**Lab 1**

2 October 2024

Authors: C1C Lino De Ros & C1C Connor Emmons



Professor: Dr. Brown

Engr 443 T2

**Documentation:** Used course notes and previous lab reports for formatting purposes. ChatGPT used to validate feedback gain and observer gain values and to help write the conclusion, though final edits were made by us. All other sections of the report were generated solely by us. No unauthorized resources used. Git repository can be found at <https://github.com/Connor-Lemons/Advanced-Controls-Labs>.

**Objective**

The objective of this lab is to investigate the behavior of a continuous-time full-state feedback control system, first using full-state measurement, and then using only partial-state measurement and a full-state observer. A state-feedback controller is designed to control the “short-period” pitch rate behavior of a Boeing 787 Dreamliner subjected to a wind gust.

The “short-period” pitch rate behavior of a Boeing 787 is represented by the following system model.



where 

and:

for full-state measurements

C = 

and for partial-state measurements

C = 

**Approach**

The plant model of the “short-period” pitch rate behavior is modeled into Simulink. In full-state feedback control using full-state measurement, a continuous-time full-state feedback controller is implemented so that the controlled system has a damping ratio of and a natural frequency of . The controlled system has zero initial conditions and a pulse input (representing a wind gust) of magnitude 1 and duration 0.5 seconds. Finally, the history of the states during and after the wind gust is plotted against the same states generated by MatLab’s lsim() command to validate the model.

In full-state feedback control using partial-state measurement and a full-state observer, a slightly different approach is taken to control the system to meet the given specifications. The feedback gain matrix from the full-state controller in the first scenario is used in conjunction with the full-state observer to control the system to meet the given specifications. The purpose of the observer is to provide a surrogate model of the system from which control can be derived by observing the states. To generate the most accurate observer, its poles are placed at three different locations (both poles at , , and ) to determine which configuration allows the observer to eliminate the difference between its own output and the system’s the fastest. Again, the compensated system has zero initial conditions and a pulse input of magnitude 1 and duration 0.5 seconds to represent the wind gust. The initial conditions of the observer are .

**Assumptions**

One important assumption for this lab is that the plant matrix is controllable for the full-state feedback implementation, and observable for the partial-state feedback with the full-state observer implementation. This is verified by the use of Ackermann’s method when solving for the feedback gain matrix and the observer gain matrix, where the controllability and observability matrices are calculated and are shown to be non-singular for the calculation to work.

**Math Technique**

**Task 1:**

For full-state feedback, the control law is:

Substituting this into the state-space equation gives:

The desired characteristic equation for the closed-loop system is:

We want to design 𝐾 so that the closed-loop system achieves the desired performance of and . Finding the roots gives the desired pole placements:

We want to find the state feedback gain such that the eigenvalues of match the desired characteristic equation:

Using MatLab’s “acker” function shown in Appendix B, we find that the state feedback gain matrix *K* is:

**Task 2:**

The full-state feedback observer is described by the following equation:

where is the estimated state, is the estimated output, is the actual state, and is the actual output. is the observer gain matrix, which we will design to place two identical observer poles at , and for each of the three cases.

The error dynamics for the observer are:

where is given as for partial-state measurements.

To solve for the gain matrix 𝐿, we use the characteristic equation for the observer which is:

Using the desired characteristic equations given by the three sets of observer poles, we use MatLab’s “acker” function shown in Appendix B to solve for the gain matrix :

**Theoretical Predictions**

The generic form of Ackerman’s method, which was used to find the values for the feedback gain matrix is shown for a matrix.

1. Calculate and where is the controllability matrix
2. Calculate the desired closed-loop characteristic equation from the given poles
3. Find the state transition matrix,
4. Find using

Ackerman’s method changes slightly when finding the observability gain matrix . The derivation for this matrix is shown below.

1. Calculate and where is the observability matrix

and

1. Calculate the desired closed-loop characteristic equation from the given observer poles
2. Find the state transition matrix,
3. Find using

It is predicted that as the poles of the observer move farther into the left-hand plane of the real axis, the observer will drive its outputs to match the actual outputs faster.

**Experimental Results/Analysis**

When comparing the results of the Simulink model to the output from the lsim() function in MatLab, the greatest observed error in either of the two outputs was on the order of . This error is small enough to be considered negligible and conclude that the Simulink model is validated for the given system and the given control. Figure 1 shows the two states as given by the Simulink model and those same two states as given by lsim(). Note that because of the small error (on the order of or smaller), the graphs of the states produced by lsim() (the validation graphs as shown in the model) completely cover the graphs of the states produced by the Simulink model at any reasonable scale.

A graph of a function

Description automatically generated

Figure 1. Input and States vs. Time

When comparing the effects of different observer poles, the model shows that as the poles of the observer move further into the left-hand plane (i.e., the observer responds faster to any difference between the actual output and the observer’s output), the observer’s outputs match the actual outputs sooner. Figure 2 shows both the system’s output and the observer output when the poles of the observer are placed at –2. Note that the observer’s output takes over one second to match the system’s output. When the poles are moved to –4 (as shown in Figure 3), the observer performs much better, matching the system’s output in less than half a second. As expected, when the poles are moved to –8 (in Figure 4), the observer matches the system’s output even faster, though this difference is less noticeable.

A graph of a function

Description automatically generated

Figure 2. System Output and Observer Output vs. Time for Pole

A graph of a function

Description automatically generated

Figure 3. System Output and Observer Output vs. Time for Pole

A graph of a function

Description automatically generated

Figure 4. System Output and Observer Output vs. Time for Pole

**Conclusions and Recommendations**

The objective of this project was to design and validate a full-state feedback controller and a full-state observer for a system representing the "short-period" pitch rate behavior of a Boeing 787 aircraft. Using MatLab, we have implemented a continuous-time controller to achieve the specified damping ratio and natural frequency of and validated the design using the lsim() function. The compensator output and the simulation results matched to a precision of confirming the accuracy of the control system. Furthermore, we have developed a full-state observer using partial-state measurements with the observer poles placed at varying locations in the left-hand plane. The results demonstrated that as the observer poles were moved further left (i.e., farther from the imaginary axis), the observer became more accurate, converging to the true system states more quickly.

Although the controller performance and observer design meet the objectives, further improvements could include investigating the effects of disturbances beyond the pulse input or a more comprehensive review of the controlled system’s specifications, to include steady-state error.

In conclusion, the compensator and observer designs effectively control the system and accurately estimate states, with performance improving as the observer poles are moved farther left. These findings confirm the utility of full-state feedback and observer-based control in managing the dynamic behavior of aerospace systems.

Appendix A: Simulink Screen Shots

Appendix B: Task Live Scripts