AE353: Step Response and Specifications

(to be treated as an appendix to the presentation)

A. Borum T. Bretl M. Ornik P. Thangeda

1 Step Response

In the notes on solutions with input, we've seen that the solution of the system

$$\dot{x}_m = A_m x_m + B_m u_m$$
$$y_m = C_m x_m,$$

where u_m is a constant, is given by

$$x_m(t) = e^{A_m t} x_m(0) + A_m^{-1} e^{A_m t} B_m u_m - A_m^{-1} B_m u_m \tag{1}$$
$$y_m(t) = C_m(e^{A_m t} x_m(0) + A_m^{-1} e^{A_m t} B_m u_m - A_m^{-1} B_m u_m). \tag{2}$$

The unit step response of this system is the output $y_m(t)$ in response to the input $u_m = 1$, with the initial condition $x_m(0) = 0$. Using (2), this response is given by

$$y_m(t) = C_m A_m^{-1}(e^{A_m t} - I) B_m$$

An example of a step response is shown on page 6-21 of *Feedback Systems* by Åström and Murray. As shown in this example and discussed in the lecture, there are a few quantities that are typically used to describe the behavior of the step response:

- time-to-peak T_p: the time taken for $y_m(t)$ to attain its maximum value
- overshoot M_p: the maximum amount $y_m(t)$ overshoots its steady-state y_{ss} value as a fraction of the steady-state value. Overshoot is often defined as a percentage, i.e., $100 \times \frac{y_m(T_p) - y_{ss}}{y_{ss}}$.
- 10% - 90% rise time: the time taken for $y_m(t)$ to rise from 10% of the steady-state output to 90% of the steady-state output, i.e., from $0.1 y_{ss}$ to $0.9 y_{ss}$
- 5% settling time: the time taken for $y_m(t)$ to settle to within $\pm 5\%$ of its steady-state value, i.e., the time after which $y_m(t)$ always lies between $0.95 y_{ss}$ and $1.05 y_{ss}$.
- steady-state error: the difference between the steady-state output y_{ss} and r, i.e., $e_{ss} = y_{ss} - r$.

2 Example second order system

In class, we considered the example system

$$\begin{aligned} \dot{x} &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \\ y &= \begin{bmatrix} 1 & 0 \end{bmatrix} x \end{aligned} \tag{3}$$

with the control input $u = -Kx + k_{\text{ref}} r$. We know that the stability of the closed-loop system is governed by the eigenvalues of the matrix $A - BK$. By properly choosing the components of K, we can place the eigenvalues at desired locations. We've seen in class that the eigenvalues of $A - BK$ can be placed at desired locations using the Matlab functions place or acker. For asymptotic stability, we want the eigenvalues to have negative real parts. What other conditions might we want these eigenvalues to satisfy?

We can show that the eigenvalues of $A - BK$ are related to the peak time and the overshoot of system (3)'s step response. To see that this is true, let's suppose that the eigenvalues of $A - BK$ are $-\sigma \pm \omega j$, where σ and ω are real numbers. When we choose k_{ref} so that there is no steady-state error, the step response of the system (3) (i.e., the output with $x(0) = 0$ and $r = 1$) is given by

$$y(t) = \alpha_1 e^{-\sigma t} \cos(\omega t) + \alpha_2 e^{-\sigma t} \sin(\omega t) + 1.$$

Since the initial condition is $x(0) = 0$, we see that $y(0) = x_1(0) = 0$ and $\dot{y}(0) = x_2(0) = 0$. Imposing these initial conditions on y gives

$$\alpha_1 = -1 \qquad \alpha_2 = -\frac{\sigma}{\omega},$$

and the step response is

$$y(t) = -e^{-\sigma t} \cos(\omega t) - \frac{\sigma}{\omega} e^{-\sigma t} \sin(\omega t) + 1.$$

Now let's try to find the peak time T_p. At the peak time, $y(t)$ will have a maximum. From calculus, we know that at this maximum, the derivative of $y(t)$ must be zero, and therefore $\dot{y}(T_p) = 0$. Computing the derivative of $y(t)$ using the product rule gives

$$\dot{y}(t) = \left(\omega + \frac{\sigma^2}{\omega} \right) e^{-\sigma t} \sin(\omega t).$$

The only way that $\dot{y}(t)$ can be zero is if $\sin(\omega t) = 0$, which occurs when $\omega t = n\pi$, where n is a non-negative integer. The maximum of $y(t)$ corresponds to $n = 1$ and $\omega t = \pi$. Therefore, the peak time is

$$T_p = \frac{\pi}{\omega}.$$

Plugging this time into the expression for the output gives

$$y(T_p) = e^{-\sigma T_p} + 1.$$

Since the steady-state output is $y_{ss} = 1$ and the overshoot is defined as

$$M_p = \frac{y(T_p) - y_{ss}}{y_{ss}},$$

we have

$$M_p = e^{-\sigma T_p}.$$

We can invert the expressions for T_p and M_p to find

$$\omega = \frac{\pi}{T_p} \qquad \sigma = -\frac{\ln(M_p)}{T_p}. \tag{4}$$

While these expression are not exact for every second-order system, they provide a rule of thumb for adjusting the eigenvalues to achieve desired peak times and overshoots.

3 Dominant eigenvalues

For many third-order systems (i.e, systems that have three states), if k_{ref} is chosen so that there is zero steady-state error, the system's step response is given by

$$y(t) = \alpha_1 e^{-\sigma t}\cos(\omega t) + \alpha_2 e^{-\sigma t}\sin(\omega t) + \alpha_3 e^{-\lambda t} + 1,$$

where α_1, α_2, and α_3 are constants, and the system's eigenvalues are $-\sigma \pm \omega j$ and $-\lambda$. (Can you say why the eigenvalues of a third-order system must have this structure?) Can we use the expression in (4) to achieve a desired peak time and overshoot?

Clearly, this output is different than the output in Section 2. However, suppose that λ is much larger than σ. The term $\alpha_3 e^{-\lambda t}$ would then decay much quicker than the two terms $\alpha_1 e^{-\sigma t}\cos(\omega t)$ and $\alpha_2 e^{-\sigma t}\sin(\omega t)$. In this case, we say that the system has a dominant second-order response, since the effects of the third eigenvalue $-\lambda$ decay much quicker than the effects of the eigenvalues $-\sigma \pm \omega j$.

In class we've seen how to place eigenvalues at desired locations. Therefore, for many third order systems, we can place two eigenvalues at $-\sigma \pm \omega j$ using (4) to achieve a desired peak time and overshoot. Then we can place the third eigenvalue $-\lambda$ so that it has a small effect on the output. This method can be extended to higher-order systems, where we place two eigenvalues to achieve a desired step response, and then place the remaining eigenvalues so that they have a small effect on the output.

4 Tuning heuristics

For the higher-order systems discussed in Section 3, the expressions (4) are not exact. They do, however, provide a rule of thumb for achieving a desired peak time and overshoot. We might think of the expressions (4) as "knobs" for adjusting the peak time and overshoot for higher-order systems. For example, a procedure for tuning the eigenvalue placement to achieve a desired peak time T_p^* and overshoot M_p^* may go as follows:

1. Specify a desired peak time $\widehat{T}_p$ and overshoot $\widehat{M}_p$ (a good initial guess for $\widehat{T}_p$ and $\widehat{M}_p$ is T_p^* and M_p^*).
2. Compute the corresponding eigenvalues $-\sigma \pm \omega j$ using (4).
3. Place all other eigenvalues so that their real parts are much more negative than $-\sigma$ (which makes $-\sigma \pm \omega j$ the dominant eigenvalues).
4. Compute the resulting step response and find the peak time T_p and overshoot M_p (which may be different than $\widehat{T}_p$ and $\widehat{M}_p$).
5. If $T_p < T_p^*$, increase $\widehat{T}_p$. If $T_p > T_p^*$, decrease $\widehat{T}_p$. Adjust $\widehat{M}_p$ similarly.
6. Repeat steps 2-5 until T_p and M_p are sufficiently close to T_p^* and M_p^*.

Many other methods are also used. You may, of course, choose whatever method you find the most helpful.

5 Example and MATLAB Snippets

In this section, we work on an example of calculating the specifications for a given controller using MATLAB. Consider the system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

with the input

$$u = -Kx + K_{ref} r.$$

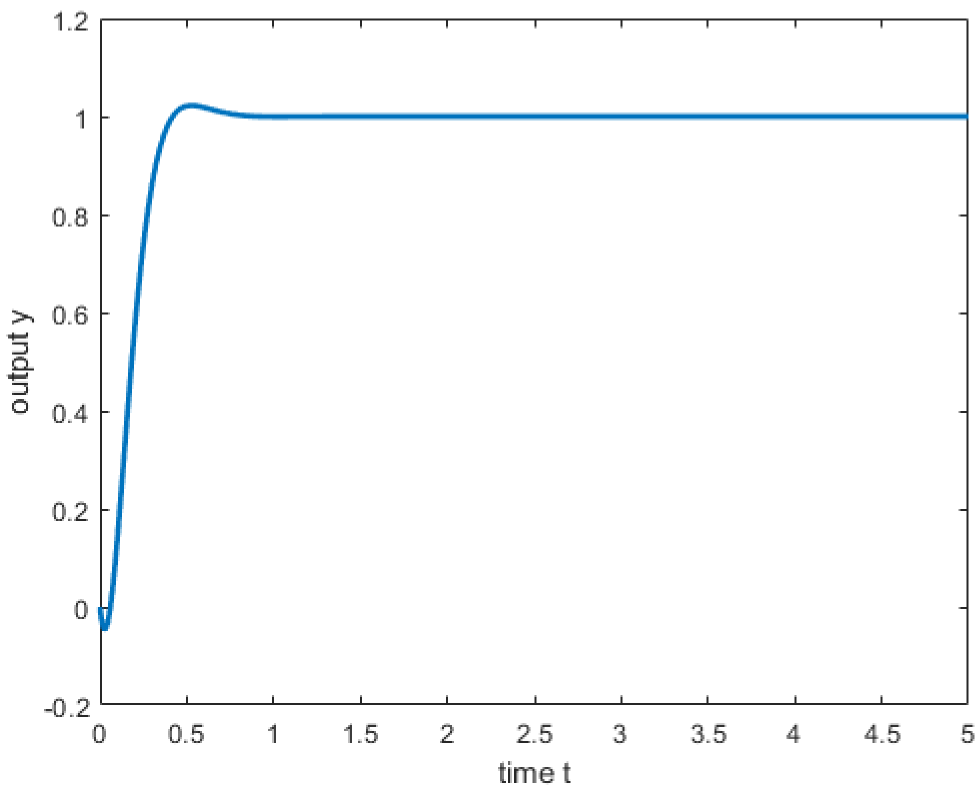
where given

```
>> A = [-0.06 1.05; 0.03 0.02];
>> B = [0.01; 0.98];
>> C = [1.05 -0.05];
>> D = [0.00];
>> K = [95.17 14.70];
```

We consider $r = 1$ and zero initial conditions and solve for the output numerically for the duration $t = [0, 5]$ using the method discussed in lecture 9. Note that you need to have ‘odedynref.m‘ file provided in ‘Lecture9Files.zip‘ in your working directory for this code to work.

```
% initial state
x_init = [0; 0];

% reference
r = 1

% calculate Kref
Kref=-1/(C*inv(A-B*K)*B);

% solve for x at different time instances in the duration [0,5]
[t,x]=ode45(@(t,x)(odedynref(A,B,K,r,Kref,x,t)),[0 5],x_init);

% calculate and plot the output y
plot(t,C*x');
```

We obtain the following plot of output y with respect to time t:



We can calculate quantities such as time-to-peek, overshoot, etc. either directly from the plot or by using the solution data as done below. Note that the following snippets won't always work: they assume that the output reaches steady-state in the time duration of the simulation and also assume that there is an overshoot in the output (which might not always be the case).

```
% calculate y
y = C*x';

% find steady state (only valid if y converged)
y_ss = y(end);

% find time-to-peak and overshoot
[ max_y, max_indx ] = max(y);
t_peek = t(max_indx);

% not so elegant attempt at detecting there is no overshoot
if t_peek == t(end)
    t_peek = -1;
end
overshoot = ((max_y - y_ss)/(y_ss))*100;

% find 10% to 90 % rise time
[ ~ , low_indx ] = min(abs(y-0.1*y_ss));
t_10yss = t(low_indx);
[ ~ , high_indx ] = min(abs(y-0.9*y_ss));
t_90yss = t(high_indx);
t_rise = t_90yss - t_10yss;

% find 5% settling time
for i = 1 : length(t)
    if all(y(i:end) >= 0.95*y_ss) && all(y(i:end)<= 1.05*y_ss)
        t_settling = t(i);
        break
    end
end

% find steady state error
e_ss = round(y_ss - r,1);
```