THE PID CONTROLLER

• A well known structure for the controller is proportional + integral + derivative (i.e., PID) control.

\[ u(t) = k_p \cdot e + k_I \int_0^t e(\tau)d\tau + k_D \frac{de(t)}{dt}, \quad (1) \]

- \text{prop.} \quad \text{Integral} \quad \text{derivative}

\[ \frac{U}{E} = D(s) = k_p + \frac{k_I}{s} + k_D s. \quad (2) \]

- reset value \quad derivative rate

\[ D_c (s) = k_p \left[ 1 + \frac{1}{T_I s} + T_D s \right], \quad (3) \]

• Increasing the proportional feedback gain reduces steady-state errors, but high gains almost always destabilize the system.
• Integral control provides robust reduction in steady-state errors, but often makes the system less stable.
• Derivative control increases damping and improves stability.
• In a number of important cases the reference input will not be constant but can be approximated as a polynomial in time long enough to the system to effectively reach steady state.

• In Eq. (3),
  ○ $T_I$ is called the "reset rate" in seconds;
  ○ $T_D$ is called the "derivative rate" in seconds

• The effect of the derivative control term depends on the rate of change of the error. Because of the sharp effect of the derivative control on suddenly changing signals, the "D" term is sometimes introduced in the feedback path (Ex. a tachometer on a shaft of a motor).

**EXAMPLE -- P, PI, PID CONTROL OF A DC MOTOR SPEED**

• Using MATLAB plot the DC motor speed control response to steps in disturbance and steps in reference input. Use the general transfer function or the simplified one.

• Consider the DC motor speed control parameters as:
  • $J_m = 1.13 \cdot 10^{-2}$ N-m-sec$^2$/rad; $b = 0.028$ N-m-sec/rad;
  • $L_a = 10^{-1}$ Henry; $R_a = 0.45$ $\Omega$;
  • $K_t = 0.067$ N-m/amp; $K_e = 0.067$ V-sec/rad.

• Consider the controller parameters as:
  • $k_P = 3$; $k_I = 15$ sec$^{-1}$; $k_D = 0.3$ sec.
SOLUTION TO EXAMPLE

- System + MATLAB program ...
- The step responses can be computed by forming the numerator and denominator coefficient vectors in descending power of s and using the step function in MATLAB.
  - P control does not eliminate the steady-state error
  - PI control increases the oscillatory behaviour but eliminates the steady-state error
  - PID control reduces the oscillation while maintaining zero steady-state error

![System Diagram]

Unused controller parameters are zero

\[ \theta \left( \frac{Y_1}{U_1} \right) \Rightarrow Y_1 = 0.1 U_1 \]
Case: \( W \neq 0, \ D = 0 \)

Equations:

\[ \begin{align*}
    u_1 &= r + y_3 = y_3 \\
    y_3 &= -u_3 = -y_2 \\
    y_2 &= u_2 \cdot \omega = (w + y_1) \cdot \omega = (w + u_1 \cdot D) \cdot \omega \\
    u_1 &= y_3 = -(w + u_1 \cdot D) \cdot \omega = -w \cdot \omega - u_1 \cdot D \cdot \omega
\end{align*} \]

From (1) \( \Rightarrow \)

\[ u_1 + u_1 \cdot D \cdot \omega = -w \cdot \omega \]

\[ u_1 (1 + D \cdot \omega) = -w \cdot \omega \quad (2) \]

From (2) \( \Rightarrow \)

\[ \frac{u_1}{w} = \frac{-G}{1 + D \cdot \omega} = \frac{-1}{D + \frac{1}{G}} = T_w(s) \]

DC MOTOR TRANSFER FUNCTION (See Model)

\[ \frac{\Theta_m(s)}{V_a(s)} \rightarrow \text{angular rotation} \quad \{ \text{plant} \} \]

\[ \frac{\Theta_m(s)}{V_a(s)} \rightarrow \text{Input voltage} \]

- In our case we need the transfer function between the motor input \( v_a \) and the output speed \( w \).
We know that \( \omega = \dot{\Theta}_m \)

\[
\frac{\Omega(s)}{V_a(s)} = s \frac{\Theta_m(s)}{V_a(s)} = G \quad ; \quad \text{From the model}
\]

\[
G = \frac{K_t}{J_m \cdot L_a \cdot s^2 + (b \cdot L_a + J_m \cdot R_a) \cdot s + (b \cdot R_a + K_t \cdot k_p)}
\]

Transfer function with \( P \) controller from (3)

\[
T_w(s) = \frac{-1}{K_p + \frac{a_1 \cdot s^2 + a_2 \cdot s + a_3}{A}} = \frac{-A}{a_1 \cdot s^2 + a_2 \cdot s + a_3 + A \cdot k_p}
\]

\[
T_w(s) = \frac{-A}{a_1 \cdot s^2 + a_2 \cdot s + b_2}
\] (4)
Similarly for the PI and PID controllers the transfer function $T_w(s)$ is:

$$T_w(s) = \frac{-1}{\frac{K_p + \frac{K_i}{s}}{s} + \frac{a_1 s^2 + a_2 s + a_3}{A}} = \frac{-A \cdot j}{a_1 s^3 + a_2 s^2 + (K_p A + a_3) s + K_i A} \quad (5)$$

$$T_w(s) = \frac{-1}{\frac{K_p + \frac{K_i}{s}}{s} + \frac{a_1 s^2 + a_2 s + a_3}{A}} = \frac{-A \cdot j}{a_1 s^3 + (A \cdot K_p + a_2) s^2 + (A \cdot K_i + a_3) s + K_i A} = \frac{-A \cdot j}{a_1 s^3 + b_1 s^2 + b_2 s + b_3} \quad (6)$$
SOLUTION: MATLAB program for obtaining the output response to a step input disturbance -- in this case the reference input R is set to 0.

HOMEWORK: develop the transfer functions for the DC motor control with P, PI, and PID controllers when the disturbance \( w = 0 \) and the input is applied at the reference R. Derive the overall transfer function of the system (controller + plant) considering the R as input and \( Y_2 (=Y) \) as output. Plot the step responses for your transfer functions using MATLAB -- solution is similar to the one presented in the example above.

NOTE: if you don't know the meaning of a MATLAB function type: help function in the workspace.
DIGITAL IMPLEMENTATION OF CONTROLLERS

• Many controllers in feedback systems are implemented as an embedded application
  ○ Advantage: the control law for calculating the control signal can be easily modified.

• The DSPs are in general specially designed for supporting real-time signal processing and real-time digital controllers

• A linear continuous analog design can be easily translated into a discrete equivalent.

• A DIGITAL CONTROLLER unlike the analog controller uses sampled and quantized signals.
  ○ A controller that operates on signals that are sampled but not quantized is called discrete, while one that operates on signals that are both sampled and quantized is called digital.
  ○ A signal to be used in digital logic needs to be sampled first, and then the samples need to be converted by an analog-to-digital (A/D) converter into a quantized digital number.
  ○ Once the digital computer has calculated the proper next control value, this value needs to be converted back into an analog signal in order to be applied to the actuator of a process.
SAMPLING TIME RULE

• The control signal is not changed until the next sampling period.
• A reasonable rule of thumb for selecting the sampling period is that during the rise time of the response to a step, the input to the discrete controller should be sampled approximately 6 times.
• The sampling can be reduced to 2 to 3 times per rise time if the controller is adjusted to cover the effects of the sampling
  ○ This corresponds to a sampling frequency that is 10 to 20 times the system's closed-loop bandwidth.

SYSTEM WITH A DIGITAL CONTROLLER

• A discrete equivalent to a given continuous controller-sampled but not quantized (quantization ignored).
• The method depends on the sampling period $T_s$ being short enough so the reconstructed control signal is close to the signal produced by the original analog controller.
Finding a discrete equivalent to a given analog controller is equivalent to finding a recurrence equation for the samples of the control which will approximate the differential equations of the controller.

ASSUME that we have the transfer function $D(s)$, and wish to replace it with a discrete controller that will compute the next control signal to be sent to actuator by:
- Accepting samples of the controller input, $e(k \cdot T_s)$;
- Using past values of the control signal $u(k \cdot T_s)$;
- Using present and past samples of the input, $e(k \cdot T_s)$.

EXAMPLE -- PID DISCRETE CONTROLLER

$$ U(s) = (k_P + \frac{k_I}{s} + k_D s) E(s), $$

$$ u(t) = k_P e(t) + k_I \int_0^t e(\tau) d\tau + k_D \dot{e}(t) = u_P + u_I + u_D, $$

where $T_s$ is the sample period.

For linear systems, the next control sample can be computed term by term (superposition) -- The proportional term $U_P$ is immediate:

$$ U_P(k \cdot T_s + T_s) = k_P \cdot e(k \cdot T_s + T_s); $$
THE INTEGRAL term can be computed by breaking the integral into two parts and approximating the second part, which is the integral over one sample period.

\[
U_I (k \cdot T_s + T_s) = K_I \cdot \int_0^{kT_s + T_s} e(\tau) \, d\tau
\]

\[
= K_I \cdot \int_0^{kT_s} e(\tau) \, d\tau + K_I \cdot \int_{kT_s}^{kT_s + T_s} e(\tau) \, d\tau
\]

\[
\approx U_I (kT_s) + K_I \cdot \frac{T_s}{2} \left[ e(kT_s + T_s) + e(kT_s) \right]
\]

In Eq. (4) the area in question is approximated by that of a trapezoid formed by the base \(T_s\) and vertices \(e(kT_s + T_s)\) and \(e(kT_s)\), as shown by the dashed line of the figure above.

Note that the area can also be approximated by the rectangle of amplitude \(e(kT_s)\) and width \(T_s\), shown by the solid blue in figure above:

\[
U_I (kT_s + T_s) = U_I (kT_s) + K_I \cdot T_s \cdot e(kT_s)
\]
• In the derivative term the roles of $u$ and $e$ are reversed from integration, and the consistent approximation can be written down at once from Eq. (4) as:

$$\frac{T_s}{2} \int U_0(k \cdot T_s + T_s) + U_0(k \cdot T_s) \, dt = k_0 \left[ e(k \cdot T_s + T_s) - e(k \cdot T_s) \right]$$

$\Rightarrow U_0(k \cdot T_s + T_s)$ can be expressed from (S)

SIMILAR to analog transfer functions, these relations are greatly simplified and generalized by the use of transform idea.

• In this course, we will introduce the discrete transform simply as a prediction operator $z$, the same as we described the Laplace transform variable $s$ as a differential operator.
THE DISCRETE TRANSFORM (Z-TRANSFORM)

- Prediction operator \( z \) is a forward shift operator in the sense that if \( U(z) \) is the transform of \( u(k \cdot T_s) \), then \( zU(z) \) will be the transform of \( u(k \cdot T_s + T_s) \)

- USING THE SHIFT OPERATOR the integral term can be written as:

\[
\begin{align*}
\text{Next value} \quad \hat{U}_1(z) &= U_1(z) + K_I \cdot \frac{T_s}{2} \left[ \frac{z}{2} \cdot E(z) + E(0) \right] \\
U_1(z) &= K_I \cdot \frac{T_s}{2} \cdot \frac{2+1}{2-1} \cdot E(z),
\end{align*}
\]

- From Eq. (5) the derivative term becomes the inverse:

\[
U_0(z) = K_D \cdot \frac{T_s}{T_s} \cdot \frac{2-1}{2+1} \cdot E(z)
\]

- The complete PID controller is thus described by:

\[
U(z) = \left( k_p + k_i \frac{T_s}{2} \frac{z+1}{z-1} + k_D \frac{2}{T_s} \frac{z-1}{z+1} \right) E(z), \quad (6)
\]

For the PID controller example, it can be seen that the effect of the discrete approximation in the z-domain is as if everywhere in the analog transfer function the operator \( s \) has been replaced by the composite operator \( (2/T_s)((z-1)/(z+1)) \). This is the trapezoid rule of discrete equivalent. The formula is also called Tustin’s Method.
EXAMPLE -- EQUIVALENT DISCRETE CONTROLLER FOR SPEED CONTROL

* Plant transfer function

\[ \frac{Y}{U} = \frac{45}{(s + 9)(s + 5)} = \frac{9}{s + 9} \cdot \frac{5}{s + 5}, \]

* PI controller designated for the plant

\[ D(s) = \frac{U}{E} = 1.4 \frac{s + 6}{s}. \]

• The closed loop system has a rise time of about 0.2 sec and an overshoot of about 20%. Design a discrete equivalent of this controller, and compare the step responses and control signals of the two system for:

a) \( T_s = 0.07 \), which is about 3 samples per rise time
b) \( T_s = 0.035 \), which is about 6 samples per rise time

SOLUTION TO EXAMPLE

Using the substitution given by the trapezoid rule the discrete equivalent for \( T_s = 0.07 \) is given by replacing \( s \) with \((2/0.07)(z-1)/(z+1)\) in \( D(s) \) as follows:
\[
\frac{U}{E} = D_d(z) = 1.4 \left[ \frac{z-1}{0.07 \left( \frac{z-1}{z+1} \right)} + 6 \right] = 1.4 \frac{1.21z - 0.79}{z-1} \quad (1)
\]

As a result, the equation for the controller is:

\[
U(z+1) = U(z) + 1.4 \left[ 1.21E(z+1) - 0.79E(z) \right] \quad (2)
\]

Note: In Eq. (2), \( T_s \) is suppressed.

Q: How did we get Eq. (2)?

A:

Careful. For \( T_s = 0.035 \), the discrete transfer function is:

\[
\frac{U}{E} = D_d = 1.4 \frac{1.105z - 0.895}{z-1} \quad (3)
\]

As a result, the difference equation is:

\[
U(z+1) = U(z) + 1.4 \left[ 1.105E(z+1) - 0.895E(z) \right] \quad (4)
\]

Note: Eqs. (2) & (4) give the control algorithm for the control system!
SUPPLEMENTAL READING: SIMULINK BLOCK DIAGRAM TO COMPARE CONTINUOUS AND DISCRETE CONTROLLERS

- MATLAB provides a command that converts a continuous transfer function \( D_c(s) = \frac{\text{numD}}{\text{denD}} \) represented as \( \text{sysDa} = \text{tf}([\text{numD}, \text{denD}]) \) to the discrete equivalent with sampling period \( T_s \):

\[
\text{sysDd} = \text{c2d} \left( \text{sysDa}, T_s, 't' \right);
\]

**Note:** only the controller is discrete, the plant is modeled by its transfer function. Having the transfer function, the MATLAB can be used to generate the time response for a step or other types of inputs.
• The discrete controller for $T_s = 0.07$ results in a substantial overshoot while for $T_s = 0.035$ the digital controller matches the performance of the analog controller fairly well.
• For controllers with many poles and zeros we can use MATLAB to generate the discrete transfer function.
ASSIGNMENTS

1. Derive the equations for the DC motor example with the controller and develop the MATLAB program for plotting the step response for both cases: disturbance input $W$ and reference input $R$.

2. [OPTIONAL] -- Implement the SIMULINK block diagram for EXAMPLE and plot the responses to a step input for the continuous and discrete controllers attached to the plant.