Summary of "A Unified Procedure for Discrete-Time Root Locus and Bode Design"

Trevor Hardy
School of Electrical and Computer Engineering
Wichita State University

I. Introduction

This is a summary of the paper "A Unified Procedure for Discrete-Time Root Locus and Bode Design." [1] Traditionally there have been two basic approaches to designing discrete-time controllers: design in the continuous-time (s) domain and then convert the design to discrete-time (z) domain (the indirect method) or develop the controller in the z-domain (the direct method). The former is used because design techniques are well established for the s-domain as is the process for conversion of the design to the z-domain. The technique, though, has a fundamental flaw: the act of sampling to move from continuous time to discrete time introduces mathematical artifacts that can potentially significantly alter the discrete time response of the system. As an alternative to this imperfect approach, this article provides a framework for direct z-domain design of the classical s-domain controllers (PD, lag compensator, etc...). The framework provided is outlined for using both root locus and Bode plot techniques.

The starting point for these designs is the s-domain design point which is converted to z-domain by:

$$z_0 = \begin{cases} e^{s_d T} & \text{root locus,} \\ e^{j\omega_{gc} T} & \text{Bode} \end{cases} \tag{1}$$

where s_d is the root locus s-domain design point and ω_{gc} is the Bode plot gain cross-over frequency. Using these, a generalized form of the angle and magnitude criterion can be formed as

$$\begin{aligned} \angle G_c(z_0) + \angle G_{sys}(z_0) = \phi \\ K|G_c(z_0)G_{sys}(z_-)| = 1 \end{aligned} \tag{2}$$

where G_c is the compensator transfer function and G_{sys} is the combined actuator, plant and sensor transfer functions and where the desired angle is

$$\phi = \begin{cases} 180^\circ & \text{root locus,} \\ 180^\circ + \text{phase margin} & \text{Bode} \end{cases} \tag{3}$$

Using these the compensator angle θ_c can be computed as follows:

$$\theta_c = \angle G_c(z_0) = \phi - \angle G_{sys}(z_0) \tag{4}$$

Note that θ_c can be determined prior to choosing a specific compensator type. The basic compensator for this approach is $G_c(z) = z - \alpha$. To find the location of the zero we use

$$\angle G_c(z_0) = \angle(z_0 - \alpha) = tan^{-1}\left(\frac{y_0}{x_0 - \alpha}\right) \tag{5}$$

where $z_0 = x_0 + jy_0$ Using our designed compensator angle θ_c the compensator zero can be found by

$$\alpha = x_0 - \frac{y_0}{tan(\theta_c)} \tag{6}$$

II. PD Compensator

In the discrete domain the PD compensator takes the basic compensator design and adds a pole at the origin. The PD compensator has a transfer function of the form

$$G_c(z) = \frac{z - \alpha}{z} \tag{7}$$

Using our known desired compensator angle θ_c we can see that

$$\angle G_c(z_0) = \angle(z_0 - \alpha) - \angle z_0 = \angle\theta_c \tag{8}$$

becomes

$$\angle(z_0 - \alpha) = \theta_c + \angle z_0 = \theta_{c,PD} \tag{9}$$

where $\theta_{c,PD}$ is the PD compensator angle. Since both θ_c and z_0 are known quantities, using (6) we can see that

$$\alpha_{PD} = x_0 - \frac{y_0}{tan(\theta_{c,PD})} \tag{10}$$

will define the location of the PD compensator zero to achieve our compensator angle.

III. PI Compensator

The general form of the discrete time PI compensator is

$$G_c(z) = \frac{z - \alpha}{z - 1} \tag{11}$$

As in above, we can use our known desired compensator angle θ_c at the chosen design point to find

$$\angle(z_0 - \alpha) = \theta_c + \angle(z_0 - 1) = \theta_{c,PI} \tag{12}$$

where $\theta_{c,PI}$ is the PI compensator angle. Since both θ_c and z_0 are known quantities, using (6) we can see that

$$\alpha_{PI} = x_0 - \frac{y_0}{tan(\theta_{c,PI}) + \frac{y_0}{x_0 - 1}} \tag{13}$$

will define the location of the PI compensator zero to achieve our compensator angle.

IV. Controller/Compensator Overview

In addition to the above two specific compensators, the paper presents design templates or procedures for an additional four others, making a total of six. Below is an outline of all six controllers/compensators along with their transfer function and the typical uses of said compensators.

Compensator type: Lag
Transfer function: $G_c(z) = \frac{z-\alpha}{z-\beta}$ with $\alpha > \beta$
Uses: Help correct (but not eliminate as in the PI or PID controller) steady-state error. [3]

Compensator type: Lead
Transfer function: $G_c(z) = \frac{z-\alpha}{z-\beta}$ with $\alpha < \beta$
Uses: Compensate for destabilizing phase shifts which leads to increased stability. Increases response time of the system. [3]

Compensator type: PD
Transfer function: $G_c(z) = \frac{z-\alpha}{z}$
Uses: Decreases rise time and overshoot. [2]

Compensator type: PI
Transfer function: $G_c(z) = \frac{z-\alpha}{z-1}$
Uses: Increase response time and eliminate steady-state error. [2]

Compensator type: PID
Transfer function: $G_c(z) = \frac{(z-\alpha_1)(z-\alpha_2)}{z(z-1)}$
Uses: When the proportional, integral, and derivative gains are properly balanced, the integral term achieves zero steady-state error while the derivative term provides reduced rise time and and overshoot control. [2]

Compensator type: PI-Lead
Transfer function: $G_c(z) = \frac{(z-\alpha_1)(z-\alpha_2)}{(z-\beta)(z-1)}$
Uses: Compensate for destabilizing phase shifts, increase response time, eliminate steady-step error. [3] [2]

References

[1] Emami, T.; Watkins, J.M.; O'Brien, R.T., "A Unified Procedure for Continuous-Time and Discrete-time Root-Locus and Bode Design," American Control Conference, 2007. ACC '07 , vol., no., pp.2509-2514, 9-13 July 2007

[2] DK, "PID Tutorial", Aug. 1997; http://www.engin.umich.edu/group/ctm/PID/PID.html.

[3] JDP, "Designing Lead and Lag Compensators", Aug. 1996; http://www.engin.umich.edu/group/ctm/extras/lead.lag.html