Part III: PID Controller Implementation with Anti-windup Mechanisms

Liuping Wang

School of Engineering
Royal Melbourne Institute of Technology University
Australia

Outline

Outline

1. Discretization of PID controllers (Position Form)
2. Discretization of PID controllers (Velocity Form)
3. Anti-windup mechanisms

Overview: Procedure

1. In the implementation using position form, steady-state values of the control signal and the output will be determined.
2. The differential equation used to capture the dynamics of PID controller will be discretized.
3. The control signal will be calculated at every sampling instant.

The Steady-state Information Needed

- In position form, the control signal $u(t)$ for a PID controller is computed using the equation,
$$u(t) = K_c e(t) + \frac{K_c}{\tau_I} \int_0^t e(\tau) d\tau - K_c \tau_D \frac{y_f(t)}{dt} \qquad (1)$$
where $e(t) = r(t) - y(t)$ is the feedback error signal between the reference signal $r(t)$ and the output $y(t)$, and $y_f(t)$ is the filtered output signal.
- It is important to make it clear that all the signals used in the computation are not the actual physical variables in numbers, instead they are the deviation variables from the physical variables in a steady-state operation.
- In other words, the control signal $u(t)$, the reference signal $r(t)$ and the output signal $y(t)$ represent the changes to the corresponding physical variables in steady-state operation.

Steady-state Values Assigned

- We determine a plant steady-state operation conditions through either mathematical analysis or experimental evaluation. For instance, through experiments and sensor/ actuator calibration, we understand in the room temperature control example that 40 percent of the valve opening of the gas furnace will correspond to $18^\circ C$ in room temperature.
- If we wish to maintain the room temperature at $18^\circ C$, the steady-state value of the control signal $U_{ss} = 40$ and $Y_{ss} = 18$.
- If we were happy with the room temperature ($r(t) = 0$) and the doors and windows remained closed, then there was no change to the room temperature $y(t) = 0$ due to the steady operation of the gas furnace.

Steady-state Values Added/Subtracted

- The position form of the PID controller is to directly calculate the deviation control variable $u(t)$ based on the feedback error $e(t)$.
- Therefore, when using the position form of PID controller for implementation, it is vital to have a priori knowledge about the steady-state information of the control signal and the output signal, so that the actual control signal will be computed as $u_{act}(t) = u(t) + U_{ss}$, in reverse, the output signal $y(t) = y_{act}(t) - Y_{ss}$.

Discretization of PID Controller

- For simplicity of expression, we let the control signal

$$u(t) = u_P(t) + u_I(t) - u_D(t)$$

  where $u_P(t)$, $u_I(t)$ and $u_D(t)$ represent the proportional, integral and derivative control terms, respectively.
- We assume that the discretization occurs in a uniformly sampling interval Δt, and the continuous-time t is sampled as $t = 0, t_1, t_2, \ldots, t_{i-1}, t_i, t_{i+1}, \ldots, .$

Discretization of Proportional Term

The proportional term is easiest to be discretized. At an arbitrary time t_i, the proportional control term $u_P(t_i)$ is calculated as

$$u_P(t_i) = K_c(r(t_i) - y(t_i)) \tag{2}$$

Discretization of Integral Term

- The integral control term $u_I(t_i)$ requires numerical approximation for the integral function, which is written as

$$u_I(t_i) = \frac{K_c}{\tau_I} \sum_{t_k=0}^{t_i} e(t_k)\Delta t \tag{3}$$

where $\int_0^t e(\tau)d\tau \approx \sum_{t_k=0}^{t_i} e(t_k)\Delta t$.

- We know

$$\lim_{\Delta t \to 0} \sum_{t_k=0}^{t_i} e(t_k)\Delta t = \int_0^t e(\tau)d\tau \tag{4}$$

- Therefore, the accuracy of the approximation increases as the sampling interval Δt reduces.

Discretization of Derivative Term (i)

- To find the derivative control term, we examine the Laplace transform of the filtered derivative control term $U_D(s)$ in relation to the output $Y(s)$:

$$
\begin{aligned}
U_D(s) &= \frac{K_c \tau_D s}{\tau_f s + 1} Y(s) \\
&= \frac{1}{\tau_f} \frac{K_c \tau_D s}{s + \frac{1}{\tau_f}} Y(s) \\
&= \frac{K_c \tau_D}{\tau_f} \frac{s + \frac{1}{\tau_f} - \frac{1}{\tau_f}}{s + \frac{1}{\tau_f}} Y(s) \\
&= \frac{K_c \tau_D}{\tau_f} Y(s) - \frac{\frac{K_c \tau_D}{\tau_f^2}}{s + \frac{1}{\tau_f}} Y(s) \qquad (5)
\end{aligned}
$$

- The inverse Laplace transform of first term in (5) gives the time domain response $\frac{K_c \tau_D}{\tau_f} y(t)$, which is easy to be discretized as $\frac{K_c \tau_D}{\tau_f} y(t_i)$ at sample time t_i.

Discretization of Derivative Term (ii)

- Let $u_D^f(t)$ denote the time domain response of the second term.

$$\frac{du_D^f(t)}{dt} + \frac{1}{\tau_f} u_D^f(t) = -\frac{K_c \tau_D}{\tau_f^2} y(t) \tag{6}$$

The solution of this differential equation gives us the term $u_D^f(t)$.

- Let the derivative $\frac{du_D^f(t)}{dt}$ be represented by the first order approximation:

$$\frac{du_D^f(t)}{dt} \approx \frac{u_D^f(t+\Delta t) - u_D^f(t)}{\Delta t} \tag{7}$$

Discretization of Derivative Term (iii)

- Then at time $t = t_i$, by substituting (7) into the differential equation (6), we obtain

$$u_D^f(t_i + \Delta t) = u_D^f(t_i) - \frac{1}{\tau_f} u_D^f(t_i)\Delta t - \frac{K_c \tau_D}{\tau_f^2} y(t_i)\Delta t \tag{8}$$

- Finally, the derivative control term is computed using

$$u_D(t_i) = \frac{K_c \tau_D}{\tau_f} y(t_i) + u_D^f(t_i) \tag{9}$$

where $u_D^f(t_i)$ is updated using the equation (8).

Key Points

- Position form of PID controller implementation is to directly calculate the control signal $u(t)$.
- Numerical approximation of integration and differentiation of the feedback error leads to the discretization of the continuous time controller for digital implementation.
- Steady-state value of the control signal U_{ss} needs to be added to $u(t)$ to form the actual control signal. Steady-state value of the output signal Y_{ss} needs to be subtracted from the measurement signal to form $y(t)$ for computation.
- Without specification of steady-state information, we implicitly assumed that they are zero, which could be correct or could be wrong.

Outline

Overview of Velocity Form

1. In the implementation using velocity form, the derivative of the control signal is determined first analytically.
2. The derivative is then discretized to give the difference of the control signal.
3. The control signal is then calculated using the past control signal adding together the difference of the control signal.
4. When implementing using velocity form, the steady-state information of the control signal and output signal is not required, which is one of the advantages for this type of implementation.
5. Proportional controller and proportional-plus-derivative controller do not have a velocity form, because they do not have an integral term required for this type of implementation.

Derivative of the Control Signal

- The Laplace transfer function of the control signal $U(s)$ is in relation to the feedback error $E(s)$:

$$\begin{aligned} U(s) &= K_c(1 + \frac{1}{\tau_I s})E(s) \\ &= \frac{K_c s E(s) + \frac{K_c}{\tau_I}E(s)}{s} \end{aligned} \tag{10}$$

- From (10), we have

$$sU(s) = K_c sE(s) + \frac{K_c}{\tau_I}E(s) \tag{11}$$

- The inverse Laplace transform of (11) leads to the following differential equation:

$$\dot{u}(t) = K_c \dot{e}(t) + \frac{K_c}{\tau_I}e(t) \tag{12}$$

- This is termed 'velocity' form because on the left-hand side of the equation the derivative of the control signal is computed.

Discretization of PI Controller

- Approximate $\dot{u}(t)$ and $\dot{e}(t)$ at sample time t_i :

$$\dot{u}(t_i) \approx \frac{u(t_i) - u(t_i - \Delta t)}{\Delta t} \tag{13}$$

$$\dot{e}(t_i) \approx \frac{e(t_i) - e(t_i - \Delta t)}{\Delta t} \tag{14}$$

- We obtain

$$u(t_i) - u(t_i - \Delta t) = K_c(e(t_i) - e(t_i - \Delta t)) + \frac{K_c}{\tau_I} e(t_i)\Delta t \tag{15}$$

- The computation of the control signal becomes

$$u(t_i) = u(t_i - \Delta t) + K_c(e(t_i) - e(t_i - \Delta t)) + \frac{K_c}{\tau_I} e(t_i)\Delta t \tag{16}$$

Alternative Structure of PI Controller

- If the proportional control term is only implemented on the output signal, a small modification of (16) is to replace the differenced feedback error $e(t_i) - e(t_i - \Delta t)$ with the differenced output signal $-y(t_i) + y(t_i - \Delta t)$.
- As a result, the implementation of the alternative PI controller structure is based on the computational equation,

$$u(t_i) = u(t_i - \Delta t) + K_c(-y(t_i) + y(t_i - \Delta t)) + \frac{K_c}{\tau_I} e(t_i)\Delta t \tag{17}$$

Considering Steady-state Information (i)

- Same as position form, all the signals in the control signal computation are the deviation signals. They are relative to their steady-state values.
- To this end, with the steady-state values of U_{ss}, Y_{ss} and R_{ss}, the actual signals corresponding to the plant operation are

$$u_{act}(t_i) = u(t_i) + U_{ss} \tag{18}$$

$$y_{act}(t_i) = y(t_i) + Y_{ss} \tag{19}$$

$$r_{act}(t_i) = r(t_i) + R_{ss} \tag{20}$$

- Adding the steady- state value of U_{ss} to both sides of (17), adding and subtracting Y_{ss} to the second term that corresponds to the proportional control lead to its equivalent expression:

$$\begin{aligned} u(t_i) + U_{ss} &= u(t_i - \Delta t) + U_{ss} + K_c(-y(t_i) - Y_{ss} + Y_{ss} + y(t_i - \Delta t)) \\ &+ \frac{K_c \Delta t}{\tau_I}(r(t_i) - y(t_i)) \end{aligned} \tag{21}$$

Considering Steady-state Information (ii)

- We assume that the steady-state of the setpoint signal R_{ss} is equal to the steady-state of the output signal Y_{ss}, which is realistic for plant operations.
- This assumption means that

$$r(t_i) - y(t_i) = r(t_i) + Y_{ss} - Y_{ss} - y(t_i) = r_{act}(t_i) - y_{act}(t_i)$$

- The computational equation for the actual implementation of PI controller is,

$$u_{act}(t_i) = u_{act}(t_i - \Delta t) + K_c(-y_{act}(t_i) + y_{act}(t_i - \Delta t)) + \frac{K_c \Delta t}{\tau_I}(r_{act}(t_i) - y_{act}(t_i)) \quad (22)$$

Considering Steady-state Information (iii)

- The extension to the original PI structure in (16) follows from the same assumption that the steady-state of the setpoint signal R_{ss} is equal to the steady-state of the output signal Y_{ss}.
- Thus, the implementation equation for the original PI structure is

$$\begin{aligned} u_{act}(t_i) &= u_{act}(t_i - \Delta t) + K_c(r_{act}(t_i) - y_{act}(t_i) \\ &- r_{act}(t_i - \Delta t) + y_{act}(t_i - \Delta t)) + \frac{K_c \Delta t}{\tau_I}(r_{act}(t_i) - y_{act}(t_i)) \end{aligned} \tag{23}$$

- Here the proportional control will directly act on the set-point change.

Discretization of Derivative Term (i)

- The Laplace transform of the PID controller is

$$U(s) = K_c E(s) + \frac{K_c}{\tau_I s} E(s) - \frac{K_c \tau_D s}{\tau_f s + 1} Y(s) \tag{24}$$

- The question is how the third term corresponding to the derivative control will be discretized. From the transfer function of the derivative control,

$$U_D(s) = \frac{1}{\tau_f} \frac{K_c \tau_D s}{s + \frac{1}{\tau_f}} Y(s) \tag{25}$$

- The differential equation that governs the relationship between the variables $u_D(t)$ and $y(t)$ is obtained as

$$\frac{du_D(t)}{dt} + \frac{1}{\tau_f} u_D(t) = \frac{K_c \tau_D}{\tau_f} \frac{dy(t)}{dt} \tag{26}$$

Discretization of Derivative Term (ii)

- By approximating

$$\frac{du_D(t)}{dt} \approx \frac{u_D(t) - u_D(t-\Delta t)}{\Delta t}; \frac{dy(t)}{dt} \approx \frac{y(t) - y(t-\Delta t)}{\Delta t} \quad (27)$$

at time t_i, the differential equation (26) becomes

$$\frac{u_D(t_i) - u_D(t_i-\Delta t)}{\Delta t} = -\frac{1}{\tau_f} u_D(t_i) + \frac{K_c \tau_D}{\tau_f} \frac{y(t_i) - y(t_i - \Delta t)}{\Delta t} \quad (28)$$

- By multiplying Δt on both sides of the equation, and re- arranging, we obtain

$$(1 + \frac{\Delta t}{\tau_f}) u_D(t_i) = u_D(t_i - \Delta t) + \frac{K_c \tau_D}{\tau_f} (y(t_i) - y(t_i - \Delta t)) \quad (29)$$

Steady-state Information for the Derivative Term

- The steady-state of $u_D(t)$ is taken as zero, because derivative of a constant term (steady-state) is zero.
- By adding and subtracting the steady-state of the output value to (29), we obtain the computation of the derivative control term using the actual output measurement,

$$u_D(t_i) = \frac{\tau_f}{\tau_f + \Delta t} u_D(t_i - \Delta t) + \frac{K_c \tau_D}{\tau_f + \Delta t} (y_{act}(t_i) - y_{act}(t_i - \Delta t)) \tag{30}$$

Expression of Discretized PID in Velocity Form

- The derivative of the control signal is expressed as

$$\dot{u}(t) = K_c \dot{e}(t) + \frac{K_c}{\tau_I} e(t) - \dot{u}_D(t) \tag{31}$$

- With the proportional term on the feedback error signal, the control signal is calculated as

$$\begin{aligned} u_{act}(t_i) &= u_{act}(t_i - \Delta t) + K_c(r_{act}(t_i) - y_{act}(t_i) - r_{act}(t_i - \Delta t) \\ &+ y_{act}(t_i - \Delta t)) + \frac{K_c \Delta t}{\tau_I}(r_{act}(t_i) - y_{act}(t_i)) - u_D(t_i) + u_D(t_i - \Delta t) \end{aligned} \tag{32}$$

- When implementing the proportional control directly on the output, the control signal is

$$\begin{aligned} u_{act}(t_i) &= u_{act}(t_i - \Delta t) + K_c(-y_{act}(t_i) + y_{act}(t_i - \Delta t)) \\ &+ \frac{K_c \Delta t}{\tau_I}(r_{act}(t_i) - y_{act}(t_i)) - u_D(t_i) + u_D(t_i - \Delta t) \end{aligned} \tag{33}$$

Choice of Sampling Interval (i)

- PID controllers are continuous-time controllers, which are designed based on continuous-time models such as Laplace transfer functions.
- In the implementation of this type of controllers, a sampling interval Δt is selected to compute the discrete-time control signal for digital implementation.
- When Δt is too large, there will be large numerical errors between the continuous-time control signal (designed for) and the discrete-time signal (actually implemented) due to discretization of derivatives and integral operations.
- Those large numerical errors will lead to closed-loop performance degradation, and in the worst case, closed-loop instability.

Choice of Sampling Interval (ii)

- So, when a PID control system becomes unstable in the implementation, the first thing to check is if the sampling interval Δt is too large.
- The rough rule is that we choose sampling interval Δt in relation to the desired closed-loop bandwidth w_n, in the order of $\frac{1}{10w_n}$. A smaller Δt may be required when some constants in the system are much smaller.

Example

A second order system is described by the transfer function

$$G(s) = \frac{-0.1}{(s+1)^2} \tag{34}$$

Design a PID controller with filter to control this plant. The desired closed-loop polynomial is specified as $(s^2 + 2\xi w_n s + w_n^2)(s + \lambda_1)$, where $\xi = 0.707$, $w_n = \lambda_1 = 5$. Simulate the unit step closed-loop response using the discrete PID computational algorithm with sampling interval Δt chosen as $\frac{1}{10w_n} = 0.02$ and $\frac{1}{5w_n} = 0.04$. Illustrate that when Δt is increased to 0.1, the closed-loop system becomes unstable although the continuous-time PID controller design produced a stable closed-loop system.

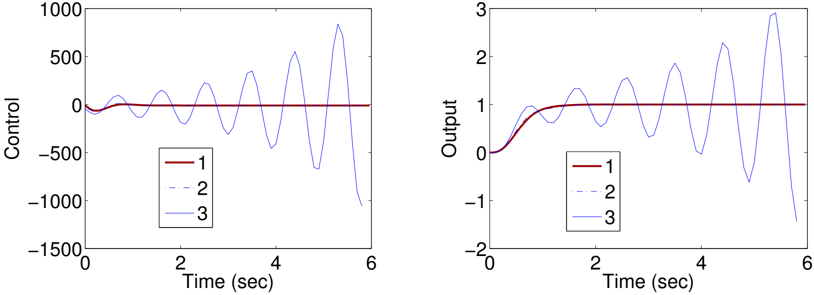
Solution

Using the pole-assignment controller design algorithm, we obtain the PID controller parameters as

$$K_c = -245.66; \tau_I = 0.59; \tau_D = 0.176; \tau_f = 0.066$$

Figure 1 compares the closed-loop responses with the three cases of the sampling intervals. It is seen that the difference between the closed-loop responses using $\Delta t = 0.02$ and 0.04 is negligible. However, when the sampling interval is increased to 0.1, the closed-loop system becomes unstable as shown in Figure 1.

Responses



(a) Input signal

(b) Output signal

Figure 1: Closed-loop response. Key: line (1) Closed-loop response with $\Delta t = 0.02$; line (2) closed-loop response with sampling interval $\Delta t = 0.04$; line (3) Closed-loop response with sampling interval $\Delta t = 0.1$.

Key Points

- Implementation of PID controllers using velocity form does not requirement steady-state information of control signal and output signal.
- The actual control signal is computed using the actual plant measurement and the actual set-point signal.
- Sampling interval Δt is the parameter used in the implementation stage for the PID controllers. Sampling interval Δt needs to be chosen properly in order to get the controller work well.

Outline

Learning Objectives

1. Understanding the scenario of integrator wind-up.
2. Ability to implement a PID controller using position form that has an anti-windup mechanism.

Scenario of Integrator Windup (i)

- Consider the integrating plus delay plant with the transfer function,

$$G(s) = \frac{1.8e^{-30s}}{s(10s+1)^2}. \tag{35}$$

- The PI controller for this plant has proportional gain $K_c = 0.0065$ and integral time constant $\tau_I = 244.5$.
- Simulate the closed-loop response for this PI control system with a unit step set-point signal.
- Supposing that the control signal amplitude is not to exceed 1.5×10^{-3}, illustrate the scenario of integrator wind-up.

Scenario of Integrator Windup (ii)

- The simulation set-up is illustrated in the Simulink diagram, where a saturation block is used to simulate the scenario of limits of control amplitude.
- Both proportional and integral terms are implemented on the feedback error signal.
- We first set the limits in the saturation blocks to be larger (± 3) than the maximum and minimum of the control signal amplitude. The closed-loop system behaves well.
- By reducing the allowable control amplitude to 1.5×10^{-3}, the actual control signal to the plant, u, is limited. When this happens, the closed-loop response becomes oscillatory.

Response Plots

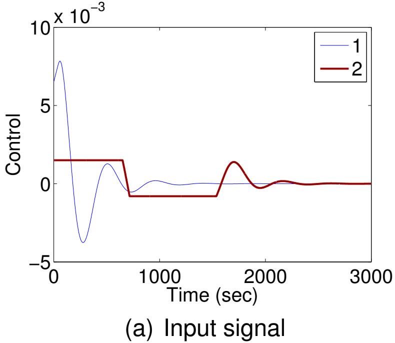


Figure 2: Closed-loop responses). Key: line (1) Closed-loop response without saturation; line (2) closed-loop response with saturation.

Investigation of Windup Scenario

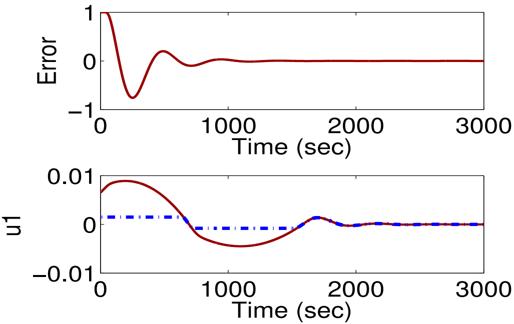


Figure 3: Error signal and control signal in integrator windup. The controller output signal (u_1) in the Simulink diagram continued to grow (see Key: solid line) after the actual control signal to the plant was limited (see Key: dash-dot line) because the feedback error is positive. The amplitude of u_1 reached its maximum when the feedback error $e(t)$ changes sign from positive to negative (see the top plot). The magnitude of the control signal was gradually reducing as the magnitude of the error increases, which is in the negative region.

Summary of Integrator Windup Scenario

- Integrator windup occurs when the amplitude of control signal is limited, and the calculated control signal exceeds this limit.
- In the position form when implementing a PI controller, the control signal is calculated using the equation,

$$u(t) = K_c e(t) + \frac{K_c}{\tau_I}\int_0^t e(\tau)d\tau \tag{36}$$

- This integral term will continue to grow in magnitude as long as the sign of the feedback error remains the same (integration is about calculation of the area of the curve). As a result, the control signal $u(t)$ calculated will continue to grow as long as the sign of the feedback error remains the same.
- Because of the saturation, the actual control signal implemented on the plant is not the same as the output of the controller when the saturation limits are reached. The controller is not informed of what is actually happening in the plant.

How to Avoid Windup

- The integral function should be stopped when the saturation limits are reached. Namely, the integration should be implemented using a stable transfer function so that the output of the controller should not grow when the saturation limits are reached.
- The controller should be informed of what is actually happening in the plant. Namely, the controller output should equal to the actual input to the plant.

Principles of Anti-windup Mechanisms

- There are many anti-windup mechanisms in PI controllers.
- More or less, they have the same principles by implementing the PI controller with a stable transfer function and letting the controller know what is actually happening in the plant.

An Anti-windup Realization (i)

- A PI controller with transfer function is assumed to have

$$C(s) = \frac{c_1 s + c_0}{s}$$

- The assumption for using this implementation is that the controller has a stable zero, namely the ratio $\frac{c_0}{c_1} > 0$, or τ_I is positive.
- Then the controller transfer function can be written as

$$\frac{U(s)}{E(s)} = \frac{c_1}{1 - \frac{c_0 c_1}{c_1 (c_1 s + c_0)}} = \frac{c_1 s + c_0}{s}.$$

- In this implementation, there is a positive feedback used in the system (positive feedback is seldom used, but here is an example).

An Anti-windup Realization (ii)

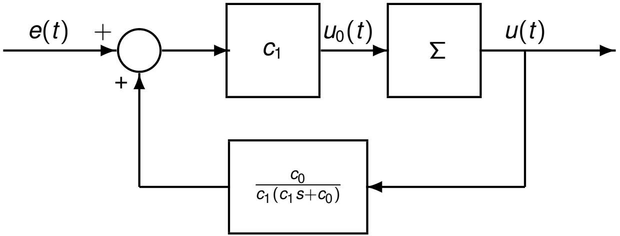


Figure 4: PI controller (position form) with anti-windup mechanism. Σ represents the saturation nonlinearity, which is defined by the following computation. If $u^{min} < u_0(t) < u^{max}$, then $u(t) = u_0(t)$; if $u_0(t) \leq u^{min}$, then $u(t) = u^{min}$; if $u_0(t) \geq u^{max}$, then $u(t) = u^{max}$.

An Anti-windup Realization (iii)

- When the saturation limits are not reached, Σ is a unity gain ($u(t) = u_0(t)$), then the transfer function from the error signal e to the control signal u is

$$\frac{U(s)}{E(s)} = \frac{c_1}{1 - \frac{c_0 c_1}{c_1(c_1 s + c_0)}} = \frac{c_1 s + c_0}{s}.$$

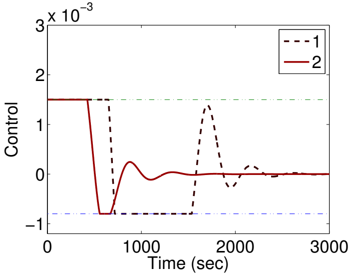
- The integral action in this configuration is achieved by putting positive feedback around a stable transfer function.
- If the control signal reaches a limit, for instance, $u(t) = U^{max}$, because the transfer function $\frac{c_0}{c_1(c_1 s + c_0)}$ is stable by the assumption made, the feedback signal $\frac{c_0}{c_1(c_1 s + c_0)}$ will become a constant after a transient response.
- With this action, the integral action will be stopped, also the controller calculation is fully informed of what is actually happening by this feedback link.

Example

We use the same example as the one to illustrate windup scenario. The parameters required for the anti-windup implementation are

$$c_1 = 0.0065; c_0 = 2.6585e - 005$$

Using the anti-windup mechanism, we simulated the closed-loop response in the same conditions as before.



(a) Input signal

(b) Output signal

Figure 5: Closed-loop response. Key: line (1) Closed-loop response without anti-windup mechanism; line (2) closed-loop response with anti-windup mechanism.

Summary

- Integrator windup scenario is to do with the control signal saturation and it occurs when the PI controller implementation uses a unstable realization and when the actual control signal is not equal to the computed control signal.
- To avoid integrator windup is a necessary consideration in the implementation of PID controllers.
- There are many strategies available for anti-windup mechanisms. One of the commonly used mechanisms is discussed here, which is to implement a PI controller using a positive feedback loop.

Learning Objectives

- Ability to implement a PID controller using velocity form that has anti-windup mechanisms for
  - constraining the amplitude of the control signal
  - constraining the rate of change of the control signal.

Anti-windup Mechanisms in Velocity Form of PID:Overview

- It is much straightforward to implement anti-windup mechanisms in velocity form of PID controllers.
- In the velocity form of PID controllers, not only the anti-windup scheme can be readily implemented on the amplitude of the control signal, but also on the derivative of the control signal.
- Similarly, the two key points in implementation of anti-windup are to stop integral action when the control signal reaches saturation and to make sure that the actual control signal equals the control signal computed.

Anti-windup Mechanism on Amplitude of the Control Signal (i)

- We assume that the actual control variable to the plant is limited by U^{min} and U^{max}:

$$U^{min} \leq u_{act}(t) \leq U^{max}$$

- The equation used to calculate the actual control signal is

$$\begin{aligned} u_{act}(t_i) &= u_{act}(t_i - \Delta t) + K_c(-y_{act}(t_i) + y_{act}(t_i - \Delta t)) \\ &+ \frac{K_c \Delta t}{\tau_I}(r_{act}(t_i) - y_{act}(t_i)) - u_D(t_i) + u_D(t_i - \Delta t) \end{aligned} \tag{37}$$

- All actual measurements of the physical variables are used in the updating of the control signal, and the control signal computed is the physical variable to be implemented.
- Thus, the implementation procedure naturally satisfies one of the requirements in an anti-windup mechanism that the actual control signal equal to the computed control signal.

Anti-windup Mechanism on Amplitude of the Control Signal (ii)

- How about the second requirement?
- In order to stop the integration, when the actual control signal reaches the limit, we impose the limits on the actual control signal with the computation that if $u_{act}(t_i) < U^{min}$, then $u_{act}(t_i) = U^{min}$; if $u_{act}(t_i) > U^{max}$, then $u_{act}(t_i) = U^{max}$.
- When the sample time t_i moves one step forward, the $u_{act}(t_i - \Delta t)$ carries the information of saturation at the previous sample time and the control signal computation is automatically informed of the saturation.

Implementation Steps

1. Calculate the actual control signal using

$$\begin{aligned} u_{act}(t_i) &= u_{act}(t_i - \Delta t) + K_c(-y_{act}(t_i) + y_{act}(t_i - \Delta t)) \\ &+ \frac{K_c \Delta t}{\tau_I}(r_{act}(t_i) - y_{act}(t_i)) - u_D(t_i) + u_D(t_i - \Delta t) \end{aligned} \tag{38}$$

2. Check if the control signal is within the limits

$$U^{min} \leq u_{act}(t_i) \leq U^{max}$$

   If the constraints are satisfied, then this is the actual control signal to the plant. If not, go to the next step,

3. If $u_{act}(t_i) < U^{min}$, then $u_{act}(t_i) = U^{min}$; if $u_{act}(t_i) > U^{max}$, then $u_{act}(t_i) = U^{max}$.

Anti-windup Mechanism for $\dot{u}(t)$ (i)

- This set of limits are typically specified as

$$DU^{min} \leq \dot{u}(t) \leq DU^{max}$$

- In the computation, we calculate $\dot{u}(t)$ using the approximation,

$$\begin{aligned} \dot{u}(t) &\approx \frac{u(t_i) - u(t_i - \Delta t)}{\Delta t} \\ &= \frac{u_{act}(t_i) - u_{act}(t_i - \Delta t)}{\Delta t} \end{aligned} \tag{39}$$

as the current sample $u_{act}(t_i)$ and the past sample $u_{act}(t_i - \Delta t)$ share the same steady-state value.

Implementation Steps

1. Calculate the actual control signal using

$$\begin{aligned} u_{act}(t_i) &= u_{act}(t_i - \Delta t) + K_c(-y_{act}(t_i) + y_{act}(t_i - \Delta t)) \\ &+ \frac{K_c \Delta t}{\tau_I}(r_{act}(t_i) - y_{act}(t_i)) - u_D(t_i) + u_D(t_i - \Delta t) \end{aligned} \quad (40)$$

2. Check if the derivative of the control signal is within the limits

$$DU^{min} \leq \frac{u_{act}(t_i) - u_{act}(t_i - \Delta t)}{\Delta t} \leq DU^{max}$$

If the constraints are satisfied, then this is the actual control signal to the plant. If not, we calculate the control signal using one of following steps.

- If $\frac{u_{act}(t_i)-u_{act}(t_i-\Delta t)}{\Delta t} < DU^{min}$, then

$$u_{act}(t_i) = u_{act}(t_i - \Delta t) + DU^{min} \Delta t;$$

- if $\frac{u_{act}(t_i)-u_{act}(t_i-\Delta t)}{\Delta t} > DU^{max}$, then

$$u_{act}(t_i) = u_{act}(t_i - \Delta t) + DU^{max} \Delta t.$$

Example

- A DC motor model is given by the transfer function

$$G(s) = \frac{0.5}{(s+2)s} \tag{41}$$

where the input is current and the output is the angular position.

- The requirement is that the angular position follows a ramp signal of a unit slope without steady-state error, and the operational requirements are that the control signal is within the limits of $(-7, 5)$, and the derivative of the control signal is within the limits of ± 20. Design a PID controller with anti-windup mechanism for this system.
- The desired closed-loop performance is determined using the desired closed-loop polynomial $(s^2 + 2\xi w_n s + w_n^2)(s + \lambda_1)^2$, where $\xi = 0.707$ and $w_n = \lambda_1 = 3$.
- Also, investigate the approach that uses a smaller controller gain to reduce both $|u(t)|$ and $|\dot{u}(t)|$ and compare the results with the anti-windup control.

Solution

- With this desired closed-loop performance, we design the PID controller using pole-assignment controller design technique and obtain the controller parameters as

$$K_c = 19.9831; \tau_I = 1.0167; \tau_D = 0.2061; \tau_f = 0.1213.$$

- In order to eliminate steady-state error in tracking a ramp signal, it is necessary to use the PID controller structure with both proportional and integral terms implemented on the feedback error, while the derivative term implemented on the output.
- What would be the problem if we implement the proportional control on $y(t)$?

Responses

The anti-windup mechanism takes effect and the results are compared in Figure 6.

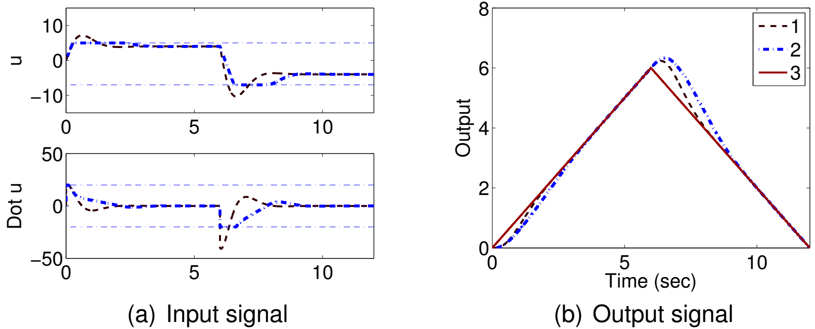
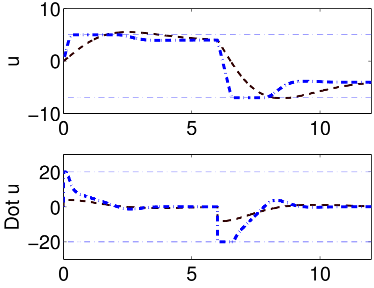


Figure 6: Closed-loop response. Key: line (1) Closed-loop response without limits; line (2) closed-loop response using anti-windup control with $w_n = 3$; line (3) reference signal.

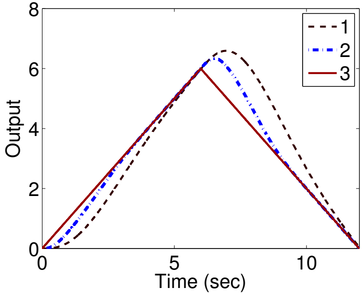
Without Anti-windup Mechanism

- Without anti-windup control, we need to reduce the parameters w_n and λ_1 to reduce the magnitudes of the control signal and the derivative of the control signal.
- By selecting $w_n = \lambda_1 = 1$, both control signal and the derivative of the control signal are within the operational limits specified in the design.
- The track performance is significantly deteriorated in comparison with the results obtained from anti-windup control.

Comparison Results



(a) Input signal



(b) Output signal

Figure 7: Closed-loop response. Key: line (1) Closed-loop response without limits ($w_n = 1$); line (2) closed-loop response using anti-windup control with $w_n = 3$; line (3) reference signal.

Summary

- Anti-windup mechanism of PID controllers is necessary in the implementation of all practical and industrial controllers, in order to ensure safety of the equipment.
- If without anti-windup mechanism, we need to reduce the closed-loop bandwidth and the controller gain so that the control signal is within the limits of safe operation.
- However, the anti-windup control, if implemented properly, will allow us to use a higher gain in closed-loop control, with safety protection of the equipment.