ENGR 105: Feedback Control Design
Winter 2013

Lecture 25 - Time Delay

Friday, March 15, 2013

Today's Objectives

1. give an example system with time delay
2. describe how to model time delay
3. examine system behavior with time delay

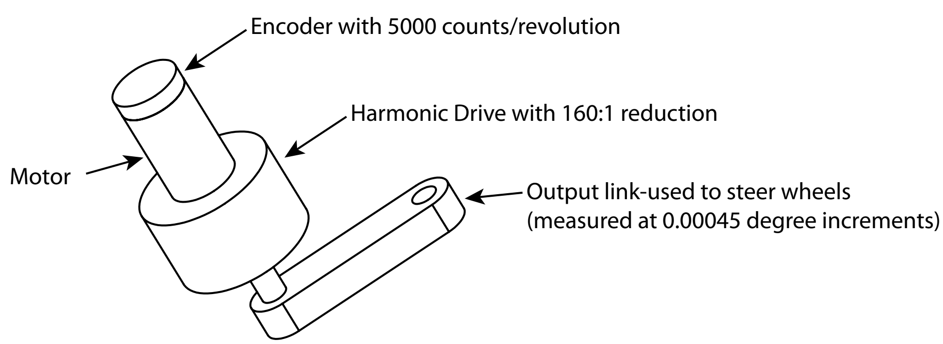
Reading: FPE Section 5.6.3

Time Delay

Time delay is a factor in many control systems. Root locus and Bode/Nyquist techniques provide a useful tool for analyzing the effect of delay.

1 Example system: Steer-by-wire cars

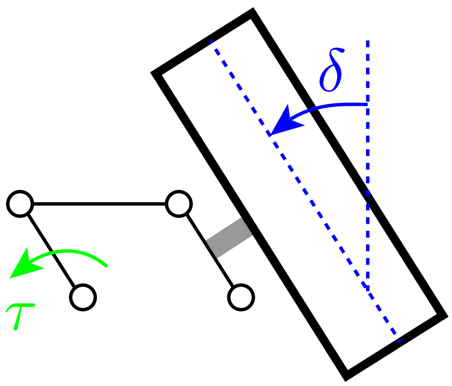
Stanford's student-built steer-by-wire cars (from Prof. Chris Gerdes' lab) run the steering control loop at 500 Hz (sampling 500 times a second) and use a 5000 counts-per-revolution encoder with a gear reduction of 160:1.



With this sampling rate and sensor resolution, we can easily use the continuous system techniques of this class to analyze the system. Furthermore, we can approximately obtain a PD controller by differencing the position output.

If we assume that the motor dynamics are much faster than those of the wheel itself, we can model the control input as our motor torque.

A schematic of the system is shown below. (This is a top view, looking down at a wheel that is steered to an angle of δ.

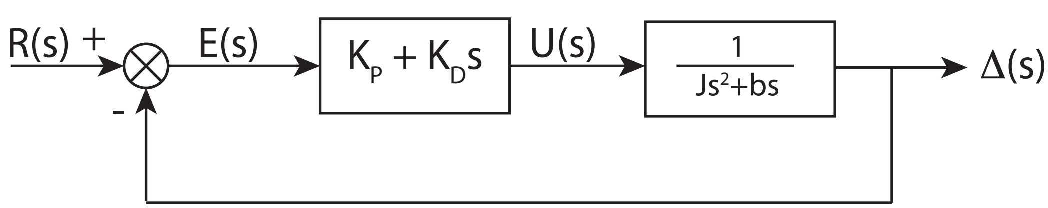


$$\tau = J(\delta)\ddot{\delta} + b(\delta)\dot{\delta}$$

By paying attention to the linkage design (close to a parallelogram), we have made $J(\delta)$ and $b(\delta)$ roughly constant, so a reasonable model is:

$$\tau = J\ddot{\delta} + b\dot{\delta}$$

The block diagram of the system is:



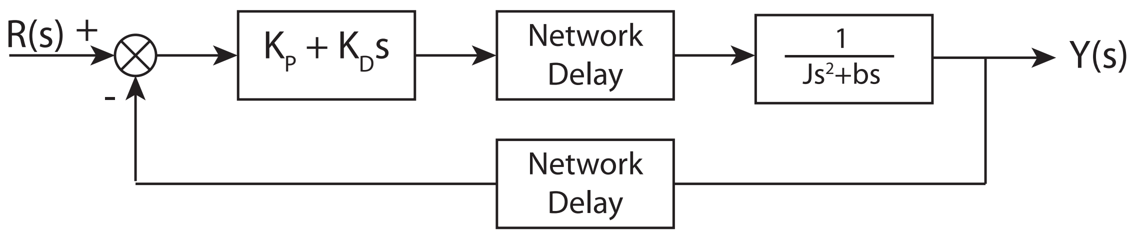
$$\frac{\Delta(s)}{T(s)} = \frac{1}{Js^2 + bs}$$

With fast sampling, high sensor resolution and enough torque, this becomes a very simple control problem. This was intentional since the students thought about "design for controllability" throughout the process. They applied similar ideas in developing the controller for the steering control on an Audi TT-S. This had to work with the steering system and electric power steering already in the car, so they could not use mechanical design to make life easier.

In modern cars, control systems are networked using an interface such as CAN (controller area network) or FlexRay. Waiting for CAN messages can add delay to the system. While this sort of problem is often addressed best using the principles of digital control (where we model sampling times explicitly and think of differences instead of derivatives), our E105 techniques show the fundamental limits well.

2 Modeling time delay

The CAN network introduces delay between the steering angle and our measurement and between our torque command and the torque sent form the motor:



How do we model the effect of this delay? If we have $f(t)$ coming into a delay of time T_d, we have $f(t - T_d)$ coming out:

$$\mathcal{L}\left\{f\left(t-T_d\right)\right\} = \int_0^\infty f\left(t-T_d\right) e^{-st}\ \mathrm{d}t$$

$$= \int_0^\infty f\left(t-T_d\right) e^{-s(t-T_d)} e^{-sT_d}\ \mathrm{d}t$$

$$= e^{-sT_d} \int_0^\infty f\left(t-T_d\right) e^{-s(t-T_d)}\ \mathrm{d}t$$

$$= e^{-sT_d}\, \mathcal{L}\left\{f(t)\right\} \qquad \text{if } f(t) = 0 \text{ for } t < 0$$

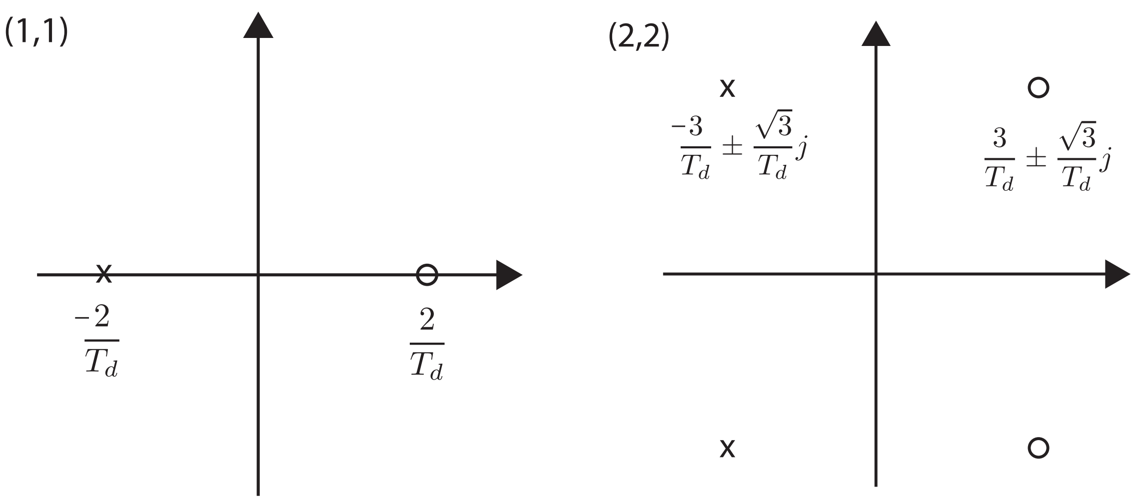
So a pure time delay is equivalent to multiplication by $e^{-T_d s}$ in the Laplace domain. (See FPE6 Example A.1, page 760 for a more detailed derivation.)

Since we find it easier to work with polynomials, this can be approximated as a ratio of polynomials. The Padé approximation can be carried out at different levels of precision:

$$(1,1) \text{ approximation: } e^{-T_d s} \approx \frac{1 - \left(\frac{T_d s}{2}\right)}{1 + \left(\frac{T_d s}{2}\right)}$$

$$(2,2) \text{ approximation: } e^{-T_d s} \approx \frac{1 - \frac{T_d s}{2} + \frac{(T_d s)^2}{12}}{1 + \frac{T_d s}{2} + \frac{(T_d s)^2}{12}}$$

These approximate the delay as adding left half plane poles and right half plane zeros:



As $T_d \to 0$, the poles and zeros move further from the origin.

For the $(2,2)$ approximation:

$$\text{zeros @} \quad s = \frac{\frac{T_d}{2} \pm \sqrt{\frac{{T_d}^2}{4} - \frac{{T_d}^2}{3}}}{\frac{2{T_d}^2}{12}}$$

$$= \frac{12T_d}{4{T_d}^2} \pm \frac{6}{{T_d}^2}\sqrt{\frac{3{T_d}^2 - 4{T_d}^2}{12}}$$

$$= \frac{3}{T_d} \pm \frac{\sqrt{3}}{T_d} j$$

3 System behavior with time delay

We can combine delays to make this analysis easier. If each network delay block has $\frac{T_d}{2}$, the closed loop system looks like

$$\frac{Y(s)}{R(s)} = \frac{G(s)D(s)Q(s)}{1 + G(s)D(s)Q^2(s)}$$

$$Q(s) = e^{\frac{-T_d}{2}s} \qquad\qquad \text{This is a delay of } \frac{T_d}{2}$$

The characteristic equation is thus:

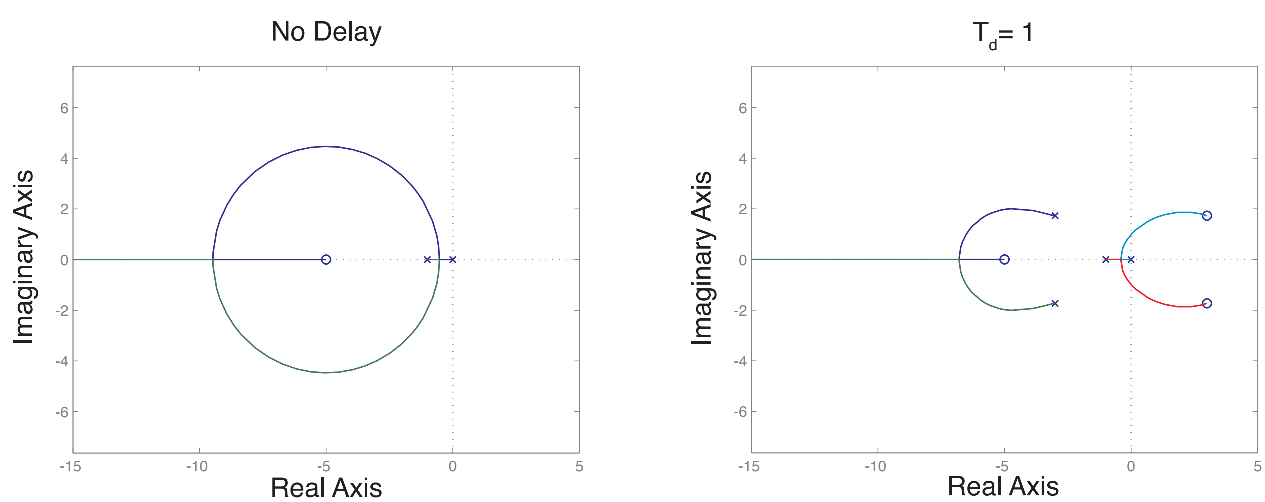
$$1 + G(s)D(s)Q^2(s) = 0$$

$$1 + \frac{K_p + K_d s}{Js^2 + bs} \cdot e^{-T_d s} = 0$$

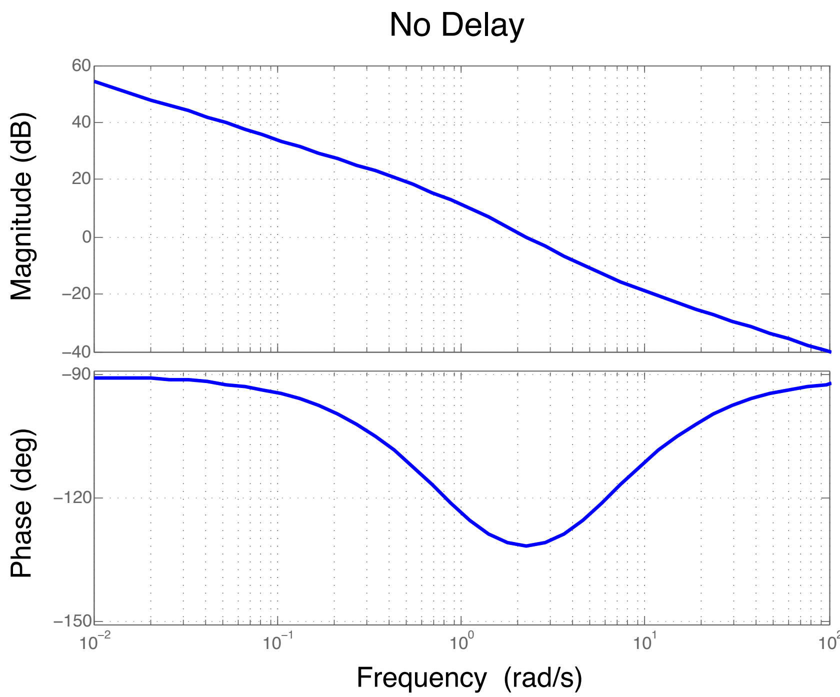
$$1 + \frac{K_d}{J} \frac{\left(s + \frac{K_p}{K_d}\right)}{s(s + \frac{b}{J})} \cdot \frac{1 - \frac{T_d s}{2} + \frac{(T_d s)^2}{12}}{1 + \frac{T_d s}{2} + \frac{(T_d s)^2}{12}} = 0$$

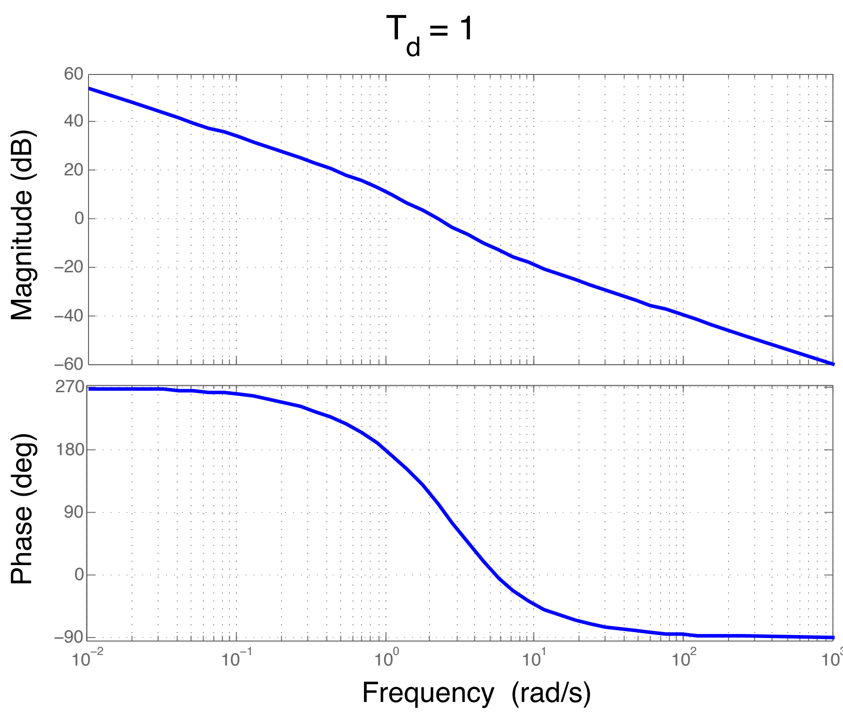
The effect of the delay depends upon its magnitude. As $T_d \to 0$, the effect becomes negligible. As T_d increases, it can eventually dominate the response.

In any case, the RHP zeros spell trouble as the gain increases. Hence, delay is a performance limit. The following root loci show the system behavior with no delay and delay of $T_d = 1$ second.

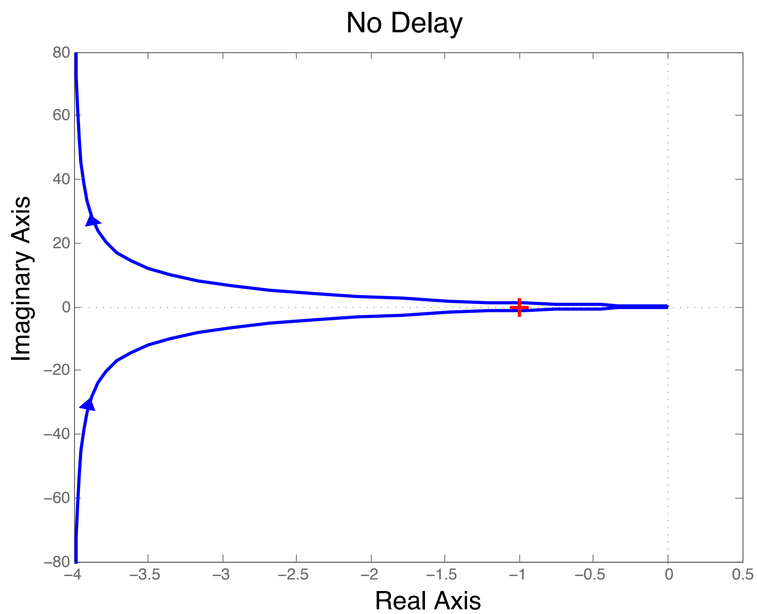


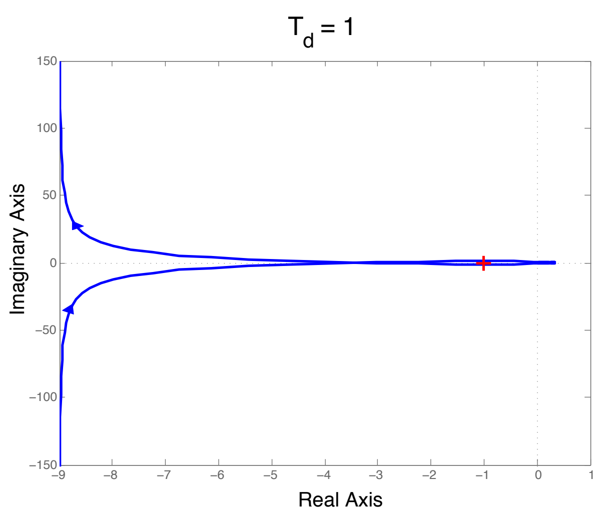
In addition, we can consider the Bode plots of this system with and without time delay:

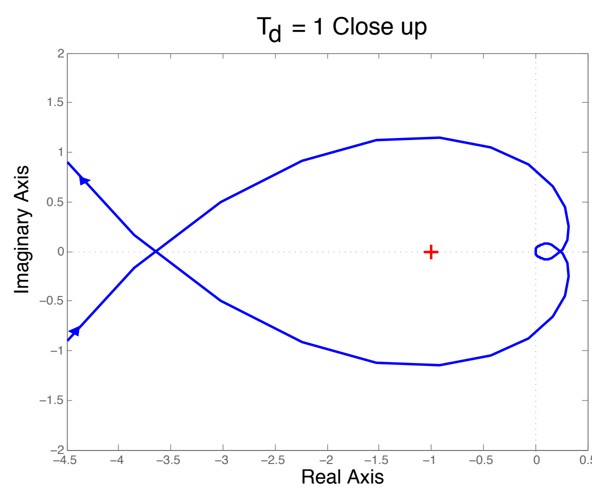




Finally, let's take a loot at the Nyquist plots of this system with and without time delay:







Matlab code:

```
clear all
clc
kp = 5;
kd = 1;
j = 1;
b = 1;
td = 1;

%% Root loci

sys1 = zpk([-kp/kd],[0 -b/j],1)
rlocus(sys1)
title('No Delay')
xlim([-15 5])
axis equal

sys2 = zpk([-kp/kd 3/td+sqrt(3)/td*i 3/td-sqrt(3)/td*i],[0 -b/j -3/td+sqrt(3)/td*i -3/td-sqrt(3)/td*i],1)
figure(2)
rlocus(sys2)
title('T_d = 1')
axis equal

%% Bode plots

figure(3)
bode(sys1)
title('No Delay')
grid on

figure(4)
bode(sys2)
title('T_d = 1')
grid on

%% Nyquist plots

figure(5)
nyquist(sys1)
title('No Delay')

figure(6)
nyquist(sys2)
title('T_d = 1')

figure(7)
nyquist(sys2)
title('T_d = 1 Close up')
axis([-4.5 0.5 -2 2])
```