Introduction

The paper “Turbofan Engine Control Design Using Robust Multivariable Control Technologies” (IEEE Trans. on Control Systems Technology, November, 2000 by D.K. Frederick, S. Garg, and S. Adibhatla) describes the design of a multivariable control system for a turbofan aircraft engine. Although the engine is a nonlinear system, linear models can be obtained at various operating points and the engine can be controlled near these operating points by a linear control system. The paper gives a state-space model at one operating condition, and the purpose of this project is to design a multivariable digital tracking system for that one linear plant model.

The input and output variables of the engine are shown below:

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_1$ = fuel flow rate</td>
<td>$y_1$ = percent corrected fan speed (PCN2R)</td>
</tr>
<tr>
<td>$w_2$ = nozzle area</td>
<td>$y_2$ = core engine pressure ratio (CEPR)</td>
</tr>
<tr>
<td>$w_3$ = by-pass duct area</td>
<td>$y_3$ = liner engine pressure ratio (LEPR)</td>
</tr>
</tbody>
</table>

A 3rd-order state-space model for a certain turbofan engine is given in the Appendix of the paper (page 969) in the following form. Note that the vector of input signals is called $w$, the vector of output signals is called $y$, and the vector of state variables is called $z$.

$$\dot{z} = A_{eng}z + B_{eng}w$$
$$y = C_{eng}z + D_{eng}w$$

where

$$A_{eng} = \begin{bmatrix} -4.1476 & 1.4108 & 0.0633 \\ 0.2975 & -3.1244 & 0.0623 \\ -0.0429 & -0.1729 & -0.1325 \end{bmatrix}, \\ B_{eng} = \begin{bmatrix} 0.2491 & 0.0969 & -0.0112 \\ 0.2336 & 0.0335 & 0.0047 \\ 0.0624 & 0 & 0 \end{bmatrix}$$

$$C_{eng} = \begin{bmatrix} 8.7379 & 0 & 0 \\ -3.3033 & 3.8052 & 0.0542 \\ 2.1940 & -2.5749 & -0.0295 \end{bmatrix}, \\ D_{eng} = \begin{bmatrix} 0 & 0 & 0 \\ 0.2383 & -0.2748 & 0.0224 \\ -0.1455 & 0.0580 & -0.2293 \end{bmatrix}$$

Plant Model

The engine model given in the paper was scaled in such a way that the commands signals are simply unit steps. However, the model given in the appendix does not include the actuators, and these must be included to obtain a complete plant model. The second column on page 962 of the paper describes the actuator models. Each engine input, $w_i$, is obtained as the output of a first-order actuator model; each actuator model requires one state variable as shown in Fig. 1. The actuator inputs, $u_1$, $u_2$, $u_3$, are the inputs to the complete plant model.

The actuator state variables are numbered beginning with $x_4$. They are concatenated with the three state variables in $z$ from the model in (1). The actuator models must be combined with the state-space model (1) to obtain a complete 6th-order state-space model:

$$\dot{x} = Ax + Bu$$
$$y = Cx$$
where the state vector, input vector, and output vector are given by:

\[
x = \begin{bmatrix} z \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}.
\]

1. The first step in this project is to derive numerical values for the complete \((A, B, C)\) state-space model: \(\dot{x} = Ax + Bu, y = Cx\). Show your work. (The tasks for this design project are written in boldface font.)

The following sentence, found on page 962 of the paper, describes the design goals. In addition to providing a stable closed-loop system, the primary objective of the engine controller was to give good decoupled command tracking of the demand values of \(y_1\) and \(y_2\), with good regulation of \(y_3\) about its nominal value. In the linearized model, the nominal value of \(y_3\) is zero. The desired settling time for this control system is 2 seconds for step input commands for either \(y_1\) or \(y_2\), with \(y_3\) and \(y_2\), or \(y_3\) and \(y_1\), held near zero. The step responses for the control system designed in the journal paper are shown below.

![Figure 1: Actuator models for the three plant inputs](image1)

![Figure 2: Left figure: unit step response for \(y_1\) with \(y_2\) and \(y_3\) held near zero. Right figure: unit step response for \(y_2\) with \(y_1\) and \(y_3\) held near zero. These figures are taken from [1].](image2)

**Tracking System Design Using Pole Placement and an Observer**

The tracking system is to be designed for step inputs, so an eigenvalue of unity is needed for the additional dynamics. Because the plant has multiple inputs (three), the additional dynamics must be replicated into three parallel systems, each with an eigenvalue of unity. That is, we choose the...
following additional dynamics matrices:

$$
\Phi_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \Gamma_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
$$

2. Choose a sampling interval and two different sets of closed-loop pole locations and calculate state-feedback tracking systems to achieve a 2-second settling time. Use tsd to calculate the tracking system gain matrices, and specify the place algorithm. For the first set of pole locations, use sufficiently damped plant poles and Bessel poles. For the second set of pole locations, use sufficiently damped plant poles, stable plant zeros, and Bessel poles. Recall that the zeros of the plant are found using the Matlab command \texttt{tzero(A,B,C,D)}. For each state-feedback tracking system: (a) print out the gain matrices \(K_1\) and \(K_2\), (b) plot \(y_1\) and \(y_2\) on the same graph using option 7 of \texttt{tssfp}, and (c) plot \(y_1\) and \(y_3\) on another graph. The goal is for \(y_2\) and \(y_3\) to remain near zero, as shown in Fig. 2 on the previous page. Compare robustness bounds and plots for the two tracking systems and discuss the good and bad aspects of each system.

Note that in order to simulate a full-state feedback tracking system using \texttt{tssf}, the following variables must be defined: \(x_0=\text{zeros}(6,1)\) and a vector-valued reference input signal called \texttt{ref}. \texttt{ref} consists of three signals, which are the reference inputs for \(y_1\), \(y_2\), and \(y_3\). Such a signal can be created using the \texttt{siggen} function. First, you must choose a value for a variable called \texttt{ftime}, which is the final time of the simulation. A rule of thumb is to choose \(ftime=2\times Ts\). To specify a unit step signal in \texttt{siggen}, the vector of numbers \([1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]\) is used. To specify a vector-valued signal consisting of a unit step as the first signal and zeros for the other two signals, the command is

\[
>> \text{ref} = \text{siggen}([1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0], ftime, T)
\]

where \(T\) is the sampling interval. To specify a step as the second signal with the first and third signals zero, use \([0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]\) in the \texttt{siggen} function.

3. Repeat Step 2 but specify the \texttt{rfbg} algorithm when using \texttt{tsd} to calculate the state-feedback tracking system gains. In addition to comparing the two tracking systems designed in this part with each other, also compare them with the tracking systems designed in Step 2. Use only the best state-feedback gains in the next two steps.

4. Calculate observer gain matrices using the \texttt{place} algorithm for two different choices of observer pole locations. The first set of observer poles is all Bessel poles with \(T_{so} = T_s/5\). For the second set of observer pole locations, use stable plant zeros and Bessel poles scaled by the same \(T_{so}\). Print out the two observer gain matrices. Use \texttt{rb_tso} to find the robustness bounds of the observer-based tracking systems using the two observer gain matrices calculated in this step with the best state feedback gains from the previous steps.

5. Use the two sets of observer pole locations from the previous step but calculate the observer gain matrices using \texttt{obg_ts}. Compare the robustness bounds of the resulting observer-based tracking systems with those from the previous step.

6. Choose the best observer-based tracking system as your final control system and use \texttt{tsob} and \texttt{tsobp} to obtain step-response plots for commands in \(y_1\) and \(y_2\), as done in item 2 above. Compare these with the (solid line) plots shown in Fig. 2.
7. Give a block diagram for the complete system including the plant, controller, D/A and A/D converters. Indicate which part of the block diagram is inside the digital control computer.

Reference