# Homework G.16 - Solution

From HW G.7, the state space equations for the lateral VTOL dynamics are given by

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

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

**Step 1.** The controllability matrix is therefore

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

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

**Step 2.** The open loop characteristic polynomial

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

which implies that

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

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

**Step 3.** When $\omega_\theta = 13.38$, $\zeta_\theta = 0.707$, $\omega_z = 0.9905$, $\zeta_z = 0.707$, the desired closed loop polynomial

$$\begin{aligned} \Delta_{cl}^d(s) &= (s^2 + 2\zeta_\theta \omega_{n_\theta} s + \omega_{n_\theta}^2)(s^2 + 2\zeta_z \omega_{n_z} s + \omega_{n_z}^2) \\ &= s^4 + 20.3203 s^3 + 206.5119 s^2 + 269.3089 s + 175.6470 \end{aligned}$$

which implies that

$$\boldsymbol{\alpha} = (20.3203, 206.5119, 269.3089, 175.6470).$$

**Step 4.** The gains are therefore given as

$$\begin{aligned} K_{lat} &= (\boldsymbol{\alpha} - \mathbf{a})\mathcal{A}^{-1}\mathcal{C}^{-1} \\ &= \begin{pmatrix} -0.8809 & 10.0940 & -1.2821 & 0.9965 \end{pmatrix} \end{aligned}$$

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 $z_d$ and since $x = (z, \theta, \dot{z}, \dot{\theta})^\top$, we have $C_{out} = (1, 0, 0, 0)$, which gives

$$\begin{aligned} k_{r_{lat}} &= \frac{-1}{C_{out}(A - BK)^{-1}B} \\ &= -0.8809. \end{aligned}$$

From HW G.7, the state space equations for the longitudinal VTOL dynamics are given by

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

**Step 1.** The controllability matrix is therefore

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

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

**Step 2.** The open loop characteristic polynomial

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

which implies that

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

**Step 3.** When $\omega_h = 0.5936$, and $\zeta_h = 0.707$ the desired closed loop polynomial

$$\begin{aligned}\Delta_{cl}^d(s) &= s^2 + 2\zeta_z\omega_{n_z}s + \omega_{n_z}^2 \\ &= s^2 + 0.8394s + 0.3524\end{aligned}$$

which implies that

$$\boldsymbol{\alpha} = (0.8394, 0.3524).$$

**Step 4.** The gains are therefore given as

$$\begin{aligned}K_{lon} &= (\boldsymbol{\alpha} - \mathbf{a})\mathcal{A}^{-1}\mathcal{C}^{-1} \\ &= \begin{pmatrix}0.5285 & 1.2590\end{pmatrix}\end{aligned}$$

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

$$\begin{aligned}k_{r_{lat}} &= \frac{-1}{C(A - BK)^{-1}B} \\ &= 0.5285.\end{aligned}$$

Alternatively, we could have used the following Matlab script

```
% state space design
A_lon = [...
    0, 1;...
    0, 0;...
    ];
B_lon = [0; 1/(P.mc+2*P.mr)];
C_lon = [1, 0];
A_lat = [...
    0, 0, 1, 0;...
    0, 0, 0, 1;...
    0, -(P.Fe/(P.mc+2*P.mr)), -(P.mu/(P.mc+2*P.mr)), 0;...
    0, 0, 0, 0;...
    ];
B_lat = [0;0;0;1/(P.Jc+2*P.mr*P.d^2)];
C_lat = [1, 0, 0, 0; 0, 1, 0, 0];

% gains for pole locations
wn_h    = 0.5936;
zeta_h  = 0.707;
```

```
wn_z    = 0.9905;
zeta_z  = 0.707;
wn_th   = 13.3803;
zeta_th = 0.707;

ol_char_poly_lon = charpoly(A_lon);
des_char_poly_lon = [1,2*zeta_h*wn_h,wn_h^2];
des_poles_lon = roots(des_char_poly_lon);

ol_char_poly_lat = charpoly(A_lat);
des_char_poly_lat = conv([1,2*zeta_z*wn_z,wn_z^2],...
                     [1,2*zeta_th*wn_th,wn_th^2]);
des_poles_lat = roots(des_char_poly_lat);

% gains for longitudinal system
if rank(ctrb(A_lon,B_lon))≠2, disp('Lon System Not Controllable'); end
P.K_lon = place(A_lon,B_lon,des_poles_lon);
P.kr_lon = -1/(C_lon*inv(A_lon-B_lon*P.K_lon)*B_lon);

% gains for lateral system
if rank(ctrb(A_lat,B_lat))≠4, disp('Lat System Not Controllable'); end
P.K_lat = place(A_lat,B_lat,des_poles_lat);
Cout = [1, 0, 0, 0];
P.kr_lat = -1/(Cout*inv(A_lat-B_lat*P.K_lat)*B_lat);
```

The Matlab code for the controller is given by

```
function u=VTOL_ctrl(in,P)
    h_d   = in(1);
    z_d   = in(2);
    z     = in(3);
    h     = in(4);
    theta = in(5);
    t     = in(6);

    % use a digital differentiator to find hdot, zdot and thetadot
    persistent hdot
    persistent h_d1
    persistent zdot
    persistent z_d1
    persistent thetadot
    persistent theta_d1
    % reset persistent variables at start of simulation
    if t<P.Ts,
        hdot        = 0;
```

```
        h_d1        = 0;
        zdot        = 0;
        z_d1        = 0;
        thetadot    = 0;
        theta_d1    = 0;
    end
    hdot = (2*P.tau-P.Ts)/(2*P.tau+P.Ts)*hdot...
        + 2/(2*P.tau+P.Ts)*(h-h_d1);
    zdot = (2*P.tau-P.Ts)/(2*P.tau+P.Ts)*zdot...
        + 2/(2*P.tau+P.Ts)*(z-z_d1);
    thetadot = (2*P.tau-P.Ts)/(2*P.tau+P.Ts)*thetadot...
        + 2/(2*P.tau+P.Ts)*(theta-theta_d1);
    h_d1 = h;
    z_d1 = z;
    theta_d1 = theta;

    % longitudinal control for alititude
    % construct the state
    x_lon = [h; hdot];
    % equilibrium force
    Fe = (P.mc+2*P.mr)*P.g/cos(theta);
        % divide Fe by cos(theta) so that force is right during
        % lateral translations.
    % compute the state feedback controller
    F_tilde = -P.K_lon*x_lon + P.kr_lon*h_d;
    F = Fe + F_tilde;

    % lateral control for position
    % construct the state
    x_lat = [z; theta; zdot; thetadot];
    % compute the state feedback controller
    tau = -P.K_lat*x_lat + P.kr_lat*z_d;

    % produce forces on right and left rotors
    u = P.mixing*[F; tau];

end
```