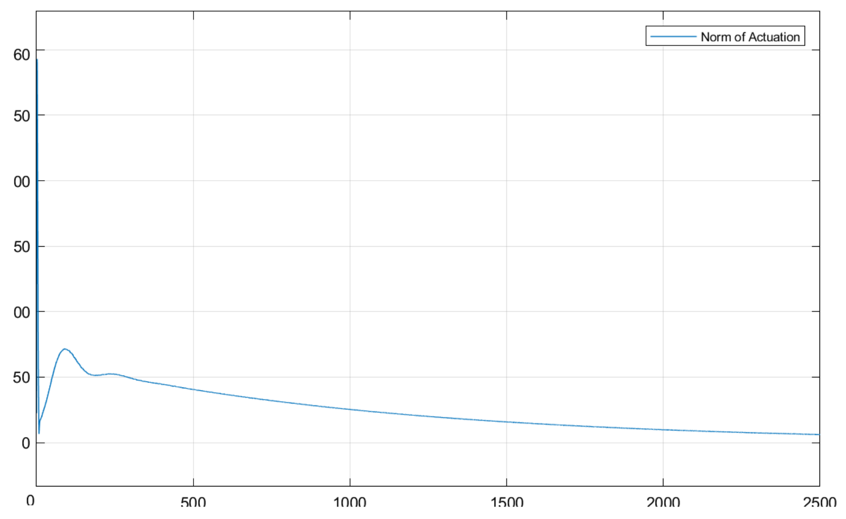

Figure 12: 2-Norm of Actuator response for PID controller