Digital Control

Summary # 8: Proportional, Integral and Derivative: Tuning methods

![Digital Control Diagram]

**Fig. 1. Ideal and actual proportional control**

Proportional control

The error between the desired set point and the actual output is defined as

\[ e(t) = r(t) - y(t) \]  (1)

where \( r(t) \) is the desired set point. The set point can be constant or time varying. The controller output is given by

\[ u(t) = u_0 + K_p e(t) \]  (2)

where \( u_0 \) is the bias and \( K_p \) is the controller gain. The actual behavior of the proportional controller is illustrated in figure 1. The saturation shown in the figure is an important limitation that needs to be taken into account.

**PI control**

The proportional integral control is widely used because of its important practical advantage: it eliminates the steady state error. The continuous time transfer function of a PI controller is

\[ K_p \left( 1 + \frac{1}{\tau_i s} \right) \]  (3)

PD control

PD control has two terms, a proportional term and a derivative term. This controller is also called rate action, pre–act control and anticipatory control. The PD continuous time transfer function is

\[ K_p \left( 1 + \tau_D s \right) \]  (4)

Proportional, integral and derivative.

PI and PID control have been dominant control algorithms for process control for decades (about 95% of process controllers utilize some form of PID). The continuous time transfer function for PID controller is given by

\[ K_p \left( 1 + \frac{1}{\tau_i s} + \tau_D s \right) \]  (5)

There exist several variants of the PID control.

**Position and velocity algorithms for digital controllers**

Two alternatives for digital PID control exist, namely position form and velocity form.

**Position algorithm**

The discrete time equation for the digital PID controller output is given by

\[ u(k) = u_0 + K_p \left[ e(k) + \frac{T}{\tau_I} \sum_{j=1}^{k} e(j) + \frac{\tau_D}{T} (e(k) - e(k-1)) \right] \]  (6)

where \( T \) is the sampling time. This form is called position algorithm.

**Velocity algorithm**

In the position PID control, we calculated the actual value of the PID controller output. In the velocity form we calculate the change in the controller output. The velocity form can be derived from the position form as follows:

\[ u(k) = u_0 + K_p \left[ e(k-1) + \frac{T}{\tau_I} \sum_{j=1}^{k-1} e(j) \right] + K_p \left[ \frac{\tau_D}{T} (e(k-1) - e(k-2)) \right] \]  (7)

The velocity form is based on the difference given by

\[ \Delta u = u(k) - u(k-1) \]  (8)

and therefore

\[ \Delta u = K_p \left[ e(k) - e(k-1) + \frac{T}{\tau_I} e(k) \right] + \left[ \frac{\tau_D}{T} (e(k) - 2e(k-1) + e(k-2)) \right] \]  (9)
Proportional, Integral and Derivative: Tuning methods

Digital Controls, spring 2015

- Robust, the closed loop system is insensitive to changes in the plant conditions and molding inaccuracies.
- Good disturbance rejection

It is not possible to achieve all these goals simultaneously because of the conflicts and trade offs that may exist.

Rearranging the terms and solving for the controller transfer function yields the following equation

$$G_c(s) = \frac{1}{G(s)} \frac{Y(s)}{R(s)}$$

(15)

In order to solve for the controller’s transfer function we need to know:
- the desired closed loop transfer function
- the open loop transfer function.

We can define a desired closed loop transfer function based on the desired characteristics of the system. We call this function

$$\left( \frac{Y(s)}{R(s)} \right)_d$$

(16)

Assuming we can determine an approximation $\hat{G}$ of the open loop transfer function, the controller equation becomes

$$G_c(s) = \frac{1}{G(s)} \left( \frac{Y(s)}{R(s)} \right)_d$$

(17)

Problems with PID controllers: Derivative kick and integral windup

A sudden change in the error that usually results from the change in the set point will cause the derivative part to become very large. This spike is undesirable. One way to solve this issue is by using $\frac{\text{d}u}{\text{d}t}$ instead of $\frac{\text{d}e}{\text{d}t}$. Integral windup (or reset windup) is another problem that happens when the integral output becomes large and the controller output becomes saturated. The build up of the integral terms is called integral windup. Consider the time response of figure 2. In figure 2–top, the integral term initially increases but begins decreasing again when the error changes sign. The positive and negative terms cancel out and the controller moves away from the saturation point. In figure 2–bottom, the integral term keeps increasing and this may result in integral wind up. The PID velocity form is called anti-integral windup because of the summation is eliminated.

Performance criteria for closed loop systems

The performance criteria are as follows:
- The closed loop system is stable
- Steady state error is eliminated
- Good transient response

PID control design methods

Several techniques are used:
- Model based design methods
- Controller tuning relationships
- Frequency response techniques
- Computer simulations
- Online tuning

Methods 1-4 are based on an approximation of the plant model and method 5 (online tuning) is purely experimental. Methods based on the plant model allow for initial setting of the controller. Computer simulation (Matlab and Labview) allow for comparison between the different alternatives.

Model based design methods

Two methods are discussed here: the direct synthesis method and the internal model control.

Direct synthesis method

The controller is designed based on the desired closed loop transfer function. The method can be used to design but not limited to PID controllers. The closed loop transfer function is given by

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}$$

(14)

where
- $G_c(s)$ is the controller transfer function
- $G(s)$ is the open loop transfer function.

As a result of the conflicts and trade offs that may exist.

It is possible to obtain an explicit formula for the controller as follows

$$u(k) = u(k - 1) + K_p \left[ e(k) - e(k - 1) + \frac{T}{\tau_i} e(k) \right]$$

(12)

$$+ K_p \left[ \frac{T_d}{T} (e(k) - 2e(k - 1) + e(k - 2)) \right]$$

(13)

Problems with PID controllers: Derivative kick and integral windup

As follows
Several possibilities exist for the desired closed loop system, ideally we want the closed loop response to be equal to the input, that is
\[
\left( \frac{Y(s)}{R(s)} \right)_d = 1 \tag{18}
\]
This is too perfect and is not possible in practice. A more realistic approach is to use a first order system.

**First order system**

In this case the desired closed loop system is a simple first order system
\[
\left( \frac{Y(s)}{R(s)} \right)_d = \frac{1}{\tau_c s + 1} \tag{19}
\]
where \( \tau_c \) is the closed loop time constant. The controller equation is reduced to the following equation
\[
G_c = \frac{1}{G(s) \tau_c s} \tag{20}
\]
If the process has a delay, we can use the closed loop transfer function with delay
\[
\left( \frac{Y(s)}{R(s)} \right)_d = \frac{e^{-\theta s}}{\tau_c s + 1} \tag{21}
\]
where \( \theta \) is a time delay. The equation for the controller is
\[
G_c = \frac{1}{G(s) (\tau_c + \theta)s} \tag{22}
\]
Assuming a small delay, the delay can be approximated as
\[
e^{-\theta s} = 1 - \theta s \tag{23}
\]
and the controller as
\[
G_c = \frac{1}{G(s) (\tau_c + \theta)s} \tag{24}
\]
Clearly, equations (20) and (24) depend on the approximation of the open loop system. This discussed below.

**A. Open loop transfer function approximation**

Several methods exist to approximate the open loop transfer function such as the step response and the frequency response. Here we consider first and second order approximations with delay.

**B. First order approximation**

The open loop system is approximated by a first order transfer function as follows
\[
\tilde{G}(s) = \frac{Ke^{-\theta s}}{(\tau s + 1)} \tag{25}
\]
where \( \tau \) is the open loop time constant. Using equation (24) and knowing that the controller's equation is
\[
G_c = K_p \left( 1 + \frac{1}{\tau_I s} \right) \tag{26}
\]
we get
\[
K_p = \frac{\tau}{K(\tau_c + \theta)} \tag{27}
\]
\[
\tau_I = \tau \tag{28}
\]

**C. Second order approximation**

The transfer function is approximated by a second order transfer function as follows
\[
\tilde{G}(s) = \frac{Ke^{-\theta s}}{(\tau_1 s + 1)(\tau_2 s + 1)} \tag{29}
\]
By substituting in the controller equation we get
\[
K_p = \frac{\tau_1 + \tau_2}{K(\tau_c + \theta)} \tag{30}
\]
\[
\tau_I = \frac{\tau_1}{\tau_1 + \tau_2} \tag{31}
\]
\[
\tau_D = \frac{\tau_1 \tau_2}{\tau_1 + \tau_2} \tag{32}
\]
and
\[
G_c = K_p \left( 1 + \frac{1}{\tau_I s} + \tau_D s \right) \tag{33}
\]

**Example**

Consider the following approximation of the open loop transfer function
\[
\tilde{G}(s) = \frac{2e^{-1s}}{(10s + 1)(5s + 1)} \tag{34}
\]
Design a PID controller when \( \tau_c = 1s, 3s, 10s \) and simulate the system behavior.
- \( \tau_c = 1s \Rightarrow K_p = 3.7500, \tau_I = 15; \tau_D = 3.3333 \)
- \( \tau_c = 3s \Rightarrow K_p = 1.8750, \tau_I = 15; \tau_D = 3.3333 \)
- \( \tau_c = 10s \Rightarrow K_p = 0.68, \tau_I = 15; \tau_D = 3.3333 \)

A simulation is shown in shown in figure 3.
Internal Model Control

The IMC method uses a simplified model of the control loop where another controller called IMC controller is introduced. The standard control loop and the IMC model are shown in figure 4. The IMC controller is denoted by $G_c$. The first step is to establish the relationship between the two controllers. Assuming the block diagrams are equivalent, it is possible to write

$$G_c = G^*_c \frac{1}{1 - G^*_c G} \quad (35)$$

There is one to one relationship between $G_c$ and $G^*_c$. This implies that for each IMC controller there is a corresponding standard controller. The IMC method first finds an expression for $G^*_c$ and then derives an equation for $G_c$. This is done in two steps:

- Step 1: The open loop model is written as

$$\tilde{G} = (\tilde{G}^+)(\tilde{G}^-) \quad (36)$$

where $\tilde{G}^+$ contains any time delays and right half plane zeros.

- Step 2: The IMC controller is derived using the following equation

$$\tilde{G}^*_c = \frac{1}{(\tilde{G}^-)(\tau cs + 1)^r} \quad (37)$$

where $\tau_c$ is the closed loop time constant and $r$ is a positive integer that characterizes the system order.

**Example**

Use IMC to design a PID controller for the following system

$$\tilde{G}(s) = \frac{Ke^{-1s}(-1 + s)}{(0.5s + 1)(3s + 1)} \quad (38)$$

We have

$$\tilde{G}^-(s) = \frac{K}{(0.5s + 1)(3s + 1)} \quad (39)$$

for $r = 1$, we have

$$G^*_c(s) = \frac{(0.5s + 1)(3s + 1)}{K(\tau cs + 1)} \quad (40)$$

After rearranging the terms, we get

$$K_p = \frac{13}{K(4\tau c + 1)} \quad (41)$$

$$\tau_l = 3.5 \quad (42)$$

$$\tau_D = 3/13 \quad (43)$$

It is important to note that the choice of $\tau_c$ plays an important role. High values of $\tau_c$ result in a more conservative controller and low values result in a more aggressive controller.

**Tuning relationship**

These are tables that give the relationship between the model parameters and the PID parameters (handouts given in class).

**Experimental online tuning**

Experimental online tuning is also called field tuning. It is based on testing, trial and error. Having a good initial model can be very helpful in experimental tuning because it makes the tuning process more straightforward and less time consuming. Here we focus our discussions on the continuous cycling and the step reactive curve methods.

**Continuous cycling**

Introduced by Ziegler and Nichols in 1942, it is based on trial and error. After the system has reached its steady state, the integral and proportional actions are eliminated.
The gain $K_p$ is set to a small value and then increased slowly until sustained oscillations with constant amplitude occur. The numerical value of the gain for which sustained oscillations with constant amplitude occur is called the ultimate gain ($K_{pu}$), its corresponding period is called the ultimate period ($P_u$). Once $K_{pu}$ and $P_u$ are determined, the table below is used to calculate the PID parameters. Illustration of the continuous cycling method is shown in figure 5.

\[
\begin{array}{|c|c|c|c|}
\hline
\text{Ziegler–Nichols} & K_p & \tau_I & \tau_D \\
\hline
P & 0.5K_{pu} & - & - \\
PI & 0.45K_{pu} & 0.83P_u & - \\
PID & 0.6K_{pu} & 0.5P_u & 0.125P_u \\
\hline
\end{array}
\] (44)

**Example**

Design a PID control for the system whose response is shown in figure 6. The ultimate gain for the sustained oscillations is $K_u = 6$.

It is clear from the oscillations that $P_u = 3.68s$. Therefore

\[
K_p = 0.6K_{pu} = 3.6 \quad (45)
\]
\[
\tau_I = P_u/2 = 1.8 \quad (46)
\]
\[
\tau_D = P_u/8 = 0.45 \quad (47)
\]

The main drawback of this method is that it pushes the system to the limit of stability, which can result in hazardous situations.

**Astrom and Hagglund tuning method**

This method is an effective alternative to the continuous cycling method. It uses an on-off controller with dead-zone. After the on-off controller is connected to the plant, the closed loop system exhibits sustained oscillations. The principle of the method is illustrated in figure 7. Astrom and Hagglund derived an approximation of the ultimate gain as follows

\[
K_{pu} = \frac{4d}{\pi a} \quad (48)
\]

where $a, d$ are shown in figure 8. Once $K_{pu}, P_u$ are determined, the Ziegler–Nichols table (44) can be used to calculate the controller parameters. One important advantage of the Astrom and Hagglund method is that it can be easily automated.
Step test method

This is another method proposed by Ziegler and Nichols in 1942. This method is also called the plant reaction curve or Ziegler and Nichols open loop method. After the system has reached its steady state, a small step change is introduced, the plant reaction curve is obtained as shown in figure 9-bottom. The controller parameters are derived based on the reaction curve parameters using table 51. The open loop transfer function whose response is shown in figure 9-bottom is

$$\hat{G} = \frac{Ke^{-\theta s}}{\tau_{zn}s + 1}$$

(49)

where $K$, $\tau_{zn}$, $\theta$ can be obtained form the reaction curve. Let us define

$$K_0 = \frac{\tau_{zn}}{K\theta}$$

(50)

The PID parameters can be determined from the table below

$$\begin{bmatrix}
Ziegler - Nichols & K_P & \tau_I & \tau_D \\
\hline
P & K_0 & & \\
PI & 0.9K_0 & 3.3\theta & \\
PID & 1.2K_0 & 2\theta & 0.5\theta \\
\end{bmatrix}$$

(51)

Example

Consider the reaction curve shown in figure 10. The goal is derive the PID control parameters. From the graph we have: $K = 3, \theta = 2, \tau_{zn} = 1$, from which we get

$$K_P = 0.2$$

(52)

$$\tau_I = 4$$

(53)

$$\tau_D = 1$$

(54)