Reducing Pilot Workload and Improving Handling Characteristics of the C-5A Galaxy

Blake Christierson
Thomas Greenhill

June 10, 2020

Abstract

The C-5A Galaxy is notorious for its high pilot workload, resulting from sub-par longitudinal and lateral stability and performance characteristics. These qualities, caused namely by the unstable Phugoid pitch mode and unstable spiral mode, lead to the plane to the lowest flight quality level for Category B flight phases. Youla and H_∞ Controllers are computed to attempt to reduce pilot workload by stabilizing ill-behaved modes, improve roll and yaw response and ultimately elevate the plane's flight quality level. Due to unstable hidden modes, the C-5A was found to be extremely difficult to control, and though some level of flight quality improvement was achieved, a plant re-design is fundamentally required to achieve a higher level of flight quality.

Table of Contents

Introduction & Literature Review

Aircraft Performance Evaluation

Aircraft are a particularly interesting class of plants due to their predisposition to display highly undesirable characteristics in instability, nonlinearity, and a large amount of parameter variation due to the wide range of operating conditions experienced during flight. To standardize the discussion and comparison of aircraft behavior standard MIL-F-8785C is applied as described by Section 7.4 of Yechout *et. al.* [1]. The process associated with this standard includes designating a flight phase category, class, and level to each aircraft to provide specific requirements on each of the following modes: short period, Phugoid, roll, spiral, and Dutch roll.

The aircraft in question for this report is the C-5A Galaxy which falls under Category B, Class III, and its airframe performance is relegated to Level 3 flight quality due to poor Phugoid and Dutch roll characteristics. These classifications are derived from its role as a heavy transport/cargo whose flight phases are nonterminal and typically completed gradually without the need for precise tracking [1]. The following table shows the requirements required to elevate the C-5A's flight quality level to improve mission effectiveness and reduce pilot workload.

Table 1 – Summary of MIL-F-8785C Handling Quality Level Requirements

Mode	Level 1	Level 2	Level 3
Short Period	$\zeta_{sp} \in [0.3,2]$	$\zeta_{sp} \in [0.2,2]$	$\zeta_{sp} > 0.15$
Phugoid	$\zeta_{ph} > 0.04$	$\zeta_{ph} > 0$	$T_{2,ph} \geq 55\ s$
Roll	$\tau_r < 1.4\ s$	$\tau_r < 3.0\ s$	$\tau_r < 10\ s$
Spiral	$T_{2,s} > 20\ s$	$T_{2,s} > 8\ s$	$T_{2,s} > 4\ s$
Dutch Roll	$\zeta_{dr} > 0.08$ $\zeta_{dr}\omega_{N,dr} > 0.15$ $\omega_{N,dr} > 0.4$	$\zeta_{dr} > 0.02$ $\zeta_{dr}\omega_{N,dr} > 0.05$ $\omega_{N,dr} > 0.4$	$\zeta_{dr} > 0$ $\omega_{N,dr} > 0.4$

The aircraft's properties are known under three operating conditions, of which the handling quality parameters are tabulated below. The first set of dynamic properties are measured under sea level conditions at an airspeed of 85.04 m/s. The second and third set are measured at an altitude of 6096 m and 220.98 m/s, and 12192 m and 206.65 m/s, respectively.

Table 2 – Summary of C-5A Stability Mode Airframe Baseline

Altitude (m)	Sea Level	6096	12192
Mach	0.25	0.7	0.7
True Airspeed (m/s)	85.04	220.98	206.65
Trim Angle of Attack	11.41 °	0.68 °	7.66 °
Short Period	$\zeta_{sp} = 0.628$	$\zeta_{sp} = 0.421$	$\zeta_{sp} = 0.299$
Phugoid	$\zeta_{ph} = -0.00245$ $T_{2,ph} = 7.802$	$\zeta_{ph} = 0.0620$	$\zeta_{ph} = 0.0656$
Roll	$\tau_r = 0.289\ s$	$\tau_r = 0.474\ s$	$\tau_r = 0.202\ s$
Spiral	$T_{2,s} = \infty\ s$	$T_{2,s} = \infty\ s$	$T_{2,s} = \infty\ s$
Dutch Roll	$\zeta_{dr} = 0.28$ $\zeta_{dr}\omega_{N,dr} = 0.13\ rad/s$ $\omega_{N,dr} = 0.45\ rad/s$	$\zeta_{dr} = 0.11$ $\zeta_{dr}\omega_{N,dr} = 0.071\ rad/s$ $\omega_{N,dr} = 0.644\ rad/s$	$\zeta_{dr} = 0.109$ $\zeta_{dr}\omega_{N,dr} = 0.051\ rad/s$ $\omega_{N,dr} = 0.465\ rad/s$

Due to the aircraft's negative Phugoid mode damping at slow speeds and low altitudes, the aircraft yields level 3 handling characteristics. MIL-F-8785C states that level 3 flying qualities *"are such that the airplane can be controlled safely, but pilot workload is excessive or mission effectiveness is inadequate, or both".* Even after addressing the aircraft's deficiencies in Phugoid mode stability, the aircraft' flight qualities still fall under level 2 due to poor Dutch roll qualities under all tested conditions and poor short period mode qualities at high altitudes and airspeeds.

In the best interest of the safety and feasibility of long missions, level 1 flight characteristics, under which *"flying qualities are clearly adequate for the mission flight phase",* are sought after.

Based on this analysis and with the goal of providing the aircraft with level 1 flying qualities, the aircraft's short period, Phugoid and Dutch roll modes must be improved.

More details on the description of these evaluations are available in the supplemental materials section.

Plant Characterization

We begin with the linearized form of the three longitudinal equations of motion and the three lateral equations of motion in Laplace from Equation 7.26 & 7.43 of Yechout *et. al.* [1]. See the derivative nomenclature table provided in the supplemental materials for descriptions of the various control derivatives.

Longitudinal Equations of Motion:

$$\dot{u} = -g\theta\cos(\Theta_1) + X_u u + X_{T_u} u + X_\alpha \alpha + X_{\delta_e}\delta_e \tag{1.1}$$

$$\dot{w} - U_1 q = -g\theta\sin(\Theta_1) + Z_u u + Z_\alpha \alpha + Z_{\dot{\alpha}}\dot{\alpha} + Z_q q + Z_{\delta_e}\delta_e \tag{1.2}$$

$$\dot{q} = M_u u + M_{T_u} u + M_\alpha \alpha + M_{T_\alpha}\alpha + M_{\dot{\alpha}}\dot{\alpha} + M_q q + M_{\delta_e}\delta_e \tag{1.3}$$

Lateral Equations of Motion:

$$\dot{v} + U_1 r = g\phi cos\Theta_1 + Y_\beta \beta + Y_p p + Y_r r + Y_{\delta_\alpha}\delta_\alpha + Y_{\delta_r}\delta_r \tag{2.1}$$

$$\dot{p} - \frac{I_{xz}}{I_{xx}}\dot{r} = L_\beta \beta + L_p p + L_r r + L_{\delta_a}\delta_\alpha + L_{\delta_r}\delta_r \tag{2.2}$$

$$\dot{r} - \frac{I_{xz}}{I_{zz}}\dot{p} = N_\beta \beta + N_{T_\beta}\beta + N_p p + N_r r + N_{\delta_\alpha}\delta_\alpha + N_{\delta_r}\delta_r \tag{2.3}$$

The full derivation of the state spaces used for plant characterization and controller design are also available in the supplemental materials section. The specific transfer function matrices are discussed further in the body due to various computational issues that arose.

The longitudinal equations of motion represent a 3x1 transfer function matrix whose sole control input is the angle of the elevator control surface, δ_e, and has the outputs of longitudinal velocity, u; vertical velocity, w; and pitch rate, q. The lateral equations of motions are a 3x2 transfer function matrix whose inputs include the angle of the aileron and rudder control surfaces, δ_a and δ_e, and has the outputs of lateral velocity, v; roll rate, p; and yaw rate, r. It should be noted that this way of modeling the plant removes any type of coupling between the longitudinal and lateral modes; however, the entries of the C-5A Galaxy's inertial tensor related to pitch coupling are a magnitude smaller than the other entries and therefore have secondary effects.

Concatenating these systems provides a 6x3 system in total, but for the purposes of the paper they will be treated separately to reduce computational complexity while formulating both longitudinal and lateral controllers. The longitudinal controller will have the sole task of stabilizing the Phugoid mode and improving short period response to increase handling quality level while the lateral controller will attempt to improve handling level quality while also attempting to decouple the roll and yaw behavior.

Step Responses of Uncontrolled Plant

Before designing controllers, the state-space representations are used to produce transfer function matrices for both longitudinal and lateral behavior. The uncontrolled output of the aircraft to step responses in elevator deflection, aileron deflection and rudder deflection are evaluated under all three operating conditions.

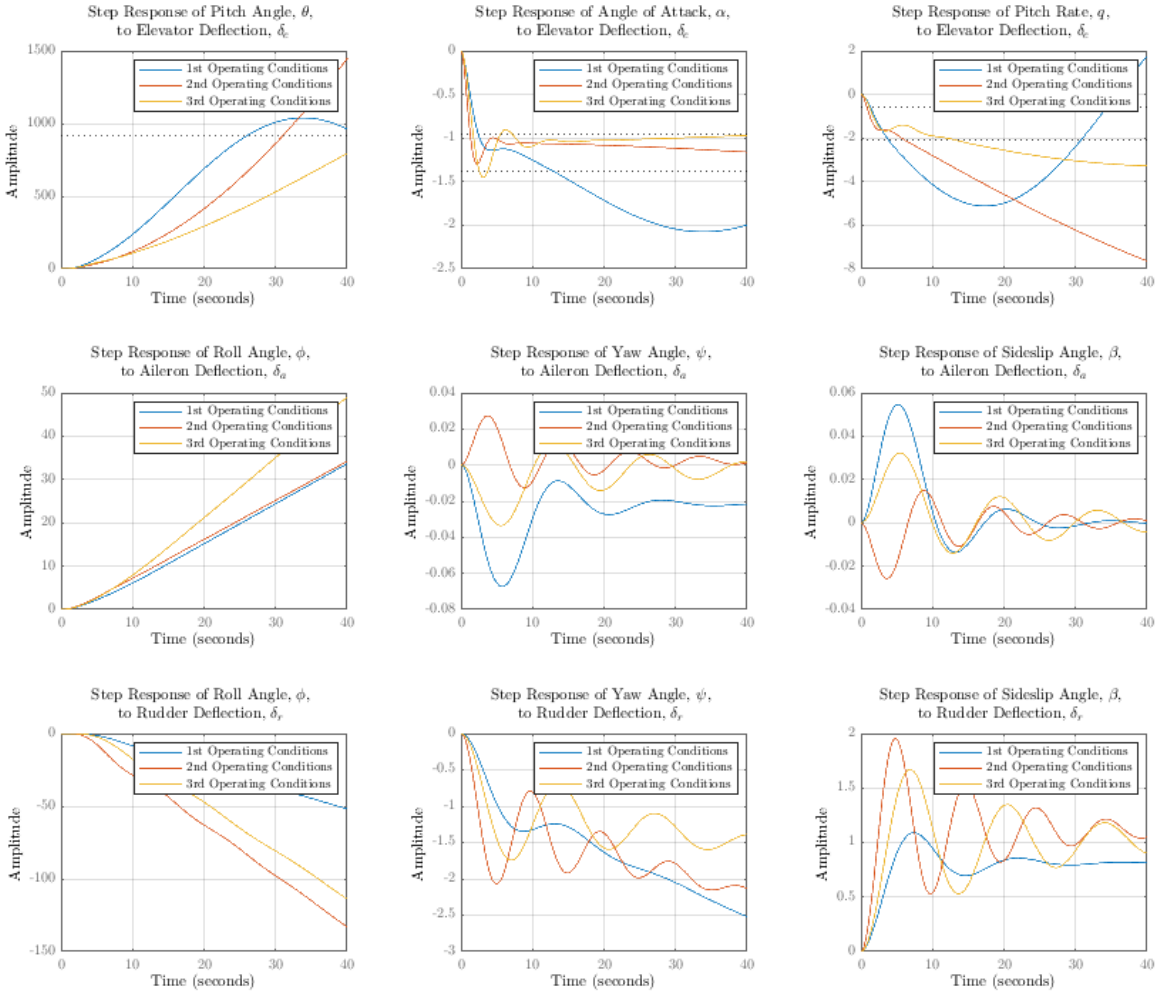

Figure 1 – Longitudinal and Lateral Responses of Uncontrolled Plant to Step Input

This figure showcases the aircraft's Phugoid mode instability and its undesirable lateral behavior due to lack of Dutch roll damping. Note that the aircraft's yaw and roll modes are weakly coupled, and that the aircraft's yaw mode is reversely coupled to the aircraft's roll mode. Further, the roll response to aileron is many orders of magnitude greater than the yaw angle and sideslip angle responses to the same input. This quickly poses difficulties in controller design due to extreme gains of certain terms.

Plant Difficulties

The C5 Galaxy's lack of Phugoid pitch stability and contrasting modal response times and amplitudes present unique difficulties for controller design. This subsection covers the difficulties encountered and the measures taken to improve our chances at successfully developing controllers.

Floating Point Error, Limitations of hermiteForm()

The C5's individual modes display an extreme contrast in response to control inputs in both lateral-directional behavior and longitudinal-directional behavior. When attempting to compute controllers with the help of MATLAB, this extreme contrast in behavior quickly proved it numerically impossible to compute the U_L and U_R matrices necessary for Youla controller design. To avoid this issue, a symbolic approach was taken and results with numeric gains for U_L were eventually produced using hermiteForm(). Unfortunately, hermiteForm()'s inability to intake symbolic gains forces us to substitute in the numerical values for the aircraft's control derivatives in U_L before computing U_R, but floating point errors persisted and caused computation of $U_R = \left(U_L * G_p\right) \backslash M_p$ to fail. Due to the size and nature of these matrices, hand computation is also unfeasible.

Lateral Hidden Modes

As our first attempts to hand compute Youla controllers for the lateral transfer function matrices immediately failed, troubleshooting led to a very uncomfortable discovery: The C-5 has many hidden modes, some of which are unstable. The lateral plant transfer function matrix for the sea level flight conditions is shown below along with a table and plots of its poles and zeros. A pole-zero plot for the lateral-directional transfer function matrix is available in the supplemental materials section.

$$G_{p,y}^{SL} = \begin{bmatrix} \dfrac{0.30571\,(s^2 + 0.196s + 0.1761)}{s\,(s+0.2885)(s^2 + 0.253s + 0.202)} & \dfrac{0.024366\,(s-2.952)(s+1.174)}{s\,(s+0.2885)(s^2 + 0.253s + 0.202)} \\ \dfrac{-0.016603\,(s+0.07599)}{(s+0.2885)(s^2 + 0.253s + 0.202)} & \dfrac{-0.14232\,(s+0.3233)(s+0.05706)}{s\,(s+0.2885)(s^2 + 0.253s + 0.202)} \\ \dfrac{0.016603\,s}{(s+0.2885)(s^2 + 0.253s + 0.202)} & \dfrac{0.016631\,(s+8.696)(s+0.3271)}{(s+0.2885)(s^2 + 0.253s + 0.202)} \end{bmatrix} \quad (3.1)$$

Poles	Zeros
0.0000 + 0.0000i	0.0000 + 0.0000i
-0.1265 + 0.4312i	-0.1265 + 0.4312i
-0.1265 - 0.4312i	-0.1265 - 0.4312i
-0.2885 + 0.0000i	-0.2885 + 0.0000i
-0.1265 + 0.4312i	-0.1265 + 0.4312i
-0.1265 - 0.4312i	-0.1265 + 0.4312i
-0.2885 + 0.0000i	-0.2885 - 0.0000i
0.0000 + 0.0000i	
-0.1265 + 0.4312i	-0.1265 + 0.4312i
-0.1265 - 0.4312i	-0.1265 - 0.4312i
-0.2885 + 0.0000i	-0.2885 + 0.0000i
-0.1265 + 0.4312i	-0.1265 + 0.4312i
-0.1265 - 0.4312i	-0.1265 - 0.4312i
-0.2885 + 0.0000i	
-0.1265 + 0.4312i	
-0.1265 - 0.4312i	
-0.2885 + 0.0000i	

Table 3 – Poles and Zeros of Lateral Transfer Function Matrix Under Sea Level Conditions

The C-5 has 17 poles and 9 zeros in its lateral transfer function matrix, with all zeros cancelling with sets of poles. Additionally, some of these cancellations occur along the *jw*-axis, which results in cancellation of BIBO unstable pole/zero pairs. Disappointingly, this results in an unobservable and/or uncontrollable lateral plant.

Longitudinal Hidden Modes

Since the lateral-directional plant is unobservable and/or uncontrollable, we are left to investigate the behavior of the longitudinal plant. The transfer function matrix of the longitudinal plant under the first operating conditions is provided in the figure below, along with a table of its poles and zeros. See the supplemental materials section for a pole-zero plot of the longitudinal-directional transfer function.

$$G_{p,x}^{SL} = \begin{bmatrix} \dfrac{0.070104\,(s+171.6)(s+0.3547)}{(s^2 - 0.0004353s + 0.007886)(s^2 + 1.295s + 1.063)} \\ \dfrac{-0.041219\,(s+26.12)(s^2 + 0.01064s + 0.01194)}{(s^2 - 0.0004353s + 0.007886)(s^2 + 1.295s + 1.063)} \\ \dfrac{-1.042\,(s+0.4226)(s+0.005902)}{(s^2 - 0.0004353s + 0.007886)(s^2 + 1.295s + 1.063)} \end{bmatrix} \tag{4.1}$$

Poles	Zeros
-0.6475 + 0.8021i	-0.6475 + 0.8021i
-0.6475 - 0.8021i	-0.6475 - 0.8021i
-0.6475 + 0.8021i	
-0.6475 - 0.8021i	
0.0002 + 0.0888i	0.0002 + 0.0888i
0.0002 - 0.0888i	0.0002 - 0.0888i
0.0002 + 0.0888i	
0.0002 - 0.0888i	

Table 4 – Poles and Zeros of Longitudinal Transfer Function Matrix Under Sea Level Conditions

Like with the lateral-directional plant, the longitudinal-directional plant has a total of four pole-zero cancellations, of which two are unstable. As a result, the longitudinal plant is uncontrollable and/or unobservable, just like the lateral plant. This is extremely disappointing, because we cannot design controllers for the aircraft's longitudinal-directional plant nor for its lateral-directional plant, and we are forced to explore other options for controller development. It was eventually determined that the only option left to design controllers for this project was by obfuscating the plant for Youla controller design.

Controller Obfuscation

After all other options are ruled out, the lateral-directional plant is obfuscated for Youla controller design. Integer numbers are used to replace the very large and very small gains from the actual lateral-directional plant, but the plant's general form is kept. The chosen obfuscated plant transfer function matrix is provided in the figure below. This plant is subsequently used for Youla controller design. While there were similar numerical difficulties in the optimal controllers, they were manageable if all coefficients were round to two significant figures, which allowed for parameter sensitivity studies.

$$\tilde{G}_{p,y} = \begin{bmatrix} \dfrac{1}{s^2 + s} & \dfrac{s-3}{s^3 + s^2 + s} \\ \dfrac{-1}{s^3 + s^2 + s} & \dfrac{s+1}{s^3 + s^2 + s} \\ \dfrac{1}{s^3 + 2s^2 + 2s + 1} & \dfrac{s+9}{s^3 + s^2 + s} \end{bmatrix} \tag{5.1}$$

Youla Controller

Three Youla controllers are computed by hand for the obfuscated lateral-directional plant. We start with the Smith-MacMillan form of the plant.

$$M_p = \begin{bmatrix} \frac{1}{s*(s+1)*(s^2+s+1)} & 0 \\ 0 & \frac{1}{s*(s^2+s+1)} \\ 0 & 0 \end{bmatrix} \tag{6.1}$$

To design the first Youla controller, we shape our M_T to resemble the form in which our desired characteristics are given.

$$M_{T_1} = \frac{\omega_{n_{DR}}^2}{\left(s^2 + 2\,\zeta_{DR}\omega_{n_{DR}}s + \omega_{n_{DR}}^2\right)*(\tau_r s+1)*(\tau_s s+1)} \tag{6.2}$$

This is a useful and very direct method because it ensures that the exact performance characteristics defined are achieved. However, as shown in the supplemental materials section, this design method fails to provide adequate robust stability, so the controller is discarded. A second controller is now designed by carefully tailoring M_T to conform to interpolation conditions.

Because we have a pole at the origin for both G_1 and G_2, we want $T_1(0) = 1$ and $T_2(0) = 1$.

We define:

$$Y_1 = \frac{s\,\eta_1(s)}{\delta_1(s)}\ and\ T_2 = T_1 \tag{6.3}$$

So, we can choose $\eta_1$and δ_1:

$$\begin{cases} \delta_1 = (s+1)^4 = s^4 + 4*s^3 + 6*s^2 + 4*s + 1 \\ \eta_1 = 5\,s + 1 \end{cases} \tag{6.4}$$

Giving us:

$$M_{T_2} = \begin{bmatrix} \frac{5*s+1}{(s+1)^4*(s^3+2*s^2+2*s+1)} & 0 \\ 0 & \frac{5*s+1}{(s+1)^4*(s^3+2*s^2+2*s+1)} \end{bmatrix} \tag{6.5}$$

We now extract all open and closed loop transfer function matrices to observe the behavior of the controlled plant in both the time and frequency domain. The figure immediately below illustrates the maximum (solid line) and minimum (dotted line) singular values of the controlled system's open and closed loop transfer function matrices. The controlled plant has mediocre closed-loop characteristics, with no disturbance rejection, no sensor noise rejection, terrible reference tracking, but good robust stability. The system behaves very unpleasantly in the time domain, with excessive overshoot and oscillation. The settling time is also relatively long, but still acceptable. The open-loop behavior of this controller is covered in the supplementary materials section. Overall, this controller does a relatively poor job.

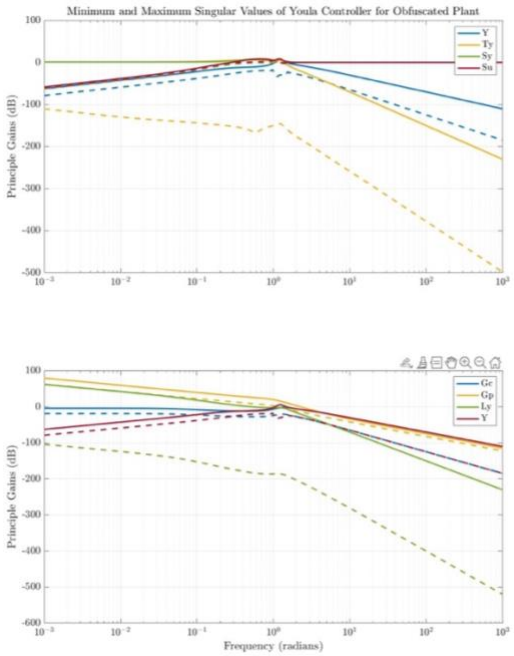

Figure 2 – Singular Values of Second Youla Controlled Obfuscated Lateral-Directional Plant

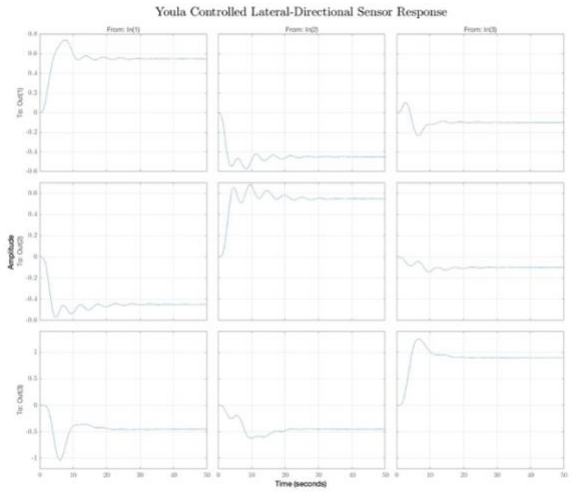

Figure 3 – Step Response of Second Youla Controlled Obfuscated Lateral-Directional Plant

A third approach is now undertaken, combining aspects from both the first and second design methodology to massage the controller's behavior. We implement the higher order of the second controller while keeping the general shape of the first controller and tune the gains until the desired behavior is achieved.

$$M_{T_3} = \begin{bmatrix} \dfrac{5 * s + 1}{(s + 1)^5 * (s^2 + 10 * s + 1)} & 0 \\ 0 & \dfrac{5 * s + 1}{(s + 1)^5 * (s^2 + 10 * s + 1)} \end{bmatrix} \tag{6.6}$$

Finally, a decent Youla controller is achieved, and the controlled plant behavior is analyzed in both the frequency and time domains. The figure immediately below illustrates the closed and open loop maximum and minimum singular values. Similarly to with the second controller, this controlled plant displays no disturbance rejection, no sensor noise rejection, terrible reference tracking, but good control effort minimization and good robust stability. In the time domain, however, the closed-loop behavior looks quite good. The response time is like before but with no overshoot or ripples, which is much more desirable.

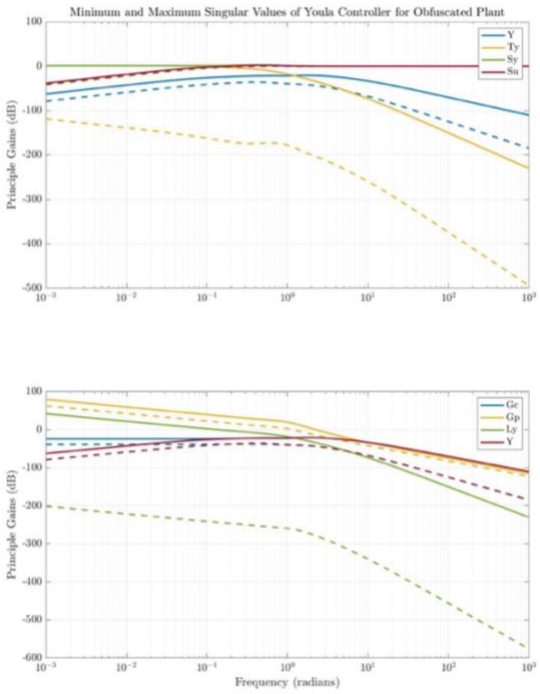

Figure 4 – Singular Values of Third Youla Controlled Obfuscated Lateral-Directional Plant

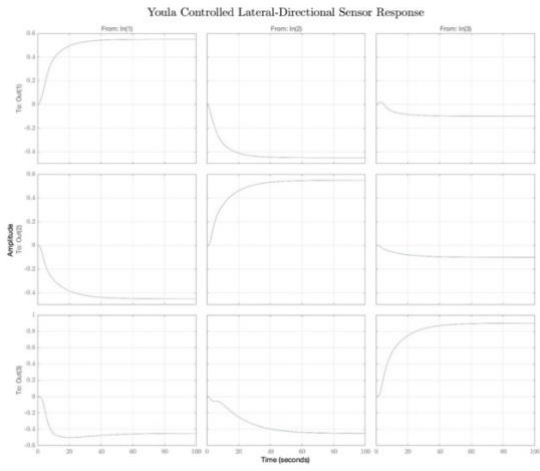

Figure 5 – Singular Values of Third Youla Controlled Obfuscated Lateral-Directional Plant

H_∞ Controller

Design Process

As stated in the section on controller obfuscation, the plants for H_∞ design were rounded to mitigate runaway floating point error while still being able to study parameter variation on the performance and robustness of the controller. Before being able to generate a H_∞ controller two aspects of the plant had to be skewed to fulfill the necessary conditions for optimal H_∞ synthesis. Namely this included skewing the plant such that the double pole on the origin and thus the *jw*-axis and ensuring that D_{12} of the augmented plant was full rank. The skewing of the system was accomplished by shifting the A matrix of the plant be a small amount and then removing the skew to the A matrix of the controller. The D_{12} matrix was forced to be full rank by enforcing a non-zero weight on the actuator effort.

The design of the filters was exceedingly unintuitive compared to the tunable forms chosen for Youla. This is due to the channels representing roll, yaw, and sideslip as opposed to the lateral stability modes—roll, spiral, and Dutch roll—which are governed by the specifications. As such there was much more time tuning the controller for underwhelming results. One of the main issues was attempting to adapt the weighting to each channel to represent that large variance in the underlying stability modes' frequency bands. Namely spiral being much longer term and lower frequency than roll and Dutch roll. Unfortunately, due to limited time and the computational issues that arose there was not enough time to completely address this issue and thus the controller was design with uniform weighting on each channel. See the future works discussed in the conclusion for slightly more discussion.

Next was the selection of the cutoff frequency of the filters. This was done mainly to accommodate a rule of thumb for a given actuator saturation of $\delta_a \in [-15,15]\ deg, \delta_r \in [-20,20]\ deg$ in the step response. This ultimately placed a maximum bound on the cutoff frequency chosen due to excessive efforts and is was settled the cutoff frequency of W_p to be 0.8 rad/s. The cutoff frequency of W_Δ to be 10 rad/s in accordance to the rule of thumb that it be approximately a decade larger than its complimentary sensitivity's counterpart to allow for overshoot of the sensitivity in the transition region.

The weights were initially chosen to be 40dB and -3dB at each frequency region as a general starting point and adjusted from there. The numerical issues seemed to be closely tied to these weight gains and the amount of constriction they placed upon the sensitivity function. If the sensitivity function had to adopt unnecessarily large orders and coefficients to find a proper solution, then the computation of Youla would result in NaN due to large number overflow. Hence allowing margin for sensitivity became one of the keys of tuning the weights. A non-constant weight on actuator effort was also attempted to reduce the amount of effort in the upper transitionary range; however, similar behavior as to what was described above was seen.

The result of the tuning is seen next in the closed loop singular values, and the nominal performance, robust stability, and robust performance plots. The open loop and controller singular values as well as the output and actuator step responses may be seen in supplemental materials.

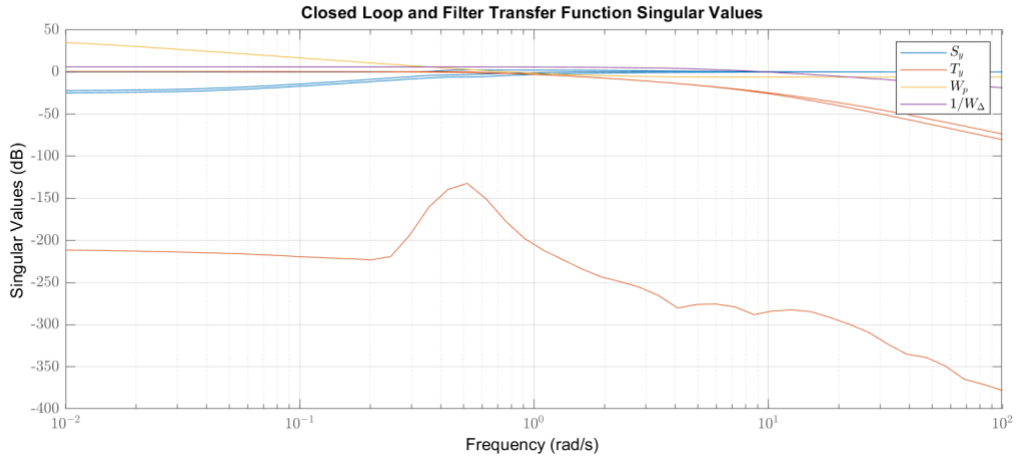

Figure 6 – Sea Level H_∞ Controller Closed Loop Singular Values

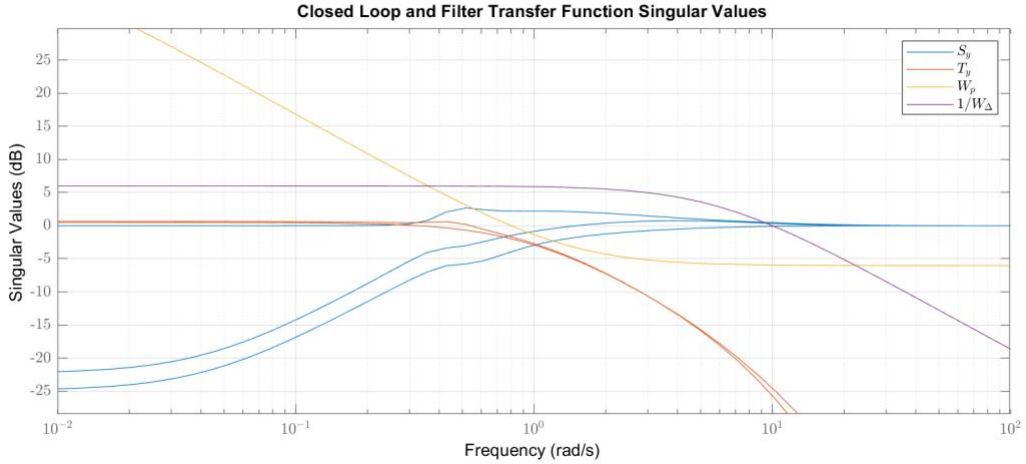

Figure 7 – Sea Level H_∞ Controller Closed Loop Singular Values (Zoomed on Crossover Region)

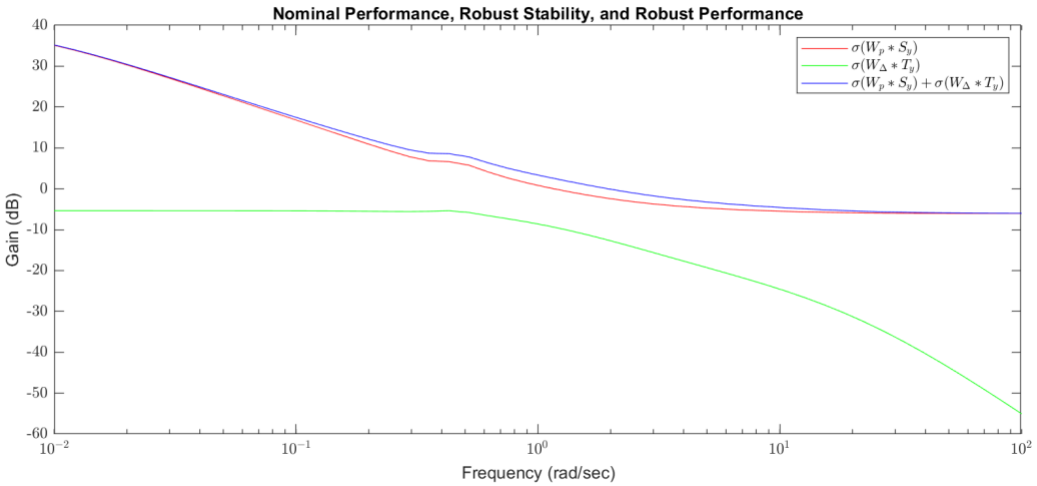

Figure 8 – Sea Level H_∞ Controller NP, RS, & RP

Result Discussion

From Figure 6, it is first apparent the condition number of the closed loop is extremely large. This is believed to be due to the parity and variance of the time constant of the various lateral modes. Unfortunately, no combination of weights seemed to come close to remedying the issue and it is most probably just a result of the plant's behavior. This unfortunately also forces the maximum singular value of the sensitivity to be approximately unity which diminishes tracking ability for that channel. Apart from this channel, the two other singular values are nicely correlated in both sensitivity and complimentary sensitivity.

From the supplemental materials, the allowed actuator displacement was maximized for the step response. This would seriously hinder performance as the actuator would saturate shortly thereafter if any more output were requested. This was done to minimize the roll time constant which can be estimated from the step response to be approximately: $\tau_r = 1.22\ sec$. While this still qualifies for Level 1 handling quality it is a severe drop from the roll response times seen in just the bare airframe in all three conditions. The only two trades could then be improvements in either the spiral mode or Dutch roll mode. Due to the model conflating their effects in yaw, sideslip, and roll, the use of system identification would be needed to accurately find the underlying mode parameters. This was not an option due to original time constraints that were only tightened with unforeseen computational issues.

It can also be seen that the plant is nominally performant; however, does not have robust stability or robust performance especially at low frequency. This is believed to be an effect of the *jw*-axis double pole at the origin which was shifted. At least one of those poles seems to be negated as the robust stability slopes downward at only -20dB/dec. The remaining spiral mode at extremely low frequencies is a very likely candidate for the behavior as it is also probably what is also induced the extremely suppressed channel allowing near direct feedthrough of disturbances at low frequency. All of these behaviors are typical for the spiral mode and thus this controller likely is suffering from either the high condition number between the spiral mode and the other channels or from the spiral mode simply being unstable.

Parameter Variation Study

While robust stability has been established to not exist for the current design, a plant parameter variation routine was written. Selected plots as well as all the rounded transfer function matrix plants used may be seen in the supplemental materials. Namely both alternative operating conditions show small peaks in the mid frequency range that indicate some amount of nominal performance loss. In the case of the second operating condition it is clear cut that nominal performance is lost; however, in the third case the nominal performance seems to be within the constraint. This breaks down in further inspection as it can be seen the step response is unbounded and that none of the complimentary sensitivity channels track to unity in low frequency there for almost definitely violating an interpolation condition similar to the one applied in the Youla control design.

Conclusions and Future Work

The common thread for the entirety of this project was the unforeseeable ability for seemingly innocuous MIMO plants to harbor many undesirable behaviors upon close analysis. The main issues at hand in this project were the ability for hidden modes to mask themselves much more discretely in MIMO systems as well as the effects of large variance in channel frequencies leading to large condition numbers. These two behaviors led to internally unstable plants in some operating conditions as well as mitigated decoupling for channels spanning different frequency bands. The main problem raised by the large condition number was the extreme numerical overflow created while attempting to obtain the unimodular matrices for the diagonalization of the plant. This specific issue required massive time investment to subtly improve until we had to relegate the plant to an extremely oversimplified version to prove a form for the Youla controller.

These issues still yield many questions, and due to the unforeseen time sinks associated with plant complications, the project had to cut back from the scope that was originally planned. This predominantly includes negligence of longitudinal dynamics and the use of gain scheduling to attempt to regain some amount of stability. However, the gain scheduling approach would have required many more operating conditions to be of any practical use. There were also plans to develop system identification models to correlate the optimal controller designs back to the MIL-F-8785C standard. In this sense, if the first Youla form could be modified it would be the most practical way of designing these types of controllers for elevation of handling level. Other topics that were also considered while establishing the project's scope included:

- Modeling actuator effort for performance limitations in short period and roll modes
- Use of Relative Gain Array to quantify the coupling of lateral modes and furthermore designing to minimize roll-yaw coupling.

This project has proved just how difficult it can be to design MIMO controllers for unstable plants with hidden modes. If we were to tackle this task in industry, we as control engineers would return our results to the aero-structural engineers and recommend several design modifications to the plant to improve its controllability and observability. Though it is likely impossible to prevent the equations of motion from propagating hidden modes into our transfer function matrices, we would at least be able to design controllers for the plant if the hidden modes were stable. To stabilize the aircraft's phugoid mode, we suggest decreasing phugoid mode natural frequency by decreasing the horizontal tail size or decreasing the distance from the aircraft center of gravity to the horizontal tail aerodynamic center. Though these changes may take a toll on static stability, we are confident that both static and dynamic stability can be achieved with these design changes. If it is found that static and dynamic stability can not both be achieved by these design modifications, the total airframe drag could be increased to boost phugoid mode damping. Approaches to improve the lateral dynamics in Dutch roll and spiral mode might include increasing wing dihedral angle, increasing vertical tail surface area, or reducing fore-of-wing fuselage side area. With these plant design changes, we are confident that high-performing, robustly stable controllers can be readily developed to produce a very high-quality product.

References

[1] T. Yechout, *et. al.*, "Introduction to Aircraft Flight Mechanics," AIAA, Blacksburg, VA, 2003.

[2] F. Assadian, Class Lectures, "MIMO Robust Optimal Control," MAE273B, College of Engineering, UC Davis, Davis, CA, Mar.-Jun., 2020.

[3] L. Filipozzi. *et. al.* "Estimation of Tire Normal Forces including Suspension Dynamics," Unpublished. IEEE Periodicals. (2018). *IEEE Reference Guide.* Transactions/Journals Department. Piscataway, NJ: IEEE. Retrieved from http://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf

Supplemental Material

Performance Requirement Evaluation Note

The performance requirements for MIL-F-8785C fall under one of three categories: 1st order time constant, τ; natural frequency, ω, & damping, ζ,for 2nd order behavior; and finally time to double, T_2 for unstable behavior. The time constant and 2nd order characteristics are evaluated as normally done while the time to double is governed by:

$$T_2 = \frac{ln\ 2}{\tau} \tag{A1.1}$$

Derivative Nomenclature

Symbol	Description
X	Longitudinal Derivative
Z	Vertical Derivative
M	Pitch Derivative
Y	Lateral Derivative
L	Roll Derivative
N	Yaw Derivative
u	Longitudinal Velocity
α	Angle of Attack
$\dot{\alpha}$	Rate of Angle of Attack
q	Pitch Rate
δ_e	Elevator Angle
β	Sideslip Angle
p	Roll Rate
r	Yaw Rate
δ_a	Aileron Angle
δ_r	Rudder Angle

Table A1 – Summary of Derivative Nomenclature

State-Space Representations

Longitudinal State-Space Representation Derivation

We begin with the longitudinal-linearized equations of motion in Laplace form as follow.

$$\dot{u} = -g\theta\cos(\Theta_1) + X_u u + X_{T_u} u + X_\alpha \alpha + X_{\delta_e}\delta_e \tag{1.1}$$

$$\dot{w} - U_1 q = -g\theta\sin(\Theta_1) + Z_u\, u + Z_\alpha \alpha + Z_{\dot{\alpha}}\dot{\alpha} + Z_q q + Z_{\delta_e}\delta_e \tag{1.2}$$

$$\dot{q} = M_u u + M_{T_u} u + M_\alpha \alpha + M_{T_\alpha}\alpha + M_{\dot{\alpha}}\dot{\alpha} + M_q q + M_{\delta_e}\delta_e \tag{1.3}$$

Small angle of attack perturbation and wings level, trimmed flight conditions are assumed. The small angle approximation for angle of attack is used.

$$\dot{w} \approx \dot{\alpha} U_1 \tag{A2.1}$$

$$q = \dot{\theta} \tag{A2,2}$$

$$\dot{q} = \ddot{\theta} \tag{A2.3}$$

From (1.2) and (A2.1), we develop a relationship for the angle of attack rate.

$$\dot{\alpha}U_1 - Z_{\dot{\alpha}}\dot{\alpha} = U_1 q - g\theta\sin\Theta_1 + Z_u u + Z_\alpha \alpha + Z_q q + Z_{\delta_e}\delta_e \tag{\boldsymbol{A2.4}}$$

$$\dot{\alpha} = \frac{U_1 q - g\theta\,\sin\Theta_1 + Z_u u + Z_\alpha \alpha + Z_q q + Z_{\delta_e}\delta_e}{U_1 - Z_{\dot{\alpha}}} \tag{\boldsymbol{A2.5}}$$

We can now choose the state variables.

$$x = \begin{bmatrix} \theta \\ \alpha \\ u \\ q \end{bmatrix} \tag{A2.6}$$

A state-space representation of the form:

$$E\,\dot{x} = A_c\,x + B_c\,u \tag{A2.7}$$

$$y = Cx + Du \tag{A2.8}$$

is subsequently created.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (U_1 - Z_{\dot{\alpha}}) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -M_{\dot{\alpha}} & 0 & 1 \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \dot{u} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ -g\sin\Theta_1 & Z_\alpha & Z_u & (Z_q + U_1) \\ -g\cos\Theta_1 & X_\alpha & (X_u + X_{T_u}) & 0 \\ 0 & (M_\alpha + M_{T_\alpha}) & (M_u + M_{T_u}) & M_q \end{bmatrix}\begin{bmatrix} \theta \\ \alpha \\ u \\ q \end{bmatrix} \ldots$$

$$+ \begin{bmatrix} 0 \\ Z_{\delta_e} \\ X_{\delta_e} \\ M_{\delta_e} \end{bmatrix}[\delta_e] \tag{A2.9}$$

We are most interested in pitch angle, pitch rate and angle of attack, so we define C as follows.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A2.10}$$

D is just a vector of zeros.

$$D = zeros(4,1) \tag{A2.11}$$

The E matrix is now embedded in the A and B matrices to complete the state-space realization.

$$A = E^{-1}A_c \tag{A2.12}$$

$$B = E^{-1}B_c \tag{A2.13}$$

Lateral State-Space Representation Derivation

We begin with the lateral-linearized equations of motion in Laplace form as follow.

$$\dot{v} + U_1 r = g\phi \cos\Theta_1 + Y_\beta \beta + Y_p p + Y_r r + Y_{\delta_a}\delta_a + Y_{\delta_r}\delta_r \tag{2.1}$$

$$\dot{p} - \frac{I_{xz}}{I_{xx}}\dot{r} = L_\beta \beta + L_p p + L_r r + L_{\delta_a}\delta_a + L_{\delta_r}\delta_r \tag{2.2}$$

$$\dot{r} - \frac{I_{xz}}{I_{zz}}\dot{p} = N_\beta \beta + N_{T_\beta}\beta + N_p p + N_r r + N_{\delta_a}\delta_a + L_{\delta_r}\delta_r \tag{2.3}$$

Small sideslip angle perturbation and wings level, trimmed flight conditions are assumed. The small angle approximation for angle of attack is used.

$$\dot{v} \approx \dot{\beta} U_1 \tag{$\boldsymbol{A3.1}$}$$

$$p = \dot{\phi} \tag{$A3.2$}$$

$$r = \dot{\psi} \tag{$A3.3$}$$

We can thus rewrite (2.1) as follows.

$$\dot{\beta} U_1 + U_1 r = g\phi \cos\Theta_1 + Y_\beta \beta + Y_p p + Y_r r + Y_{\delta_a}\delta_a + Y_{\delta_r}\delta_r \tag{$A3.4$}$$

$$\dot{\beta} = \frac{-U_1 r + g\phi \cos\Theta_1 + Y_\beta \beta + Y_p p + Y_r r + Y_{\delta_a}\delta_a + Y_{\delta_r}\delta_r}{U_1} \tag{$A3.5$}$$

We can now choose the state variables.

$$x = \begin{bmatrix} \phi \\ \psi \\ \beta \\ p \\ r \end{bmatrix} \tag{$\boldsymbol{A3.6}$}$$

A state-space representation of the form:

$$E\dot{x} = A_c x + B_c u \tag{$A3.7$}$$

$$y = Cx + Du \tag{$A3.8$}$$

is subsequently created.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -\frac{I_{xz}}{I_{xx}} \\ 0 & 0 & 0 & -\frac{I_{xz}}{I_{zz}} & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\psi} \\ \dot{\beta} \\ \dot{p} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\frac{g\phi \cos\Theta_1}{U_1} & 0 & \frac{Y_\beta}{U_1} & \frac{Y_p}{U_1} & -1 + \frac{Y_r}{U_1} \\ 0 & 0 & L_\beta & L_p & L_r \\ 0 & 0 & N_\beta + N_{T_\beta} & N_p & N_r \end{bmatrix} \begin{bmatrix} \phi \\ \psi \\ \beta \\ p \\ r \end{bmatrix} \ldots$$

$$+\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{Y_{\delta_a}}{U_1} & \frac{Y_{\delta_r}}{U_1} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} [\delta_a \quad \delta_r] \tag{A3.9}$$

We are most interested in roll angle, yaw angle and sideslip angle, so we define C as follows.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{A3.10}$$

D is just a vector of zeros.

$$D = zeros(3,2) \tag{A3.11}$$

The E matrix is now embedded in the A and B matrices to complete the state-space realization.

$$A = E^{-1}A_c \tag{A3.12}$$

$$B = E^{-1}B_c \tag{A3.13}$$

Lateral Transfer Function Matrices Used in H_∞ Controller Design

For the first operating condition:

$$\bar{G}_{p,y}^{SL} = \begin{bmatrix} \frac{0.31\,(s^2 + 0.2s + 0.18)}{s\,(s+0.29)(s^2 + 0.25s + 0.2)} & \frac{0.024\,(s-3)(s+1.2)}{s\,(s+0.29)(s^2 + 0.25s + 0.2)} \\ \frac{-0.017\,(s+0.076)}{(s+0.29)(s^2 + 0.25s + 0.2)} & \frac{-0.14\,(s+0.33)(s+0.057)}{s\,(s+0.29)(s^2 + 0.25s + 0.202)} \\ \frac{0.017\,s}{(s+0.29)(s^2 + 0.25s + 0.2)} & \frac{0.017(s+8.7)(s+0.33)}{(s+0.29)(s^2 + 0.25s + 0.2)} \end{bmatrix} \tag{A4.1}$$

For the second operating condition:

$$\bar{G}_{p,y}^{2} = \begin{bmatrix} \frac{0.39\,(s^2 + 0.26s + 0.47)}{s\,(s+0.54)(s^2 + 0.15s + 0.45)} & \frac{0.13\,(s-2.6)(s+2)}{s\,(s+0.54)(s^2 + 0.15s + 0.45)} \\ \frac{0.014\,(s-0.7)(s+0.1)}{(s+0.54)(s^2 + 0.15s + 0.45)} & \frac{-0.5(s+0.45)(s+0.076)}{s\,(s+0.54)(s^2 + 0.15s + 0.45)} \\ \frac{0.014\,(s-0.7)}{(s+0.54)(s^2 + 0.15s + 0.45)} & \frac{0.023\,(s+22)(s+0.45)}{(s+0.54)(s^2 + 0.15s + 0.45)} \end{bmatrix} \tag{A4.2}$$

For the third operating condition:

$$\bar{G}_{p,y}^{3} = \begin{bmatrix} \frac{0.30(s^2 + 0.12s + 0.2)}{s\,(s+0.25)(s^2 + 0.087s + 0.23)} & \frac{0.062\,(s-1.7)(s+1.4)}{s\,(s+0.25)(s^2 + 0.087s + 0.23)} \\ \frac{-0.007\,(s+0.66)(s+0.043)}{(s+0.25)(s^2 + 0.087s + 0.23)} & \frac{-0.2\,(s+0.32)(s+0.04)}{s\,(s+0.25)(s^2 + 0.087s + 0.23)} \\ \frac{0.007\,(s+0.66)}{(s+0.25)(s^2 + 0.087s + 0.23)} & \frac{0.003\,(s+68)(s+0.22)}{(s+0.25)(s^2 + 0.087s + 0.23)} \end{bmatrix} \tag{A4.3}$$

Additional Figures

Pole Zero Plots of Lateral and Longitudinal Transfer Function Matrices

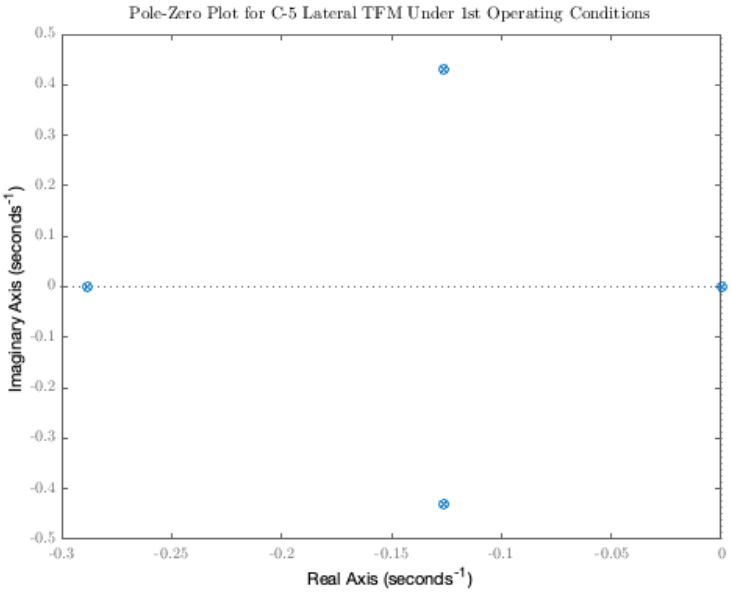

Figure A1 – Pole-Zero Plot of Lateral Transfer Function Matrix Under Sea Level Conditions

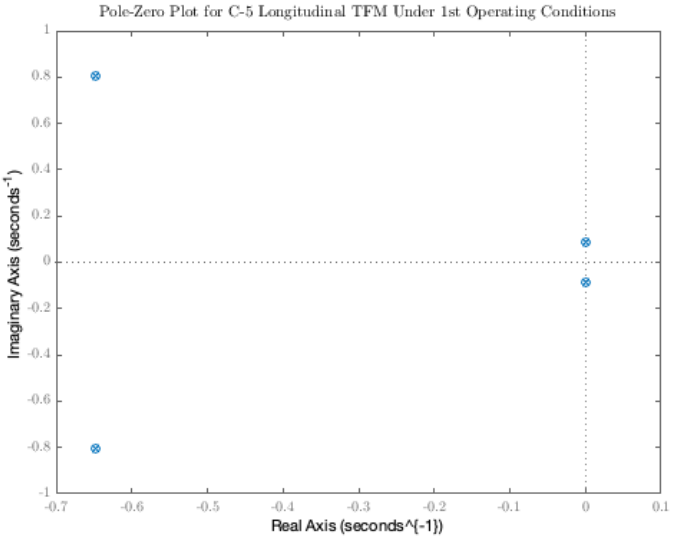

Figure A2 – Pole-Zero Plot of Longitudinal Transfer Function Matrix Under Sea Level Conditions

Frequency and Time Domain Behavior of First Youla Controller

As can be seen in the two figures below, the First Youla controller design has no robust stability and as a result, is completely unusable.

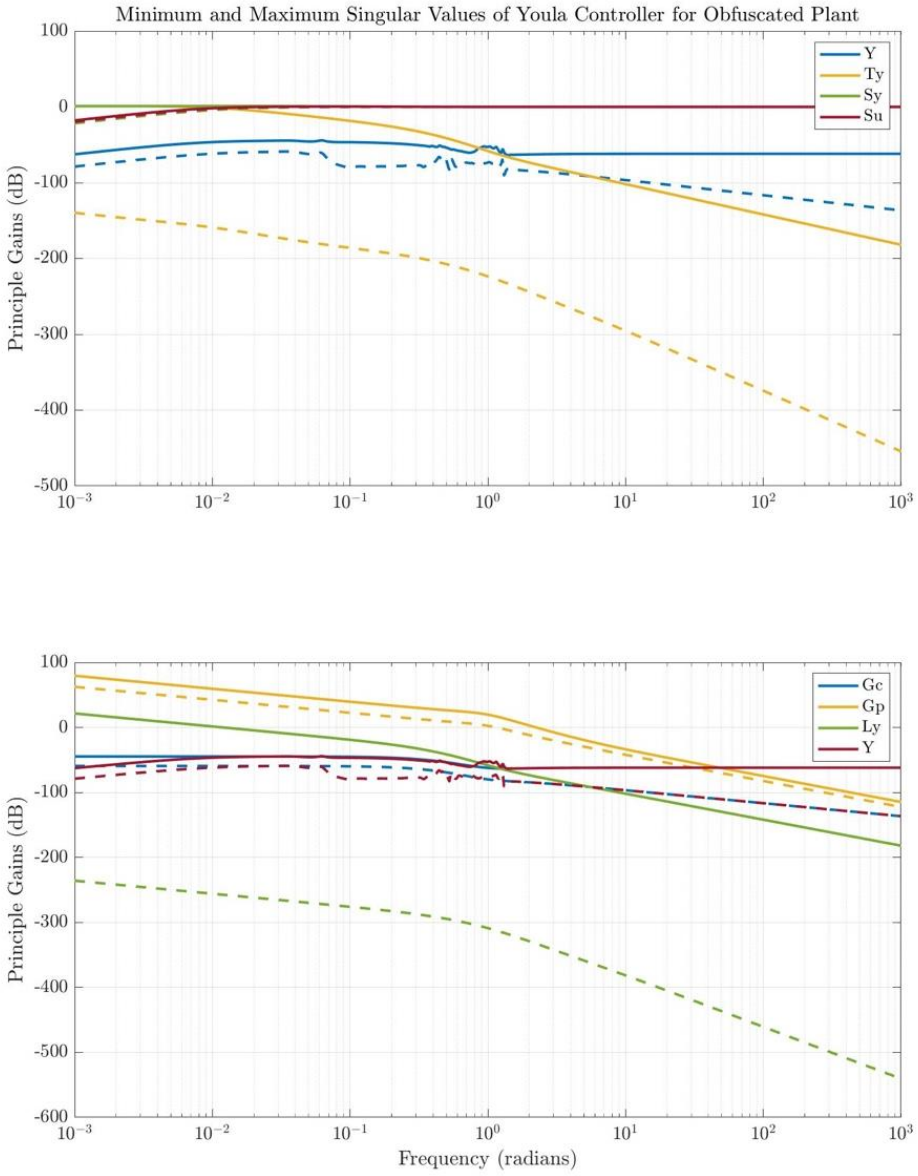

Figure A3 – Minimum and Maximum Singular Values of First (Discarded) Youla Controller

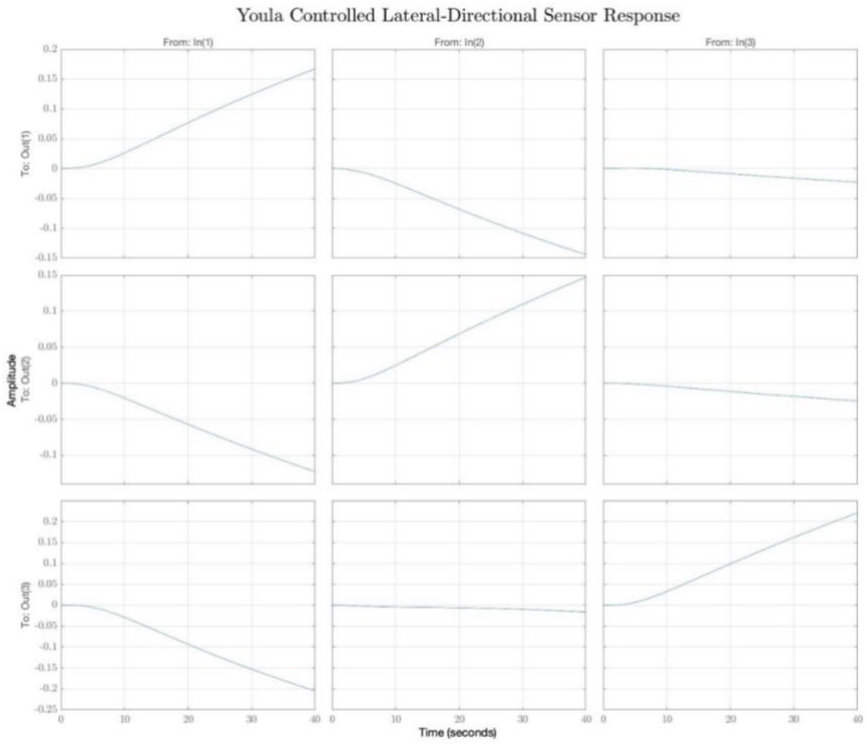

Figure A4 – Step Response of First (Discarded) Youla Controller

Open-Loop Behavior of Youla Controllers

Controller Designed Exclusively Based on Interpolation Conditions (2nd Controller)

In the open-loop, the system has bad disturbance rejection, good sensor noise rejection, bad tracking, good control effort minimization, and good robust stability.

Controller Designed Using a Combination of Both Methodologies (3rd Controller)

The open-loop behavior of this controller is identical to the open-loop behavior of the second controller. The system has bad disturbance rejection, good sensor noise rejection, bad tracking, good control effort minimization, and good robust stability.

Additional H_∞ Controller Plots

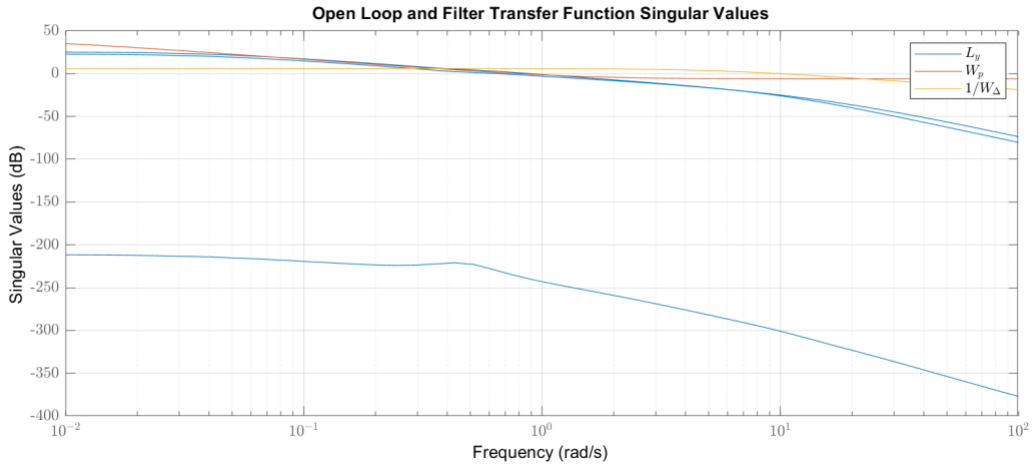

Figure A5 – Open Loop Singular Values of H_∞ Controller

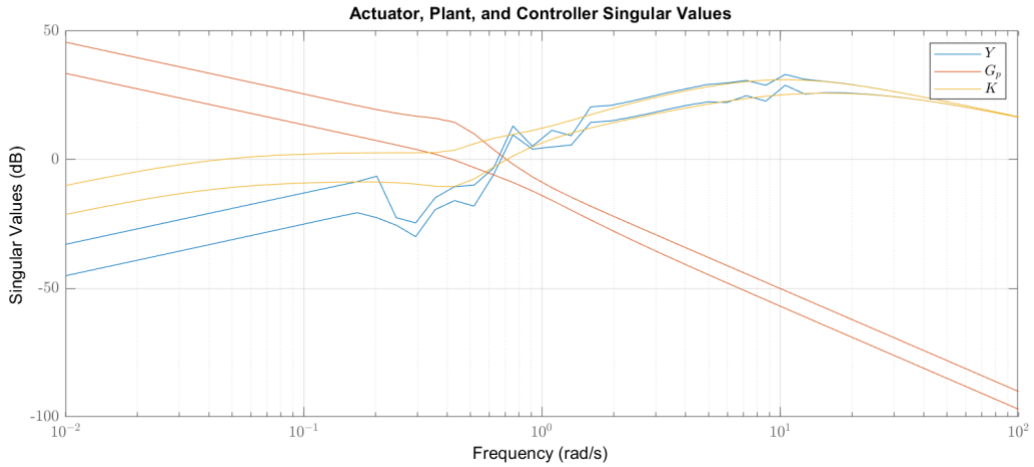

Figure A6 – Youla, Controller, and Plant Singular Values of H_∞ Controller

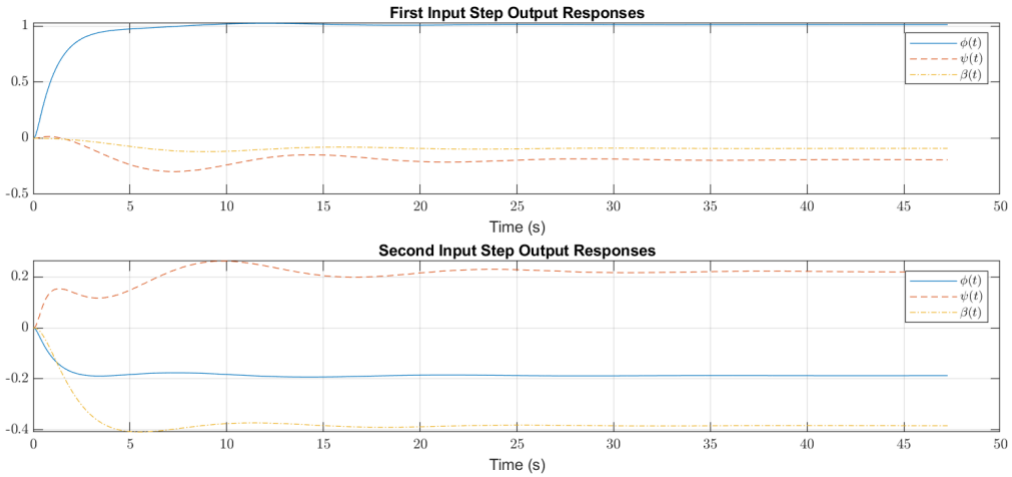

Figure A7 – Output Step Responses of H_∞ Controller

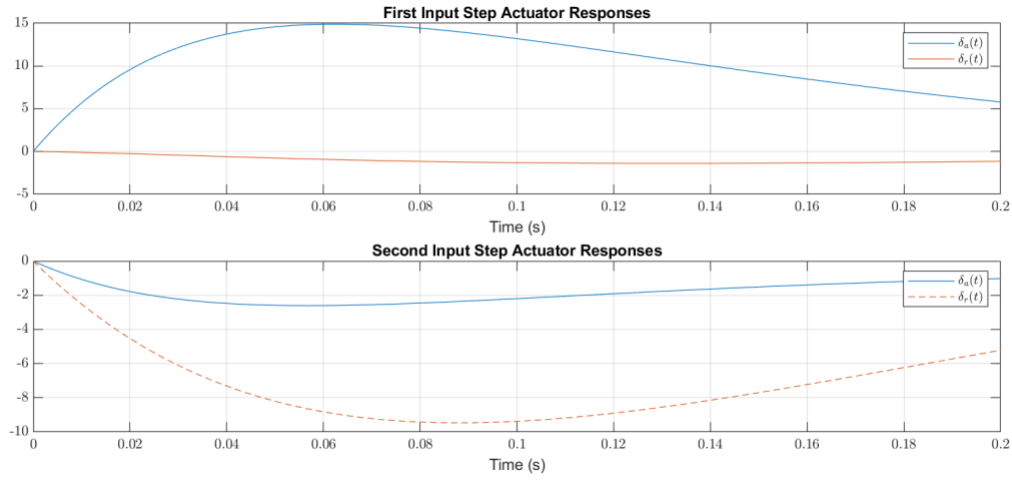

Figure A8 – Actuator Step Responses of H_∞ Controller

H_∞ Controller Plant Parameter Variation Performance

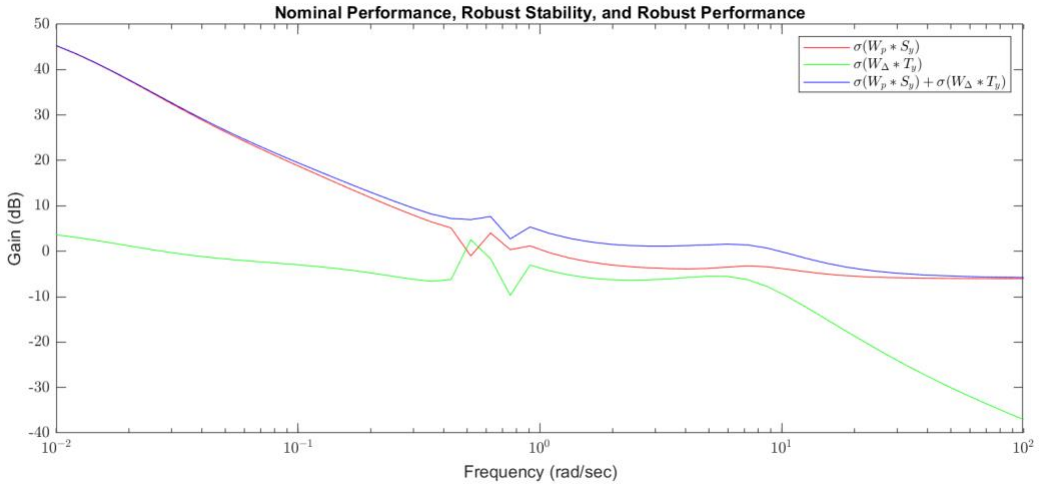

Figure A9 – Sea Level H_∞ *Controller Applied to Operating Condition #2 NP, RS, & RP*

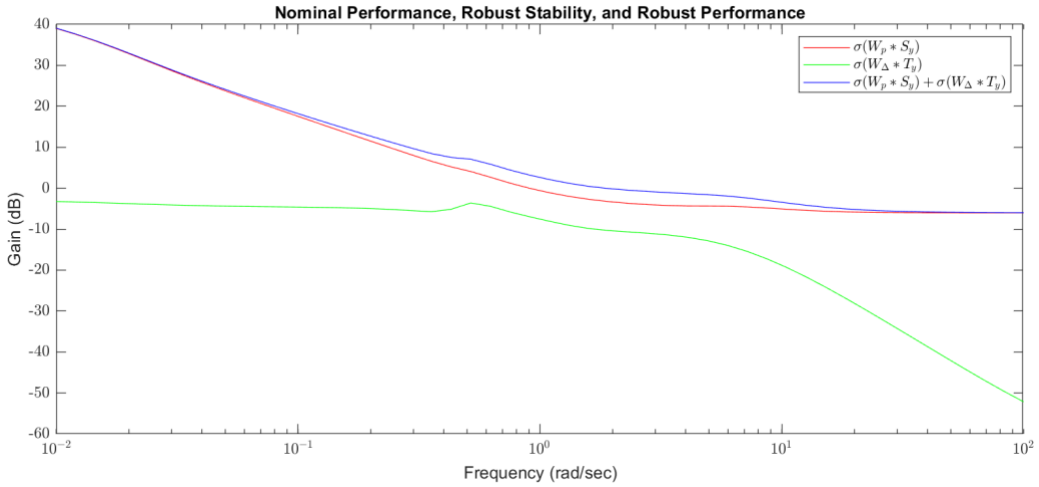

Figure A10 – Sea Level H_∞ *Controller Applied to Operating Condition #3 NP, RS, & RP*

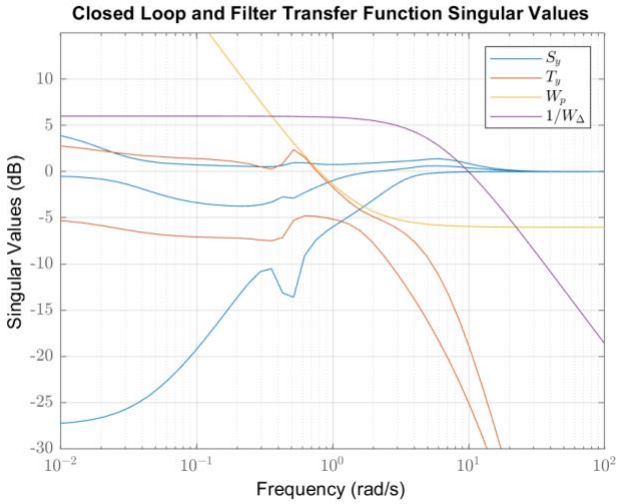

Figure A11 – Crossover Region of Sea Level H_∞ Controller Applied to Operating Condition #3

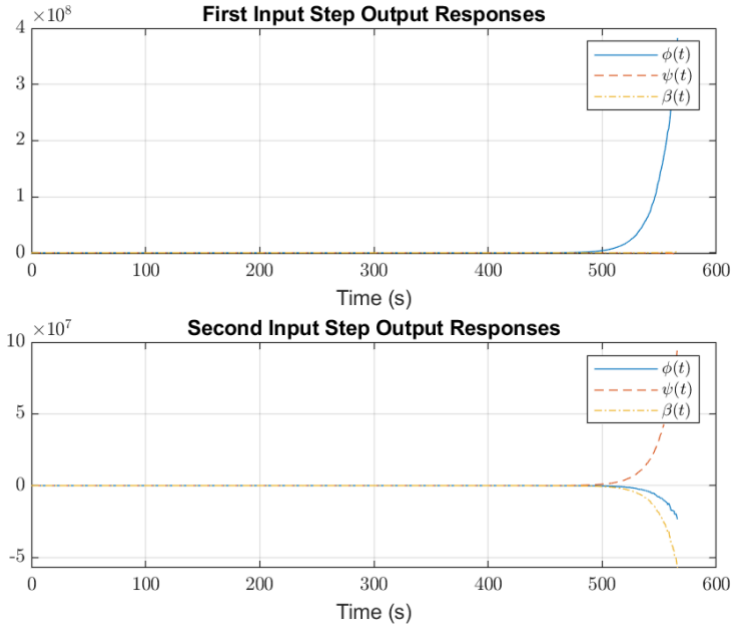

Figure A7 – Unstable Output Step Responses of Sea Level H_∞ Controller Applied to Operating Condition #3

MATLAB Code

Main: MAE273B_Final_Greenhill_Christierson

```
clc; clear; close all;

set(groot,'defaulttextinterpreter','latex');
set(groot,'defaultaxesticklabelinterpreter','latex');
set(groot,'defaultlegendinterpreter','latex');

syms s

%% C5-A MIMO Youla & H_inf Controller Project Main Script
% Thomas Greenhill
% Blake Christierson
% June 11, 2020
%
% MAE 273B SQ2020
% Professor Francis Assadian

%% State Space & Transfer Function Realization Loading
% The functions used to generate the plant realization data files being
% loaded are present in the "Plant Realization" Folder. Load one of the
% data files as an input to "Produce_C5_Gp_sym_F2.mat" to generate a new
% symbolic plant file.

% Operating Condition #1:
% Aircraft: C-5A
% Ma: 0.25
% Elevation: Sea Level
% Trim AOA: 11.41 deg
Gp(1) = load('Gp_1_sym.mat');

% Operating Condition #2:
% Aircraft: C-5A
% Ma: 0.7
% Elevation: 6096 m
% Trim AOA: 0.68 deg
Gp(2) = load('Gp_2_sym.mat');

% Operating Condition #3:
% Aircraft: C-5A
% Ma: 0.7
% Elevation: 40,000 ft
% Trim AOA: 7.66 deg
Gp(3) = load('Gp_3_sym.mat');

Gp = [Gp(:).Gp]; % Resolves embedding due to mismanagement of structure

%% Plotting Base Plant
% [P] = Plottingx3(Gp(1), Gp(2), Gp(3));
% saveas(gcf,'./Figures/Uncontrolled_Plant/StepResponse.jpg')

%% Youla Controllers
% Only a single case was considered due to the computational
% barriers that prevented the creating the plant lead to the requirement of
% oversimplification. This rounding lead to complete obfuscation of the
% plant and is discussed in the report

% Obfuscated Plant & Polynomial Matrix
Gp(1).Lat.subs = [  ( 1*(s^2+1*s+1) /(s*(s+1)*(s^2+1*s+1))) ...
                    (1*(s-3)*(s+1)  /(s*(s+1)*(s^2+1*s+1))); ...
                    (-1*(s+1)       /(s*(s+1)*(s^2+1*s+1))) ...
                    (1*(s+1)*(s+1)  /(s*(s+1)*(s^2+1*s+1))); ...
                    (1*s            /(s*(s+1)*(s^2+1*s+1))) ...
                    (1*(s+9)*(s+1)  /(s*(s+1)*(s^2+1*s+1)))];

Gp(1).Lat.P.subs = [1*(s^2+1*s+1)   1*(s-3)*(s+1); ...
                    -1*(s+1)        1*(s+01)*(s+1); ...
                    1*s             1*(s+9)*(s+1)];

%%% Simplifying the Lateral TFM and the SM form of the TFM
Gp(1).Lat.TFM = minreal(zpk(sym2tf(Gp(1).Lat.subs)));

Gp(1).Lat.Mp = minreal(zpk(smform(Gp(1).Lat.TFM)));

%%% Find the Numerator and Denominator of the TFM
[Gp(1).Lat.Num, Gp(1).Lat.Den] = numden(Gp(1).Lat.subs);

syms s
```

```
[Gp(1).Lat.UL, Gp(1).Lat.H] = hermiteForm(Gp(1).Lat.P.subs);

Gp(1).Lat.UL = simplify(Gp(1).Lat.UL);

Gp(1).Lat.Mp = tf2sym(Gp(1).Lat.Mp);
Gp(1).Lat.Mp = subs( Gp(1).Lat.Mp, 'x', 's' );

Gp(1).Lat.UR = (Gp(1).Lat.UL*Gp(1).Lat.subs)\Gp(1).Lat.Mp;

%%% Shape Our Youla Controller Based on MT for the Obfuscated Plant
syms s

T1 = (5*s+1)/((s+1)^4*(s^3+2*s^2+2*s+1));

T2 = T1;

[Gp(1).Lat] = C5_Youla(Gp(1).Lat, T1, T2);

%%% Plotting of Principal Gains and Step Response
figure('Position',[1000 100 1000 800]);

step(Gp(1).Lat.Ty)
grid on
title('Youla Controlled Lateral-Directional Sensor
Response','FontSize',18,'Interpreter','latex');
% saveas(gcf,'./Figures/Youla/Step_Youla_Lat_1.jpg')

w = logspace(-3,3,1000);
PrincipleGainAnalysis(Gp(1).Lat.subs, Gp(1).Lat.Gc, Gp(1).Lat.Ly, Gp(1).Lat.Ty, Gp(1).Lat.Sy,
Gp(1).Lat.Su, Gp(1).Lat.Y, w)
set(findall(gcf,'type','line'),'linewidth',2);
set(findall(gcf,'type','axes'),'fontsize',14);
subplot(2,1,1)
title('Minimum and Maximum Singular Values of Youla Controller for Obfuscated Plant')

% saveas(gcf,'./Figures/Youla/Sigma_Youla_Lat_1.jpg')

clear T1 T2 s

Gp(1).Lat.subs = tf2sym(Gp(1).Lat.subs);

syms s

T1 = (5*s+1)/((s+1)^4*(s+1)*(s^2+10*s+1));

T2 = T1;

[Gp(1).Lat] = C5_Youla(Gp(1).Lat, T1, T2);

%%% Plotting of Principal Gains and Step Response
figure('Position',[1000 100 1000 800]);

step(Gp(1).Lat.Ty)
grid on
title('Youla Controlled Lateral-Directional Sensor
Response','FontSize',18,'Interpreter','latex');
% saveas(gcf,'./Figures/Youla/Step_Youla_Lat_2.jpg')

w = logspace(-3,3,1000);
PrincipleGainAnalysis(Gp(1).Lat.subs, Gp(1).Lat.Gc, Gp(1).Lat.Ly, Gp(1).Lat.Ty, Gp(1).Lat.Sy,
Gp(1).Lat.Su, Gp(1).Lat.Y, w)
set(findall(gcf,'type','line'),'linewidth',2);
set(findall(gcf,'type','axes'),'fontsize',14);
subplot(2,1,1)
title('Minimum and Maximum Singular Values of Youla Controller for Obfuscated Plant')

% saveas(gcf,'./Figures/Youla/Sigma_Youla_Lat_2.jpg')

%% Hinf Controller
% Similar runaway computational errors were seen in the Hinf process,
% therefore the plant was somewhat simplified by rounding to two
% significant figures. This still allowed for the comparison of the
% different flight cases.
%
% Use minreal( zpk( sym2tf( Gp(i).Lat.subs ) ) ) to view full plant
% before plant is obfuscated in Youla section.

% Rounded Plants
Gp(1).Lat.subs = [(0.31*(s^2+0.2*s+0.18) /(s*(s+0.29)*(s^2+0.25*s+0.20))) ...
    (0.024*(s-3.0)*(s+1.2)    /(s*(s+0.29)*(s^2+0.25*s+0.20))); ...
```

```
                (-0.017*(s+0.76)       /(s*(s+0.29)*(s^2+0.25*s+0.20))) ...
    (-0.14*(s+0.32)*(s+0.057) /(s*(s+0.29)*(s^2+0.25*s+0.20))); ...
                (0.017*s               /(s*(s+0.29)*(s^2+0.25*s+0.20))) ...
    (0.017*(s+8.7)*(s+0.33)   /(s*(s+0.29)*(s^2+0.25*s+0.20)))];

Gp(2).Lat.subs = [(0.39*(s^2+0.26*s+0.47) /(s*(s+0.54)*(s^2+0.15*s+0.45))) ...
    (0.13*(s-2.6)*(s+2)  /(s*(s+0.54)*(s^2+0.15*s+0.45))); ...
                (0.014*(s-0.7)*(s+0.1)  /(s*(s+0.54)*(s^2+0.15*s+0.45))) ...
    (-0.5*(s+0.45)*(s+0.076) /(s*(s+0.54)*(s^2+0.15*s+0.45))); ...
                (-0.014*(s-0.7)         /((s+0.54)*(s^2+0.15*s+0.45))) ...
    (0.023*(s+22)*(s+0.45)  /((s+0.54)*(s^2+0.15*s+0.45)))];

Gp(3).Lat.subs = [(0.30*(s^2+0.11*s+0.2)     /(s*(s+0.25)*(s^2+0.087*s+0.23))) ...
    (0.062*(s-1.7)*(s+1.4)  /(s*(s+0.25)*(s^2+0.087*s+0.23))); ...
                (-0.007*(s+0.66)*(s+0.043) /(s*(s+0.25)*(s^2+0.087*s+0.23))) ...
    (-0.2*(s+0.22)*(s+0.04) /(s*(s+0.25)*(s^2+0.087*s+0.23))); ...
                ( 0.007*(s+0.66)           /((s+0.25)*(s^2+0.087*s+0.23))) ...
    (0.003*(s+68)*(s+0.22)  /((s+0.25)*(s^2+0.087*s+0.23)))];

% Controller Generation
Hinf(1).Lat = Produce_Lateral_Hinf_2( sym2tf( Gp(1).Lat.subs ) );

% Parameter Variation Study
Hinf(2).Lat = Robust_Parameter_Study( Hinf(1).Lat, sym2tf( Gp(2).Lat.subs ) );
Hinf(3).Lat = Robust_Parameter_Study( Hinf(1).Lat, sym2tf( Gp(3).Lat.subs ) );
```

Published with MATLAB® R2020a

Plant Generation Main: Produce_C5_Gp_sym_F2()

```
function [Gp] = Produce_C5_Gp_sym_F2(C5)
%% Aircraft State-Space Representation for C5-A Galaxy
% Thomas Greenhill
% Blake Christierson
% June 11, 2020
%
% MAE 273B SQ2020
% Professor Francis Assadian

%% Symbolic Variables
syms s

% Inertial & Trim
syms I_xx I_yy I_zz I_xz
syms g U_1 Theta_1 phi

% Longitudinal Derivatives
syms X_u X_T_u X_alpha X_delta_e
syms Z_u Z_alpha Z_alpha_dot Z_q Z_delta_e
syms M_u M_T_u M_alpha M_T_alpha M_alpha_dot M_q M_delta_e

% Lateral Derivatives
syms Y_beta Y_p Y_r Y_delta_alpha Y_delta_r
syms L_beta L_p L_r L_delta_alpha L_delta_r
syms N_beta N_T_beta N_p N_r N_delta_alpha N_delta_r

%% State Space Creation

%%% Lateral
[Ac, Bc, Gp.Lat.C, Gp.Lat.D, E] = Lateral_SSRep_F2( ...
    Y_beta, Y_p, Y_r, Y_delta_alpha, Y_delta_r, ...
    L_beta, L_p, L_r, L_delta_alpha, L_delta_r, ...
    N_beta, N_T_beta, N_p, N_r, N_delta_alpha, N_delta_r, ...
    I_xx, I_yy, I_zz, I_xz, ...
    g, U_1, Theta_1, phi);

Gp.Lat.A = E^-1*Ac;
Gp.Lat.B = E^-1*Bc;

%% Full & Substituted Symbolic Transfer Function Matrices
% To prevent any accumulation of floating point error, matrices used in
% operations are fully symbolic and then substituted with numerical values.


%%% Lateral
Gp.Lat.syms = simplify( Gp.Lat.C * (s*eye(size(Gp.Lat.A)) - Gp.Lat.A)^-1 * ...
    Gp.Lat.B + Gp.Lat.D);

Gp.Lat.P.syms = simplify(Gp.Lat.syms .* ...
    (s*(I_xz^2*U_1*s^4 - I_xz^2*Y_beta*s^3 - I_xx*I_zz*U_1*s^4 + ...
    I_xx*I_zz*Y_beta*s^3 - I_xx*I_xz*L_beta*U_1*s^2 + ...
    I_xx*I_xz*L_r*U_1*s^3 + I_xx*I_zz*L_p*U_1*s^3 - ...
    I_xx*I_zz*N_T_beta*U_1*s^2 - I_xx*I_zz*N_beta*U_1*s^2 + ...
    I_xx*I_zz*N_r*U_1*s^3 + I_xx*I_xz*L_beta*Y_r*s^2 - ...
    I_xx*I_xz*L_r*Y_beta*s^2 + I_xx*I_zz*L_beta*Y_p*s^2 - ...
    I_xx*I_zz*L_p*Y_beta*s^2 + I_xz*I_zz*N_T_beta*Y_p*s^2 + ...
    I_xx*I_zz*N_T_beta*Y_r*s^2 + I_xz*I_zz*N_beta*Y_p*s^2 + ...
    I_xx*I_zz*N_beta*Y_r*s^2 - I_xx*I_zz*N_r*Y_beta*s^2 + ...
    I_xx*I_zz*L_p*N_T_beta*U_1*s + I_xx*I_zz*L_p*N_beta*U_1*s - ...
    I_xx*I_zz*L_p*N_T_beta*Y_r*s + I_xx*I_zz*L_r*N_T_beta*Y_p*s - ...
    I_xx*I_zz*L_beta*N_r*Y_p*s - I_xx*I_zz*L_p*N_beta*Y_r*s + ...
    I_xx*I_zz*L_p*N_r*Y_beta*s + I_xx*I_zz*L_r*N_beta*Y_p*s - ...
    I_xx*I_zz*L_p*N_r*U_1*s^2 + I_xx*I_zz*L_r*N_T_beta*g*phi*cos(Theta_1) - ...
    I_xx*I_zz*L_beta*N_r*g*phi*cos(Theta_1) + ...
    I_xx*I_zz*L_r*N_beta*g*phi*cos(Theta_1) + ...
    I_xx*I_zz*L_beta*g*phi*s*cos(Theta_1) + ...
    I_xz*I_zz*N_T_beta*g*phi*s*cos(Theta_1) + ...
    I_xz*I_zz*N_beta*g*phi*s*cos(Theta_1))));
    % The heinous denominator above is taken from some command line
    % determination of lcm of denominators of the system.

% Substitution
Gp.Lat.subs = vpa( simplify( subs( Gp.Lat.syms, ...
    { Y_beta, Y_p, Y_r, Y_delta_alpha, Y_delta_r, ...
    L_beta, L_p, L_r, L_delta_alpha, L_delta_r, ...
    N_beta, N_T_beta, N_p, N_r, N_delta_alpha, N_delta_r, ...
    I_xx, I_yy, I_zz, I_xz, ...
    g, U_1, Theta_1, phi }, ...
    ...
```

```
    { C5.Y_beta, C5.Y_p, C5.Y_r, C5.Y_delta_alpha, C5.Y_delta_r, ...
    C5.L_beta, C5.L_p, C5.L_r, C5.L_delta_alpha, C5.L_delta_r, ...
    C5.N_beta, C5.N_T_beta, C5.N_p, C5.N_r, C5.N_delta_alpha, C5.N_delta_r, ...
    C5.I_xx, C5.I_yy, C5.I_zz, C5.I_xz, ...
    C5.g, C5.U_1, C5.Theta_1, C5.phi } ) ) );

Gp.Lat.P.subs = vpa( simplify( subs( Gp.Lat.P.syms, ...
    { Y_beta, Y_p, Y_r, Y_delta_alpha, Y_delta_r, ...
    L_beta, L_p, L_r, L_delta_alpha, L_delta_r, ...
    N_beta, N_T_beta, N_p, N_r, N_delta_alpha, N_delta_r, ...
    I_xx, I_yy, I_zz, I_xz, ...
    g, U_1, Theta_1, phi }, ...
    ...
    { C5.Y_beta, C5.Y_p, C5.Y_r, C5.Y_delta_alpha, C5.Y_delta_r, ...
    C5.L_beta, C5.L_p, C5.L_r, C5.L_delta_alpha, C5.L_delta_r, ...
    C5.N_beta, C5.N_T_beta, C5.N_p, C5.N_r, C5.N_delta_alpha, C5.N_delta_r, ...
    C5.I_xx, C5.I_yy, C5.I_zz, C5.I_xz, ...
    C5.g, C5.U_1, C5.Theta_1, C5.phi } ) ) );

Gp.Lat.A = vpa( simplify( subs( Gp.Lat.A, ...
    { Y_beta, Y_p, Y_r, Y_delta_alpha, Y_delta_r, ...
    L_beta, L_p, L_r, L_delta_alpha, L_delta_r, ...
    N_beta, N_T_beta, N_p, N_r, N_delta_alpha, N_delta_r, ...
    I_xx, I_yy, I_zz, I_xz, ...
    g, U_1, Theta_1, phi }, ...
    ...
    { C5.Y_beta, C5.Y_p, C5.Y_r, C5.Y_delta_alpha, C5.Y_delta_r, ...
    C5.L_beta, C5.L_p, C5.L_r, C5.L_delta_alpha, C5.L_delta_r, ...
    C5.N_beta, C5.N_T_beta, C5.N_p, C5.N_r, C5.N_delta_alpha, C5.N_delta_r, ...
    C5.I_xx, C5.I_yy, C5.I_zz, C5.I_xz, ...
    C5.g, C5.U_1, C5.Theta_1, C5.phi } ) ) );

Gp.Lat.B = vpa( simplify( subs( Gp.Lat.B, ...
    { Y_beta, Y_p, Y_r, Y_delta_alpha, Y_delta_r, ...
    L_beta, L_p, L_r, L_delta_alpha, L_delta_r, ...
    N_beta, N_T_beta, N_p, N_r, N_delta_alpha, N_delta_r, ...
    I_xx, I_yy, I_zz, I_xz, ...
    g, U_1, Theta_1, phi }, ...
    ...
    { C5.Y_beta, C5.Y_p, C5.Y_r, C5.Y_delta_alpha, C5.Y_delta_r, ...
    C5.L_beta, C5.L_p, C5.L_r, C5.L_delta_alpha, C5.L_delta_r, ...
    C5.N_beta, C5.N_T_beta, C5.N_p, C5.N_r, C5.N_delta_alpha, C5.N_delta_r, ...
    C5.I_xx, C5.I_yy, C5.I_zz, C5.I_xz, ...
    C5.g, C5.U_1, C5.Theta_1, C5.phi } ) ) );

Gp.Lat.C = vpa( simplify( subs( Gp.Lat.C, ...
    { Y_beta, Y_p, Y_r, Y_delta_alpha, Y_delta_r, ...
    L_beta, L_p, L_r, L_delta_alpha, L_delta_r, ...
    N_beta, N_T_beta, N_p, N_r, N_delta_alpha, N_delta_r, ...
    I_xx, I_yy, I_zz, I_xz, ...
    g, U_1, Theta_1, phi }, ...
    ...
    { C5.Y_beta, C5.Y_p, C5.Y_r, C5.Y_delta_alpha, C5.Y_delta_r, ...
    C5.L_beta, C5.L_p, C5.L_r, C5.L_delta_alpha, C5.L_delta_r, ...
    C5.N_beta, C5.N_T_beta, C5.N_p, C5.N_r, C5.N_delta_alpha, C5.N_delta_r, ...
    C5.I_xx, C5.I_yy, C5.I_zz, C5.I_xz, ...
    C5.g, C5.U_1, C5.Theta_1, C5.phi } ) ) );

Gp.Lat.D = vpa( simplify( subs( Gp.Lat.D, ...
    { Y_beta, Y_p, Y_r, Y_delta_alpha, Y_delta_r, ...
    L_beta, L_p, L_r, L_delta_alpha, L_delta_r, ...
    N_beta, N_T_beta, N_p, N_r, N_delta_alpha, N_delta_r, ...
    I_xx, I_yy, I_zz, I_xz, ...
    g, U_1, Theta_1, phi }, ...
    ...
    { C5.Y_beta, C5.Y_p, C5.Y_r, C5.Y_delta_alpha, C5.Y_delta_r, ...
    C5.L_beta, C5.L_p, C5.L_r, C5.L_delta_alpha, C5.L_delta_r, ...
    C5.N_beta, C5.N_T_beta, C5.N_p, C5.N_r, C5.N_delta_alpha, C5.N_delta_r, ...
    C5.I_xx, C5.I_yy, C5.I_zz, C5.I_xz, ...
    C5.g, C5.U_1, C5.Theta_1, C5.phi } ) ) );

%% Pole-Zero Locations
%%% Lateral
Gp.Lat.Poles = pole( sym2tf( Gp.Lat.subs ) );
Gp.Lat.Zeros = tzero( sym2tf( Gp.Lat.subs ) );

end
```

Published with MATLAB® R2020a

Symbolic Lateral State Space Definition: Lateral_SSREP_F2()

```
function [A, B, C, D, E] = Lateral_SSRep_F2( ...
    Y_beta, Y_p, Y_r, Y_delta_alpha, Y_delta_r, ...
    L_beta, L_p, L_r, L_delta_alpha, L_delta_r, ...
    N_beta, N_T_beta, N_p, N_r, N_delta_alpha, N_delta_r, ...
    I_xx, ~, I_zz, I_xz, ...
    g, U_1, Theta_1, phi)
% Aircraft State-Space Representation Creation
% Thomas Greenhill
% Blake Christierson
% June 11, 2020
%
% MAE 273B SQ2020
% Professor Francis Assadian

%%% A Matrix Creation
A = sym(zeros(5));

A(1,4) = 1;

A(2,5) = 1;

A(3,1) = g * phi * cos(Theta_1) / U_1;
A(3,3) = Y_beta / U_1;
A(3,4) = Y_p / U_1;
A(3,5) = -1 + Y_r / U_1;

A(4,3) = L_beta;
A(4,4) = L_p;
A(4,5) = L_r;

A(5,3) = N_beta + N_T_beta;
A(5,4) = N_p; % Term seems to be accidentally excluded in Yechout et. al.
A(5,5) = N_r;

%%% B Matrix Creation
B = sym(zeros(5,2));

B(3,1) = Y_delta_alpha / U_1;
B(3,2) = Y_delta_r / U_1;

B(4,1) = L_delta_alpha;
B(4,2) = L_delta_r;

B(5,1) = N_delta_alpha;
B(5,2) = N_delta_r;

%%% C Matrix Creation
C = sym(zeros(3,5));

C(1:3,1:3) = sym(eye(3));

%%% D Matrix Creation
D = sym(zeros(3,2));

%%% E Matrix Creation
E = sym(eye(5));

E(4,5) = -I_xz / I_xx;

E(5,4) = -I_xz / I_zz;

end
```

Published with MATLAB® R2020a

Youla Controller Generation: C5_Youla()

```
function [Gp] = C5_Youla(Gp, T1, T2)
% Expects Gp, z, wn to come in as type sym.

[Gp.Num, Gp.Den] = numden(Gp.subs);

Gp.P = Gp.Num;

[Gp.UL, Gp.H] = hermiteForm(Gp.P);

Gp.subs = sym2tf(Gp.subs);

Gp.Mp = smform(Gp.subs);
Gp.Mp = tf2sym(Gp.Mp);

Gp.UL = simplify(Gp.UL);

Mt=[T1 0;0 T2];

Y1 = T1/Gp.Mp(1,1);

Y2 = T2/Gp.Mp(2,2);

Gp.My = sym(zeros(2,3));

Gp.My(1,1) = Y1;

Gp.My(2,2) = Y2;


Gp.Gc = simplify(Gp.UR*(eye(2)-Gp.My*Gp.Mp)^-1*Gp.My*Gp.UL);

%% Open And Closed Loop Variables
I = tf(eye(3));

Gp.Gc = sym2tf(subs(Gp.Gc,'x','s'));

Gp.Ly = minreal(zpk(Gp.subs*Gp.Gc));
Gp.Ty = (I+Gp.Ly)^-1*Gp.Ly;
Gp.Sy = (I+Gp.Ly)^-1;

Gp.Lu = Gp.Gc*Gp.subs;
I = tf(eye(2));
Gp.Y = (I+Gp.Lu)^-1*Gp.Gc;
Gp.Tu = (I+Gp.Lu)^-1*Gp.Gc*Gp.subs;
Gp.Su = (I+Gp.Gc*Gp.subs)^-1;

Gp.Ty = minreal(zpk((Gp.Ty)))


Gp.Gc = zpk(Gp.Gc);
Gp.Y = zpk(Gp.Y);
Gp.Ty = zpk(Gp.Ty);
Gp.Sy = zpk(Gp.Sy);
Gp.Ly = zpk(Gp.Ly);
Gp.Tu = zpk(Gp.Tu);
Gp.Su = zpk(Gp.Su);
end
```

Published with MATLAB® R2020a

Hinf Controller Generation: Produce_Lateral_Hinf_2()()

```
function Hinf = Produce_Lateral_Hinf_2( Gp )
%% Hinf Synthesis
% Thomas Greenhill
% Blake Christierson
% June 11, 2020
%
% MAE 273B SQ2020
% Professor Francis Assadian

%% Notes:
%{
Modified and formatted from "InclassExample.m" provided by Professor
    Assadian on June 4, 2020
%}

%% Main Function
% Setup Variables
w = logspace(-2,2);

% Extract State Space Data From Plant
[Ag, Bg, Cg, Dg] = ssdata(Gp);

[Ag, Bg, Cg, Dg] = minreal(Ag, Bg, Cg, Dg);

Ag = Ag - 0.05*eye(size(Ag)); % Bilinear Transformation Approximation to
    % Avoid jw-axis Poles (Reversed After Controller is Computed)

%% Weight Construction
%%% Complimentary Sensitivity Weighting [3 x 3]
Wd1 = makeweight(0.5,10,50);
% Wd1 = makeweight(0.99,10,50); % Attempt for Full Plant, Not Rounded
Wd1tf = tf(Wd1);

Wd2 = makeweight(0.5,10,50);
%Wd2 = makeweight(0.99,10,50); % Attempt for Full Plant, Not Rounded
Wd2tf = tf(Wd2);

Wd3 = makeweight(0.5,10,50);
%Wd3 = makeweight(0.99,10,50);  % Attempt for Full Plant, Not Rounded
Wd3tf = tf(Wd3);

Hinf.Wd = [Wd1tf 0 0; 0 Wd2tf 0; 0 0 Wd3tf];

Wdss = ss(Hinf.Wd);
[Ad, Bd, Cd, Dd] = ssdata(Wdss);
[Ad, Bd, Cd, Dd] = minreal(Ad, Bd, Cd, Dd);

%%% Sensitivity Weighting [3 x 3]
Wp1 = makeweight(100,0.8,0.5);
Wp1tf = tf(Wp1);

Wp2 = makeweight(100,0.8,0.5);
Wp2tf = tf(Wp2);

Wp3 = makeweight(100,0.8,0.5);
Wp3tf = tf(Wp3);

Hinf.Wp = [Wp1tf 0 0;0 Wp2tf 0; 0 0 Wp3tf];

Wpss = ss(Hinf.Wp);
[Ap, Bp, Cp, Dp] = ssdata(Wpss);
[Ap, Bp, Cp, Dp] = minreal(Ap, Bp, Cp, Dp);

%%% Youla Weighting [2 x 2]
eps = 0.001;
Wynum = {eps 0; 0 eps};
Wyden = {1 1; 1 1};

Hinf.Wy = tf(Wynum, Wyden);

Wyss = ss(Hinf.Wy);
[Au, Bu, Cu, Du] = ssdata(Wyss);
[Au, Bu, Cu, Du] = minreal(Au, Bu, Cu, Du);

%% Compute Augmented Plant
[A, B1, B2, C1, C2, D11, D12, D21, D22] = ...
    augss(Ag, Bg, Cg, Dg,...
          Ap, Bp, Cp, Dp,...
          Au, Bu, Cu, Du,...
```

```
          Ad, Bd, Cd, Dd);

B = [B1, B2];
C = [C1; C2];
D = [D11, D12; D21, D22];

Gaug = ss(A, B, C, D);
Hinf.Gaug = minreal(Gaug);

%% Compute Hinf Controller
% [acp, bcp, ccp, dcp, acl, bcl, ccl, dcl] = ...
%      hinf(A, B1, B2, C1, C2, D11, D12, D21, D22);

%[Hinf.Gam, acp, bcp, ccp, dcp, acl, bcl, ccl, dcl] = ...
%    hinfopt(A, B1, B2, C1, C2, D11, D12, D21, D22);

[Kss, ~, Hinf.Gam] = hinfsyn( Hinf.Gaug, 3, 2 );
[acp, bcp, ccp, dcp] = ssdata( Kss );

acp = acp + 0.05*eye(size(acp)); % Removing Bilinear Transformation

Kss = ss(acp, bcp, ccp, dcp);
K = tf(Kss);
Hinf.K = minreal(K);

%% Compute Relevant Transfer Functions
Ly = Gp * Hinf.K;
Hinf.Ly = minreal(Ly);

Ty = feedback(Hinf.Ly, eye(3));
Hinf.Ty = minreal(Ty);

Sy = (eye(3) - Hinf.Ty);
Hinf.Sy = minreal(Sy  );

Y = Hinf.K * (eye(size(Hinf.Ly)) + Hinf.Ly)^(-1);
Hinf.Y = minreal(Y );

Hinf.Tu = minreal( Y * Gp );

Hinf.Su = eye(size(Hinf.Tu)) - Hinf.Tu;

[SVWpSy] = sigma(Hinf.Wp * Hinf.Sy, w);
LmWpSy = 20*log10(SVWpSy(1,:));

[SVWdTy]=sigma(Hinf.Wd * Hinf.Ty,w);
LmWdTy=20*log10(SVWdTy(1,:));

LmRp=20*log10(SVWpSy(1,:)+SVWdTy(1,:));

%% Plotting
figure
sigma(Hinf.Ly, w);
hold on
sigma(Hinf.Wp, w);
sigma(1 / Hinf.Wd, w);
title('Open Loop and Filter Transfer Function Singular Values');
h = legend({'$L_y$','$W_p$','$1/W_{\Delta}$'}, 'interpreter', 'latex');
set(h,'string',{'$L_y$','$W_p$','$1/W_{\Delta}$'});
grid
set(gca,'Color','w')

figure
sigma(Hinf.Sy,w);
hold on
sigma(Hinf.Ty,w);
hold on
sigma(Hinf.Wp,w);
hold on
sigma(1/ Hinf.Wd,w);
title('Closed Loop and Filter Transfer Function Singular Values');
h = legend({'$S_y$','$T_y$','$W_p$','$1/W_{\Delta}$'}, 'interpreter', 'latex');
set(h,'string',{'$S_y$','$T_y$','$W_p$','$1/W_{\Delta}$'});
grid
set(gca,'Color','w')

figure
sigma(Hinf.Y,w)
hold on
sigma(Gp,w)
hold on
```

```
sigma(Hinf.K,w)
title('Actuator, Plant, and Controller Singular Values','interpreter', 'tex');
h = legend({'$Y$','$G_p$','$K$'}, 'interpreter', 'latex');
set(h,'string',{'$Y$','$G_p$','$K$'});
grid
set(gca,'Color','w')

figure
subplot(2,1,1)
[y,t] = step(Hinf.Ty);
k=1:length(t);
plot(t,y(k,1,1),'-');
title('First Input Step Output Responses','interpreter', 'tex')
hold on
plot(t,y(k,2,1),'--')
plot(t,y(k,3,1),'-.')
legend('$\phi(t)$','$\psi(t)$','$\beta(t)$');
xlabel('Time (s)','interpreter', 'tex')
grid
set(gca,'Color','w')

subplot(2,1,2)
[y,t] = step(Hinf.Ty);
k=1:length(t);
plot(t,y(k,1,2),'-');
title('Second Input Step Output Responses','interpreter', 'tex')
hold on
plot(t,y(k,2,2),'--')
plot(t,y(k,3,2),'-.')
legend('$\phi(t)$','$\psi(t)$','$\beta(t)$');
xlabel('Time (s)','interpreter', 'tex')
grid
set(gca,'Color','w')

figure
subplot(2,1,1)
t=0:0.001:0.2;
[ya] =step(Y,t);
k=1:length(t);
plot(t,ya(k,1,1),'-');
title('First Input Step Actuator Responses','interpreter', 'tex')
hold on
plot(t,ya(k,2,1),'-')
h = legend('$\delta_{a}(t)$','$\delta_{r}(t)$');
set(h,'string',{'$\delta_{a}(t)$','$\delta_{r}(t)$'});
xlabel('Time (s)','interpreter', 'tex')
grid
set(gca,'Color','w')

subplot(2,1,2)
t=0:0.001:0.2;
[ya] =step(Y,t);
k=1:length(t);
plot(t,ya(k,1,2),'-');
title('Second Input Step Actuator Responses','interpreter', 'tex')
hold on
plot(t,ya(k,2,2),'--')
h = legend('$\delta_{a}(t)$','$\delta_{r}(t)$');
set(h,'string',{'$\delta_{a}(t)$','$\delta_{r}(t)$'});
xlabel('Time (s)','interpreter', 'tex')
grid
set(gca,'Color','w')

figure
semilogx(w,LmWpSy,'r',w,LmWdTy,'g',w,LmRp,'b')
grid;
title('Nominal Performance, Robust Stability, and Robust Performance','interpreter', 'tex')
h =
legend('$\sigma(W_{p}*S_{y})$','$\sigma(W_{\Delta}*T_{y})$','$\sigma(W_{p}*S_{y})+\sigma(W_{\Delt
a}*T_{y})$');
set(h,'string',{'$\sigma(W_{p}*S_{y})$','$\sigma(W_{\Delta}*T_{y})$','$\sigma(W_{p}*S_{y})+\sigma
(W_{\Delta}*T_{y})$'});
xlabel('Frequency (rad/sec)','interpreter', 'tex');
ylabel('Gain (dB)','interpreter', 'tex');
grid
set(gca,'Color','w')
```

Published with MATLAB® R2020a

Plant Parameter Variation: Robust_Parameter_Study()

```
function Hinf = Robust_Parameter_Study( HinfOld, Gp )
%% Hinf Parameter Variation Study
% Thomas Greenhill
% Blake Christierson
% June 11, 2020
%
% MAE 273B SQ2020
% Professor Francis Assadian

% Setup Variables
w = logspace(-2,2);

%% Compute Relevant Transfer Functions
Hinf = HinfOld;

Ly = Gp * Hinf.K;
Hinf.Ly = minreal(Ly);

Ty = feedback(Hinf.Ly, eye(3));
Hinf.Ty = minreal(Ty);

Sy = (eye(3) - Hinf.Ty);
Hinf.Sy = minreal(Sy  );

Y = Hinf.K * (eye(size(Hinf.Ly)) + Hinf.Ly)^(-1);
Hinf.Y = minreal(Y );

Hinf.Tu = minreal( Y * Gp );

Hinf.Su = eye(size(Hinf.Tu)) - Hinf.Tu;

% PrincipleGainAnalysis( Gp, Hinf.K, Hinf.Ly, ...
%    Hinf.Ty, Hinf.Sy, Hinf.Su, Hinf.Y, logspace(-2, 2) )

[SVWpSy] = sigma(Hinf.Wp * Hinf.Sy, w);
LmWpSy = 20*log10(SVWpSy(1,:));

[SVWdTy]=sigma(Hinf.Wd * Hinf.Ty,w);
LmWdTy=20*log10(SVWdTy(1,:));

LmRp=20*log10(SVWpSy(1,:)+SVWdTy(1,:));

%% Plotting
figure
sigma(Hinf.Ly, w);
hold on
sigma(Hinf.Wp, w);
sigma(1 / Hinf.Wd, w);
title('Open Loop and Filter Transfer Function Singular Values');
h = legend({'$L_y$','$W_p$','$1/W_{\Delta}$'}, 'interpreter', 'latex');
set(h,'string',{'$L_y$','$W_p$','$1/W_{\Delta}$'});
grid
set(gca,'Color','w')


figure
sigma(Hinf.Sy,w);
hold on
sigma(Hinf.Ty,w);
hold on
sigma(Hinf.Wp,w);
hold on
sigma(1/ Hinf.Wd,w);
title('Closed Loop and Filter Transfer Function Singular Values');
h = legend({'$S_y$','$T_y$','$W_p$','$1/W_{\Delta}$'}, 'interpreter', 'latex');
set(h,'string',{'$S_y$','$T_y$','$W_p$','$1/W_{\Delta}$'});
grid
set(gca,'Color','w')


figure
sigma(Hinf.Y,w)
hold on
sigma(Gp,w)
hold on
sigma(Hinf.K,w)
title('Actuator, Plant, and Controller Singular Values','interpreter', 'tex');
h = legend({'$Y$','$G_p$','$K$'}, 'interpreter', 'latex');
set(h,'string',{'$Y$','$G_p$','$K$'});
```

```
grid
set(gca,'Color','w')


figure
subplot(2,1,1)
[y,t] = step(Hinf.Ty);
k=1:length(t);
plot(t,y(k,1,1),'-');
title('First Input Step Output Responses','interpreter', 'tex')
hold on
plot(t,y(k,2,1),'--')
plot(t,y(k,3,1),'-.')
legend('$\phi(t)$','$\psi(t)$','$\beta(t)$');
xlabel('Time (s)','interpreter', 'tex')
grid
set(gca,'Color','w')

subplot(2,1,2)
[y,t] = step(Hinf.Ty);
k=1:length(t);
plot(t,y(k,1,2),'-');
title('Second Input Step Output Responses','interpreter', 'tex')
hold on
plot(t,y(k,2,2),'--')
plot(t,y(k,3,2),'-.')
legend('$\phi(t)$','$\psi(t)$','$\beta(t)$');
xlabel('Time (s)','interpreter', 'tex')
grid
set(gca,'Color','w')


figure
subplot(2,1,1)
t=0:0.001:0.2;
[ya] =step(Y,t);
k=1:length(t);
plot(t,ya(k,1,1),'-');
title('First Input Step Actuator Responses','interpreter', 'tex')
hold on
plot(t,ya(k,2,1),'-')
h = legend('$\delta_{a}(t)$','$\delta_{r}(t)$');
set(h,'string',{'$\delta_{a}(t)$','$\delta_{r}(t)$'});
xlabel('Time (s)','interpreter', 'tex')
grid
set(gca,'Color','w')

subplot(2,1,2)
t=0:0.001:0.2;
[ya] =step(Y,t);
k=1:length(t);
plot(t,ya(k,1,2),'-');
title('Second Input Step Actuator Responses','interpreter', 'tex')
hold on
plot(t,ya(k,2,2),'--')
h = legend('$\delta_{a}(t)$','$\delta_{r}(t)$');
set(h,'string',{'$\delta_{a}(t)$','$\delta_{r}(t)$'});
xlabel('Time (s)','interpreter', 'tex')
grid
set(gca,'Color','w')

figure
semilogx(w,LmWpSy,'r',w,LmWdTy,'g',w,LmRp,'b')
grid;
title('Nominal Performance, Robust Stability, and Robust Performance','interpreter', 'tex')
h =
legend('$\sigma(W_{p}*S_{y})$','$\sigma(W_{\Delta}*T_{y})$','$\sigma(W_{p}*S_{y})+\sigma(W_{\Delt
a}*T_{y})$');
set(h,'string',{'$\sigma(W_{p}*S_{y})$','$\sigma(W_{\Delta}*T_{y})$','$\sigma(W_{p}*S_{y})+\sigma
(W_{\Delta}*T_{y})$'});
xlabel('Frequency (rad/sec)','interpreter', 'tex');
ylabel('Gain (dB)','interpreter', 'tex');
grid
set(gca,'Color','w')

end
```

Published with MATLAB® R2020a

Project Outline

Group members: Blake Christierson, Thomas Greenhill

Plant: C-5A Galaxy

Background:

The C-5A is known for having high pilot workload due to its pitch characteristics along with highly coupled roll & yaw modes with high levels of adverse yaw. It is also unstable in its phugoid pitch mode and has poor handling qualities (level 3 handling quality) due to its phugoid mode damping properties. It is also unstable in its spiral mode.

Performance Criteria:

- Roll/Yaw decoupling
- Phugoid mode stability
- Spiral mode stability

Robustness criteria

- Disturbances
- Increased payloads at different altitudes.

Option 2: Learjet C-21

Background:

The Learjet C-21 is known for having high pilot workload due to its poor Dutch roll mode damping. Its other handling characteristics are relatively good, but its Dutch roll mode puts it in the worst category for handling (level 3), so improving its dutch roll mode would mean greatly improving its handling ranking. It also has an unstable phugoid pitch mode.

Performance Criteria:

- Dutch roll stability
- Phugoid mode stability

Robustness criteria

- Disturbances
- Increased fuel levels & different air densities.

Option 3: F-4 Phantom

Background:

The F-4 Phantom is famous for its roll/pitch coupling. This undesirable characteristic causes the aircraft to pitch downwards when initiating a roll due to inertial effects. It also has relatively large levels of roll/yaw coupling for a fighter jet, which is undesirable. This is in my opinion the most interesting plant, but including inertial effects into the transfer function matrices is actually nontrivial due to approximations made to linearize the equations of motion. The equations of motion are linearized by using a small angle approximation, and as a result, the terms that cause this inertial coupling are neglected in the final linearized equations of motion. This can be fixed by assuming a numerical value for pitch/roll/yaw rates while performing the linearization, to keep the terms that cause this inertial effect. However, this involves assuming rates *before we know them* which is somewhat dangerous… Otherwise we would have to solve the equations of motion using PDEs which is out of the scope of this project.

I would be interested in discussing whether this is still a viable option or not.

Performance Criteria:

- Roll/pitch decoupling
- Roll/yaw decoupling

Robustness criteria

- Disturbances
- Weapons payloads & different air densities.

We will continue to explore these three options and let the professor know which option we finally settle for.