A Design Study Approach to Classical Control

Randal W. Beard   Timothy W. McLain
Brigham Young University

Updated: December 28, 2020

Homework F.11

The objective of this problem is to implement a state feedback controller for the VTOL system. Start with the simulation files developed in Homework F.10.

(a) Using the values for ω_{n_h}, ζ_h, ω_{n_z}, and ζ_z from Homework F.8, choose the closed-loop pole locations to ensure that the longitudinal poles have damping ratios greater than ζ_h and natural frequencies greater than ω_{n_h}. Similarly choose the closed-loop poles for the lateral dynamics to ensure that their damping ratios are greater than ζ_z and natural frequencies greater than ω_{n_z}

(b) Add the state space matrices A, B, C, D derived in Homework F.6 to your param file.

(c) Verify that the state space system is controllable by checking that $\text{rank}(\mathcal{C}_{A,B}) = n$.

(d) Find the feedback gain K such that the eigenvalues of $(A - BK)$ are equal to desired closed loop poles. Find the reference gains k_{r_h} and k_{r_z} so that the DC-gain from h_r to h is equal to one, and the DC-gain from z_r to z is equal to one.

(e) Implement the state feedback scheme and tune the closed-loop poles to get acceptable response. What changes would you make to your closed-loop pole locations to decrease the rise time of the system? How would you change them to lower the overshoot in response to a step command?

Solution

From HW F.6, the state space equations for the lateral VTOL dynamics are given by

$$\dot{x}_{lat} = \begin{pmatrix} 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 1.0000 \\ 0 & -9.8100 & -0.0667 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} x_{lat} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 20.3252 \end{pmatrix} u$$

$$y = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} x_{lat}.$$

Step 1. The controllability matrix is therefore

$$\mathcal{C}_{A,B} = [B, AB, A^2B, A^3B] = \begin{pmatrix} 0 & 0 & 0 & -199.3902 \\ 0 & 20.3252 & 0 & 0 \\ 0 & 0 & -199.3902 & 13.2927 \\ 20.3252 & 0 & 0 & 0 \end{pmatrix}.$$

The determinant is $\det(\mathcal{C}_{A,B}) = -1.63e+7 \neq 0$, therefore the system is controllable.

Step 2. The open loop characteristic polynomial is

$$\Delta_{ol}(s) = \det(sI - A) = s^4 + 0.0667s^3$$

which implies that

$$\mathbf{a}_A = (0.0667, 0, 0, 0)$$

$$\mathcal{A}_A = \begin{pmatrix} 1 & 0.0667 & 0 & 0 \\ 0 & 1 & 0.0667 & 0 \\ 0 & 0 & 1 & 0.0667 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Step 3. When $\omega_\theta = 2.75$, $\zeta_\theta = 0.707$, $\omega_z = 0.275$, $\zeta_z = 0.707$, the desired closed loop polynomial is

$$\begin{aligned}\Delta_{cl}^d(s) &= (s^2 + 2\zeta_\theta\omega_{n_\theta}s + \omega_{n_\theta}^2)(s^2 + 2\zeta_z\omega_{n_z}s + \omega_{n_z}^2) \\ &= s^4 + 4.2773s^3 + 9.1502s^2 + 3.2347s + 0.5719\end{aligned}$$

which implies that

$$\boldsymbol{\alpha} = (4.2773, 9.1502, 3.2347, 0.5719).$$

Step 4. The gains are therefore given as

$$\begin{aligned} K_{lat} &= (\boldsymbol{\alpha} - \mathbf{a}_A)\mathcal{A}_A^{-1}\mathcal{C}_{A,B}^{-1} \\ &= \begin{pmatrix} -0.0029 & 0.4364 & -0.0133 & 0.2072 \end{pmatrix} \end{aligned}$$

To compute the reference gain $k_r = -1/C_{out}(A - BK)^{-1}B$, we need to use C_{out}, the output matrix matching the reference input, which since the desired reference input is z_r and since $x = (z, \theta, \dot{z}, \dot{\theta})^\top$, we have $C_{out} = (1, 0, 0, 0)$, which gives

$$\begin{aligned} k_{r_{lat}} &= \frac{-1}{C_{out}(A - BK)^{-1}B} \\ &= -0.0029. \end{aligned}$$

From HW F.6, the state space equations for the longitudinal VTOL dynamics are given by

$$\begin{aligned} \dot{x}_{lon} &= \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x_{lon} + \begin{pmatrix} 0 \\ 0.6667 \end{pmatrix} u \\ y &= \begin{pmatrix} 1 & 0 \end{pmatrix} x_{lon}. \end{aligned}$$

Step 1. The controllability matrix is therefore

$$\mathcal{C}_{A,B} = [B, AB] = \begin{pmatrix} 0 & 0.6667 \\ 0.6667 & 0 \end{pmatrix}.$$

The determinant is $\det(\mathcal{C}_{A,B}) = -0.444 \neq 0$, therefore the system is controllable.

Step 2. The open loop characteristic polynomial is

$$\Delta_{ol}(s) = \det(sI - A) = s^2$$

which implies that

$$\begin{aligned} \mathbf{a}_A &= (1, 0) \\ \mathcal{A}_A &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \end{aligned}$$

Step 3. When $\omega_h = 0.275$, and $\zeta_h = 0.707$ the desired closed loop polynomial is

$$\begin{aligned}\Delta_{cl}^d(s) &= s^2 + 2\zeta_z\omega_{n_z}s + \omega_{n_z}^2 \\ &= s^2 + 0.3889s + 0.0756\end{aligned}$$

which implies that

$$\boldsymbol{\alpha} = (0.3889, 0.0756).$$

Step 4. The gains are therefore given as

$$\begin{aligned}K_{lon} &= (\boldsymbol{\alpha} - \mathbf{a}_A)\mathcal{A}_A^{-1}\mathcal{C}_{A,B}^{-1} \\ &= \begin{pmatrix}0.1134 & 0.5833\end{pmatrix}\end{aligned}$$

The reference gain $k_r = -1/C(A-BK)^{-1}B$, is

$$\begin{aligned}k_{r_{lat}} &= \frac{-1}{C(A-BK)^{-1}B} \\ &= 0.1134.\end{aligned}$$

Alternatively, we could have used the following Matlab script

```
# VTOL Parameter File
import numpy as np
import control as cnt
import sys
sys.path.append('..')  # add parent directory
import VTOLParam as P


####################################################
#                  State Space
####################################################
# tuning parameters
# tuning parameters
tr_h    = 2.0
wn_h    = 2.2/tr_h
zeta_h  = 0.707

tr_z    = 2.0
wn_z    = 2.2/tr_z
```

```
zeta_z  = 0.707

M = 10.0
tr_th   = tr_z/M
wn_th   = 2.2/tr_th
zeta_th = 0.707

# State Space Equations
A_lon = np.array([[0.0, 1.0],
                  [0.0, 0.0]])
B_lon = np.array([[0.0],
                  [1.0/(P.mc+2.0*P.mr)]])
C_lon = np.array([[1.0, 0.0]])
A_lat = np.array([[0.0, 0.0, 1.0, 0.0],
                  [0.0, 0.0, 0.0, 1.0],
                  [0.0, -P.Fe/(P.mc+2.0*P.mr), -(P.mu/(P.mc+2.0*P.mr)), 0.0],
                  [0.0, 0.0, 0.0, 0.0]])
B_lat = np.array([[0.0],
                  [0.0],
                  [0.0],
                  [1.0/(P.Jc+2*P.mr*P.d**2)]])
C_lat = np.array([[1.0, 0.0, 0.0, 0.0],
                  [0.0, 1.0, 0.0, 0.0]])

# gain calculation
des_char_poly_lon = [1.0, 2.0*zeta_h*wn_h, wn_h**2]
des_poles_lon = np.roots(des_char_poly_lon)

des_char_poly_lat = np.convolve([1.0, 2.0*zeta_z*wn_z, wn_z**2],
                                [1.0, 2.0*zeta_th*wn_th, wn_th**2])
des_poles_lat = np.roots(des_char_poly_lat)


# Compute the gains if the system is controllable
if np.linalg.matrix_rank(cnt.ctrb(A_lon, B_lon)) != 2:
    print("The longitudinal system is not controllable")
else:
    K_lon = cnt.acker(A_lon, B_lon, des_poles_lon)
    kr_lon = -1.0/(C_lon @ np.linalg.inv(A_lon-B_lon @ K_lon) @ B_lon)

if np.linalg.matrix_rank(cnt.ctrb(A_lat, B_lat)) != 4:
    print("The lateral system is not controllable")
else:
    K_lat = cnt.acker(A_lat, B_lat, des_poles_lat)
    Cr = np.array([[1.0, 0.0, 0.0, 0.0]])
```

```
    kr_lat = -1.0/(Cr @ np.linalg.inv(A_lat-B_lat @ K_lat) @ B_lat)

print('K_lon: ', K_lon)
print('kr_lon: ', kr_lon)
print('K_lat: ', K_lat)
print('kr_lat: ', kr_lat)
```

The Matlab code for the controller is given by

```
import numpy as np
import VTOLParam as P
import VTOLParamHW11 as P11

class VTOLController:
    def __init__(self):
        self.limit = P.fmax

    def update(self, r, x):
        z_r = r.item(0)
        h_r = r.item(1)
        z = x.item(0)
        h = x.item(1)
        theta = x.item(2)
        # Construct the states
        x_lon = np.array([[x.item(1)], [x.item(4)]])
        x_lat = np.array([[x.item(0)], [x.item(2)], [x.item(3)], [x.item(5)]])
        # Compute the state feedback controllers
        F_tilde = -P11.K_lon @ x_lon + P11.kr_lon*h_r
        F = P.Fe/np.cos(theta) + F_tilde.item(0)
        tau = -P11.K_lat @ x_lat + P11.kr_lat*z_r
        return np.array([[F], [tau.item(0)]])

    def saturate(self,u):
        if abs(u) > self.limit:
            u = self.limit*np.sign(u)
        return u
```