ME 233 Spring 2010
Solution to Homework #10

1. (a) We first assemble the nominal system and the neglected dynamics using the commands

```
>> Gp = tf(wb^2,[1 2*zb*wb wb^2]);
>> Delta = tf([wr*zr wr^2], [1 2*zr*wr wr^2]) ...
   * tf([(wt/wn)^2 2*zt*wt^2/wn wt^2], [1 2*zt*wt wt^2]) - 1;
```

The output filter to enforce zero steady state error for a constant reference input in constructed by

```
>> Qr = tf(1, [1 0]);
```

After doing this, we need find a value of ρ which gives a gain crossover frequency of $60 rad/s$. Intuitively, as we increase ρ, there is more weight on the control input, which results in less control effort, which in turn causes a less aggressive controller. Since a higher gain crossover frequency intuitively means higher performance, we should decrease ρ in order to increase the gain crossover frequency (or increase ρ in order to decrease the gain crossover frequency). For a given value of ρ, we can find the gain crossover frequency using the code

```
>> [Cr, C1, C2] = fslqr(Gp, Qr, zeros(1,2), 1, rho);
>> [Go_R, T_R, S_R, Go_D, T_D] = fslqr_reg(Gp, Cr, C1, C2);
>> margin(Go_R)
```

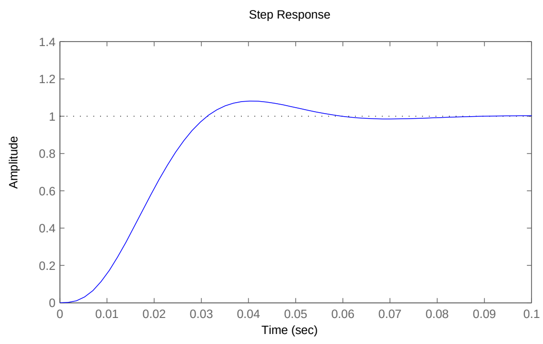

Figure 1: Closed loop step response from $R \to Y$ for $\rho = 3.2 \times 10^{-9}$, $R_f(s) = 1$

The first line of code finds the optimal state feedback controller K_e for the chosen values of ρ and frequency weights and then finds the blocks $C_r(s)$, $C_1(s)$, and $C_2(s)$ in slide 52 of lecture 14 using this value of K_e. The second line of code takes these blocks and then forms the relevant open loop transfer functions, complementary sensitivity functions, and sensitivity functions by performing the relevant block diagram manipulations. (There is no design occurring in this line of code, only block diagram manipulations.) The third line of code creates a Bode plot of the open loop transfer function from $E \to Y$ and shows its stability margins and associated frequencies. Note that the gain crossover frequency is the frequency associated with the phase margin. After a bit of trial and error, it is possible to find a value of ρ that gives a gain crossover frequency close to $60 rad/s$. Figure 2 shows the open loop Bode plot from $E \to Y$ for $\rho = 3.2 \times 10^{-9}$. As desired, the gain crossover frequency for this case is $60 rad/s$. The closed loop step response for this system from $R \to Y$ is computed by

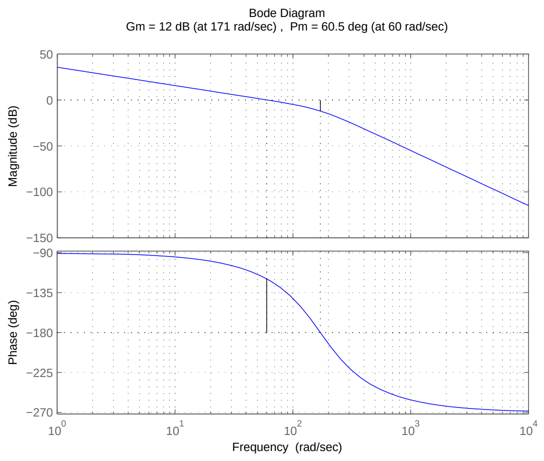

Figure 2: Open loop Bode plot from $E \to Y$ for $\rho = 3.2 \times 10^{-9}$ and no input shaping

```
>> step(T_R)
```

and is shown in figure 1.

To analyze the robustness of the nominal system to the unmodeled dynamics, we look at the open loop transfer function from $(-A) \to B$. The reason we look at this transfer function rather than the one from $A \to B$ has to do with how we broke the loop. If we were to reconnect the loop by connecting A and B, this would be equivalent to closing a loop with *positive* feedback. Since our robustness results are based on negative feedback, we need to instead look at an open loop transfer function broken at the control input which would be reconnected using negative feedback. Thus, looking at the block diagram in figure 3, we can see that the transfer function from $(-A) \to B$ would be reconnected using negative feedback, which justifies why we look at this open loop transfer function and its associated complementary sensitivity function. Figure

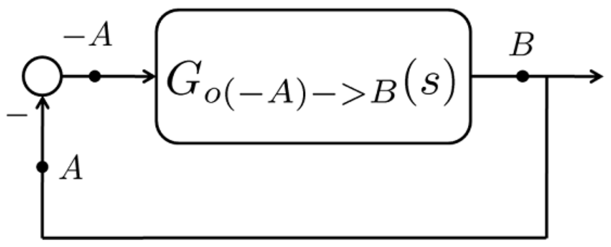

Figure 3: Block diagram for the closed loop system

4 shows the Bode magnitude plot of the open loop transfer function from $(-A) \to B$ (denoted $G_{o(-A)->B}(s)$), its associated complementary sensitivity function ($T_{(-A)->B}(s)$), and the inverse of $\Delta(s)$. This was computed by

```
>> bodemag(Go_R, T_R, inv(Delta))
```

Since $T_{(-A)->B}(j\omega)$ has a larger magnitude than $1/\Delta(j\omega)$ at some frequencies, we are unable to use the small gain theorem to make any conclusions regarding stability of the actual plant. In fact, if we check the stability of the actual closed loop system, e.g. using the commands

```
>> [Go_RA, T_RA, S_RA, Go_DA, T_DA] = ...
   fslqr_reg_robust_test(Gp*(1+Delta), Gp, Cr, C1, C2);
>> isstable(T_RA)
```

we will see that the closed loop system is, in fact, unstable. (The first line just performs block diagram manipulations for the actual system to obtain the transfer functions that we are interested

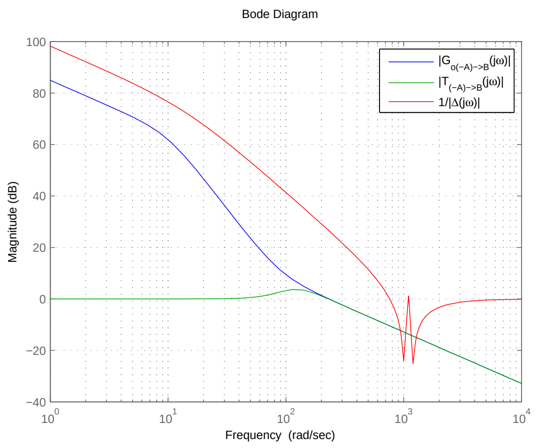

Figure 4: Bode magnitude plot of the open loop transfer function from $(-A) \to B$ $(G_{o(-A)->B}(s))$, its associated complementary sensitivity function $(T_{(-A)->B}(s))$, and the inverse of $\Delta(s)$

in.) Note that although we could not infer instability from the fact that the small gain condition was violated, it was a good indicator in this case that the actual closed loop system might be unstable.

(b) In this part, we design an input filter, $R_f(s)$ to increase the robustness of the system to the unmodeled dynamics in the nominal. In general, there are many approaches to finding a filter to achieve this. There are, however, a few guiding principles that should be followed. At frequencies where we want good performance (usually low frequency), we should choose the input weight to be small so that the controller can use a lot of control effort to achieve the desired level of performance. At frequencies where we instead want good robustness properties (usually high frequency), we should choose the input filter weight to be large. This has the effect of forcing the controller to use relatively little control effort in these frequency ranges, which in turn tends to give better robustness properties. At very high frequencies, in order to guarantee that the frequency shaped LQR problem will have a solution, $R_f(s)$ must have a constant magnitude (i.e. $R_f(s)$ should be causal but not strictly causal). Also, from a practical standpoint, it is desirable to have low-order filters. The reason for this is that the number of states in $R_f(s)$ is equal to the number of states in $C_2(s)$. Since the resulting controller must contain all of these states, the resulting controller must have at least as many states as $R_f(s)$. We would like to keep the number of states in the controller small for two reasons. First of all, in controller with a large number of states, it takes a relatively long time to update the state because the A matrix in the state space representation of the controller is very large. This means that we cannot discretize the the controller and implement it at very fast sampling rates because the computational delay will be too large. Second of all, since working with larger matrices and vectors tends to result in larger numerical errors (e.g. round-off error), we should keep the controller relatively small to keep the numerical errors small.

For this particular problem, we would like to choose $R_f(s)$ such that it has relatively large magnitude at frequencies above $10^3 rad/s$ to increase robustness to the unmodeled dynamics. One approach would be to use a second-order filter, as was done in lecture 14. Another approach would be to use a product of first-order filters. For instance, consider

$$R_f(s) = 26\left(\frac{s+200}{s+2000}\right)^2$$

which is constructed by the command

```
>> Rf = 26 * tf([1 200],[1 2000])^2;
```

and has the Bode magnitude plot shown in figure 5. This filter was found in two steps. First a

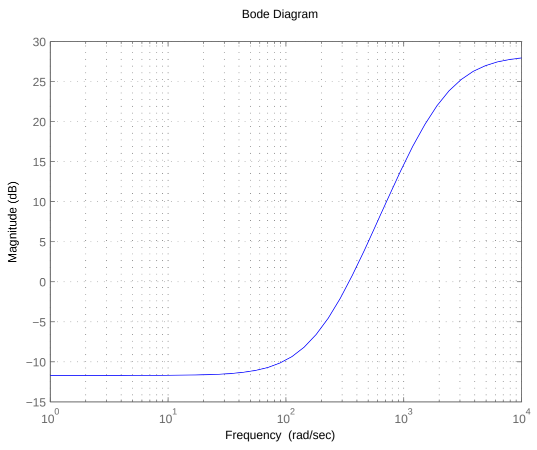

Figure 5: Bode magnitude plot of input filter, $R_f(s)$

reasonable shape was selected for the filter. This filter penalizes control input more heavily at frequencies higher than $200 rad/s$. Since the filter magnitude must level off somewhere, we chose the filter to level off at $2000 rad/s$, which is past the unmodeled resonances. Once this was done, we chose the constant out front (26, in this case) to achieve the desired gain crossover frequency of the transfer function $G_{oE->Y}(s)$. For particular choice of the input filter, we executed the commands

```
>> [Cr, C1, C2] = fslqr(Gp, Qr, zeros(1,2), Rf, rho);
>> [Go_R, T_R, S_R, Go_D, T_D] = fslqr_reg(Gp, Cr, C1, C2);
>> margin(Go_R)
```

to check the gain crossover frequency. The intuition for choosing the constant out front in the expression for $R_f(s)$ is the same as the intuition for choosing ρ. In general, if the performance and robustness constraint cannot be met just by changing the constant out front, it may be necessary to change the filter shape. (In this case, we found that the filter shape $(s + 200)/(s + 2000)$ was not aggressive enough, so we squared it to make the difference between its high and low frequency magnitudes more pronounced.)

With this choice of $R_f(s)$, the open loop Bode plot of $G_{oE->Y}(s)$ is shown in figure 6. As desired,

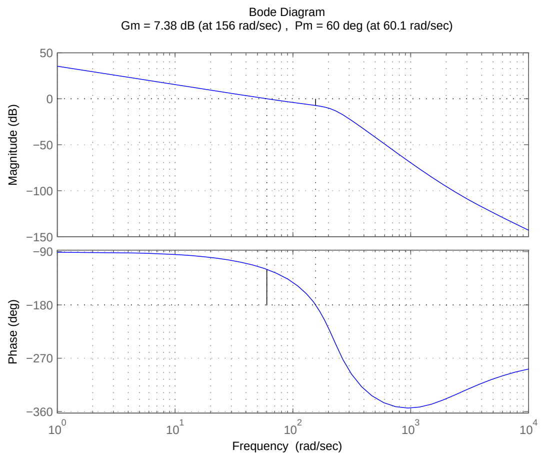

Figure 6: Open loop Bode plot from $E \to Y$ with input shaping

the gain crossover frequency is still close to $60 rad/s$. Figure 7 shows the Bode magnitude plots that are relevant to robustness to the unmodeled dynamics. In this case, since the magnitude of

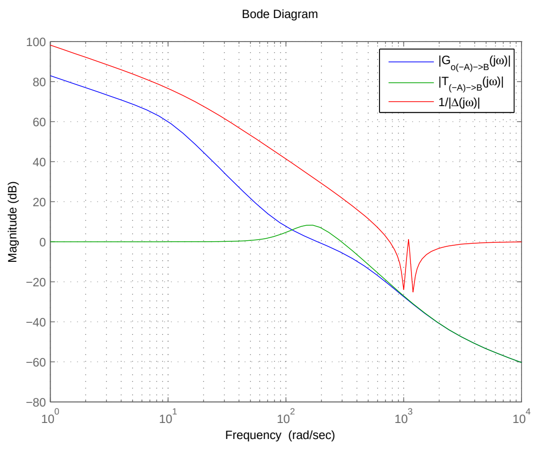

Figure 7: Bode magnitude plot of the open loop transfer function from $(-A) \rightarrow B$ ($G_{o(-A)->B}(s)$), its associated complementary sensitivity function ($T_{(-A)->B}(s)$), and the inverse of $\Delta(s)$ with input weighting

$T_{(-A)->B}(j\omega)$ is less than the magnitude of $1/\Delta(j\omega)$ at all frequencies, the small gain theorem applies and we can conclude that the actual closed loop system is stable. This is easily verified using the commands

```
>> [Go_RA, T_RA, S_RA, Go_DA, T_DA] = ...
   fslqr_reg_robust_test(Gp*(1+Delta), Gp, Cr, C1, C2);
>> isstable(T_RA)
```

(c) In this part, we use the value of ρ from part (a) and the value of $R_f(s)$ in part (b) to design a controller which does not require direct state access by using the LQG-LTR recovery process. In this part, as we decrease the covariance of the fictitious measurement noise, we should recover the robustness results achieved in part (b). For a particular value of μ, we compute the relevant transfer functions using the commands

```
>> [A,B,C,D] = ssdata(Gp);
>> Kest = kalman(ss(A, [B B], C, [0 0]), 1, mu^2);
>> [Go_RKF, T_RKF, S_RKF, Go_DKF, T_DKF] = ...
   fslqr_reg_est(Gp, Cr, C1, C2, Kest);
```

Figure 8 shows the recovery process for $G_{o(-A)->B}(s)$ (`Go_DKF` in the MATLAB commands above). Since $\mu = 10^{-7}$ gives good gain and phase recovery until frequencies higher than the unmodeled dynamics, we will use that value. Figure 9 shows the Bode plot of $G_{oE->Y}(s)$ (`Go_RKF` in the MATLAB commands above). As desired, the gain crossover frequency is still close to $60 rad/s$. Figure 10 shows the Bode magnitude plots that are relevant to robustness to the unmodeled dynamics. As desired, the magnitude of $T_{(-A)->B}(j\omega)$ is smaller than the magnitude of $\Delta(j\omega)$ at all frequencies, which allows us to conclude that the actual closed loop system will be stable. This could be verified using the code

```
>> [Go_RKFA, T_RKFA, S_RKFA, Go_DKFA, T_DKFA] = ...
   fslqr_reg_est(Gp*(1+Delta), Cr, C1, C2, Kest);
>> isstable(T_RFKA)
```

Now, we look at the robustness of the actual closed loop system. Throughout this problem, we have assumed that the unmodeled dynamics could be modeled as multiplicative input uncertainty. To analyze the robustness of the actual closed loop system, we will keep this same assumption. Thus, we are interested in the gain and phase margins of the open loop transfer function from $(-A) \rightarrow B$. Figure 11 shows the Bode plot of $G_{o(-A)->B}(s)$ for the actual system with the Kalman Filter. For this open loop transfer function, there are three gain margins and one phase margins. The smallest positive and negative gain margins are $13.5db$ and $-8.82db$ and the phase margin is $23.7°$. The presence of the negative gain margin makes sense because $G_{o(-A)->B}(s)$ for

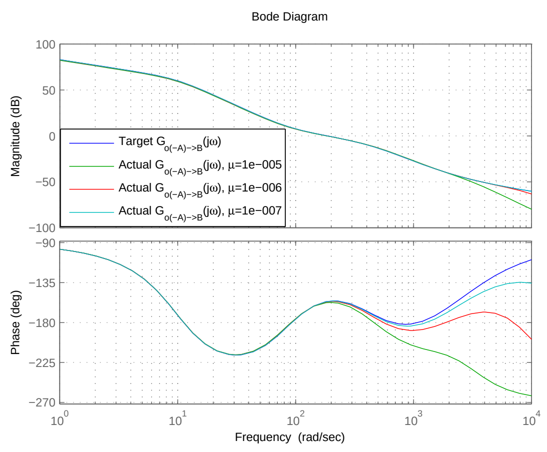

Figure 8: Recovery process for $G_{o(-A)->B}(s)$

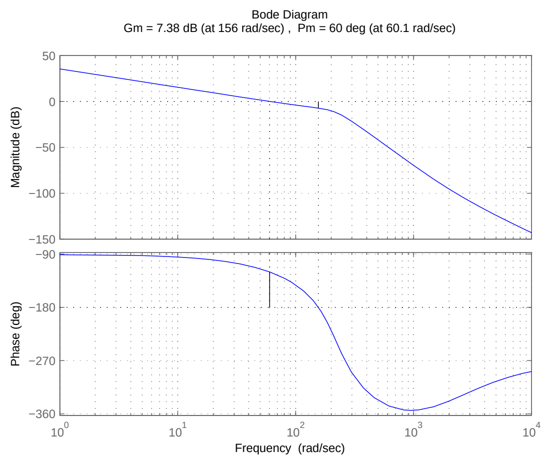

Figure 9: Open loop Bode plot from $E \to Y$ with Kalman Filter and input shaping

this system contains one unstable open loop zero. The closed loop unit step response from $R \to Y$ generated by `step(T_RKFA)` is shown in figure 12 and verifies that the controller is reasonable.

2. (a) To find the discretized plant, we use the MATLAB code

```
>> Gd = c2d( tf(1,[1 1 0]), 0.5 )
```

This gives that

$$\begin{aligned} G(z) &= \frac{0.1065z + 0.0902}{z^2 - 1.607z + 0.6065} \\ \Rightarrow A(q^{-1}) &= 1 - 1.607q^{-1} + 0.6065q^{-2} \\ B(q^{-1}) &= 0.1065 + 0.0902q^{-1} \\ d &= 1. \end{aligned}$$

(b) The continuous time poles are given by

$$\begin{aligned} p_c &- \zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \\ &= -0.707 \pm j\sqrt{1-0.707^2} \\ &= -0.707 \pm 0.7072j. \end{aligned}$$

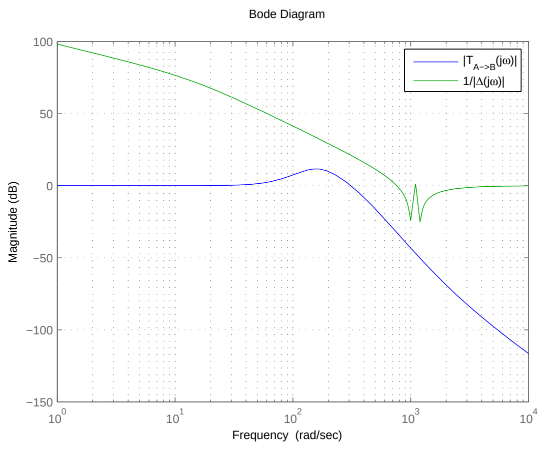

Figure 10: Bode magnitude plot of the open loop transfer function from $(-A) \to B$ ($G_{o(-A)->B}(s)$), its associated complementary sensitivity function ($T_{(-A)->B}(s)$), and the inverse of $\Delta(s)$ with Kalman Filter and input weighting

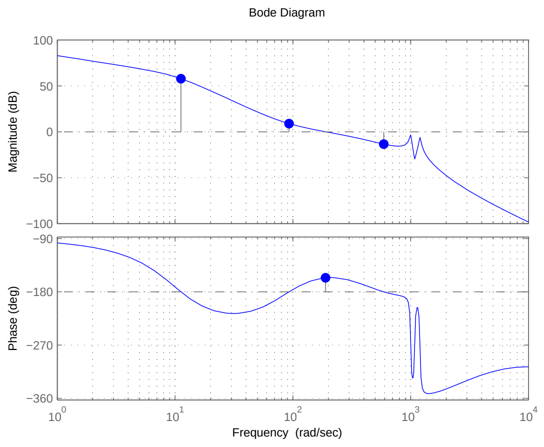

Figure 11: Open loop Bode plot from $(-A) \to B$ with Kalman Filter and input shaping

Therefore, the discrete time poles are given by

$$\begin{aligned} p_d &= e^{p_c T} \\ &= 0.6588 \pm 0.2432j. \end{aligned}$$

Using the MATLAB command `poly`, we find that the polynomial with these roots is given by

$$z^2 - 1.3176z + 0.4931 = 0.$$

Therefore, $A_m(q^{-1})$ is given by

$$A_m(q^{-1}) = 1 - 1.3176q^{-1} + 0.4931q^{-2}.$$

(c) We choose

$$b_{mo} = A_m(1) = 0.1755.$$

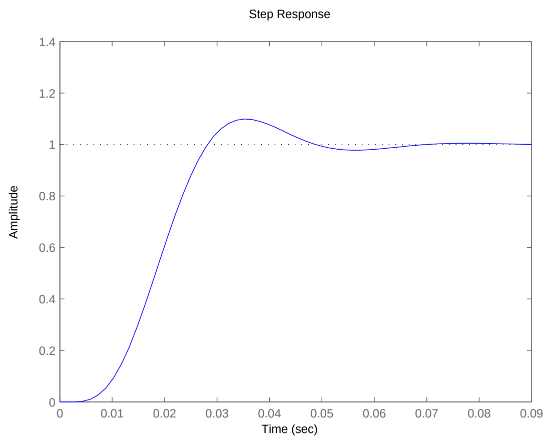

Figure 12: Actual closed loop step response from $R \rightarrow Y$

(d) As before

$$\begin{aligned} p_c &= -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \\ &= -1 \pm 2j\sqrt{1-0.5^2} \\ &= -1 \pm 1.7321j \\ \Rightarrow p_d &= e^{p_c T} \\ &= 0.3929 \pm 0.462j. \end{aligned}$$

Using the MATLAB command `poly`, we find that the polynomial with these roots is given by

$$z^2 - 0.7859z + 0.3679 = 0$$

which means that we should choose

$$A_c'(q^{-1}) = 1 - 0.7859q^{-1} + 0.3679q^{-2}.$$

(e) In this case, $A_d(q^{-1}) = 1$ and $B^u(q^{-1}) = b_0 = 0.1065$. Thus, the Bezout (Diophantine) equation for this case is given by

$$A_c'(q^{-1}) = A(q^{-1})R'(q^{-1}) + q^{-1}b_0S(q^{-1})$$
$$\Rightarrow 1 - 0.7859q^{-1} + 0.3679q^{-2} = \left(1 - 1.607q^{-1} + 0.6065q^{-2}\right)(1) + q^{-1}0.1065\left(s_0 + s_1q^{-1}\right).$$

Equating coefficients gives

$$\begin{aligned} s_0 &= 7.7108 \\ s_1 &= -2.2404. \end{aligned}$$

Therefore

$$\begin{aligned} S(q^{-1}) &= 7.7108 - 2.2404q^{-1} \\ R(q^{-1}) &= R'(q^{-1})B^s(q^{-1}) \\ &= 1 + 0.8467q^{-1}. \end{aligned}$$

With this choice of $R(q^{-1})$ and $S(q^{-1})$, the dynamics from $r(z)$ to $y(z)$ are given by

$$\begin{aligned} y(z) &= \frac{z^{-1}b_0B^s(z^{-1})}{B^s(z^{-1})A_c'(z^{-1})}r(z) \\ &= \frac{b_0}{zA_c'(z^{-1})}r(z). \end{aligned}$$

In order to obtain the dynamics $y(z) = y_d(z)$, we see that we should choose

$$\begin{aligned} r(z) &= \left(zA'_c(z^{-1})\right) y_d(z) \\ \Rightarrow r(k) &= \left(qb_0^{-1}A'_c(q^{-1})\right) y_d(k) \\ \Rightarrow T(q^{-1}, q) &= qb_0^{-1}A'_c(q^{-1}) \\ &= 9.3897q - 7.3793 + 3.4545q^{-1}. \end{aligned}$$

(f) See Fig. 13 for simulation results.

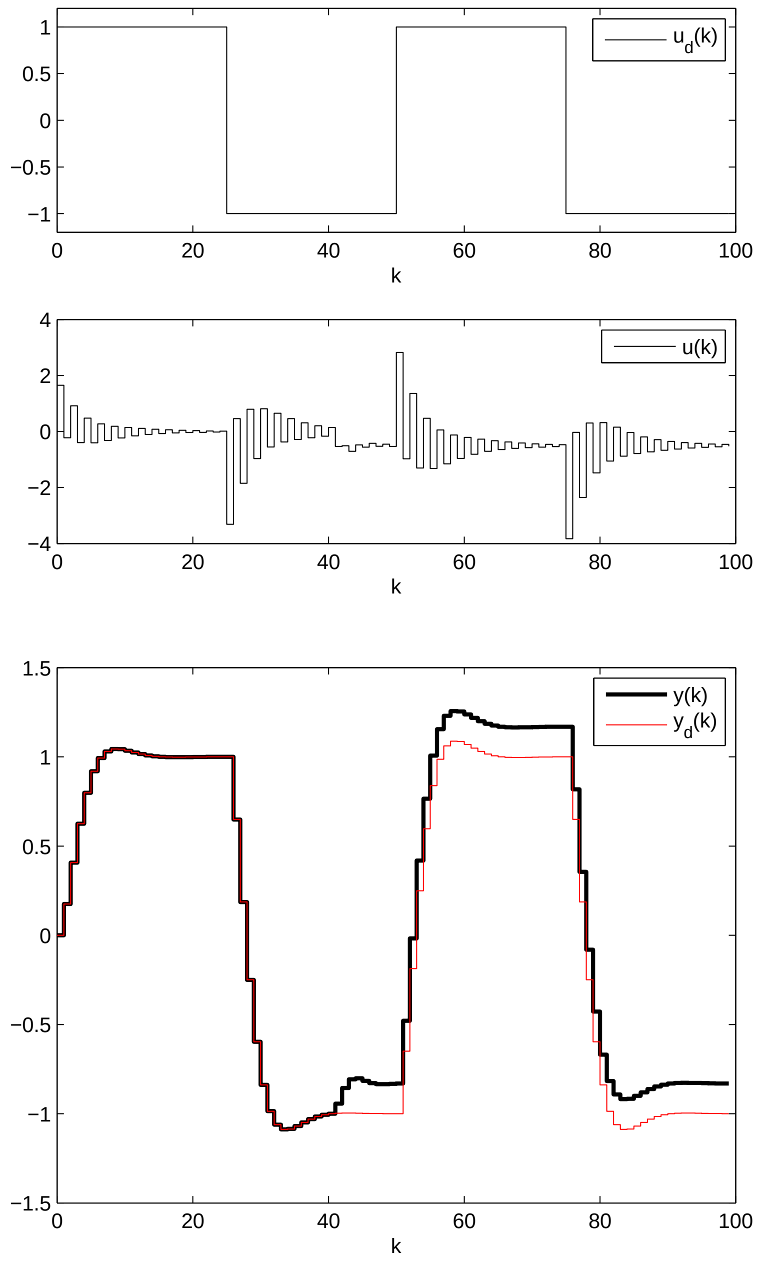

Figure 13: Closed loop simulation for $(A_d, B^u) = (1, b_0)$

(g) In this case, $A_d = 1 - q^{-1}$ and $B^u = b_0 = 0.1065$. Thus, the Bezout (Diophantine) equation for this case is given by

$$\begin{aligned} A_c' &= A_d A R' + q^{-1} b_0 S \\ \Rightarrow 1 - 0.7859q^{-1} + 0.3679q^{-2} &= \left(1 - q^{-1}\right)\left(1 - 1.607q^{-1} + 0.6065q^{-2}\right)(1) \\ &\quad + q^{-1} b_0 \left(s_0 + s_1 q^{-1} + s_2 q^{-2}\right). \end{aligned}$$

Equating coefficients gives

$$\begin{aligned} s_0 &= 17.1005 \\ s_1 &= -17.3296 \\ s_2 &= 5.6948 \end{aligned}$$

Therefore

$$\begin{aligned} S &= 17.1005 - 17.3296q^{-1} + 5.6948q^{-2} \\ R &= R' A_d B^s \\ &= (1)\left(1 - q^{-1}\right)\left(1 + 0.8467q^{-1}\right) \\ &= 1 - 0.1531q^{-1} - 0.8467q^{-2} \end{aligned}$$

As before, since the dynamics from $r(k)$ to $y(k)$ are now given by

$$b_0^{-1} A_c' y(k) = q^{-1} r(k)$$

we still set

$$T = q b_0^{-1} Ac' = 9.3897q - 7.3793 + 3.4545q^{-1}.$$

to get the desired dynamics.

(h) See Figure 14 for simulation results.

(i) In this case, $A_d = 1 - q^{-1}$, $B^s = 1$, and $B^u(q^{-1}) = 0.1065 + 0.0902q^{-1}$. Thus, the Bezout (Diophantine) equation for this case is given by

$$\begin{aligned} A_c' &= A_d A R' + q^{-1} B^u S \\ \Rightarrow 1 - 0.7859q^{-1} + 0.3679q^{-2} &= \left(1 - q^{-1}\right)\left(1 - 1.607q^{-1} + 0.6065q^{-2}\right)\left(1 + r_1 q^{-1}\right) \\ &\quad + q^{-1}\left(0.1065 + 0.0902q^{-1}\right)\left(s_0 + s_1 q^{-1} + s_2 q^{-2}\right). \end{aligned}$$

Equating coefficients gives

$$\begin{aligned} r_1 &= 0.5937 \\ s_0 &= 11.5259 \\ s_1 &= -12.5585 \\ s_2 &= 3.9919. \end{aligned}$$

Therefore

$$\begin{aligned} S &= 11.5259 - 12.5585q^{-1} + 3.9919q^{-2} \\ R &= R' A_d B^s \\ &= \left(1 + 0.5937q^{-1}\right)\left(1 - q^{-1}\right) \\ &= 1 - 0.4063q^{-1} - 0.5937^{-2}. \end{aligned}$$

The zero-phase feedforward compensator is then given by

$$\begin{aligned} T = \frac{q A_c' B_u(q)}{[B_u(1)]^2} &= \frac{\left(q - 0.7859 + 0.3679q^{-1}\right)(0.1065 + 0.0902q)}{0.1967^2} \\ &= 2.3313q^2 + 0.9204q - 1.3056 + 1.0127q^{-1}. \end{aligned}$$

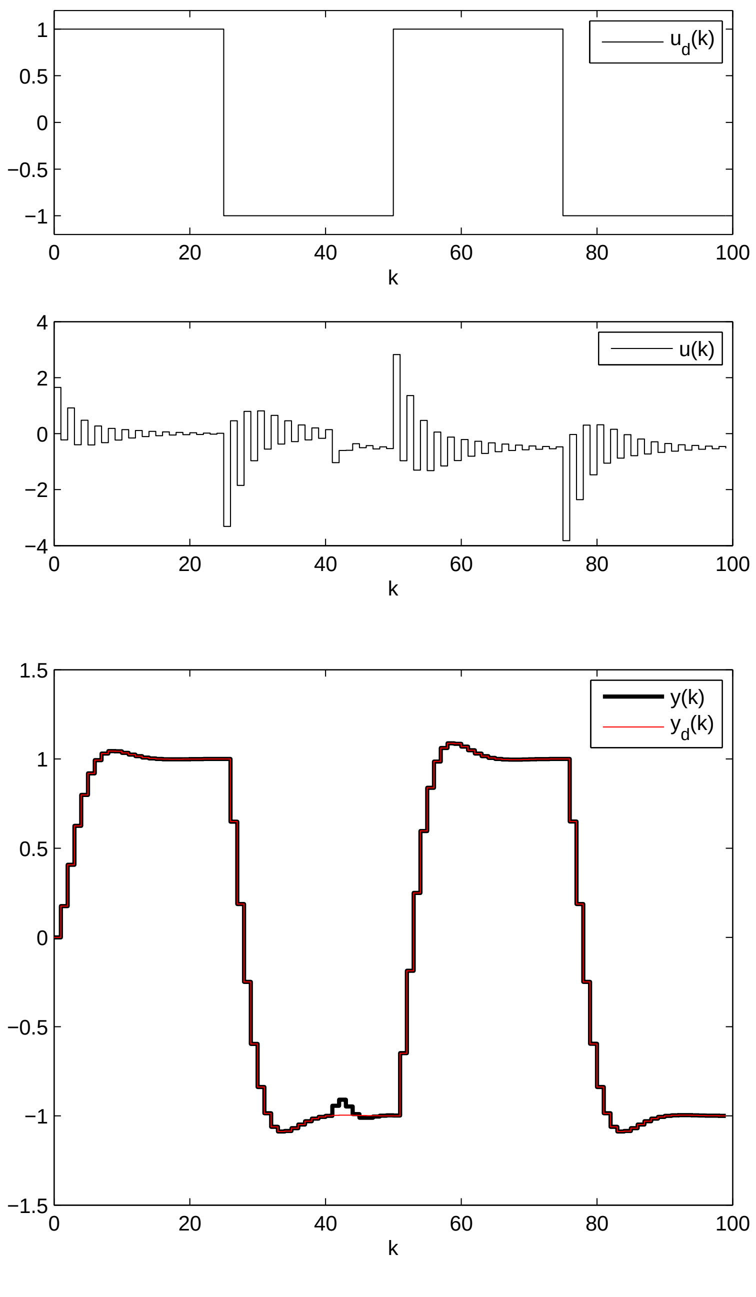

Figure 14: Closed loop simulation for $(A_d, B^u) = (1 - q^{-1}, b_0)$

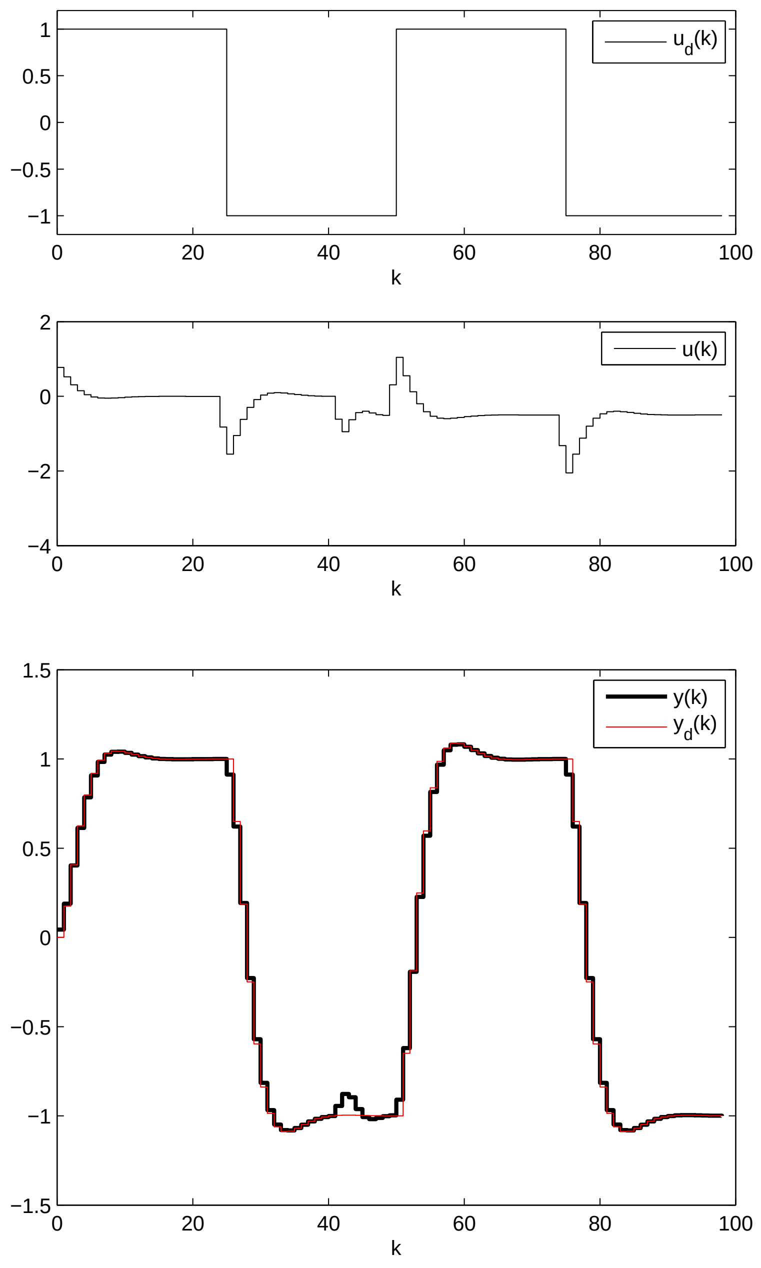

Figure 15: Closed loop simulation for $(A_d, B^u) = (1 - q^{-1}, 0.1065 + 0.0902q^{-1})$

(j) See Figure 15 for simulation results.

(k) If the disturbance is not considered in the controller design, there will be steady state error, as shown in Figure 13. When the disturbance model is considered in the controller design process, the steady state error is removed. If the stable pole-zero cancellation is performed, the tracking performance is better than the one in which pole-zero cancellation is not done, but the control input oscillates a little bit due to the lightly damped pole in the controller. With the zero-phase error tracking controller, the same performance can be achieved with smaller control effort, as can be seen by comparing Figure 14 with Figure 15.