

HND Electrical and Electronic Engineering

PLC's and Control

PLC Assignment 2

By Aneurin Creed

Contents

Task 1	3
Principles of PID control.....	3
Proportional control.....	3
Integral control	4
Differential control.....	5
PID tuning methods	6
Trial and error tuning method	6
Ziegler-Nichols tuning method	6
Zeigler Nichols oscillation method.....	7
Task 2	8
Signal flow diagram.....	8
Mason Gain formula	9
Task 3	11
Root locus	11
Drawing root locus diagram for a system.	12
Task 4	14
State Space modelling.....	14
State space model.....	15
State space transfer function.....	17
Bibliography	18

Task 1

Principles of PID control

The basic idea of control within an open loop system is to figure out how to generate the appropriate input signal into the plant, so the system will produce the desired objective. The input signal is often found through experimental trial and error, trying different inputs to produce the required objective. The plant is a name for the system which can be thought of as two separate systems, the first is the actuator or actuators - devices that provide the force or energy to change the system. The second is the process – the system that the actuator is acting upon. Open loop systems are usually easy and conceptually simple. However, they are known to have shortcomings which make them unreliable due to variations and unexpected environmental changes.

Closed loop systems build upon the design of an open loop system by introducing feedback. Feedback within a closed loop system means that output signal is fed back and used to adapt the input. The adaptation to the input is calculated by comparing the fed back output with the desired objective, the difference in this comparison is known as the error. The closer to the output is to the desired objective, the smaller the error gets. Therefore, converting the error value to zero is a feedback system's objective. This can be achieved by using a P.I.D controller.

Proportional Integral Differential controllers are versatile controllers as they use the present error, past error, and a prediction of the future error to calculate the appropriate actuator command.

Proportional control

The Proportional controller produces a signal to the actuator which is directly proportional to the error signal. The actuating signal is equal to the product of the error signal and the gain of the proportional controller.

$A(t)$ is the actuating signal $e(t)$ is the error signal

$$A(t) \propto e(t) \quad \Rightarrow \quad A(t) = K_p \times e(t)$$

Where K_p is a proportional constant, which is the controllers gain.

Proportional controllers have varying success depending on the system and situation. For example, a proportional controller would be extremely effective if the systems objective were to move an object laterally 100m from point A to point B. This would be simple to calculate as it would be the multiplication of the remaining distance (error) by the gain. If the gain were 0.1 and the initial distance to travel was 100m, the output would be 10m/s. As the object would get closer to the target the error would reduce and so would the output. For example, 30m remaining output 3m/s, 10m remaining output 1m/s. Until finally reaching the desired location (100m), the remaining distance would then be 0m and thus the output would be 0m/s and the object would stop. In this scenario the proportional controller works as required.

However, in a scenario where the system's objective is to reach a vertical distance of 100m with a helicopter, the problem then changes. Presuming that at 100rpm the helicopter hovers, if our proportional gain is 1 then the helicopter would hover at ground level. This is because the gain of 1 multiplied by the error 100m would give an output of 100rpm causing the helicopter to hover.

Height	Error	Gain	Output=Error × Gain
0m	100m	1	100 rpm
50m	50m	2	100 rpm
80m	20m	5	100 rpm

Even a gain of 100 would mean that the helicopter would only reach 99m. No matter how large the gain becomes the error will never be at 0, it will only get smaller with a proportional controller. This is known as a steady state error. To overcome this, the controller needs to use past information - this can be achieved by using an integral controller in the system.

Integral control

The Integral controller produces a signal to the actuator which is directly proportional to the integral of the error signal. The actuating signal is equal to the integral of the error signal multiplied by the gain of the integral controller.

$A(t)$ is the actuating signal

$e(t)$ is the error signal

$$A(t) \propto \int_0^t e(t) \Rightarrow A(t) = K_i \times \int_0^t e(t)$$

Where K_i is a proportional constant, which is the integrals gain also known as the reset controller.

An integrator controller keeps a running total of the inputs - the error signals over the running time - and sums them together. If the system reaches a steady state below the desired objective, then the error signal is nonzero, and when a nonzero value is integrated the output will increase. This value from the integrator will then be combined with the proportional value, thus driving the error value to zero. Once the error value is at zero, the proportional controller is no longer engaged but the integral controller sums and subtracts values until it reaches the desired objective. This output from the integrator will not change as the input is zero.

However, the path to the desired objective might not have been ideal. The integrator may have overshoot the objective causing a negative error, which when summed would cause the output signal to drop - potentially back below the desired objective. Eventually, the output signal will find the desired output. Overshooting is not preferable and sometimes can be dangerous in some systems. A simple way to overcome this is with a differential controller which can predict the future and respond to the rate at which the error is being reduced.

Differential control

The Differential controller produces a signal to the actuator which is directly proportional to the differential of the error signal. The actuating signal is equal to the differential of the error signal multiplied by the gain of the differential controller.

A(t) is the actuating signal

e(t) is the error signal

$$A(t) \propto \frac{de(t)}{dt} \quad \Rightarrow \quad A(t) = K_d \times \frac{de(t)}{dt}$$

Where K_d is a proportional constant, which is the differential gain also known as the rate controller.

The derivative produces a measure of the rate of change of the error. For example, if the error is quickly decreasing then it has a negative rate of change, meaning the derivatives output will be negative. That negative value will be added to the controller's output, thus reducing the speed at which system is reaching the desired objective. This prevents the system from overshooting.

PID tuning methods

Trial and error tuning method

There are several tuning methods that are used in order to obtain efficient PID performance. The first is to test the PID manually through a process of trial and error. This works through setting the integral and derivative terms to zero and then increasing the proportional gain until the output of the control loop oscillates at a constant rate. This increase in proportional gain should allow the system to become faster, providing that it is not made unstable. Then, once the P-response is fast enough, the integral term is set, and the oscillations can be gradually reduced. The I-Value should be altered until the steady state error is reduced, but this may increase overshoot. (PID Controller-Working and Tuning Methods, 2015) Finally, once P and I parameters have been altered to obtain the system's desired values, the derivative gain should be increased until the system reacts quickly to its set point.

One of the main advantages of this method is that it requires a minimum amount of preparation work and no plant testing is required to achieve the results needed. Whether used with a closed loop or open loop system, this is advantageous because tests are often a burden to the operation staff.

However, in terms of disadvantages, the trial and error method of tuning can be very time-consuming. There is no systematic path to the final answer and thus it is left up to the experience of the user. In other words, it comes down to an individual's ability to extract the information contained in the behaviour of the PV and the controller output and to turn these findings into meaningful adaptations of the controller settings.

Ziegler-Nichols tuning method

Another tuning method for a PID is the Ziegler-Nichols or 'continuous cycling' method. As this method works off a closed loop system, this means that the system is being controlled by a controller in the analysis. In order to use this tuning method it is essential that we have some understanding of the dynamic characteristic of how the control loop behaves, which can be done through the following procedure: firstly, the I control and D controls are shut off in order to leave a P only controller. Then the controller should be put into automatic, thus making the system into a closed loop. The next step is to alter the controller gain, k_c , until the response that is seen is undamped. This leaves the system in a state where the oscillations are neither growing nor receding.

One of the main advantages of the Ziegler-Nichols method is its simplicity in terms of tuning rules. However, in terms of the disadvantages, this method often results in further fine tuning being needed. Furthermore, the controller settings can be seen as aggressive, which can result in large overshoot and oscillatory responses. (Eder, 2010)

Zeigler Nichols oscillation method

An open loop plant has a transfer function of $\frac{(2-s)}{(s+3)^2}$. Using the Zeigler Nichols oscillation method, the parameters for the PID controller can be found.

The closed loop characteristic polynomial = $1 + K_p G_0(s) = 0$

$$G_0(s) = \frac{(2-s)}{(s+3)^2}$$

$$1 + K_p \frac{(2-s)}{(s+3)^2} = 0$$

$$(s+3)^2 + K_p(2-s) = 0 \times (s+3)^2$$

$$(s+3)^2 + K_p(2-s) = 0$$

At the critical point $K_p = K_c$ and $s = j\omega_c$, the critical frequency.

$$(s+3)^2 + K_p(2-s) = (j\omega_c+3)^2 + K_c(2-j\omega_c) = 0$$

$$K_c = \frac{-(j\omega_c+3)^2}{(2-j\omega_c)}$$

Multiply by the conjugate to remove the imaginary “j” from the denominator.

$$\frac{-(j\omega_c+3)^2}{(2-j\omega_c)} \times \frac{(2+j\omega_c)}{(2+j\omega_c)} = \frac{-(j^2\omega_c^2+9j\omega_c+9) \times (2+j\omega_c)}{(4+\omega_c^2)}$$

$$K_c = \frac{-(j^2\omega_c^2+9j\omega_c+9) \times (2+j\omega_c)}{(4+\omega_c^2)} = \frac{-(-2\omega_c^2+18j\omega_c+18-j^3\omega_c^3+9j^2\omega_c^2+9j\omega_c)}{(4+\omega_c^2)}$$

$$K_c = \frac{-(-2\omega_c^2+18-j^3\omega_c^3+9j^2\omega_c^2+27j\omega_c)}{(4+\omega_c^2)} = \frac{-(-11\omega_c^2+18+j\omega_c^3+27j\omega_c)}{(4+\omega_c^2)}$$

$$K_c = \frac{(11\omega_c^2-18-j(\omega_c^3-27\omega_c))}{(4+\omega_c^2)} \Rightarrow 11\omega_c^2 - 18, \text{ Real.} \quad -j(\omega_c^3 - 27\omega_c), \text{ Imaginary.}$$

For K to be real, $\omega_c^3 - 27\omega_c = 0$

$$\omega_c^3 = 27\omega_c$$

$$\omega_c^2 = 27$$

$$\omega_c = \sqrt{27}$$

$$K_c = \frac{(11\omega_c^2-18)}{(4+\omega_c^2)} = \frac{((11 \times 27) - 18)}{(4+27)} = \frac{(279)}{31} = 9$$

$$\text{Period} = \frac{(2\pi)}{(\omega_c)} = \frac{(2\pi)}{(\sqrt{27})}$$

Using the bottom row of the table, values for K_p , T_i and T_d can be found.

$$K_p = 0.6 \times 9$$

$$T_i = 0.5 \times \frac{(2\pi)}{(\sqrt{27})}$$

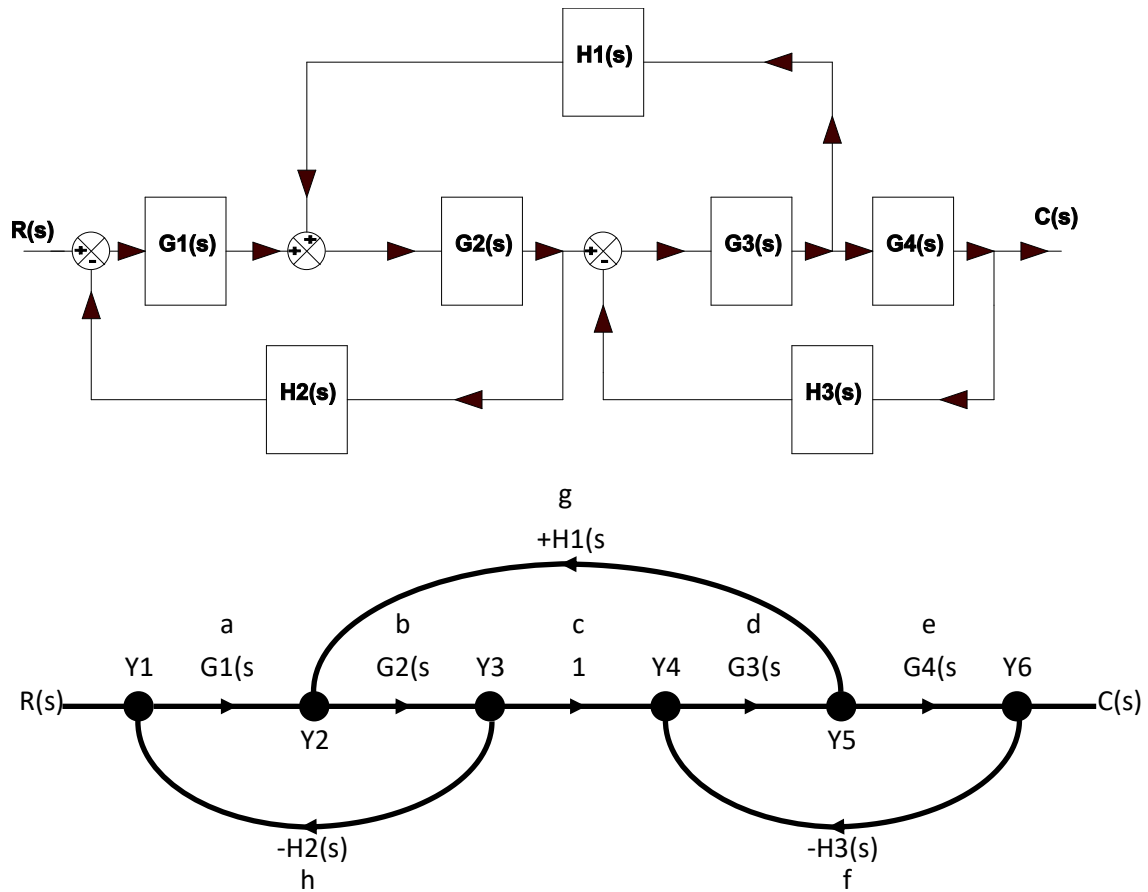
$$T_d = 0.125 \times \frac{(2\pi)}{(\sqrt{27})}$$

Type of Controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$\frac{1}{1.2}P_{cr}$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Task 2

Signal flow diagram

System block diagram converted to signal flow diagrams



Mason Gain formula

$$T = \frac{C(s)}{R(s)} = \frac{\sum_{i=1}^N P_i \Delta_i}{\Delta}$$

C(s) is the output node

R(s) is the input node

T is the transfer function or gain between R(s) and C(s)

P_i is the *i*th forward path gain

$$\begin{aligned} \Delta = & 1 - (\text{sum of all individual loop gains}) \\ & + (\text{sum of gain products of all possible two non-touching loops}) \\ & - (\text{sum of gain products of all possible three non-touching loops}) \\ & + \dots \end{aligned}$$

Δ_i is obtained from Δ by removing the loops which are touching the *i*th forward path.

Applying Mason Gain formula to above signal flow diagram.

Number of forward paths, $N = 2$.

- First forward path - $y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_4 \rightarrow y_5 \rightarrow y_6$.
- First forward path gain, $P_1 = abcde$.
- Second forward path - $y_1 \rightarrow y_2 \rightarrow y_5 \rightarrow y_6$.
- Second forward path gain, $P_2 = age$.

Number of individual loops, $L = 3$.

- First loop - $y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_1$.
- First loop gain, $L_1 = -abh$.
- Second loop - $y_4 \rightarrow y_5 \rightarrow y_6 \rightarrow y_4$.
- Second loop gain, $L_2 = -def$.
- Third loop - $y_2 \rightarrow y_3 \rightarrow y_4 \rightarrow y_5 \rightarrow y_2$.
- Third loop gain, $L_3 = bcdg$.

Number of two non-touching loops = 0

$$\Delta = 1 - (\text{sum of all individual loop gains})$$

$$\Delta = 1 - (-abh + -def + bcdg) + 0$$

$$\Delta = 1 + abh + def - bcdg$$

There is no loop which is non-touching to the first forward path.

$$\text{So, } \Delta_1 = 1$$

There is no loop which is non-touching to the second forward path.

$$\text{So, } \Delta_2 = 1$$

$$T = \frac{C(s)}{R(s)} = \frac{\sum_{i=1}^N P_i \Delta_i}{\Delta}$$

Substitute N=2 into Mason's gain formula

$$T = \frac{\sum_{i=1}^2 P_i \Delta_i}{\Delta}$$

$$T = \frac{P_1 \Delta_1 + P_2 \Delta_2}{\Delta}$$

$$T = \frac{abcde.1 + age.1}{1 + abh + def - bcdg}$$

$$T = \frac{abcde + age}{1 + abh + def - bcdg}$$

Task 3

Root locus

Root locus analysis refers to a method used to examine how the roots of a system variation in relation to the changing of a certain system parameter, usually a gain within a feedback system. It is an important tool in the analysis and design of control systems.

The root locus is a graphical representation that usually depicts a control system. The depiction displays the location of the plotted poles and zeros on both the real and the imaginary axis. Depending on the arrangement of said poles and zeros, root locus helps to gain a better understanding of how the closed loop system behaves. This information can then be used to decide how to set the gain of a controller to achieve the desired closed-loop response.

Drawing root locus diagram for a system.

i)

When $K=0$

$$G(s) = \frac{1}{(s+1)(s+3)(s+5)}$$

$$(s+1)(s+3)(s+5) = 0$$

$n=3$ Poles at -1, -3, -5

$m=0$ There are no zeros

$$\frac{\frac{K}{(s+1)(s+3)(s+5)}}{1 + \frac{K}{(s+1)(s+3)(s+5)}} \times (s+1)(s+3)(s+5) = \frac{K}{(s+1)(s+3)(s+5) + K}$$

$$\therefore (s+1)(s+3)(s+5) + K = 0$$

Equation is of 3rd order. So, 3 loci.

Angle of asymptote- $n-m=3-0=-3$ $\pi/3=60^\circ$

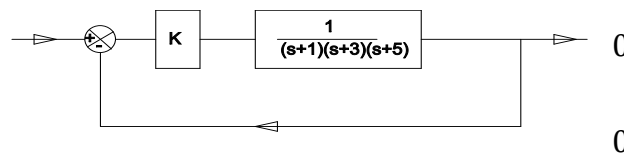
As $S = j\omega$

$$(j\omega + 1)(j\omega + 3)(j\omega + 5) + K = 0$$

$$23j\omega + 15 + K =$$

$$\text{As } j^2 = -1$$

$$23j\omega + 15 + K =$$



$$j^3\omega^3 + 9j^2\omega^2 +$$

$$-j\omega^3 - 9\omega^2 +$$

j parts

$$-\omega^3 + 23\omega = 0$$

$$\div \omega$$

$$-\omega^2 + 23 = 0$$

$$23 = \omega^2$$

$$\sqrt{23} = \omega$$

Real parts

$$-9\omega^2 + 15 + K = 0$$

$$-9(23) + 15 + K = 0$$

$$-207 + 15 + K = 0$$

$$-192 + K = 0$$

$$K = 192$$

Break away point

$$(s+1)(s+3)(s+5) + K = 0$$

$$K = -(s+1)(s+3)(s+5)$$

$$K = -(s^3 + 9s^2 + 23s + 15)$$

$$K = -s^3 - 9s^2 - 23s - 15$$

$$\frac{dk}{ds} = -3s^2 - 18s - 23$$

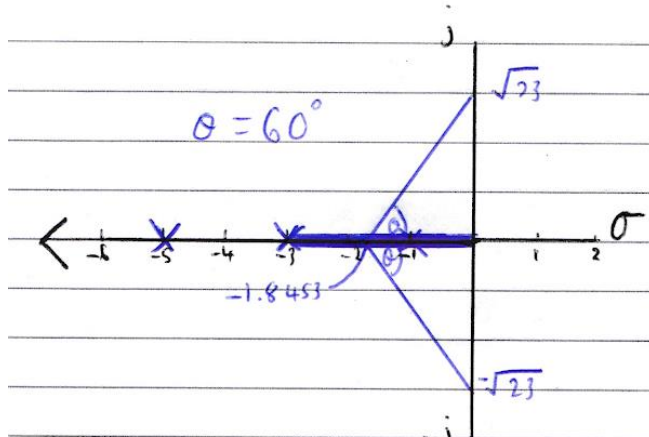
$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$x = \frac{-18 \pm \sqrt{(-18)^2 - 4(-3)(-23)}}{2(-3)}$$

$$\text{Either } x = \frac{18}{-6} + \frac{\sqrt{48}}{-6} \quad \text{Or } x = \frac{18}{-6} - \frac{\sqrt{48}}{-6}$$

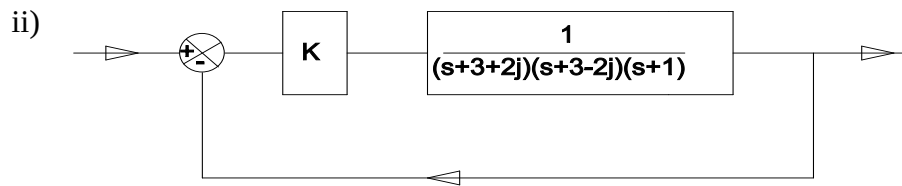
$$x = -4.1547$$

$$x = -1.8453$$



$$K = \frac{1}{G(s)} = \frac{\frac{1}{1}}{(s+1)(s+3)(s+5)} = \frac{1}{15}$$

$$K = 15$$



When $k=0$

$$G(s) = \frac{1}{(s+3+2j)(s+3-2j)(s+1)} \quad (s+3+2j)(s+3-2j)(s+1) = 0 \quad n=3 \text{ Poles at } -3, \pm 2j, -1$$

$m=0$ There are no zeros

$$\frac{\frac{K}{(s+3+2j)(s+3-2j)(s+1)}}{1 + \frac{K}{(s+3+2j)(s+3-2j)(s+1)}} \times (s+3+2j)(s+3-2j)(s+1) = \frac{K}{(s+3+2j)(s+3-2j)(s+1)+K}$$

$$\therefore (s+3+2j)(s+3-2j)(s+1) + K = 0$$

Equation is of 3rd order. So, 3 loci.

$$\text{Angle of asymptote- } n-m=3-0=-3 \quad \pi/3=60^\circ$$

$$\text{As } S = j\omega$$

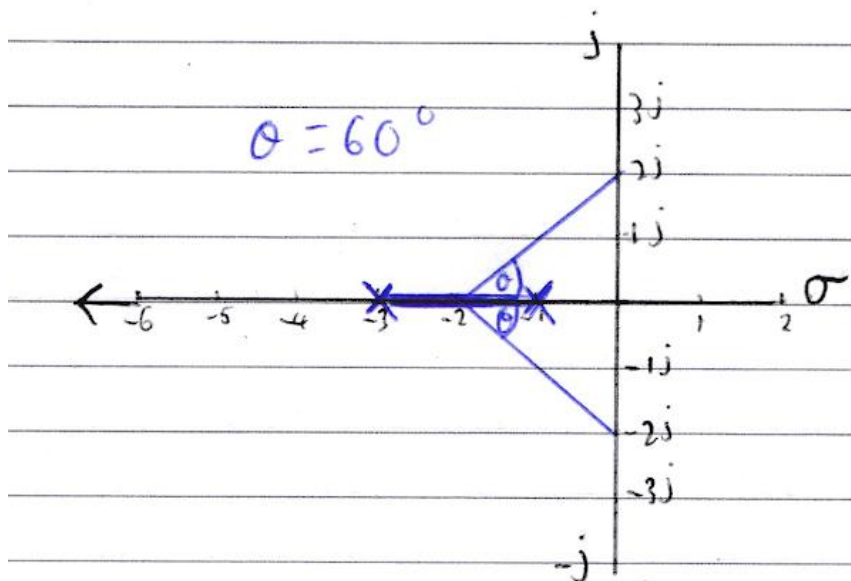
$$(j\omega + 3 + 2j)(j\omega + 3 - 2j)(j\omega + 1) + K = 0$$

$$j^3\omega^3 - 2j^2\omega + 3j^2\omega^2 - 9j\omega + 6j - 4j^3\omega - 4j^2 + 9 + K = 0$$

$$\text{As } j^2 = -1$$

$$-j\omega^3 + 2\omega - 3\omega^2 - 9j\omega + 6j - 4j\omega + 4 + 9 + K = 0$$

$$-j\omega^3 - 3\omega^2 - 13j\omega + 6j - 3\omega^2 + 2\omega + 13 + K = 0$$

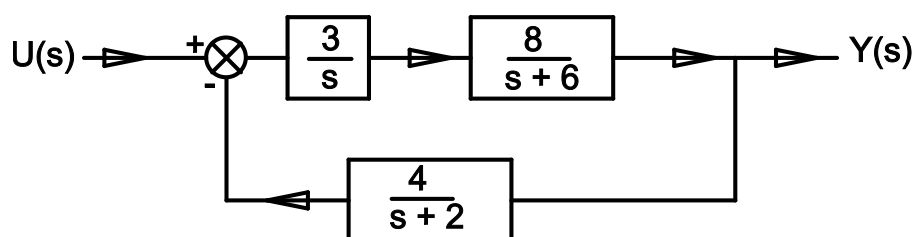


Task 4

State Space modelling

State space model (SSM) refers to a class of probabilistic graphical model (Koller and Friedman, 2009) that describes the probabilistic dependence between the latent state variable and the observed measurement. The state or the measurement can be either continuous or discrete. State-space models work through using state variables in order to define a system by a set of first-order differential equations, instead of n th-order differential or difference equations.

State space model



$$\left. \begin{array}{l} x_2 \\ \} \\ s \end{array} \right\} \quad \begin{array}{l} x_1 \\ \frac{8}{s+6} \end{array}$$

$$\begin{array}{l} x_3 \\ \frac{4}{s+2} \end{array}$$

$$x_1 = \frac{8}{s+6} x_2$$

$$x_1(s) = \frac{8}{s+6} x_2(s)$$

$$\frac{x_1(s)}{x_2(s)} = \frac{8}{s+6} \quad (1)$$

$$\frac{x_2(s)}{u(s) - x_3(s)} = \frac{1}{s} \quad (2)$$

$$\frac{x_3(s)}{x_1(s)} = \frac{4}{s+2} \quad (3)$$

$$y(s) = x_1(s) \quad (4)$$

$$\text{Eq (1)} \quad \frac{x_1(s)}{x_2(s)} = \frac{8}{s+6} \Rightarrow x_1(s)(s+6) = 8x_2(s)$$

$$\begin{aligned} s x_1(s) + 6 x_1(s) &= 8 x_2(s) \\ s x_1(s) &= -6 x_1(s) + 8 x_2(s) \end{aligned} \quad (5)$$

$$\text{Eq (2)} \quad \frac{x_2(s)}{u(s) - x_3(s)} = \frac{1}{s} \Rightarrow s x_2(s) = u(s) - x_3(s) \quad (6)$$

$$\text{Eq (3)} \quad \frac{x_3(s)}{x_1(s)} = \frac{4}{s+2} \Rightarrow x_3(s)(s+2) = 4 x_1(s)$$

$$\begin{aligned} s x_3(s) + 2 x_3(s) &= 4 x_1(s) \\ s x_3(s) &= -2 x_3(s) + 4 x_1(s) \end{aligned} \quad (7)$$

$$\text{Eq (4)} \quad y(s) = x_1(s)$$

take Laplace⁻¹ of ⑤ ⑥ ⑦ ④

$$⑤ \Rightarrow x_1 = -6x_1 + 8x_2$$

$$⑥ \Rightarrow x_1 = 4 - x_3$$

$$⑦ \Rightarrow x_3 = -2x_3 + 4x_1$$

$$④ \Rightarrow y = x_1$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -6 & +8 & 0 \\ 0 & 0 & -1 \\ +4 & 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

State space transfer function

$$\dot{X} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -3 & 2 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [u]$$

$$Y = [0 \ 1] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad D = [0]$$

$$G_p(s) = [0 \ 1] \begin{bmatrix} s+3 & -2 \\ -1 & s-4 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + C$$

$$E = \begin{bmatrix} s+3 & -2 \\ -1 & s-4 \end{bmatrix} \quad \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$E^{-1} = \frac{1}{\det(E)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

$$= \frac{1}{(s+3)(s-4) - (-2 \times -1)} \begin{bmatrix} s-4 & 2 \\ 1 & s+3 \end{bmatrix}$$

$$G_p(s) = [0 \ 1] \frac{1}{(s+3)(s-4) - (-2 \times -1)} \begin{bmatrix} s-4 & 2 \\ 1 & s+3 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$G_p(s) = [0 \ 1] \frac{1}{s^2 - s - 14} \begin{bmatrix} 0+2 \\ 0+(s+3) \end{bmatrix}$$

~~$$G_p(s) = \frac{0 \times (0+2)}{1 \times (s+3)} \times \frac{1}{s^2 - s - 14}$$~~

$$G_p(s) = \frac{1}{s^2 - s - 14} [0 \times (0+2) \quad 1 \times (s+3)]$$

$$G_p(s) = \frac{s+3}{s^2 - s - 14}$$

Bibliography

Eder, H., 2010. [online] Act-control.com. Available at: <<https://www.act-control.com/publications/PID%20Tuning.pdf>> [Accessed 18 June 2020].

Electronics Hub. 2015. *PID Controller-Working And Tuning Methods*. [online] Available at: <<https://www.electronicshub.org/pid-controller-working-and-tuning-methods/>> [Accessed 18 June 2020].

Koller, D. and Friedman, N., 2009. [online] List01.bio.ens.psl.eu. Available at: <https://list01.bio.ens.psl.eu/wws/d_read/machine_learning/BayesianNetworks/koller.pdf> [Accessed 21 June 2020].