ECE 3510 PID Tuning

A.Stolp a
3/24/20

For some time now we have been studying root-locus plots because these plots give us information about the closed-loop system response using only the open-loop transfer function and the system gain. We even extended the basic idea so that we can draw unconventional root-locus plots for variables other than gain. We also found that we could use the knowledge of how the open-loop poles and zeros affected the closed-loop response to design compensators which added new poles and zeros. The proportional-integral-differential (PID) compensator turned out to be especially useful and, not surprisingly, is also one of the most and common. All of this depended on knowing the open-loop transfer function. This makes it look like we can't get the benefits of a PID compensator (or controller) without the transfer function. Oh, how wrong you are. Smarter people than us found that you can put an adjustable PID in a feedback system and twiddle the knobs 'til you get the response you want. Even smarter people than them developed ways to get a good starting settings for the knobs and more systematic ways to twiddle from there. These methods are called "PID Tuning" and you should be aware of their existence.

Ziegler-Nichols PID Tuning Methods

Reaction-Curve Method Measurements are made on the **open-loop** system to determine controller parameters.

Can only be used:

1. Open-loop system is stable, and it's step response doesn't ring. (Typically worded as "doesn't have integrators or dominant complex-conjugate poles".)
2. The open-loop system (without any feedback), has a simple S-shaped unit step response like the one shown below. This curve is called "the reaction curve".

Measurements

Draw a tangent line at the inflection point of the curve.

Measure A, L, and T, as shown.

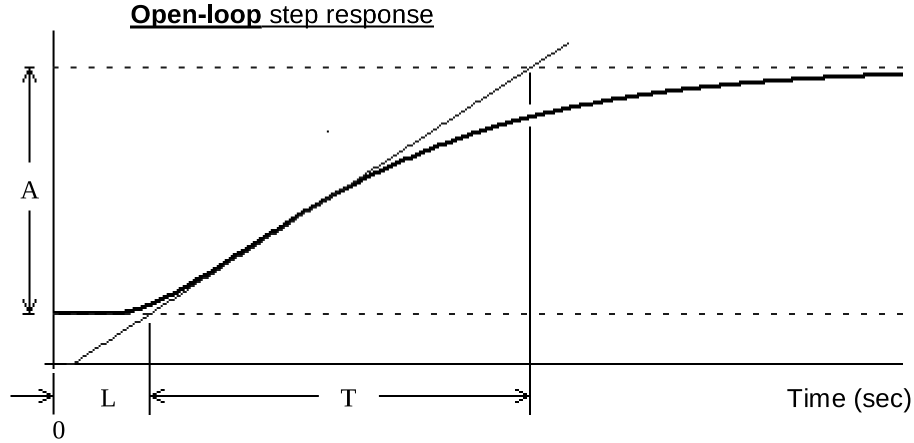
Calculate the slope at inflection point:

$$R = \frac{A}{T}$$

If the input is not a unit step (is a step of x_m instead of 1), modify R like this:

$$R = \frac{A}{x_m \cdot T}$$

Open-loop step response



The units of R should come out to be $\frac{1}{\text{sec}}$

Decide on what type of controller you would like to use. Proportional only, Proportional with Integral (PI) to eliminate steady-state error, OR Full PID to inprove dynamic response as well.

Type of Controller	Parameters of Controller (These are initial settings and may be subject to finer adjustments later)		
Proportional	$k_p = \frac{1}{R \cdot L}$	$k_i = 0$ k_i and k_d are both 0 because this is just proportional control	$k_d = 0$
PI	$k_p = \frac{0.9}{R \cdot L}$	$k_i = k_p \cdot \frac{0.3}{L} = \frac{0.27}{R \cdot L^2}$ OR $T_I = \frac{L}{0.3}$	$k_d = 0$
PID	$k_p = \frac{1.2}{R \cdot L}$	$k_i = \frac{k_p}{2 \cdot L} = \frac{0.6}{R \cdot L^2}$ OR $T_I = 2 \cdot L$	$k_d = \frac{k_p \cdot L}{2}$ OR $T_D = \frac{L}{2}$

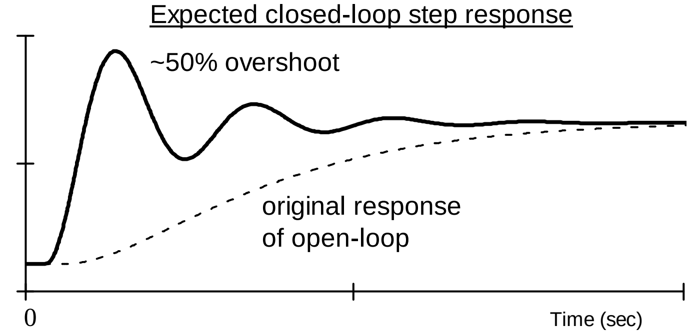
Controller transfer function

$$\mathbf{C}(s) = k_p \cdot \left(1 + \frac{1}{T_I \cdot s} + T_D \cdot s\right)$$ using T parameters (eq. 4.110 in other notes)

$$= k_p + \frac{k_p}{T_I \cdot s} + k_p \cdot T_D \cdot s$$

$$= k_p + \frac{k_i}{s} + s \cdot k_d$$ using k parameters, like we do in our class

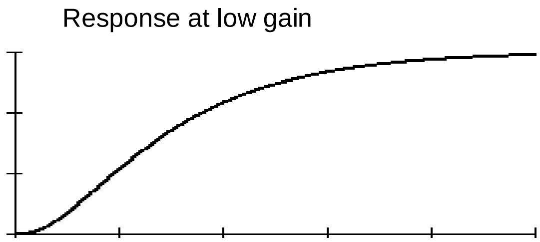
Expected closed-loop step response

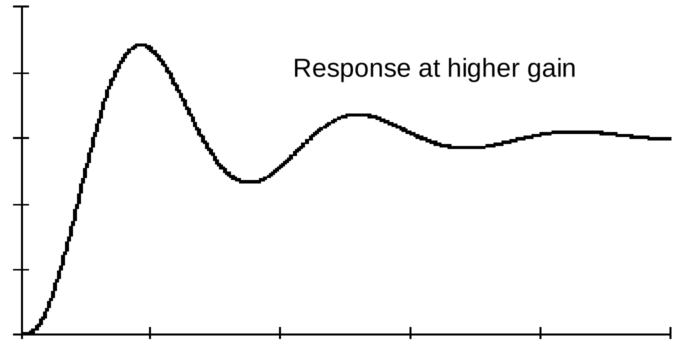


Ultimate-Sensitivity Method Measurements are made on the **closed-loop** system to determine controller parameters.

1. Can be used when the open-loop system is unstable, and requires feedback to be stable.
2. Use **only proportional gain** to make initial measurements.

Slowly turn up the gain while observing the step response.

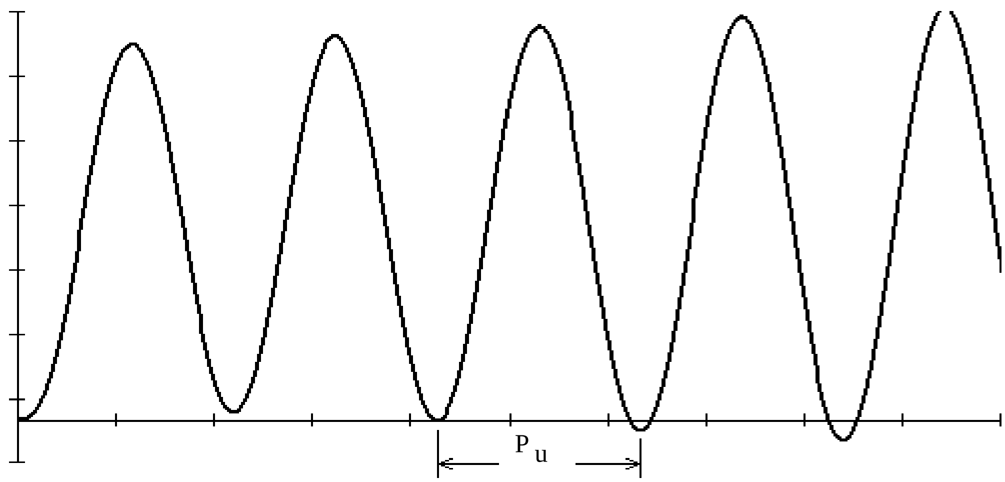




Response at the "ultimate gain" (K_U), when the system begins to oscillate.

Record the "ultimate gain" (K_U).

Measure the "ultimate period" (P_U), of the oscillation.



Decide on what type of controller you would like to use.

Type of Controller	Parameters of Controller (These are initial settings and may be subject to finer adjustments later)		
Proportional	$k_p = 0.5 \cdot K_u$	$k_i = 0$ k_i and k_d are both 0 because this is just proportional control	$k_d = 0$
PI	$k_p = 0.45 \cdot K_u$	$k_i = k_p \cdot \frac{1.2}{P_u}$ OR $T_I = \frac{P_u}{1.2}$	$k_d = 0$
PID	$k_p = 0.6 \cdot K_u$	$k_i = k_p \cdot \frac{2}{P_u}$ OR $T_I = \frac{P_u}{2}$	$k_d = k_p \cdot \frac{P_u}{8}$ OR $T_D = \frac{P_u}{8}$

The Ziegler-Nichols PID Tuning Methods usually result in systems that have quite a bit of overshoot, so you will probably want to make minor adjustments after the initial settings.

Effects of increasing a parameter independently

Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
k_p	Decrease	Increase	Small change	Decrease	Degrade
k_i	Decrease	Increase	Increase	Eliminate faster	Degrade
k_d	Little effect	Decrease	Decrease	Little effect	Improve if k_d is small

More about PID Tuning: https://cdn.instructables.com/ORIG/FC1/NAZC/IVA51KF1/FC1NAZCIVA51KF1.pdf
(and source material) https://en.wikipedia.org/wiki/PID_controller#PID_tuning_software
Google "PID Tuning" for much more.