A Design Study Approach to Classical Control

Randal W. Beard   Timothy W. McLain
Brigham Young University

Updated: May 18, 2016

Homework A.11

The objective of this problem is to implement state feedback controller using the full state. Start with the simulation files developed in Homework A.10.

(a) Select the closed loop poles as the roots of the equations $s^2+2\zeta\omega_n+\omega_n^2 = 0$ where ω_n, and ζ were found in Homework A.8.

(b) Add the state space matrices A, B, C, D derived in Homework A.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 so that the eigenvalues of $(A-BK)$ are equal to desired closed loop poles. Find the reference gain k_r so that the DC-gain from θ_r to θ is equal to one. Note that $K = (k_p, k_d)$ where k_p and k_d are the proportional and derivative gains found in Homework A.8. Why?

(e) Modify the control code to implement the state feedback controller. To construct the state $x = (\theta, \dot{\theta})^\top$ use a digital differentiator to estimate $\dot{\theta}$.

Solution

From HW A.6, the state space equations for the single link robot arm are given by

$$\dot{x} = \begin{pmatrix} 0 & 1.0000 \\ 0 & -0.667 \end{pmatrix} x + \begin{pmatrix} 0 \\ 66.667 \end{pmatrix} u$$
$$y = \begin{pmatrix} 1 & 0 \end{pmatrix} x.$$

Step 1. The controllability matrix is therefore

$$\mathcal{C}_{A,B} = [B, AB] = \begin{pmatrix} 0 & 66.6667 \\ 66.6667 & -44.44440 \end{pmatrix}.$$

The determinant is $det(\mathcal{C}_{A,B}) = -4444 \neq 0$, therefore the system is controllable.

Step 2. The open loop characteristic polynomial is

$$\Delta_{ol}(s) = \det(sI - A) = \det \begin{pmatrix} s & -1 \\ 0 & s + 0.667 \end{pmatrix} = s^2 + 0.667s,$$

which implies that

$$\mathbf{a}_A = (-0.667, 0)$$
$$\mathcal{A}_A = \begin{pmatrix} 1 & -0.667 \\ 0 & 1 \end{pmatrix}$$

Step 3. The desired closed loop polynomial is

$$\Delta_{cl}^d(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 6.3621s + 20.2445$$

which implies that

$$\boldsymbol{\alpha} = (6.3621, 20.2445).$$

Step 4. The gains are therefore given as

$$K = (\boldsymbol{\alpha} - \mathbf{a}_A)\mathcal{A}_A^{-1}\mathcal{C}_{A,B}^{-1}$$
$$= \begin{pmatrix} 0.3037 & 0.0854 \end{pmatrix}$$

$$k_r = \frac{-1}{C(A - BK)^{-1}B}$$
$$= 0.3037.$$

Alternatively, we could have used the following Matlab script

```matlab
% state space model
A = [0, 1; 0, -3*P.b/P.m/(P.ell^2)];
B = [0; 3/P.m/(P.ell^2) ];
C = [1, 0];

% gains for pole locations
charpoly = [1,2*zeta*wn,wn^2];
des_poles = roots(charpoly);

% is the system controllable?
if rank(ctrb(A,B))~=2,
    disp('System Not Controllable');
else
    P.K = place(A,B,des_poles);
    P.kr = -1/(C*inv(A-B*P.K)*B);
end
```

The Matlab code for the controller is given by

```matlab
function tau=arm_ctrl(in,P)
    theta_c = in(1);
    theta   = in(2);
    t       = in(3);

    % use a digital differentiator to find thetadot
    persistent thetadot
    persistent theta_d1
    % reset persistent variables at start of simulation
    if t<P.Ts,
        thetadot    = 0;
        theta_d1    = 0;
    end
    thetadot = (2*P.tau-P.Ts)/(2*P.tau+P.Ts)*thetadot...
        + 2/(2*P.tau+P.Ts)*(theta-theta_d1);
    theta_d1 = theta;

    % construct the state
    x = [theta; thetadot];
    % compute equilibrium torque tau_e
    tau_e = P.m*P.g*(P.ell/2)*cos(theta);
    % compute the state feedback controller
    tau_tilde = - P.K*x + P.kr*theta_c;
```

```
    % compute total torque
    tau = sat( tau_e + tau_tilde, P.tau_max);
end
```