

2.004: Modeling, Dynamics, & CONTROL II

Spring Term 2003

Experiment 5: MatLab Tutorial : Control Toolbox

Introduction

In this third tutorial session, we will examine some important tools provided by MatLab in designing control systems. The control toolbox in MatLab provides some very useful tools in representing and manipulating transfer functions. This program also has convenient methods for visualizing the responses of linear time-invariant (LTI) systems. In particular, we will examine the use of LTI viewer. We will consider frequency domain feedback controller design using Root Locus Design GUI.

The goals of this tutorial session are:

- (1) Familiarize basic MatLab capability in designing control systems. Familiarize with basic function calls such as `tf()`, `series()`, `feedback()`, `step()` etc.
- (2) Familiar with two useful MatLab graphic interfaces in the control toolbox: LTI Viewer and Root Locus Tool.
- (3) Understand Root Locus representation graphically using `rltool`.
- (4) Modifying system root locus by adding poles and zeros.

Procedure:

Consider second order transfer functions:

$$F_1(s) = \frac{1056}{0.85s^2 + 14s + 1056}$$

$$F_2(s) = \frac{132000}{0.85s^2 + 14s + 132000}$$

- (a) Define these transfer function in MatLab using function TF.
- (b) Visualize the step response of these transfer function using step().
- (c) Familiarize LTI Viewer.
 - (i) Plot step, impulse responses marking peak response, settling time, rise time, and steady state as appropriate.
 - (ii) Create Bode plots for both transfer functions.
 - (iii) Plot poles and zeros for both transfer functions.

(d) Create transfer function with constant gain:

$$K(s) = kI \text{ for } kI=1, 10, 100.$$

- (i) Create transfer functions cascading $F_1(s)$ and $K(s)$ with the three kI values. Find pole locations for all three cascaded functions. Where are the zeros. Try both pole() and zero() functions and try using LTI Viewer.
- (ii) Incorporate unity feedback of all three cascaded functions [hint: use function feedback()]. Again, identify poles and zeros locations.
- (iii) Observe the effect of changing cascade gain using root locus tool.
- (iv) How does changing proportional feedback gain affect system dynamics? Observe step responses.

(e) Create transfer function with a zero:

$$D(s) = s + k2 \text{ for } k2 = 1, 10, 100$$

- (i) Create unity feedback transfer functions with cascading differential controller with the three $k2$ values.
- (ii) Identify pole and zero locations and step responses for all three cases. Try creating these functions either by hand [i.e. using series() and feedback()]. However, you will find that using root locus tool is much more convenient.
- (iii) In root locus tool, try examining the effect of changing gain on system response.
- (iv) Describe the effect of adding a zero on root locus.

(f) Create transfer function with an additional pole:

$$I(s) = \frac{1}{s + k3} \text{ for } k3 = 1, 10, 100$$

- (i) Create unity feedback transfer functions with cascading integral controller with the three $k3$ values.
 - (ii) Identify pole and zero locations and step responses for all three cases. Try creating these functions either by hand [i.e. using `series()` and `feedback()`]. However, you will find that using root locus tool is much more convenient.
 - (iii) In root locus tool, try examining the effect of changing gain on system response.
 - (iv) Describe the effect of adding a pole on root locus.
- (g) Time permitting, you may examine the effect of adding additional poles and zeros.

Deliverables:

Poles and zeros locations for the relevant transfer functions. Create appropriate step response plots, impulse response plots, and bode plots for these transfer functions.