ELEC ENG 4CL4:

Control System Design

Notes for Lecture #18

Friday, February 13, 2004

Dr. Ian C. Bruce
Room: CRL-229
Phone ext.: 26984
Email: ibruce@mail.ece.mcmaster.ca

PI and PID Synthesis Revisited using Pole Assignment

The reader will recall that PI and PID controller synthesis using classical methods were reviewed in Chapter 6. In this section we place these results in a more modern setting by discussing the synthesis of PI and PID controllers based on pole assignment techniques.

We begin by noting that any controller of the form

$$C(s) = \frac{n_2 s^2 + n_1 s + n_o}{d_2 s^2 + d_1 s}$$

is identical to the PID controller, where

$$K_p = \frac{n_1 d_1 - n_o d_2}{d_1^2}$$
$$K_I = \frac{n_o}{d_1}$$
$$K_D = \frac{n_2 d_1^2 - n_1 d_1 d_2 + n_o d_2^2}{d_1^3}$$
$$\tau_D = \frac{d_2}{d_1}$$

Hence all we need do to design a PID controller is to take a second order model of the plant and use pole assignment methods.

Example

A plant has a nominal model given by

$$G_o(s) = \frac{2}{(s+1)(s+2)}$$

Synthesize a PID controller which yields a closed loop with dynamics dominated by the factor $s^2 + 4s + 9$.

Solution

The controller is synthesized by solving the pole assignment equation, with the following quantities

$$A_{cl}(s) = (s^2 + 4s + 9)(s + 4)^2; \quad B_o(s) = 2; \quad A_o(s) = s^2 + 3s + 2$$

Solving the pole assignment equation gives

$$C(s) = \frac{P(s)}{s\bar{L}(s)} = \frac{14s^2 + 59s + 72}{s(s + 9)}$$

We observe that C(s) is a PID controller with

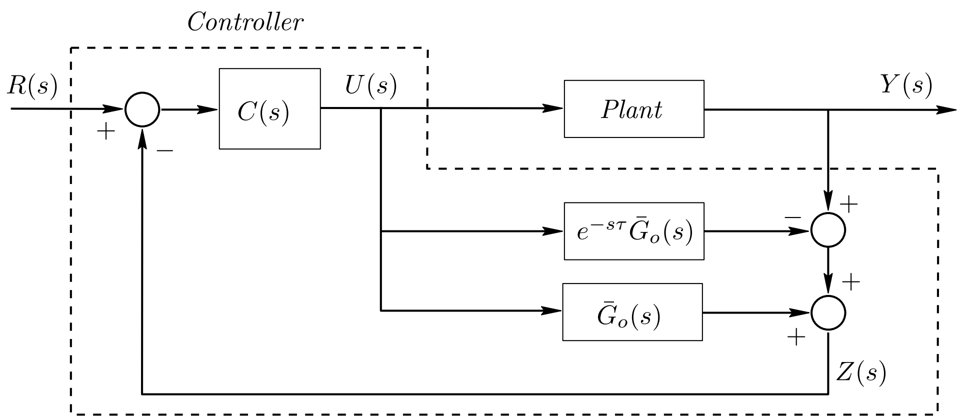
$$K_p = 5.67; \quad K_I = 8; \quad K_D = 0.93; \quad \tau_D = 0.11$$

Smith Predictor

Since time delays are very common in real world control problems, it is important to examine if one can improve on the performance achievable with a simple PID controller. This is specially important when the delay dominates the response.

For the case of *stable* open loop plants, a useful strategy is provided by the Smith predictor. The basic idea here is to build a parallel model which cancels the delay, see figure 7.1.

Figure 7.1: *Smith predictor structure*



We can then design the controller using a a pseudo complementary sensitivity function, $T_{zr}(s)$, between r and z which has no delay in the loop. This would be achieved, for example, via a standard PID block, leading to:

$$T_{zr}(s) = \frac{\bar{G}_o(s)C(s)}{1 + \bar{G}_o(s)C(s)}$$

In turn, this leads to a nominal complementary sensitivity, between r and y of the form:

$$T_o(s) = e^{-s\tau}T_{zr}(s)$$

Four observations are in order regarding this result:

(i) Although the scheme appears somewhat ad-hoc, it will be shown in Chapter 15 that the architecture is inescapable in so far that it is a member of the set of *all possible stabilizing controllers* for the nominal system.

(ii) Provided $\tilde{G}_0(s)$ is simple (e.g. having no nonminimum phase zero), then $C(s)$ can be designed to yield $T_{zr}(s) \approx 1$. However, we see that this leads to the ideal result $T_0(s) = e^{-s\tau}$.

(iii) There are significant robustness issues associated with this architecture. These will be discussed later.

(iv) One cannot use the above architecture when the open loop plant is unstable. In the latter case, more sophisticated ideas are necessary.

Summary

- This chapter addresses the question of synthesis and asks: *Given the model $G_0(s) = B_0(s)/A_0(s)$, how can one synthesize a controller, $C(s) = P(s)/L(s)$ such that the closed loop has a particular property.*
- Recall:
  - the poles have a profound impact on the dynamics of a transfer function;
  - the poles of the four sensitivities governing the closed loop belong to the same set, namely the roots of the characteristic equation $A_0(s)L(s) + B_0(s)P(s) = 0$.

- Therefore, a key synthesis question is:
  Given a model, can one synthesize a controller such that the closed loop poles (i.e. sensitivity poles) are in pre-defined locations.

- Stated mathematically:
  Given polynomials $A_0(s)$, $B_0(s)$ (defining the model) and given a polynomial $A_{cl}(s)$ (defining the desired location of closed loop poles), is it possible to find polynomials $P(s)$ and $L(s)$ such that $A_0(s)L(s) + B_0(s)P(s) = A_{cl}(s)$? This chapter shows that this is indeed possible.

- The equation $A_0(s)L(s) + B_0(s)P(s) = A_{cl}(s)$ is known as a Diophantine equation.

- Controller synthesis by solving the Diophantine equation is known *as pole placement*. There are several efficient algorithms as well as commercial software to do so

- Synthesis ensures that the emergent closed loop has particular constructed properties (namely the desired closed loop poles).
  - However, the overall system performance is determined by a number of further properties which are consequences of the constructed property.
  - The coupling of constructed and consequential properties generates trade-offs.

- Design is concerned with
  - Efficient detecting if there is no solution that meets the design specifications adequately and what the inhibiting factors are,
  - Choosing the constructed properties such that, whenever possible, the overall behavior emerging from the interacting constructed and the consequential properties meets the design specifications adequately.
- This is the topic of the next chapter.