Robust Optimal Autopilot Design for Hypersonic Reentry Thrust Vector Control of Rockets

MAE 273A SISO Optimal Robust Control Final Project

Abhinav Kamath, Trevor Vidano

1 Abstract

Autonomous precision landing of first-stage rocket boosters is one of the most challenging control problems of this decade [1]. The development of this technology is enabling re-usability of rockets and thus lowering the cost of launch by a drastic margin. Also, this technology is key to achieving trans-planetary travel. A major component of this process is hypersonic reentry.

In this project, we design controllers to control the output angle of attack of a rocket during descent using gimbaled engines, with the angle of thrust with respect to the body of the rocket as the input.

In order to achieve this, we model the hypersonic reentry dynamics of a SpaceX Falcon 9 FT first-stage booster, taking into account a gimbaled Merlin 1D engine. We then linearize the plant using Taylor Series Expansion about the equilibrium point of the nonlinear plant, and arrive at the transfer function of the linearized plant.

We design three controllers for the same application, using different approaches: 'Robust Control Design Using Youla Parametrization' [2], 'Robust Optimal (H_∞) Control Design Using MATLAB', and 'PID Control Design Using Simulink'. We then compare the stability, robustness, and performance of these controllers by using them in conjunction with the derived true (nonlinear) model.

Contents

2 Background

2.1 Introduction

Figure 1: Artist's rendition of a Falcon 9 first-stage booster returning to Earth. *Credits: ZLSA Design*

The study of dynamics of a rocket begins with traditional aerodynamics. Similar to a fixed wing aircraft, a rocket experiences lift, drag and gravitational forces. SpaceX's Falcon 9 has been carefully designed so that its performance is highly controllable, enabling both optimized trajectory reentry and precision landing. This project addresses the problem of reentry control, as it is most easily reduced to a single input single output (SISO) controls problem.

The literature surrounding SISO problems is vast and focuses heavily on designing a PID controller. The method of this design is simple, but it does not provide the designer with information about robustness of the system. An alternate method, which is a Modern-control approach, the Youla Parametrization method, provides the designer with the insight to enable the design of robust, internally stable, and high-performing control systems. This method derives its name from the Youla parameter, which is defined as:

$$Y(s) = \frac{\hat{u}}{\hat{r}} = \frac{G_c}{1 + L(s)} \tag{1}$$

where $L(s)$ is the return ratio of the closed loop shown in figure(2). $\hat{u}$ is the controller output and $\hat{r}$ is the reference signal.

Additional transfer functions that are important in Youla Parametrization are the sensitivity transfer function, $S(s)$, and the complementary-sensitivity transfer function, $T(s)$. These are defined as:

$$S(s) = \frac{\hat{e}}{\hat{r}} = \frac{1}{1 + L(s)} \tag{2}$$

where $\hat{e}$ is the error signal, and

$$T(s) = \frac{\hat{y}}{\hat{r}} = \frac{L(s)}{1 + L(s)} \tag{3}$$

where $\hat{y}$ is the output signal.

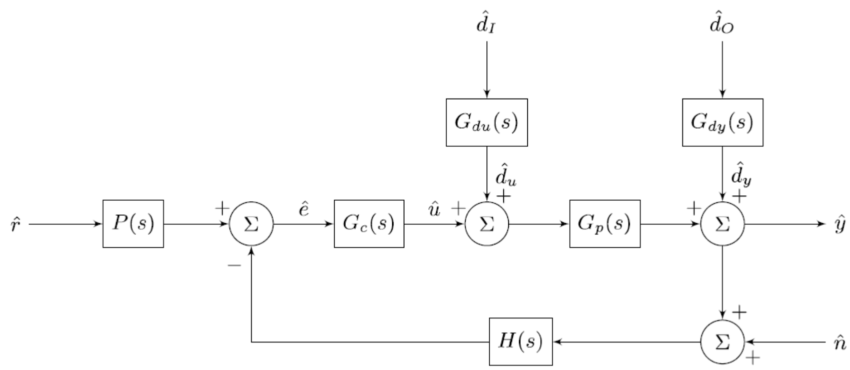

Figure 2: SISO Feedback Loop [2]

These three transfer functions allow the direct shaping of closed-loop responses to all the inputs shown in figure (2). The relationships are as follows:

$$\begin{bmatrix} \hat{e} \\ \hat{y} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} S(s) & -S(s) & -S(s) & -G_p S(s) \\ T(s) & -T(s) & S(s) & G_p S(s) \\ Y(s) & -Y(s) & -Y(s) & -T(s) \end{bmatrix} \begin{bmatrix} \hat{r} \\ \hat{n} \\ \hat{d}_y \\ \hat{d}_u \end{bmatrix} \tag{4}$$

where $\hat{d}_y$ is the output disturbance and $\hat{d}_u$ is the controller output disturbance.

From these relationships, important points with respect to the physical aspect of systems can be noted about the three transfer functions $S(s)$, $T(s)$, and $Y(s)$. $S(s)$ represents how sensitive the output of the feedback system is to output disturbances. $T(s)$ is the closed-loop transfer function and represents the behavior of the system to reference signals. $Y(s)$ represents the actuator output based on the reference signals.

These three transfer functions also provide information about internal stability. A system is internally stable if the following three conditions are met:

1. The Youla parameter $Y(s)$, is BIBO stable.
2. $G_p(s)Y(s)$, or $T(s)$, is BIBO stable.
3. $G_p(s)[1 - G_p(s)Y(s)]$, or $G_p(s)S(s)$, is BIBO stable.

An additional method, and the final one discussed in this project, is the H_∞ optimization. This is an optimization technique that utilizes MATLAB's Robust Control Toolbox, to design a controller based on minimizing the $\|H\|_\infty$. The method requires the design of filters to bound, weight and shape the transfer functions mentioned above. Three filters W_p, W_u and W_d are designed in this project, to achieve a robustly stable and nominally performant feedback control system, by shaping $S(s)$, $Y(s)$ and $T(s)$ respectively. These metrics are evaluated based on the following constraints:

$$\left\|W_p S(s)\right\|_\infty < 1 \Rightarrow \text{ Nominal Performance} \tag{5}$$

$$\left\|W_d T(s)\right\|_\infty < 1 \Rightarrow \text{ Robust Stability} \tag{6}$$

This report employs all three of these methods and compares them by applying the computed controllers to the non-linear dynamic equation for the Falcon 9.

2.2 Literature Review

Derivation of the rocket dynamics and modeling of the Falcon 9 are borrowed heavily from the 'Thrust Vector Control of a Hypersonic Reentry Vehicle' project undertaken at Stanford's Charm Lab [3].
In setting a constraint for actuator effort, we investigated off-the-shelf rocket-engine gimbal actuators which are used in rockets such as the Delta IV, NASA's Space Launch System (SLS) and previously the Space Shuttle, and based on their requirements, we restricted the actuator effort of our system to $300V$ or $49.54dB$ [4].

3 System Modeling

3.1 Hypersonic Reentry Dynamics

3.1.1 Plant Model

The descending rocket is modeled as a rigid cylindrical rod, rotating freely about its center of mass in response to torques generated by aerodynamic and thrust forces. The output parameter of interest is the angle of attack of the rocket, θ. The angle of the thrust vector with respect to the body of the rocket, α, will be taken as the plant input parameter and will be used to control the attitude of the rocket.

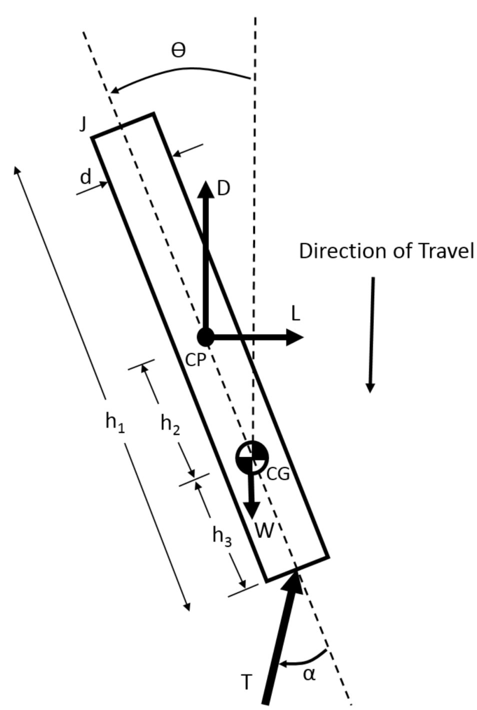

Figure 3: Free Body Diagram of The Rocket System

Figure (3) depicts the free body diagram of a Falcon 9 first-stage booster in descent.

3.1.2 Assumptions

- The vehicle operates at a high altitude in a free molecular flow field, such that the aerodynamic forces acting on the body can be readily determined.

- Control is for atmospheric reentry orientation (1-DOF) of a hypersonic vehicle with a high length to diameter ratio.
- The reentry burn is performed by a single gimbaled Merlin 1D engine, and the thrust remains constant (not impulsive).
- Vehicle velocity is constant.
- The plant mimics an inverted pendulum, in this case, with four relevant forces acting on the vehicle: gravity, thrust, lift, and drag.
- The lift and drag forces are estimated as those acting on a flat plate in free molecular flow at an angle of attack θ.
- Small-angle approximation is valid on the cosine terms.
- All powers of θ greater than 2 are negligible.

3.1.3 Derivation

Considering the summation of moments about the rocket's center of gravity:

$$\Sigma M_{CG} = J\ddot{\theta} = -h_2 \left(L\cos\theta + D\sin\theta\right) + h_3 T \sin\alpha$$

The forces are given by:

Lift:

$$L = L(\theta) = C_L(\theta) \times q \times A = C_L(\theta) \times \left(\frac{1}{2}\rho V^2\right) \times (h_1 d \sin\theta)$$

Drag:

$$D = D(\theta) = C_D(\theta) \times q \times A = C_D(\theta) \times \left(\frac{1}{2}\rho V^2\right) \times (h_1 d \sin\theta)$$

$$\text{where, } C_L(\theta) = \left[\sigma_n \frac{V_w}{V} + (2 - \sigma_n - \sigma_t) \times \sin\theta\right] \times \sin(2\theta),$$

$$C_D(\theta) = 2 \times \left[\sigma_t + \sigma_n \frac{V_w}{V} \times \sin\theta + (2 - \sigma_n - \sigma_t) \times \sin^2\theta\right] \times \sin(2\theta)$$

$$\text{and } V_w = \sqrt{\frac{\pi R_u T_w}{2 \times MW}}$$

Also, we let $\sigma_n = \sigma_t = \sigma,$ where $0 \leq \sigma \leq 1$ for specular and diffuse reflections, respectively.

Combining these results and applying small angle approximation on the cosine terms ($\cos\theta \approx 1$), we get

$$J\ddot{\theta}(t) = -h_1 h_2 d\sigma\rho V \sqrt{\frac{\pi R_u T_w}{2 \times MW}} \times \sin^2\big(\theta(t)\big) + h_3 T \times \alpha(t), \tag{7}$$

which will be considered to be the true model of the plant for the purpose of this project.

3.1.4 Parameters

Variable	Value	Description
h_1	$70\ m$	Length of the Falcon 9
h_2	$25\ m$	Distance between the center of gravity and the center of pressure
h_3	$10\ m$	Distance between the center of gravity and the engine
J	$37.576837 \times 10^6\ kgm^2$	Moment of inertia of the Falcon 9 about its vertical axis
T	$845.22\ kN$	Merlin 1D engine thrust
V	$890\ m/s$	Reentry velocity
d	$3.66\ m$	Airframe diameter
T_w	$300\ K$	Airframe wall temperature
σ	0.7	Free molecular reflection coefficient
ρ	$0.1\ kg/m^3$	Air density
R_u	$8.31\ Jmol^{-1}K^{-1}$	Universal gas constant
MW	$28.97\ g/mol$	Atmospheric molecular weight

Table 1: Parameters of interest in the derivation

3.2 Linearization

Based on the parameter values listed in Table 1, the true nonlinear equation (equation (7)) boils down to:

$$\ddot{\theta}\left(t\right) = -C_1 \sin^2\big(\theta\left(t\right)\big) + C_2\,\alpha\left(t\right), \tag{8}$$

$$\text{where, } C_1 = 0.1234648871 \ \&\ C_2 = 0.2249317105$$

In state-space form, with $x_1 = \theta,\ \dot{x_1} = \dot{\theta} = x_2,\ y = \theta$ and $u(t) = \alpha\left(t\right)$, we get:

$$\dot{x_1} = x_2$$

$$\dot{x_2} = -C_1 \sin^2\big(x_1\left(t\right)\big) + C_2\,\alpha\left(t\right)$$

$$y = x_1$$

The equilibrium point: $(x_1, x_2) = (0, 0) \ \mapsto\ (\dot{x_1}, \dot{x_2}) = (0, C_2\,\alpha\left(t\right))$

Performing Taylor Series Expansion about this equilibrium point, we get:

$$\begin{bmatrix}\dot{x_1}\\ \dot{x_2}\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ C_2\end{bmatrix}\alpha\left(t\right) \tag{9}$$

$$y = x_1$$

The state equations after linearization:

$$\dot{x_1} = x_2$$

$$\dot{x_2} = C_2\,\alpha\left(t\right)$$

$$y = x_1$$

Therefore, the linearized plant equation is:

$$\ddot{\theta}\left(t\right) = C_2\,\alpha\left(t\right) \tag{10}$$

Taking the Laplace Transform of equation (10) & setting the initial conditions to zero, we get:

$$s^2\,\Theta\left(s\right) = C_2\,A\left(s\right)$$

Therefore, the transfer function of the linearized plant model is:

$$\frac{\Theta\left(s\right)}{A\left(s\right)} = \frac{C_2}{s^2} \;\Rightarrow\; \frac{\Theta\left(s\right)}{A\left(s\right)} = \frac{0.2249317105}{s^2} \tag{11}$$

4 Control Design Methods

4.1 Robust Control Design Using Youla Parametrization

As the rocket system (plant) is unstable with a double-pole at the origin of the s-plane, we choose a Youla-parameter $Y(s)$ to cancel out the unstable poles. We make sure that the closed-loop step-response is similar to that of a second-order Butterworth filter (equation 12).

$$Y\left(s\right) = \frac{Ks^2\left(\tau_z s + 1\right)}{\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)\left(\tau_p s + 1\right)} \tag{12}$$

where,

$K \rightarrow$ gain, which will be chosen so as to satisfy the interpolation conditions (equation 15)
$\tau_z \rightarrow$ time-constant of the chosen zero
$\tau_p \rightarrow$ time-constant of the chosen pole
$\zeta \rightarrow$ damping ratio of the second order pole
$\omega_n \rightarrow$ natural frequency of the second order pole

The complementary-sensitivity transfer function *T(s)* is given by:

$$T\left(s\right) = \frac{Y\left(s\right)}{G_p\left(s\right)},\ \text{where } G_p\left(s\right) = \frac{0.2249317105}{s^2}\ \text{(from equation (11))}$$

$$\therefore\; T\left(s\right) = \frac{K \times 0.2249317105\left(\tau_z s + 1\right)}{\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)\left(\tau_p s + 1\right)} \tag{13}$$

To ensure that the system behaves like a second-order Butterworth filter, the additional pole should be 4 times away from the poles that govern the desired behavior. Therefore the following criterion should be met:

$$\tau_p < \frac{1}{4\zeta\omega_n} \tag{14}$$

The interpolation conditions to be satisfied in order to ensure internal stability of the closed-loop system in this case (unstable double-pole at s = 0) are as follows:

$$T\left(0\right) = 1\ \text{ and }\ \frac{\partial}{\partial s}T\left(0\right) = 0 \tag{15}$$

ω_n is chosen to be 0.85 *rad/s* to make sure that the system response is as fast as possible, while keeping the actuator effort in check. ζ is set to 0.707. τ_p is set to $1/10\omega_n$, i.e. 0.1176 *s*, to satisfy equation (14).

The remaining parameters K and τ_z are tuned to ensure that the interpolation conditions are met. Here, we end up with $K = 3.212$ and $\tau_z = 1.7814\ s$. The first interpolation condition also ensures accurate command tracking in this case.

The sensitivity transfer-function *S(s)* is given by:

$$S(s) = 1 - T(s) \tag{16}$$

The controller (autopilot) for this system is computed by:

$$G_c(s) = \frac{Y(s)}{S(s)}$$

$$\therefore\ G_c(s) = \frac{48.638\ (s + 0.5613)}{(s + 9.702)} \tag{17}$$

The Youla parameter $Y(s)$, the sensitivity transfer function $S(s)$, the complementary-sensitivity transfer function $T(s)$ and the product $G_p(s)S(s)$ are all BIBO stable. Thus, the conditions for internal stability are satisfied. The peak actuator effort is $48.6380dB$, which is less than the chosen constraint of $49.54dB$.

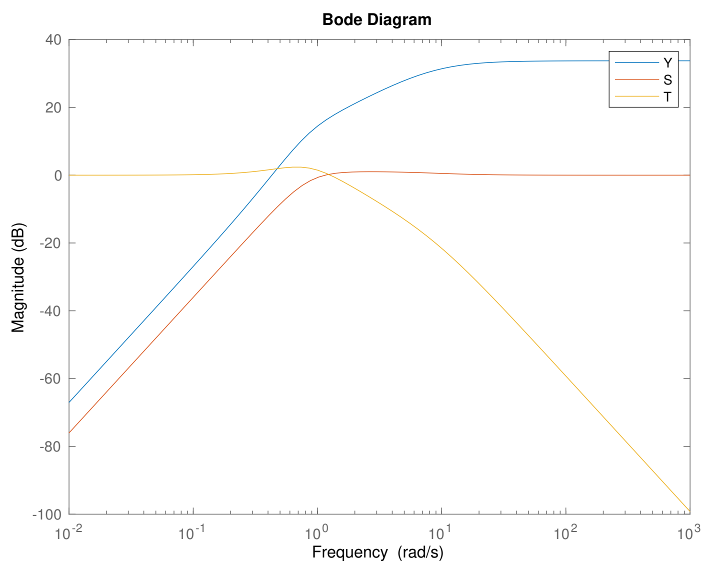

Figure 4: Bode plots of $Y(s)$, $T(s)$ and $S(s)$, obtained using Youla parametrization

4.2 Robust Optimal (H_∞) Control Design Using MATLAB

The MATLAB script for this design approach is in the appendix. We design weighting filters W_p, W_u and W_d to shape $S(s)$, $Y(s)$ and $T(s)$, respectively.

Figure (5) shows the bode diagram for the three important transfer functions. These plots were carefully shaped to meet the internal stability criteria. A key feature is that the plots of $S(s)$ and $T(s)$ cross each other below 0 dB, which ensures a phase margin greater than 60 degrees.

As shown in figure (6), the W_p filter was designed using a double pole and double zero, to achieve a steep slope and precise flattening out of $S(s)$. This ensures rejection of low frequency output disturbances and allows $T(s)$ to ensure good command following. Also with $S(s)$ high at high frequencies, noise rejection is

achieved. Next, the W_d filter, shown in figure (8), was designed to bound the complementary-sensitivity transfer function. This forced strong target following for a large bandwidth at low frequencies. It also rejects high frequency noise signals. Finally a simple gain was chosen for the W_u filter to reduce the actuator effort at all frequencies. The peak actuator effort in this case is $49.4262dB$, which is also less than the constraint of $49.54dB$.
The controller designed using this method is as follows:

$$G_c = \frac{7580.8\ (s+0.6341)\,(s+0.08)^2}{(s+149)\,(s+6.563)\,(s+0.01)^2} \tag{18}$$

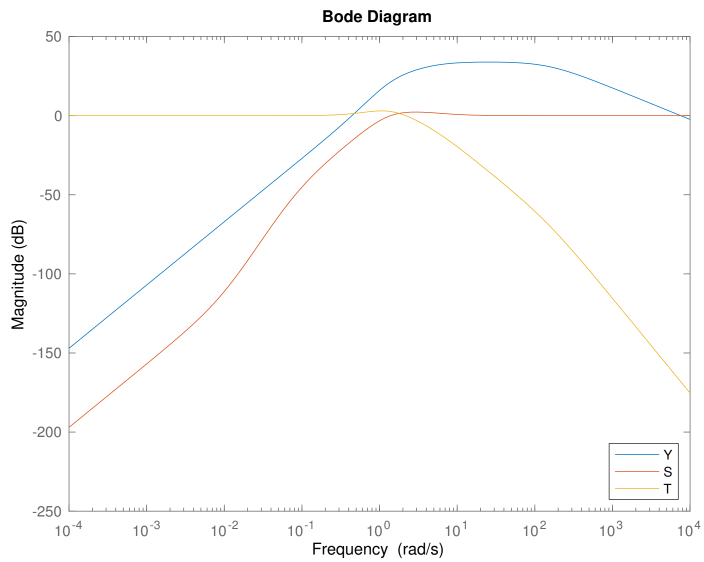

Figure 5: Bode plots of $Y(s)$, $T(s)$ and $S(s)$, obtained using H_∞ design

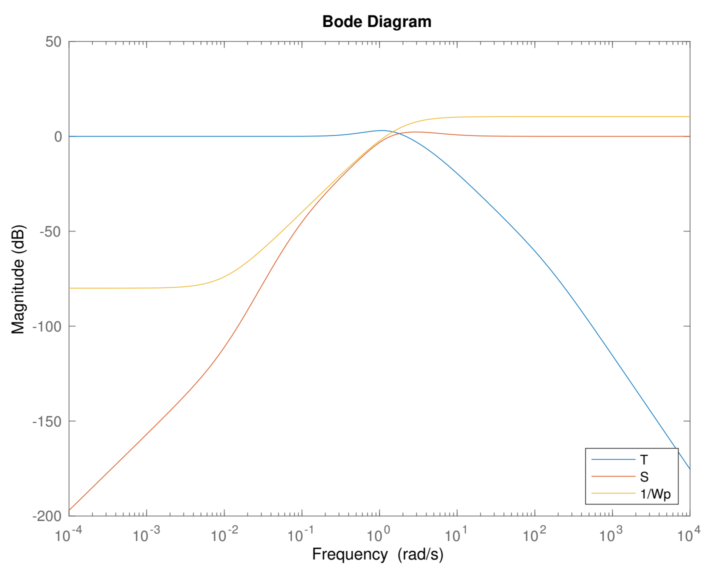

Figure 6: Bode plot of the inverse of the weighting filter W_p designed to bound *S(s)*

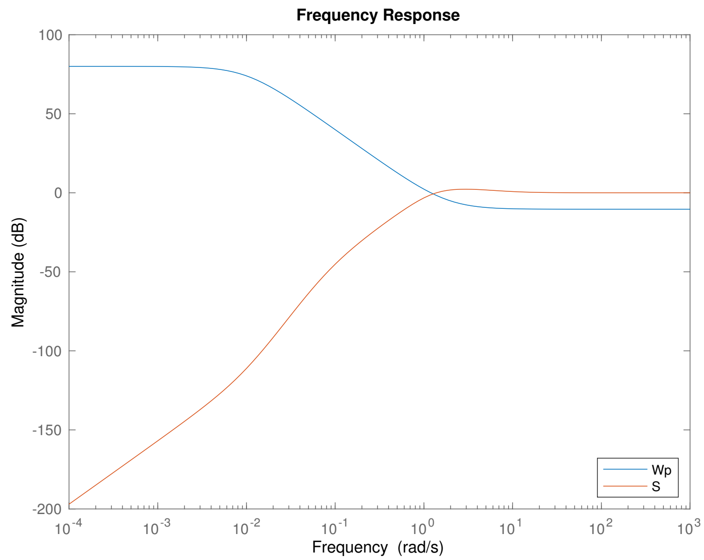

Figure 7: Bode plot of the weighting filter W_p designed to shape $S(s)$

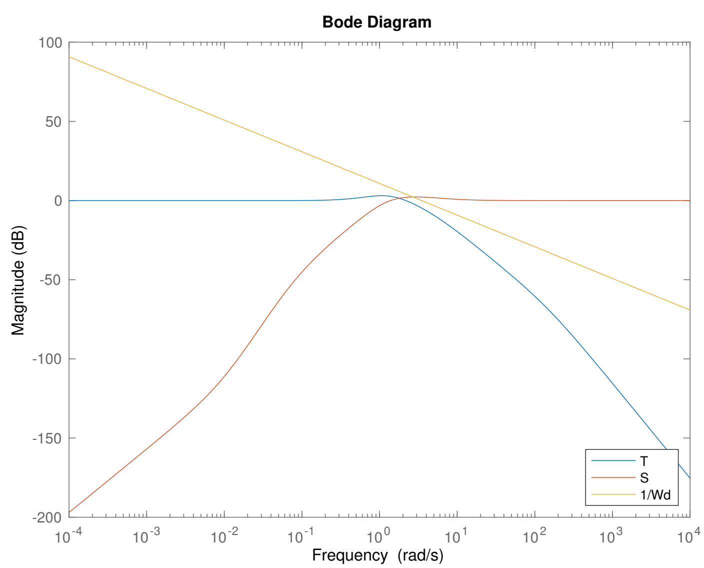

Figure 8: Bode plot of the inverse of the weighting filter W_d designed to bound $T(s)$

4.3 PID Control Design Using Simulink

Simulink simplifies the design of PID controllers. However, this triviality comes at the cost of significantly increasing actuator effort. This is shown by the large slope in Y(s) as frequency decreases. The shapes of the $T(s)$ and $S(s)$ plots suggest that the closed-loop system will be insensitive to disturbances and noise, while maintaining reference tracking. However, $Y(s)$ is unstable, thus undermining the internal stability of the feedback system.

The computed controller is:

$$G_c = \frac{1.7872\ (s^2 + 0.1109s + 0.004803)}{s\ (s + 0.1834)} \tag{19}$$

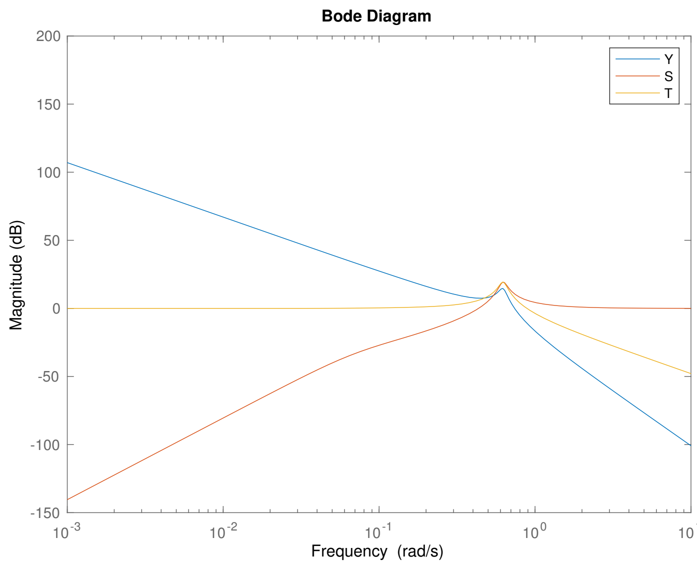

Figure 9: Bode plots of $Y(s)$, $T(s)$ and $S(s)$, obtained using PID design

5 Results

5.1 Comparisons

In addition to comparing the bode plots for the three important transfer functions for each controller design method, the step responses of the nonlinear true model show differences in performance of the different controllers. While the H_∞ designed controller achieves steady state the quickest, it also has the highest overshoot. However, this is acceptable, as fast response is desirable in this case. The PID controller reaches steady state the slowest, but has the smallest overshoot. The Youla controller computed "by hand" results in a reasonably good response as well.

Controller	M_2	Bandwidth
Youla	0.8918	1.8177
H_∞	0.7698	2.9813
PID	0.1076	0.9761

Table 2: Stability measurements for each controller design method

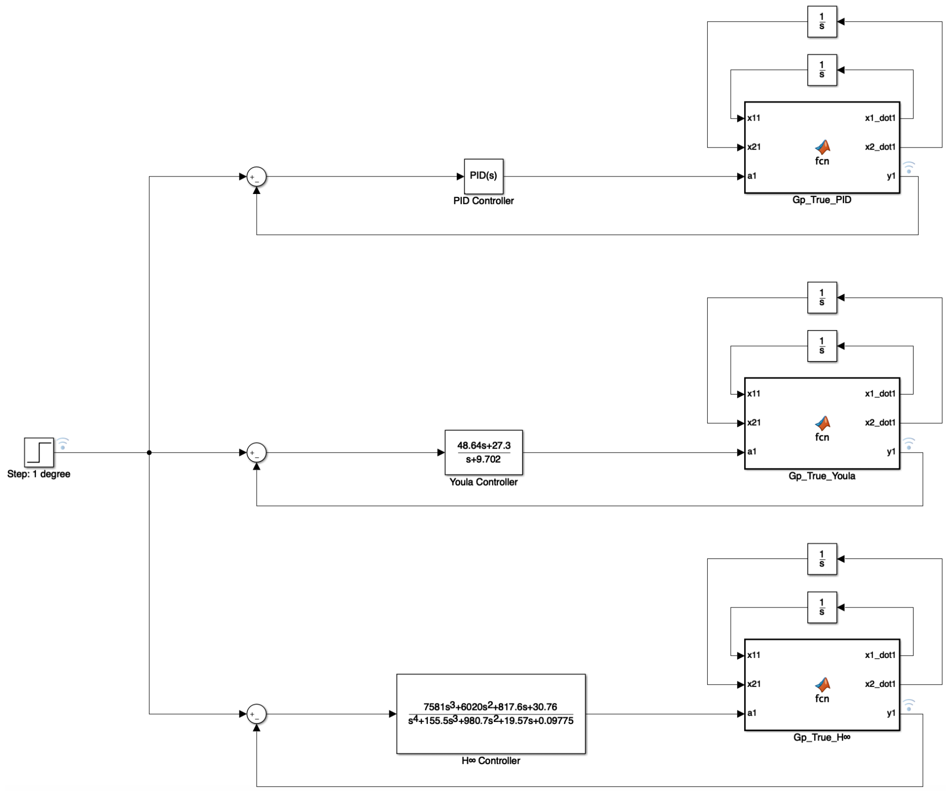

Figure 10: Simulink block diagram to test the designed controllers with the true (nonlinear) model of the rocket system

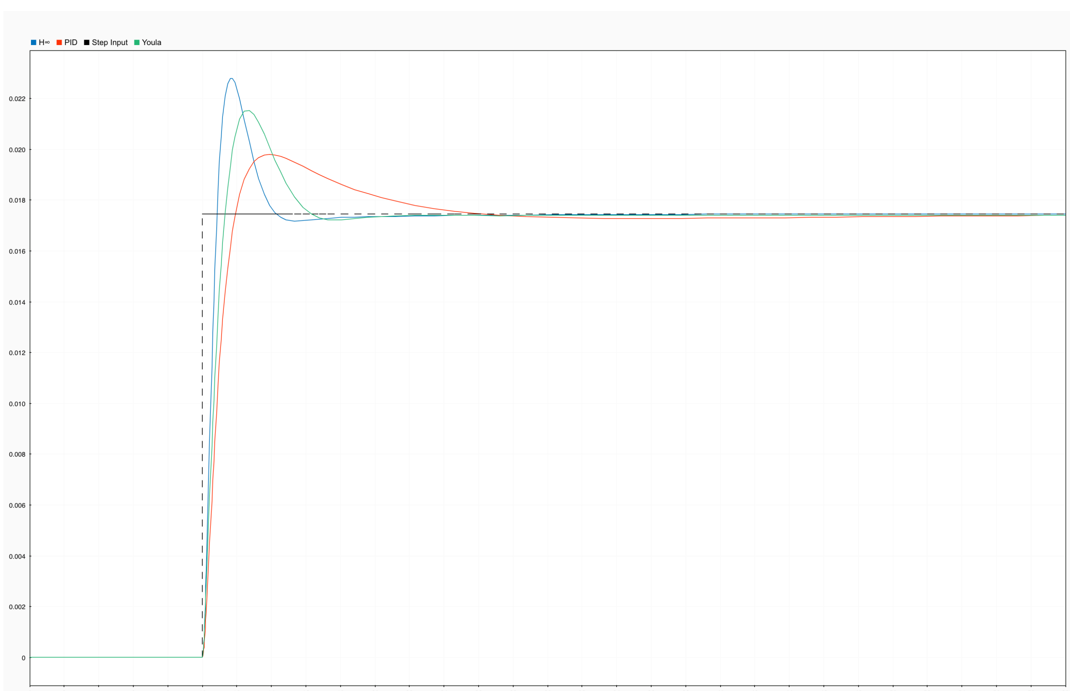

Figure 11: The responses of the true (nonlinear) model of the plant in conjunction with the different designed controllers to a step (angle) input of 1 degree

5.2 Conclusions

Based on the bode plots, the PID system fails to stay within the actuator effort limitation. The hand-calculated Youla parameter has a smaller bandwidth and a more gradual slope. The step responses show the trade offs of performance in the time domain. Based on actuator effort, stability margins, and the confidence brought about by satisfying the robust stability and nominal performance criteria, the optimal Youla control design method is the most preferable.

5.3 Future Work

Plans for the future involve modeling the rocket system to account for bending modes, and designing robust optimal controllers around the complex augmented model. Also, being constrained by SISO control limited the project to hypersonic reentry thrust vector control. Knowledge of MIMO control would aid in solving the problem of autonomous precision landing as well.

6 References

References

[1] L. Blackmore, "Autonomous precision landing of space rockets," vol. 46, pp. 15–20, 01 2016.

[2] F. Assadian and K. Mallon, "Robust control: Youla parameterization approach," 2020.

[3] C. Cox and Z. Clausing, "Thrust vector control of a hypersonic reentry vehicle," in *http://charm.stanford.edu/ENGR1052016/CharlesCoxZacClausing*.

[4] "Thrust vector control systems - moog, inc." in *https://www.moog.com/products/actuation-systems/space/tvc-systems.html*.

7 Appendix

7.1 Code

Youla Control Design

```
s = tf('s');

% Constants & Design Parameters

h3 = 10; % Distance between the Center of Gravity of the Rocket & the Gimbaled Merlin 1D Engine in consideration (meters)
T = 845.22 * 10^3; % Thrust of a Falcon 9 FT Stage 1 Merlin 1D Engine (Newtons)
J = 37576837; % Moment of Inertia of the Falcon 9 about the vertical axis (Assumption: Cylindrical Body)
C = (h3*T)/J; % Constant
Wn = 0.85; % Natural Frequency of the Control System
K = (Wn^2)/C; % Controller Gain
Z = 2^-0.5; % Damping Ratio ?
tp = 1/(10*Wn); % Time Constant of the added pole
tz = (20*2^(1/2))/17 + 2/17; % Time Constant of the added pole (to satisfy the 2nd interpolation condition (shown later))

% Plant TF, 'Gp'
Gp = zpk(minreal(C/s^2))

% Chosen Youla Parameter, 'Y' -> Y(0) = 0
Y = zpk(minreal(((K*s^2)*(tz*s + 1)/((s^2 + 2*Z*Wn*s + Wn^2)*(tp*s + 1))),1e-05))

% Complementary Sensitivity TF, 'T' -> T(0) = 1 (1st interpolation
% condition)
T = zpk(minreal((Y*Gp),1e-05))

% Sensitivity TF, 'S'
S = zpk(minreal((1-T),1e-05))

% Controller TF, 'Gc'
Gc = zpk(minreal((Y/S),1e-05))

% Return Ratio, 'L'
L = zpk(minreal((Gc*Gp),1e-05))

GpS = zpk(minreal((Gp*S),1e-05))

% Internal stability check
Y_stability = isstable(Y)
T_stability = isstable(T)
S_stability = isstable(S)
GpS_stability = isstable(GpS)

M2 = 1/getPeakGain(S) % M2-margin
BW = bandwidth(T) % Bandwidth of the closed-loop
AE = getPeakGain(Y) % Maximum actuator effort

figure(1)
bodemag(Y, S, T);
legend('Y','S','T');
```

```
Gp =

  0.22493
  -------
    s^2

Continuous-time zero/pole/gain model.


Y =

       48.638 s^2 (s+0.5613)
  ------------------------------
  (s+8.5) (s^2 + 1.202s + 0.7225)

Continuous-time zero/pole/gain model.


T =

         10.94 (s+0.5613)
  ------------------------------
  (s+8.5) (s^2 + 1.202s + 0.7225)

Continuous-time zero/pole/gain model.
```

```
S =

  (s-2.598e-08) (s+2.598e-08) (s+9.702)
  -------------------------------------
     (s+8.5) (s^2 + 1.202s + 0.7225)

Continuous-time zero/pole/gain model.


Gc =

  48.638 (s+0.5613)
  -----------------
      (s+9.702)

Continuous-time zero/pole/gain model.


L =

  10.94 (s+0.5613)
  ----------------
   s^2 (s+9.702)

Continuous-time zero/pole/gain model.


GpS =

         0.22493 (s+9.702)
  -------------------------------
  (s+8.5) (s^2 + 1.202s + 0.7225)

Continuous-time zero/pole/gain model.

Y_stability =
  logical
   1
T_stability =
  logical
   1
S_stability =
  logical
   1
GpS_stability =
  logical
   1
M2 =
    0.8918
BW =
    1.8177
AE =
   48.6380
```

Run this section first to calculate 'tz' to ensure that the second interpolation condition is satisfied

```
% d^k(T)/ds^k|(s=0) = 0, where k = 1 (since there is a double unstable pole
% (multiplicity ap = 2) in the plant at s = 0; k = ap - 1) -> 2nd
% interpolation condition

h3 = 10; % Distance between the Center of Gravity of the Rocket & the Gimbaled Merlin 1D Engine in consideration (meters)
T = 845.22 * 10^3; % Thrust of a Falcon 9 FT Stage 1 Merlin 1D Engine (Newtons)
J = 37576837; % Moment of Inertia of the Falcon 9 about the vertical axis (Assumption: Cylindrical Body)
C = (h3*T)/J; % Constant
Wn = 0.85; % Natural Frequency of the Control System
K = Wn^2/C; % Controller Gain
Z = 2^-0.5; % Damping Ratio ?
tp = 1/(10*Wn); % Time Constant of the added pole
syms s tz
TF = ((K*C)*(tz*s + 1))/((s^2 + 2*Z*Wn*s + Wn^2)*(tp*s + 1))
dTF = diff(TF,s)
eqn = subs(dTF,s,0) == 0;
tz = solve(eqn,tz)
```

H_∞ Control Design

```
% Courtesy of Professor Francis Assadian & TA Kevin Mallon

s=tf('s');

Gp = 0.224931119135972/s^2;

sysg=ss(Gp);
[Ag,Bg,Cg,Dg]=ssdata(sysg);

Ag=Ag-0.08*eye(2); % slightly shift A to avoid poles on jw axis

[num,den]=ss2tf(Ag,Bg,Cg,Dg);
Gpn=tf(num,den); % perturbed plant

%Hinf shaping filter
Wp = (db2mag(80)*((1/1.8177)*s + 1)^2)/((100)*s + 1)^2;
Wu = 0.01675;
Wd = s/3.469;

%Hinf Controller Computation
ssga_=augtf(Gpn,Wp,Wu,Wd);
[sys3,sscl]=hinfsyn(ssga_);

% Hinf Controller
Gc=zpk(minreal(tf(sys3)))

% Results
L=zpk(minreal((Gc*Gp),1e-05))
Y=zpk(minreal((Gc/(1+Gc*Gp)),1e-05))
T=zpk(minreal((Y*Gp),1e-05))
S=zpk(minreal((1-T),1e-05))
GpS=zpk(minreal((Gp*S),1e-05))

% Internal stability check
Y_stability = isstable(Y)
T_stability = isstable(T)
S_stability = isstable(S)
GpS_stability = isstable(GpS)

M2 = 1/getPeakGain(S) % M2-margin
BW = bandwidth(T) % Bandwidth of the closed-loop
AE = getPeakGain(Y) % Maximum actuator effort

NP = getPeakGain(Wp*S) % Nominal Performance
RS = getPeakGain(Wd*T) % Robust Stability
RP = getPeakGain((Wp*S) + (Wd*T)) % Robust Performance

figure(1)
bodemag(Y,S,T)
legend('Y','S','T','location','southeast');

figure(2)
bodemag(T,S,1/Wp)
legend('T','S','1/Wp','location','southeast')

figure(3)
bodemag(Wp,S)
title('Frequency Response')
legend('Wp','S','location','southeast')

figure(4)
bodemag(T,S,1/Wd)
legend('T','S','1/Wd','location','southeast')
```

```
Gc =

  7580.8 (s+0.6341) (s+0.08)^2
  ----------------------------
  (s+149) (s+6.563) (s+0.01)^2

Continuous-time zero/pole/gain model.


L =
```

```
    1705.2 (s+0.6341) (s+0.08)^2
  --------------------------------
  s^2 (s+149) (s+6.563) (s+0.01)^2

Continuous-time zero/pole/gain model.


Y =

                   7580.8 s^2 (s+0.6341) (s+0.08)^2
  -------------------------------------------------------------------
  (s+149) (s+4.302) (s^2 + 0.1583s + 0.006295) (s^2 + 2.042s + 1.714)

Continuous-time zero/pole/gain model.


T =

                     1705.2 (s+0.6341) (s+0.08)^2
  -------------------------------------------------------------------
  (s+149) (s+4.302) (s^2 + 0.1583s + 0.006295) (s^2 + 2.042s + 1.714)

Continuous-time zero/pole/gain model.


S =

  (s+6.563) (s+149) (s^2 + 0.02s + 0.0001) (s^2 + 7.483e-10s + 5.353e-14)
  -----------------------------------------------------------------------
    (s+149) (s+4.302) (s^2 + 0.1583s + 0.006295) (s^2 + 2.042s + 1.714)

Continuous-time zero/pole/gain model.


GpS =

          0.22493 (s+6.563) (s+149) (s^2 + 0.02s + 0.0001)
  -------------------------------------------------------------------
  (s+149) (s+4.302) (s^2 + 0.1583s + 0.006295) (s^2 + 2.042s + 1.714)

Continuous-time zero/pole/gain model.

Y_stability =
  logical
   1
T_stability =
  logical
   1
S_stability =
  logical
   1
GpS_stability =
  logical
   1
M2 =
    0.7698
BW =
    2.9813
AE =
   49.4262
NP =
    0.9081
RS =
    0.6274
RP =
    1.2105
```

PID Control Design

```
% Constants & Design Parameters

h3 = 10; % Distance between the Center of Gravity of the Rocket & the Gimbaled Merlin 1D Engine in consideration (meters)
T = 845.22 * 10^3; % Thrust of a Falcon 9 FT Stage 1 Merlin 1D Engine (Newtons)
J = 37576837; % Moment of Inertia of the Falcon 9 about the vertical axis (Assumption: Cylindrical Body)
C = (h3*T)/J; % Constant

s = tf('s');

% Plant TF, 'Gp'
Gp = zpk(minreal(C/s^2))

% These values were obtained from the auto-tuning the PID controller in Simulink
P = 0.825493884458449;
I = 0.0468147782797737;
D = 5.24517904929436;
N = 5.45397025421619;
Gc = zpk(minreal(pid(P,I,D,N),1e-05))

L = zpk(minreal((Gc*Gp),1e-05))
T = zpk(minreal((L/(1 + L)),1e-05))
S = zpk(minreal((1 - T),1e-05))
Y = zpk(minreal((T*Gp),1e-05))
GpS = zpk(minreal((Gp*S),1e-05))

% Internal stability check
Y_stability = isstable(Y)
T_stability = isstable(T)
S_stability = isstable(S)
GpS_stability = isstable(GpS)

M2 = 1/getPeakGain(S) % M2-margin
BW = bandwidth(T) % Bandwidth of the closed-loop
AE = getPeakGain(Y) % Maximum actuator effort

figure(1)
bodemag(Y, S, T);
legend('Y','S','T');
```

```
Gp =

  0.22493
  -------
    s^2

Continuous-time zero/pole/gain model.


Gc =

  1.7872 (s^2 + 0.1109s + 0.004803)
  ---------------------------------
            s (s+0.1834)

Continuous-time zero/pole/gain model.


L =

  0.402 (s^2 + 0.1109s + 0.004803)
  --------------------------------
          s^3 (s+0.1834)

Continuous-time zero/pole/gain model.


T =

            0.402 (s^2 + 0.1109s + 0.004803)
  ---------------------------------------------------
  (s^2 + 0.1137s + 0.004962) (s^2 + 0.06969s + 0.3891)

Continuous-time zero/pole/gain model.
```

```
S =

  (s-1.191e-06) (s+0.1834) (s^2 + 1.191e-06s + 1.42e-12)
  -----------------------------------------------------
   (s^2 + 0.1137s + 0.004962) (s^2 + 0.06969s + 0.3891)

Continuous-time zero/pole/gain model.


Y =

            0.090422 (s^2 + 0.1109s + 0.004803)
  -------------------------------------------------------
  s^2 (s^2 + 0.1137s + 0.004962) (s^2 + 0.06969s + 0.3891)

Continuous-time zero/pole/gain model.


GpS =

            0.22493 (s+0.1834) (s+3.972e-07)
  ----------------------------------------------------
  (s^2 + 0.1137s + 0.004962) (s^2 + 0.06969s + 0.3891)

Continuous-time zero/pole/gain model.

Y_stability =
  logical
   0
T_stability =
  logical
   1
S_stability =
  logical
   1
GpS_stability =
  logical
   1
M2 =
    0.1076
BW =
    0.9761
AE =
   Inf
```

True (Nonlinear) Model $\rightarrow$ *Simulink*

```
function [x1_dot1, x2_dot1, y1] = fcn(x11, x21, a1)
c1 = 0.1234648871;
c2 = 0.224931119135972;
x1_dot1 = x21;
x2_dot1 = -c1*(sin(x11))^2 + c2*a1;
y1 = x11;
```

All code used in this project can be found at `https://github.com/abhinav.gk/Rocket-Controls`.