MECH 467/MECH589 -Computer Control of Mechatronics Systems (Copy Write Protected)

Yusuf Altintas, Professor
University of British Columbia, Mechanical Engineering Department,
Vancouver, B.C. Canada www.mal.mech.ubc.ca

September 2020

Contents

1 Introduction **4**
1.1 List of core study topics . . . 5

2 Review of System Modeling and Laplace Transforms **7**
2.1 Block Diagram Algebra . . . 7
2.2 Review of Laplace Transforms . . . 9

2.2.1 Inverse Laplace Transform by Partial Fraction Expansion . . . 12
2.2.2 Solution of Ordinary Linear Differential Equations with Laplace Transforms . . . 14
2.3 Review of Complex Numbers . . . 19
2.4 Frequency Response Function (FRF) of Continuous Systems . . . 20

3 Continuous and Discrete Time Control Systems **23**
3.1 z Transforms . . . 23

3.2 Zero Order Hold Equivalent of Continuous Systems . . . 26
3.3 Approximate Transformations between Continuous (s) and Discrete Time Domain Systems . 29
3.4 Summary . . . 36

4 Closed Loop Control System **39**
4.1 Transfer Function of Closed Loop Systems . . . 39
4.2 Time Domain Response of Closed Loop Control Systems . . . 39
4.2.1 Steady State Error of the Closed Loop Control System . . . 44
4.3 Discrete Identification of Dynamic Systems . . . 46
4.3.1 Off-Line Least Squares Estimation . . . 46
4.3.2 Recursive Parameter Estimation Algorithm . . . 47
4.4 Dynamics of a Position Control Servo System . . . 48

4.4.1 Mechanical Components and Torque Requirements . . . 50
4.4.2 Feedback Devices . . . 52
4.4.3 Electrical Drives . . . 53
4.4.4 Permanent Magnet Armature-Controlled dc Motors . . . 53
4.4.5 dc Motor Power Amplifiers and Velocity Control Loop . . . 55
4.4.6 Position Control Loop . . . 59
4.4.7 Digital Compensation Filter . . . 59
4.4.8 Digital to Analog (D/A) Converter . . . 59
4.5 Transfer Function of the Position Loop . . . 59
4.6 State Space Model of Dynamic Systems . . . 62

5 Stability Analysis of Continuous and Discrete Time Closed Loop Control Systems 79
5.1 Introduction . . . 79
5.1.1 $z-$Domain . . . 80
5.2 Stability maps in s and z planes . . . 82
5.3 Summary: Stability of Closed Loop Control Systems . . . 84
5.4 Root Locus in Laplace (s) and Discrete (z) Domain . . . 87
5.4.1 Introduction . . . 87
5.4.2 Root Locus . . . 89
5.5 Nyquist Stability Criterion . . . 93
5.5.1 Vector Representation of Poles and Zeros . . . 94
5.5.2 Conformal Mapping . . . 95
5.5.3 Nyquist Stability Law . . . 96
5.6 Bode Plot . . . 99
5.6.1 Approximate Bode Plots . . . 107
5.7 Frequency Response of a Closed Loop System . . . 109

6 Design of Control Laws 111
6.1 Introduction . . . 111
6.2 Proportional Controller Design . . . 112
6.2.1 Proportional Controller Design in s Domain . . . 112
6.2.2 Proportional Controller Design in z Domain . . . 118
6.2.3 Proportional Integral Derivative (PID) Controller Design . . . 122
6.3 Lead-Lag Compensator Design in Frequency Domain . . . 124
6.3.1 Design of Low Pass Filter . . . 126
6.3.2 Design of Phase-Lead Controller . . . 128
6.3.3 Design of Phase-Lag Controller . . . 131
6.4 Summary of Stability of Closed Loop Control Systems . . . 143
6.4.1 Controller Design in Laplace (continuous time) Domain : . . . 143
6.4.2 Controller Design in z (discrete time) Domain . . . 144
6.5 Pole Placement Control System Design . . . 145
6.6 Sliding Mode Controller . . . 153

7 Trajectory Generation for Computer Controlled Machines 158
7.1 Introduction . . . 158
7.1.1 Acceleration - Limited Velocity Profile Generation with Constant Interpolation Period 158
7.2 Jerk-Limited Velocity Profile Generation . . . 162
7.2.1 Kinematic Profiles. . . . 162
7.2.2 Initialization. . . . 165
7.2.3 Number of Interpolation Steps in Each Phase. . . . 165
7.2.4 Recursively Executed Real Time Part. . . . 167
7.2.5 Reconstruction of Reference Trajectory at Servo Loop Control Frequency. . . . 167
7.3 Real Time Interpolation Methods . . . 169
7.3.1 Linear Interpolation Algorithm . . . 169
7.3.2 Circular Interpolation Algorithm . . . 173
7.3.3 Quintic Spline Interpolation within CNC Systems . . . 180

8 Problems 187

Preface

These notes are prepared to provide basics of motion control system design which consists of actuator sizing, selection and modeling; their digital control, trajectory generation and multi axis path interpolation in real time. The contents, laboratory projects and problem sets have largely been developed in the Manufacturing Automation Laboratory, The University of British Columbia where the instructor teaches.

Chapter 1

Introduction

The topic of Computer Control of Mechatronics Systems covers a very wide area, ranging from modeling, dynamics and control to the implementation of such fundamental engineering areas using actuators, sensors, embedded systems and real time software engineering. This particular text is geared towards teaching the digital control of multi-axis motion control systems used in machine tools, robots, conveyors, positioning tables and CD drives.

The concept of digital control engineering is explained on a feed drive system given in Figure 1.1. Operator enters the motion commands to the CNC System (i.e. computer controller). The software in CNC decodes the commands as linear, circular or splined paths. The paths are digitized in small, discrete time intervals (i.e. 1ms) as a function of trajectory motion profiles, i.e. time dependent smooth velocity, acceleration, and jerk curves. The time stamped motion path is decoupled into axis commands by using the kinematic model of the machine. The resulting position commands are sent to each axis controller which is equipped with position feedback sensors. The velocity of the machine is manipulated by sending the amount of position commands sent at each control interval. The axis controller is an equation which is solved in each control interval (i.e. 1ms), and the resulting number is sent to motor amplifier as a voltage command. The amplifier converts the voltage command into a current on the armature of the motor, creating a magnetic torque. Torque rotates the motor shaft which is connected to ball screw. The ball screw is connected to the table via a nut. When the ball screw rotates one full revolution, the table moves one pitch length of the screw. The angular motion of the ball screw is monitored by an optical encoder which may have a resolution of one micron or less. The encoder is fed back to computer where the error between the commanded and actual position is computed. The error is processed by the control law, and the process is continued until the machine arrives at the commanded position. This text will troche the design of such a control system step by step as follows.

The text starts with the review of dynamic system modeling, open and closed loop system dynamics, transient and frequency response, proportional control, and the stability of the systems in Laplace domain. The concept of discrete time modeling of continuous systems is introduced by Zero Order Hold Equivalent and z transforms, and the concept of digital control system design is presented. The mapping of poles and zeros in s and z planes are shown along with the system stability and transient response characteristics. Controller design using root locus technique is given.

The identification of system dynamics using the laws of physics as well as experimental techniques are presented. Several common control law design techniques are explained. Proportional (P), Proportional - Derivative (PD), Proportional - Integral (PI) and Proportional -Integral - Derivative (PID) techniques are covered. Pole placement control law in discrete time domain is given with an application to an electro-hydraulic system. Sliding Mode Controller design is presented to introduce nonlinear control techniques which has improved robustness against hysteresis, nonlinearity, and disturbance while having high frequency bandwidth.

Motion control systems need to receive smooth, time stamped position commands in order to avoid discontinuity in displacement, velocity and preferably acceleration and jerk. Trajectory generation techniques with constant acceleration and jerk are introduced. The interpolation of multi - axis machines using linear and circular paths is illustrated. Dynamic states, such as position, velocity, acceleration and current of the

system, are estimated using State Space Models.

Each theoretical subject is applied experimentally in the laboratory. One and two axis drive systems are modeled and controlled in real time by each student.

1.1 List of core study topics

1. Laplace Transforms, Block Diagram Algebra. Open and Closed Loop Systems. Proportional, Proportional - Integral and Proportional - Derivative- Integral Control structures.
2. Transient Response of First, Second and Third Order Systems using Inverse Laplace Transforms
3. Frequency Response of Systems: First order, second order, time delay, high order systems. Magnitude and phase diagrams using Bode Plots, Logarithmic and linear scales in Frequency Domain representation of systems. Real and Imaginary parts, polar plots.
4. Stability of Closed Loop Systems: Root Locus, and Nyquist Stability criterion. Gain and Phase Margins.
5. Design of Proportional Controller using Root locus and Nyquist charts. Pole cancellation, pole assignment based control system design.
6. Discrete Time Systems, z transforms, zero order hold (ZOH) equivalent of continuous systems. Mapping between s and z planes.
7. Pole Placement Control System Design. Sliding Mode Controller design with feed-forward disturbance cancellation.
8. State Space Model of the systems in continuous and discrete time domain.
9. Acceleration and Jerk limited trajectory generation
10. Linear and circular interpolation for two axis motion control systems.
11. Integrated design of a complete, multi axis motion control system.

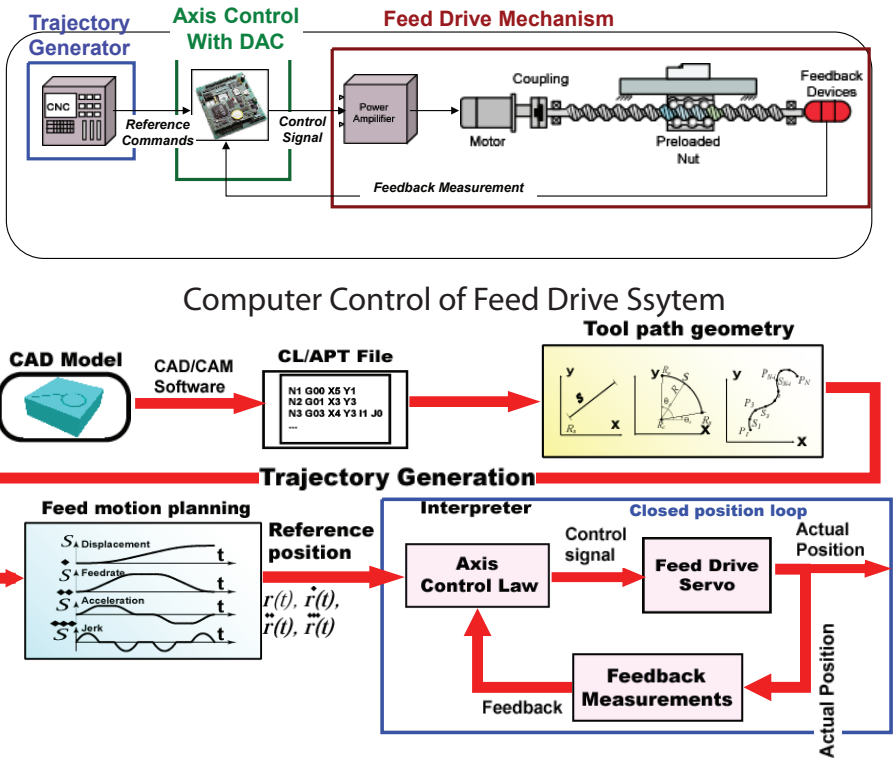


Figure 1.1: Architecture of a computer control system for drives.

Chapter 2

Review of System Modeling and Laplace Transforms

2.1 Block Diagram Algebra

Automatic control systems are graphically represented using block diagrams shown in Figure 2.1

Input to the system is $u(s)$, and the output from the system is $v(s)$, where s is the Laplace operator. The system dynamics are represented by its transfer function $G(s)$.When the blocks are in serial, the equivalent transfer function becomes the product of two transfer functions, i.e. $G(s) = G_1(s)G_2(s)$, see Figure 2.1. When they are in parallel, the transfer functions are summed to obtain the equivalent transfer function, i.e. $G(s) = G_1(s) + G_2(s)$.

The block diagrams can be manipulated to convert them into more convenient forms as shown in Figure 2.2.

A typical closed loop system is shown in Figure 2.3. The output from the physical system $y(s)$ is measured with a sensor and signal conditioning circuit/algorithm which yields feedback signal $v(s)$. The feedback is subtracted from the reference input $r(s)$ if the feedback is negative $(-)$, or added to the input $r(s)$ if the feedback is positive $(+)$. When the output is desired to follow the input to minimize the tracking error $(e(s))$, which is the case in most mechatronics systems, the feedback signal is always subtracted from the reference input $r(s)$.The difference between the command input and feedback is the servo error $e(s) = r(s) - v(s)$. The error is processed by the controller $D(s)$, and the resulting controller command $u(s)$ is sent to the physical system actuator $G(s)$. It is important to remember the following convention in control systems :

$r(s)$	:	Reference input	$C(s)$	:	Control law/algorithm
$e(s)$	:	Tracking error	$G(s)$	:	Plant's transfer function
$u(s)$	:	Controller command	$H(s)$	:	Feedback transfer function
$y(s)$	:	System output	$C(s)G(s)$	:	Feed forward transfer function
$v(s)$	:	Feedback signal	$C(s)G(s)H(s)$	:	Open loop transfer function
$d(s)$	:	External Disturbance	$\frac{y(s)}{r(s)}$	:	Closed loop transfer function
$1 + C(s)G(s)H(s)$	:	Characteristic equation	$\frac{y(s)}{d(s)}$	:	Disturbance transfer function

By neglecting the Laplace transform operator (s) in the notation, the system's transfer functions can be evaluated from the block diagram as follows:

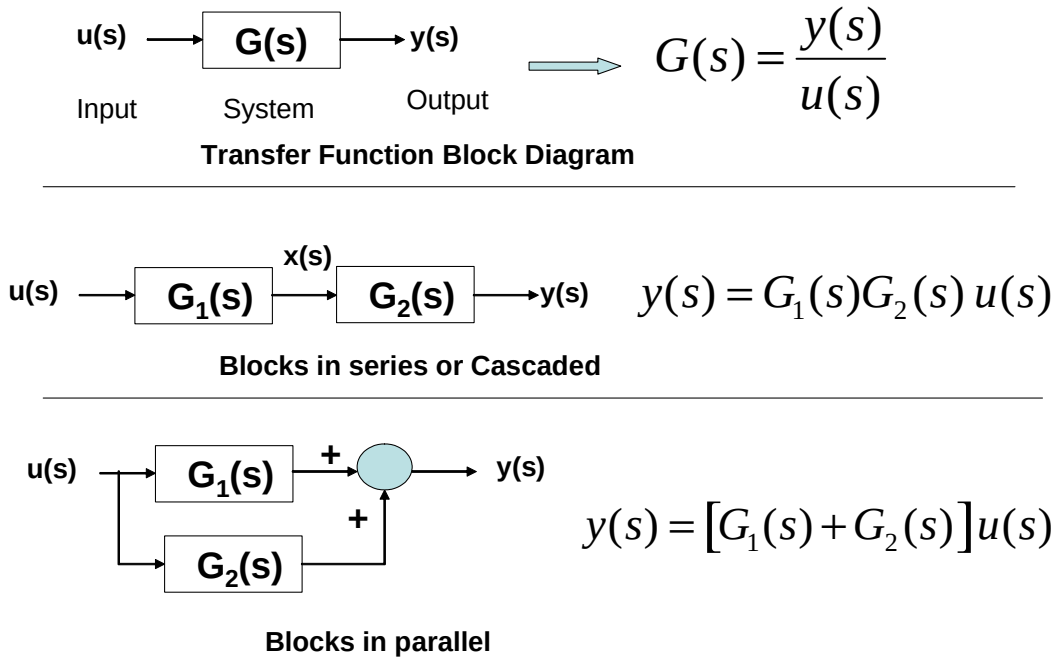


Figure 2.1: Blocks conected in series and parallel fashion.

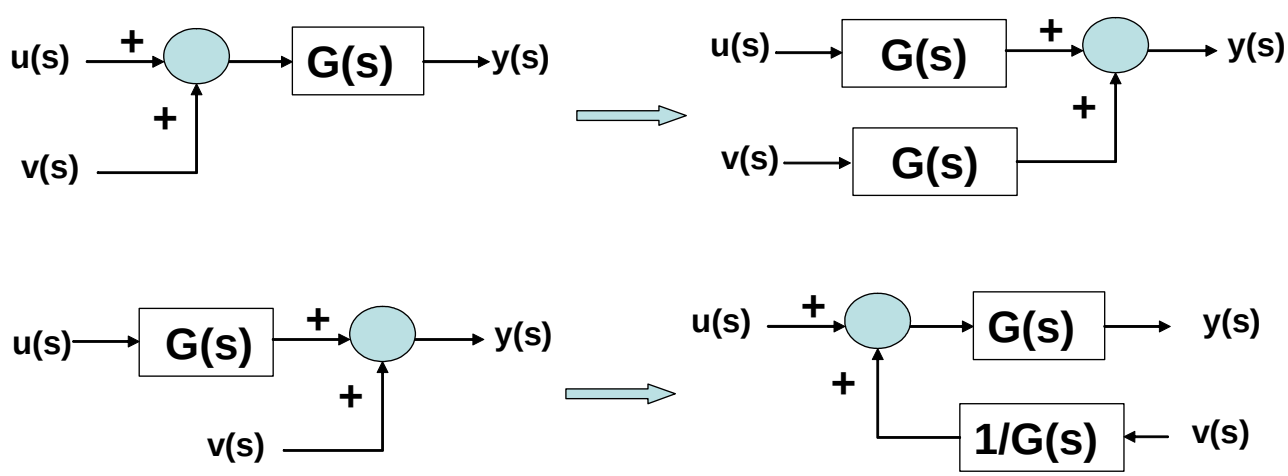


Figure 2.2: Manipulation of block diagrams.

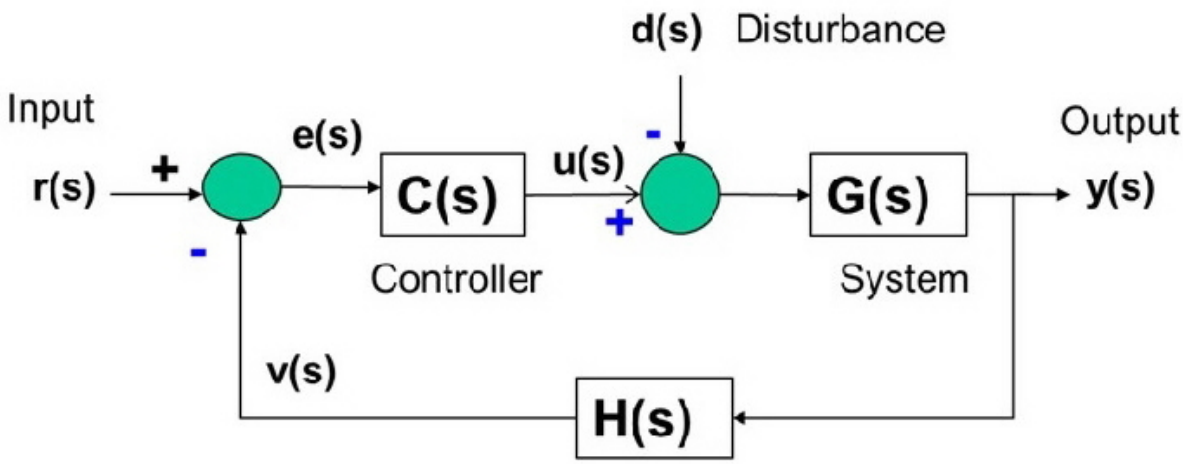


Figure 2.3: Closed Loop system dynamics.

$$
\begin{aligned}
e &= r - v = r - Hy \\
u &= Ce = C\left(r - Hy\right) \\
y &= \left(u + d\right)G = \left[C\left(r - Hy\right) + d\right]G = CGr - CGHy + dG \\
\left[1 + CGH\right]y &= CGr + Gd \\
y &= \frac{CG}{1 + CGH}r + \frac{G}{1 + CGH}d
\end{aligned}
$$

The output $y(s)$ is affected by both input $r(s)$ and disturbance $d(s)$. If we wish to analyze only the effect of input $r(s)$ on the output $y(s)$, the closed loop transfer function of the system is:

$$G_{yr}(s) = \frac{y(s)}{r(s)} = \frac{CG}{1 + CGH}$$

When the effect of disturbance needs to be analyzed, the closed loop disturbance transfer function becomes,

$$G_{yd}(s) = \frac{y(s)}{d(s)} = \frac{G}{1 + CGH}$$

The characteristic equation of the closed loop system is $1 + CGH$, which determines the controlled system's stability.

2.2 Review of Laplace Transforms

The Laplace transforms are used to solve linear, ordinary differential equations which represent the mathematical model of continuos, dynamic systems. The Laplace transform of a time dependent function $f(t)$ is evaluated as follows:

$$F(s) = \mathcal{L}[f(t)] = \int_0^{+\infty} f(t)e^{-st}dt \tag{2.1}$$

where s is the Laplace operator which is a complex variable and may have real and imaginary parts, i.e. $s = \sigma \pm j\omega$. The inverse Laplace transform of a function is evaluated as:

$$f(t) = \mathcal{L}^{-1}F(s) = \frac{1}{2\pi j}\int_{-\infty}^{+\infty} F(s)e^{st}ds \tag{2.2}$$

where σ is a real number

Example 1 *Laplace transform of an impulse function*

Impulse function is not a typical mathematical function but an abstract one used as a mathematical tool.

$$\begin{aligned} \delta(t) &= \begin{cases} +\infty & t = 0 \\ 0 & t \neq 0 \end{cases}, \int_{-\infty}^{+\infty} \delta(t)dt = 1 \\ F(s) &= \int_{0}^{+\infty} \delta(t)e^{-st}dt \end{aligned} \tag{2.3}$$

Since the impulse is zero everywhere but at $t = 0$, the integral boundaries can be changed as

$$F(s) = \int_{0^-}^{0^+} \delta(t)e^{-st}dt \tag{2.4}$$

As e^{-st} is continuous between $0^- < t < 0^+$, we can substitute $e^{-st}|_{t=0} = 1$, which leads to

$$F(s) = \int_{0^-}^{0^+} \delta(t)e^{-st}dt = \int_{0^-}^{0^+} \delta(t)dt = 1 \tag{2.5}$$

Alternatively, the impulse function can be defined as :

$$\begin{aligned} \delta(t) = \lim_{\epsilon\to 0} r_\epsilon(t) \to r_\epsilon(t) &= \begin{cases} 1/\epsilon & 0 \leq t \leq \epsilon \\ 0 & \text{otherwise} \end{cases} \\ \text{or } r_\epsilon(t) &= \frac{1}{\epsilon}\left[1(t) - 1(t-\epsilon)\right] \end{aligned} \tag{2.6}$$

or $r_\epsilon(t)$ can be expressed as a unit pulse function as

$$r_\epsilon(t) = \frac{1}{\epsilon}\left[1(t) - 1(t-\epsilon)\right]$$

where ϵ is an infinitely small time interval and $1(t)$ is a unit step function.

$$F(s) = \mathcal{L}(\delta(t)) = \int_{0}^{+\infty} \left[\lim_{\epsilon\to 0} r_\epsilon(t)\right] e^{-st}dt \tag{2.7}$$

$$= \lim_{\epsilon\to 0}\left\{\int_{0}^{+\infty} \frac{1}{\epsilon}\left[1(t) - 1(t-\epsilon)\right] e^{-st}dt\right\} \tag{2.8}$$

$$= \lim_{\epsilon\to 0}\left\{\frac{1}{\epsilon}\left[\frac{1}{s} - \frac{e^{-\epsilon s}}{s}\right]\right\} = \lim_{\epsilon\to 0}\left\{\frac{1 - e^{-\epsilon s}}{\epsilon s}\right\} \tag{2.9}$$

$$= \lim_{\epsilon\to 0}\left\{\frac{-(-se^{-\epsilon s}}{s}\right\} = e^0 = 1 \tag{2.10}$$

Example 2 *Laplace transform of a step function*

$$f(t) = \begin{cases} U & t \geq 0 \\ 0 & t < 0 \end{cases}$$

$$F(s) = \int_0^{+\infty} U e^{-st} dt = -U \frac{1}{s} e^{-st} \Big|_0^{+\infty} = -\frac{U}{s}\frac{1}{e^{\infty}} + \frac{U}{s} e^0 = 0 + \frac{U}{s} \tag{2.11}$$

Example 3 *Laplace transform of a ramp function*

$$f(t) = \begin{cases} ft & t \geq 0 \\ 0 & t < 0 \end{cases}$$

$$F(s) = \int_0^{+\infty} fte^{-st} dt = \frac{f}{s^2} \tag{2.12}$$

Note that the integrating by parts using the following rule is useful:

$$\begin{aligned}
\frac{d}{dt}(uv) &= udv + vdu \rightarrow \int udv = uv - \int vdu \\
F(s) &= \int_0^{+\infty} \underbrace{t}_{u} \underbrace{e^{-st} dt}_{dv} = t\frac{-1}{s} e^{-st} \Big|_0^{+\infty} - \int_0^{+\infty} \frac{-1}{s} e^{-st} dt \\
&= \left(\frac{-te^{-st}}{s}\right)\Big|_0^{+\infty} + \int_0^{+\infty} \frac{e^{-st}}{s} dt \\
&= \lim_{t\to+\infty}\left(\frac{-te^{-st}}{s}\right) - \lim_{t\to+0}\left(\frac{-te^{-st}}{s}\right) + \left(-\frac{1}{s^2}e^{-st}\right)\Big|_0^{+\infty} \\
&= -\frac{1}{s}\lim_{t\to+\infty}\left(\frac{t}{e^{st}}\right) - \frac{1}{s^2}e^{-s\infty} + \frac{1}{s^2}e^0 \\
&= -\frac{1}{s}\lim_{t\to+\infty}\left(\frac{1}{se^{st}}\right) - 0 + \frac{1}{s^2} = 0 - 0 + \frac{1}{s^2} \\
&= \frac{1}{s^2}
\end{aligned} \tag{2.13}$$

Example 4 *Sinusoidal signal*

$$\mathcal{L}\left[\sin(\omega t)\right] = \frac{\omega}{s^2 + \omega^2} \tag{2.14}$$

Example 5 *Functions with a time delay. Let a function $f(t)$ has a time delay of τ. The variable time can be replaced by $t \rightarrow t - \tau$ in the function, i.e. $f(t-\tau) = f(t^{'})$. The Laplace transform can be expressed as:*

$$F_1(s) = \int_0^{+\infty} f(t-\tau)e^{-st} dt \tag{2.15}$$

By changing the variables as $t^{'} = t - \tau$, $t = t^{'} + \tau$, $dt^{'} = dt$, $(t = 0 \rightarrow t^{'} = -\tau)$ and $(t = +\infty \rightarrow t^{'} = +\infty)$,

$$\begin{aligned}
F_1(s) &= \int_{-\tau}^{+\infty} f(t^{'})e^{-s(t^{'}+\tau)} dt^{'} = e^{-s\tau}\int_0^{+\infty} f(t^{'})e^{-st^{'}} dt^{'} \\
F_1(s) &= \mathcal{L}\left[f(t-\tau)\right] = \int_0^{+\infty} f(t-\tau)e^{-st} dt = e^{-s\tau}F(s)
\end{aligned} \tag{2.16}$$

Hence, the time delay enters as $e^{-s\tau}$ to Laplace transform of systems with τ [sec] delay.

Example 6 *Laplace transform of derivatives ($\int udv = uv - \int vdu$)*

$$\begin{aligned}\mathcal{L}\left(\frac{df}{dt}\right) &= \int_0^{+\infty} \frac{df(t)}{dt} e^{-st} dt = e^{-st} f(t)\Big|_0^{+\infty} + s\int_0^{+\infty} f(t)e^{-st}dt \\ &= 0 - f(0) + sF(s)\end{aligned}$$

$$\begin{aligned}\mathcal{L}\left(\frac{df}{dt}\right) &= sF(s) - f(0) \\ \mathcal{L}\left(\frac{d^2f}{dt^2}\right) &= s^2F(s) - sf(0) - \frac{df(0)}{dt} \\ \mathcal{L}\left(\frac{d^mf}{dt^m}\right) &= s^mF(s) - s^{m-1}f(0) - s^{m-2}\frac{df(0)}{dt} - \cdots - \frac{d^{m-1}f(0)}{dt^{m-1}}\end{aligned}$$

Example 7 *Laplace transform of integrals*

$$\begin{aligned}\mathcal{L}\left(\int_0^t f(\tau)d\tau\right) &= \frac{F(s)}{s} \\ \mathcal{L}\left(\int_0^{t_1}\left(\int_0^{t_2} f(\tau)d\tau\right)d\tau\right) &= \frac{F(s)}{s^2}\end{aligned}$$

Theorem 8 ***Initial value theorem***

$$\lim_{t\to 0} f(t) = \lim_{s\to +\infty} sF(s)$$

Theorem 9 ***Final value theorem*** *If $F(s)$ does not have poles on or to the right of the imaginary axis on s plane:*

$$\lim_{t\to +\infty} f(t) = \lim_{s\to +0} sF(s)$$

2.2.1 Inverse Laplace Transform by Partial Fraction Expansion

Consider a transfer function $G(s)$ with a numerator $B(s)$ and denominator $A(s)$,

$$G(s) = \frac{B(s)}{A(s)} \tag{2.17}$$

The order or the degree of $A(s)$ is greater than $B(s)$, $\deg(A) > \deg(B)$. The denominator is expressed as a polynomial function of s,

$$A(s) = s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_1 s + a_0 \tag{2.18}$$

where the coefficients a_i are real. Polynomial $A(s)$ has n number of roots, hence the transfer function can be expressed as

$$G(s) = \frac{B(s)}{(s+p_1)(s+p_2)\cdots(s+p_n)} \tag{2.19}$$

The roots of the denominator $A(s)$ are called *poles* of $G(s)$, whereas the roots of numerator $B(s)$ are called *zeros* of $G(s)$ in control engineering. The system may have simple real, complex conjugate or repeated poles.

Example 10 *Simple, real poles (i.e. $p_1 \neq p_2 \neq ... \neq p_n$ are real and distinct numbers which are not duplicate of each other.*

$$G(s) = \frac{B(s)}{(s+p_1)(s+p_2)\cdot\cdot(s+p_i)\cdot\cdot(s+p_n)} = \frac{C_1}{s+p_1} + \frac{C_2}{s+p_2} + \cdot\cdot + \frac{C_i}{s+p_i} + \cdot\cdot + \frac{C_n}{s+p_n} \tag{2.20}$$

The coefficients C_i can be evaluated from the partial fraction expansion rule as follows:

$$C_i = \lim_{s\to -p_i}(s+p_i)\frac{B(s_i)}{(s+p_1)(s+p_2)\cdot\cdot(s+p_i)\cdot\cdot(s+p_n)} \tag{2.21}$$

Example 11 *Distinct complex poles*

$$\begin{aligned}
F(s) &= \frac{1}{s(s^2+s+1)} = \frac{1}{s(s-p_1)(s-p_2)} = \frac{C_1}{s} + \frac{C_2}{s-p_1} + \frac{C_2^*}{s-p_2} \\
s^2+s+1 = 0 \to p_{1,2} &= \frac{-1\pm\sqrt{1-4}}{2} = -\frac{1}{2} \pm j\frac{\sqrt{3}}{2} \\
C_1 &= \lim_{s\to 0} s\frac{1}{s(s^2+s+1)} = 1 \\
C_2 &= \lim_{s\to p_1}(s-p_1)\frac{1}{s(s-p_1)(s-p_2)} = \lim_{s\to -\frac{1}{2}+j\frac{\sqrt{3}}{2}} \frac{1}{s(s-p_2)} \\
&= \frac{1}{(-\frac{1}{2}+j\frac{\sqrt{3}}{2})(-\frac{1}{2}+j\frac{\sqrt{3}}{2}+\frac{1}{2}+j\frac{\sqrt{3}}{2})} = -\frac{1}{2} + j\frac{1}{2\sqrt{3}} \\
C_2^* &= \lim_{s\to p2}(s-p_2)\frac{1}{s(s-p_1)(s-p_2)} = -\frac{1}{2} - j\frac{1}{2\sqrt{3}} \\
f(t) &= \mathcal{L}^{-1}[F(s)] = \mathcal{L}^{-1}\left[\frac{C_1}{s} + \frac{C_2}{s-p_1} + \frac{C_2^*}{s-p_2}\right] = C_1 + C_2 e^{-p_1 t} + C_2^* e^{-p_2 t}
\end{aligned}$$

:

Example 12 *Repeated poles*

$$\begin{aligned}
F(s) &= \frac{Q(s)}{(s+p_1)(s+p_2)\cdot\cdot(s+p_{n-r})(s+p_i)^r} \\
&= \underbrace{\frac{C_1}{s+p_1} + \frac{C_2}{s+p_2} + \cdot\cdot + \frac{C_{n-r}}{s+p_{n-r}}}_{n-r \text{ distinct, simple poles}} \\
&+ \underbrace{\frac{A_1}{s+p_i} + \frac{A_2}{(s+p_i)^2} + \cdot\cdot + \frac{A_r}{(s+p_i)^r}}_{r \text{ terms of repeated poles}}
\end{aligned}$$

$n-r$ distinct, simple poles can be evaluated as shown before. r terms of repeated poles are evaluated as follows:

$$
\begin{aligned}
A_r &= \lim_{s\to -p_i} (s+p_i)^r F(s) \\
A_{r-1} &= \lim_{s\to -p_i} \frac{d}{ds}\left[(s+p_i)^r F(s)\right] \\
A_{r-2} &= \lim_{s\to -p_i} \frac{1}{2!}\frac{d^2}{ds^2}\left[(s+p_i)^r F(s)\right] \\
&. \\
A_1 &= \lim_{s\to -p_i} \frac{1}{(r-1)!}\frac{d^{r-1}}{ds^{r-1}}\left[(s+p_i)^r F(s)\right]
\end{aligned}
$$

Example 13 *Lets assume the following differential equation with initial conditions :* $\frac{dx(0)}{dt} = 0, x(0) = x_0$

$$
\begin{aligned}
\frac{d^2x}{dt^2} + a\frac{dx}{dt} + bx &= f(t) \\
s^2X(s) - sX(0) - \frac{dx(0)}{dt} + asX(s) - aX(0) + bX(s) &= F(s) \\
s^2X(s) - sx_0 - 0 + asX(s) - ax_0 + bX(s) &= F(s) \\
(s^2 + as + b)X(s) &= F(s) + (s+a)x_0 \\
X(s) &= \frac{1}{s^2+as+b}F(s) + \underbrace{\frac{s+a}{s^2+as+b}x_0}_{\text{initial condition effect}}
\end{aligned}
$$

Here the first part corresponds to general solution, and the second part corresponds to particular solution of the differential equation.

2.2.2 Solution of Ordinary Linear Differential Equations with Laplace Transforms

The transfer function of a system is represented in Laplace domain as $G(s)$, and the input (or the excitation) to the system is transformed from time domain to Laplace domain as $\mathcal{L}[u(t)] = U(s)$. The output $Y(s)$ is expressed by the product of transfer function and input in Laplace domain.

$$Y(s) = G(s)U(s)$$

The output $y(t)$ is evaluated by taking inverse Laplace transform of $Y(s)$,

$$y(t) = \mathcal{L}^{-1}Y(s) = \mathcal{L}^{-1}\left[G(s)U(s)\right]$$

which corresponds to the combined particular and homogenous solution $y(t)$ of the differential equation with excitation $u(t)$.

Example 14 *First order system*

$$
\begin{aligned}
\frac{dy}{dt} + ay(t) &= u(t) \rightarrow u(t) = U \\
sy(s) + ay(s) &= \frac{U}{s} \\
(s+a)y(s) &= \frac{U}{s} \\
y(s) &= \frac{1}{s+a}\frac{U}{s} \\
y(s) &= U\left(\frac{C_1}{s} + \frac{C_2}{s+a}\right) \\
C_1 &= \lim_{s=0} s\frac{1}{s+a}\frac{1}{s} = \frac{1}{a} \\
C_2 &= \lim_{s=-a} (s+a)\frac{1}{s+a}\frac{1}{s} = -\frac{1}{a} \\
y(s) &= \frac{U}{a}(\frac{1}{s} - \frac{1}{s+a}) \\
y(t) &= \mathcal{L}^{-1}y(s) = \frac{U}{a}\left(1 - e^{-at}\right)
\end{aligned}
$$

Example 15 *Solution of differential equations with initial conditions.*

Consider the following differential equation with an initial condition y_0:

$$\frac{dy(t)}{dt} + 3y(t) = 3r(t) \quad \text{with } y(0) = y_0$$

Transfer function of the system is:

$$sy(s) + 3y(s) = 3r(s) \rightarrow G(s) = \frac{y(s)}{r(s)} = \frac{3}{s+3}$$

That means the transfer function of the system represents general dynamics (i.e. differential equation), which does not consider the initial conditions. However, it can be included in the solution as follows.

Let the input be a step function with an amplitude $r(t) = R_0$. If we start solving the differential equation using Laplace transforms:

$$
\begin{aligned}
\frac{dy(t)}{dt} + 3y(t) &= 3r(t) \quad \text{with } y(0) = y_0 \quad \text{and } r(t) = R_0 \\
sy(s) - y(0) + 3y(s) &= 3R(s) \rightarrow R(s) = \frac{R_0}{s} \\
y(s) &= \frac{R_0}{s(s+3)} + \frac{y_0}{s+3} = \underbrace{\frac{R_0}{3}(\frac{1}{s} - \frac{1}{s+3})}_{\text{step response}} \quad \underbrace{\frac{1}{s+3}y_0}_{\text{initial condition part}} \\
y(t) &= \frac{R_0}{3}\left(1 - e^{-3t}\right) + y_0 e^{-3t}
\end{aligned}
$$

The effect of initial condition is represented by $y_0 e^{-3t}$ which will diminish over time ($t \rightarrow \infty, y_0 e^{-3t} \rightarrow 0$). In a way, the initial condition brings a second part, which represents particular solution of the differential equation. The characteristic equation (the denominator term) is the same, but the numerator depends on the order of the system and initial conditions.

Although this is a general tutorial in Laplace transforms, we hardly use initial conditions in control systems

Example 16 *Complex Roots.*

$$
\begin{aligned}
\frac{dy^2(t)}{dt^2} + 2\zeta\omega_n \frac{dy(t)}{dt} + \omega_n^2 y(t) &= \omega_n^2 f(t) \rightarrow f(t) = \text{impulse function} \\
\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right) y(s) &= \omega_n^2 F(s) \rightarrow F(s) = 1 \\
y(s) &= G(s)F(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \rightarrow \zeta < 1 \\
\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} &= \frac{C_1}{s - p_1} + \frac{C_2}{s - p_1^*} \\
s^2 + 2\zeta\omega_n s + \omega_n^2 &= 0 \rightarrow p_1 = -\sigma + j\omega_d,\ p_1^* = -\sigma - j\omega_d \\
\sigma &= \zeta\omega_n,\ \ \omega_d = \omega_n\sqrt{1-\zeta^2} \\
C_1 &= \lim_{s \to p_1} (s - p_1) \frac{\omega_n^2}{(s-p_1)(s-p_1^*)} = \frac{\omega_n^2}{(p_1 - p_1^*)} = \frac{\omega_n^2}{j2\omega_d} = \frac{\omega_n}{j2\sqrt{1-\zeta^2}} \\
C_2 &= \lim_{s \to p_1^*} (s - p_1^*) \frac{\omega_n^2}{(s-p_1)(s-p_1^*)} = \frac{\omega_n^2}{p_1^* - p_1} = -\frac{\omega_n^2}{j2\omega_d} = -\frac{\omega_n}{j2\sqrt{1-\zeta^2}} \\
y(s) &= \frac{\omega_n}{j2\sqrt{1-\zeta^2}} \left[\frac{1}{s-p_1} - \frac{1}{s-p_1^*}\right] \\
y(t) &= \frac{\omega_n}{j2\sqrt{1-\zeta^2}} \mathcal{L}^{-1}\left[\frac{1}{s-p_1} - \frac{1}{s-p_1^*}\right] = \frac{\omega_n}{j2\sqrt{1-\zeta^2}} \left[e^{p_1 t} - e^{p_1^* t}\right] \\
&= \frac{\omega_n e^{-\zeta\omega_n t}}{j2\sqrt{1-\zeta^2}} \left[e^{j\omega_d t} - e^{-j\omega_d t}\right] \\
&= \frac{\omega_n e^{-\zeta\omega_n t}}{j2\sqrt{1-\zeta^2}} \left[\cos\omega_d t + j\sin\omega_d t - \cos\omega_d t + j\sin\omega_d t\right] \\
&= \frac{\omega_n}{j2\sqrt{1-\zeta^2}} \left[j2\sin\omega_d t\right] \\
y(t) &= \frac{\omega_n}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t} \sin\omega_d t
\end{aligned}
$$

Example 17 *Consider a single degree of freedom system with an external force $F(t)$ applied on mass (m), damping (c) and spring (k)*

$$
m\frac{d^2y}{dt^2} + c\frac{dy}{dt} + ky = F(t) \tag{2.22}
$$

which can be expressed by normalizing both sides by mass:

$$
\frac{d^2y}{dt^2} + 2\zeta\omega_{\mathrm{n}}\frac{dy}{dt} + \omega_{\mathrm{n}}^2 y = \frac{\omega_{\mathrm{n}}^2}{k}F(t) \tag{2.23}
$$

where $\omega_{\mathrm{n}} = \sqrt{k/m}$, $c/m = 2\zeta\omega_{\mathrm{n}}$ and $1/m = \omega_{\mathrm{n}}^2/k$.The Laplace transform of the equation of motion with initial displacement $y\,(0)$, vibration velocity $y'(0)$ and external force $F(t)$ is expressed as

$$
\begin{aligned}
\mathcal{L}\left(\frac{d^2y}{dt^2} + 2\zeta\omega_{\mathrm{n}}\frac{dy}{dt} + \omega_{\mathrm{n}}^2 y\right) &= \mathcal{L}\left(\frac{\omega_{\mathrm{n}}^2}{k}F(t)\right) \\
s^2y(s) - sy\,(0) - y'(0) + 2\zeta\omega_{\mathrm{n}}sy(s) - 2\zeta\omega_{\mathrm{n}}y(0) + \omega_{\mathrm{n}}^2 y(s) &= \frac{\omega_{\mathrm{n}}^2}{k}F(s)
\end{aligned} \tag{2.24}
$$

The system's general response, the vibrations of the structure with a single degree of freedom, can be expressed as,

$$y(s) = \frac{\omega_{\mathrm{n}}^2}{k}\frac{1}{s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2}F(s) + \frac{(s+2\zeta\omega_{\mathrm{n}})y(0)-y'(0)}{s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2} \tag{2.25}$$

The transfer function of the system is represented by neglecting the effect of initial conditions which will eventually disappear as transient vibrations.

$$\Phi(s) = \frac{y(s)}{F(s)} = \frac{\omega_{\mathrm{n}}^2}{k}\frac{1}{s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2}$$

The characteristic equation of the system has complex conjugate roots (p, p^*) because the mechanical systems are underdamped $(\zeta \ll 1)$ and vibrate :

$$\begin{aligned} s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2 &= (s-p)(s-p^*) = 0 \\ p &= -\zeta\omega_{\mathrm{n}} + j\omega_{\mathrm{d}}, \quad p^* = -\zeta\omega_{\mathrm{n}} - j\omega_{\mathrm{d}} \end{aligned}$$

where $\omega_{\mathrm{d}} = \omega_{\mathrm{n}}\sqrt{1-\zeta^2}$ is the damped natural frequency of the structure.

- Response of the system to step load $(F(t) = F_0 \rightarrow t \geq 0, \mathcal{L}(F_0) = F_0/s)$ with zero initial conditions $(y(0) = y'(0) = 0)$.

$$\begin{aligned} y(s) &= \frac{1}{k}\frac{\omega_{\mathrm{n}}^2}{s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2}\frac{F_0}{s} = \frac{F_0}{k}\left(\frac{A}{s} + \frac{Bs+C}{s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2}\right) \\ A &= \lim_{s\to 0} s\frac{\omega_{\mathrm{n}}^2}{s\left[s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2\right]} = 1 \\ \omega_{\mathrm{n}}^2 &= s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2+Bs^2+Cs \\ -Bs^2-Cs+\omega_{\mathrm{n}}^2 &= s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2 \qquad (2.26) \\ B &= -1, C = -2\zeta\omega_{\mathrm{n}} \qquad (2.27) \\ y(s) &= \frac{1}{k}\frac{\omega_{\mathrm{n}}^2}{s^2+2\zeta\omega_{\mathrm{n}}s+\omega_{\mathrm{n}}^2}\frac{F_0}{s} = \frac{F_0}{k}\left(\frac{1}{s} - \frac{s+2\zeta\omega_n}{(s+\zeta\omega_{\mathrm{n}})^2+\omega_{\mathrm{d}}^2}\right) \qquad (2.28) \\ y(s) &= \frac{F_0}{k}\left(\frac{1}{s} - \frac{\zeta\omega_n}{(s+\zeta\omega_{\mathrm{n}})^2+\omega_{\mathrm{d}}^2} - \frac{s+\zeta\omega_n}{(s+\zeta\omega_{\mathrm{n}})^2+\omega_{\mathrm{d}}^2}\right) \end{aligned}$$

Noting that $\mathcal{L}^{-1}(\frac{b}{(s+a)^2+b^2}) = e^{-at}\sin bt$ and $\mathcal{L}^{-1}(\frac{s+a}{(s+a)^2+b^2}) = e^{-at}\cos bt$:

$$\begin{aligned} y(s) &= \frac{F_0}{k}\left(\frac{1}{s} - \frac{\zeta\omega_n}{\omega_d}\frac{\omega_d}{(s+\zeta\omega_{\mathrm{n}})^2+\omega_{\mathrm{d}}^2} - \frac{s+\zeta\omega_n}{(s+\zeta\omega_{\mathrm{n}})^2+\omega_{\mathrm{d}}^2}\right) \\ y(t) = \mathcal{L}^{-1}x(s) &= \frac{F_0}{k}\left(1 - \frac{\zeta\omega_n}{\omega_n\sqrt{1-\zeta^2}}e^{-\zeta\omega_n t}\sin\omega_{\mathrm{d}}t - e^{-\zeta\omega_n t}\cos\omega_{\mathrm{d}}t\right) \\ &= \frac{F_0}{k}\left[1 - e^{-\zeta\omega_n t}\frac{1}{\sqrt{1-\zeta^2}}\left(\zeta\sin\omega_{\mathrm{d}}t + \sqrt{1-\zeta^2}\cos\omega_{\mathrm{d}}t\right)\right] \end{aligned}$$

Considering that $b\cos b + a\sin b = c\sin(b+\phi) = c[\sin\phi\cos b + \cos\phi\sin b] \rightarrow c\sin\phi = b, c\cos\phi = a$, which leads to $c = \sqrt{b^2+a^2}$ and $\tan\phi = b/a$,

$$\sqrt{1-\zeta^2}\cos\omega_{\mathrm{d}}t + \zeta\sin\omega_{\mathrm{d}}t = \sin(\omega_{\mathrm{d}}t+\phi) \quad \rightarrow \tan\phi = \sqrt{1-\zeta^2}/\zeta$$

Hence, the response becomes:

$$y(t) = \frac{F_0}{k}\left[1 - e^{-\zeta\omega_n t}\frac{1}{\sqrt{1-\zeta^2}}\sin\left(\omega_{\mathrm{d}} t + \phi\right)\right] \rightarrow \phi = \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta} \tag{2.29}$$

Example 18 *Consider a second order dynamic system (i.e. mass, spring, damper) with a ramp input.*

$$\frac{dy^2(t)}{dt^2} + 2\zeta\omega_n\frac{dy(t)}{dt} + \omega_n^2 y(t) = \frac{\omega_n^2}{k}f(t) \rightarrow f(t) = ft$$

Laplace transform of both sides with zero initial conditions lead to:

$$\begin{aligned}
\left[s^2 + 2\zeta\omega_n s + \omega_n^2\right]Y(s) &= \frac{\omega_n^2}{k}F(s) \\
G(s) &= \frac{Y(s)}{F(s)} = \frac{\omega_n^2/k}{s^2 + 2\zeta\omega_n s + \omega_n^2},\ F(s) = \frac{f}{s^2} \\
Y(s) &= G(s)F(s) = \frac{\omega_n^2/k}{s^2 + 2\zeta\omega_n s + \omega_n^2}\frac{f}{s^2} \\
Y(s) &= \frac{f}{k}\left(\frac{A}{s} + \frac{B}{s^2} + \frac{Cs + D}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right) \\
&= \frac{f}{k}\left(\frac{As\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right) + B\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right) + Cs^3 + Ds^2}{s^2\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)}\right) \\
&= \frac{f}{k}\left(\frac{(A+C)s^3 + (2\zeta\omega_n A + B + D)\,s^2 + (A\omega_n^2 + B2\zeta\omega_n)s + B\omega_n^2}{s^2\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)}\right) \\
&= \frac{f}{k}\frac{\omega_n^2}{s^2\left(s^2 + 2\zeta\omega_n s + \omega_n^2\right)} \\
B &= 1\ ;\ A\omega_n^2 + B2\zeta\omega_n = 0 \rightarrow A = -\frac{2\zeta}{\omega_n}\ ; \\
2\zeta\omega_n A + B + D = 0 &\rightarrow D = -2\zeta\omega_n A - B = 4\zeta^2 - 1\ ;\ A + C = 0 \rightarrow C = -A = \frac{2\zeta}{\omega_n} \\
Y(s) &= \frac{f}{k}\left(\frac{1}{s^2} - \frac{2\zeta}{\omega_n}\frac{1}{s} + \frac{\frac{2\zeta}{\omega_n}s + \left(4\zeta^2 - 1\right)}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right) \\
y(t) &= \mathcal{L}^{-1}Y(s) \\
y(t) &= \frac{A}{k}\left[t - \frac{2\zeta}{\omega_n} + \frac{e^{-\zeta\omega_n t}}{\omega_n\sqrt{1-\zeta^2}}\sin\left(\omega_n\sqrt{1-\zeta^2}t + \theta\right)\right], \theta = \cos^{-1}(2\zeta^2 - 1)
\end{aligned}$$

The steady state value of the output can be evaluated by taking the limit as :

$$y(t \rightarrow +\infty) = \lim_{t\rightarrow\infty} y(t) = \frac{A}{k}\left[t_\infty - \frac{2\zeta}{\omega_n}\right]$$

The steady state error between the input $(f(t)/k)$ and output $(y(t))$ can be evaluated as :

$$\begin{aligned}
e(t \rightarrow \infty) = e_{ss} &= \lim_{t\rightarrow\infty}\left(\frac{At}{k} - y(t \rightarrow +\infty)\right) \\
&= \lim_{t\rightarrow\infty}\left(\frac{At}{k} - \frac{A}{k}\left[t - \frac{2\zeta}{\omega_n} + \frac{e^{-\zeta\omega_n t}}{\omega_n\sqrt{1-\zeta^2}}\sin\left(\omega_n\sqrt{1-\zeta^2}t + \theta\right)\right]\right) \\
e_{ss} &= \frac{A}{k}\frac{2\zeta}{\omega_n}
\end{aligned}$$

The steady state error can be evaluated from the final value theorem as well as follows:

$$
\begin{aligned}
e(s) &= y_r(s) - y(s) = \frac{1}{k}\frac{A}{s^2} - G(s)\frac{1}{k}\frac{A}{s^2} = \frac{1}{k}\frac{A}{s^2}\left[1 - G(s)\right] \\
e(s) &= \frac{1}{k}\frac{A}{s^2}\left[1 - \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right] = \frac{1}{k}\frac{A}{s^2}\left[\frac{s^2 + 2\zeta\omega_n s + \omega_n^2 - \omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right] \\
e(s) &= \frac{1}{k}\frac{A}{s^2}\left[\frac{s^2 + 2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right] = \frac{1}{k}\frac{A}{s}\left[\frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right] \\
e_{ss} &= \lim_{s\to 0} s e(s) = \lim_{s\to 0} s\frac{1}{k}\frac{A}{s}\left[\frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right] = \frac{A}{k}\frac{2\zeta}{\omega_n}
\end{aligned}
$$

- Free vibrations with initial displacement $x(0) = x_0$. (External force $F(t) = 0$, initial velocity $x'(0) = 0$)

$$
\begin{aligned}
x(s) &= \frac{s + 2\zeta\omega_{\mathrm{n}}}{(s + \zeta\omega_{\mathrm{n}})^2 + \omega_{\mathrm{d}}^2} x_0 = \left[\frac{\zeta\omega_n}{(s + \zeta\omega_{\mathrm{n}})^2 + \omega_{\mathrm{d}}^2} + \frac{s + \zeta\omega_n}{(s + \zeta\omega_{\mathrm{n}})^2 + \omega_{\mathrm{d}}^2}\right] x_0 \qquad (2.30) \\
x(t) &= x_0 e^{-\zeta\omega_n t}\frac{1}{\sqrt{1-\zeta^2}}\sin\left(\omega_{\mathrm{d}} t + \phi\right) \to \phi = \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}
\end{aligned}
$$

2.3 Review of Complex Numbers

Consider a complex number $y = a + jb$ where a is the real and b is the complex part. A complex number can be represented either by its real and imaginary parts, or by its magnitude and phase as follows.

$$
\begin{aligned}
y &= a + jb \\
\text{Magnitude} \quad |y| &= \sqrt{a^2 + b^2} \text{ and Phase } \phi = \tan^{-1}\frac{b}{a}
\end{aligned}
$$

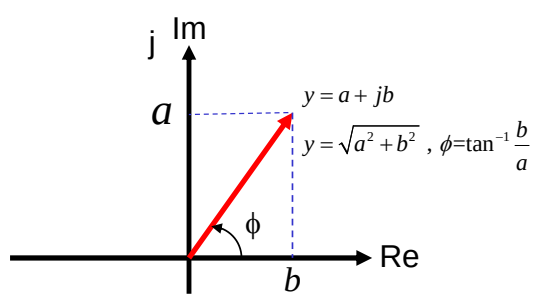


Figure 2.4: Complex plane algebra

Complex function at the denominator

$$
\begin{aligned}
y &= \frac{1}{a + jb} = \frac{1}{a + j}\frac{a - jb}{a - jb} = \frac{a - jb}{a^2 + b^2} = \frac{a}{a^2 + b^2} - j\frac{b}{a^2 + b^2} \\
y &= \frac{1}{a + jb} \to |y| = \frac{1}{\sqrt{a^2 + b^2}} \text{and } \phi = \tan^{-1}\frac{-b}{a} = -\tan^{-1}\frac{b}{a}
\end{aligned}
$$

2.4 Frequency Response Function (FRF) of Continuous Systems

FRF of a system exhibits its steady state response under sinusoidal excitation at a range of excitation frequency. Consider a dynamic system with a transfer function represented in continuos (Laplace) domain as:

$$\begin{aligned} G(s) = \frac{x(s)}{u(s)} &= \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \\ &= K\frac{(s-z_m)(s-z_{m-1})\cdots(s-z_1)}{(s-p_n)(s-p_{n-1})\cdots(s-p_1)} \end{aligned} \tag{2.31}$$

where the order of denominator is greater than the order of numerator ($n > m$) for a *proper system*. The system has m zeros and n poles. The system response (output) is $x(s)$ for a given excitation (input) $u(s)$. FRF assumes that the input to the system is harmonic (i.e. sinusoidal), so as the steady state output where the transient response is neglected.

$$\begin{aligned} u(t) &= U \sin \omega t \\ x(t) &= X(\omega)\sin\left[\omega t + \phi(\omega)\right] = X(\omega)\sin\omega\left[t + t_d(\omega)\right], \quad t \to +\infty \end{aligned} \tag{2.32}$$

However, the amplitude ($X(\omega)$) and phase delay ($\phi(\omega)$) are dependent on the frequency of excitation (ω). Note that the phase difference $\phi(\omega)$ represents the time shift between the output and input, $\phi(\omega) = \omega t_d$. FRF represents the changes in X, ϕ as the excitation frequency (ω) is varied between zero and finite value until there is no change in the trend of the system's steady state response. The steady state response of a system for a harmonic input can be evaluated by simply replacing $s \to j\omega$, where ω is the frequency of input in [rad/s].

$$\begin{aligned} G(s \to j\omega) = G(j\omega) &= \frac{x(\omega)}{u(\omega)} \\ G(j\omega) &= K\frac{(s-z_m)(s-z_{m-1})\cdots(s-z_1)}{(s-p_n)(s-p_{n-1})\cdots(s-p_1)}\Bigg|_{s\to j\omega} \\ &= K\frac{(j\omega-z_m)(j\omega-z_{m-1})\cdots(j\omega-z_1)}{(j\omega-p_n)(j\omega-p_{n-1})\cdots(j\omega-p_1)} \end{aligned} \tag{2.33}$$

FRF $G(j\omega)$ is a complex function, which can be represented by its magnitude and phase as follows ($(j\omega)^2 = -\omega^2$):

$$\begin{aligned} G(s \to j\omega) = G(j\omega) &= \frac{x(j\omega)}{u(j\omega)} \\ |G(j\omega)| = \frac{x(\omega)}{u(\omega)} &= \frac{K\cdot\left[\sqrt{z_m^2+\omega^2}\cdot\sqrt{z_{m-1}^2+\omega^2}\cdot\cdot\sqrt{z_1^2+\omega^2}\right]}{\left[\sqrt{p_n^2+\omega^2}\cdot\sqrt{p_{n-1}^2+\omega^2}\cdot\cdot\sqrt{p_1^2+\omega^2}\right]} \\ \phi(\omega) &= \left[\tan^{-1}\frac{\omega}{-z_m} + \tan^{-1}\frac{\omega}{-z_{m-1}} + \cdot\cdot + \tan^{-1}\frac{\omega}{-z_1}\right] - \left[\tan^{-1}\frac{\omega}{-p_n} + \tan^{-1}\frac{\omega}{-p_{n-1}} \cdot + \cdot \tan^{-1}\frac{\omega}{-p_1}\right] \end{aligned} \tag{2.34}$$

If the FRF is plotted graphically in Logarithmic scale, it is called Bode Plot.

$$\begin{aligned} |G(j\omega)| = &\left[20\log_{10}K + 20\log_{10}\sqrt{z_m^2+\omega^2} + 20\log_{10}\sqrt{z_{m-1}^2+\omega^2}\cdot\cdot + 20\log_{10}\sqrt{z_1^2+\omega^2}\right] \\ &- \left[20\log_{10}\sqrt{p_n^2+\omega^2} + 20\log_{10}\sqrt{p_{n-1}^2+\omega^2} + \cdot\cdot + 20\log_{10}\sqrt{p_1^2+\omega^2}\right] \quad \text{[dB]} \\ \phi(\omega) = &\left[\tan^{-1}\frac{\omega}{-z_m} + \tan^{-1}\frac{\omega}{-z_{m-1}} + \cdot\cdot + \tan^{-1}\frac{\omega}{-z_1}\right] - \left[\tan^{-1}\frac{\omega}{-p_n} + \tan^{-1}\frac{\omega}{-p_{n-1}} \cdot\cdot + \tan^{-1}\frac{\omega}{-p_1}\right] \end{aligned} \tag{2.35}$$

where dB (decibel) is defined as

$$\log_{10} 10 = 1 \text{ [dB]}$$

The FRF can be expressed by its real and imaginary parts,

$$G(j\omega) = \mathrm{Re}\,[G(j\omega)] + j\,\mathrm{Im}\,[G(j\omega)]$$

where $\mathrm{Re}\,[G(j\omega)]$ and $\mathrm{Im}\,[G(j\omega)]$ parts can be separately plotted as a function of frequency. Alternatively, the FRF can be shown in polar coordinates by plotting $\mathrm{Re}\,[G(j\omega)]$ against $\mathrm{Im}\,[G(j\omega)]$ as the frequency increases from zero to infinity ($\omega \in (0, +\infty)$). The polar plot of FRF is also called as Nyquist plot.

Example 19 *The frequency response of a first order system with gain $K = 1$ and time constant $\tau = 0.1$ is evaluated as follows*

$$G(s) = \frac{K}{\tau s + 1} \rightarrow G(j\omega) = \frac{K}{j\tau\omega + 1}$$
$$|G(j\omega)| = \frac{K}{\sqrt{(\tau\omega)^2 + 1}}, \qquad \phi = -\tan^{-1}(\tau\omega)$$
$$|G(j\omega)| = \frac{1}{\sqrt{(0.1\omega)^2 + 1}}, \qquad \phi = -\tan^{-1}(0.1\omega)$$

If we scan the frequency with decade increments

$\omega[rad/s]$	$\lvert G(j\omega)\rvert$	$20log\,\lvert G(j\omega)\rvert$ $[dB]$	ϕ
0.1	1	0	0
1	~ 1	0	~ 0
10	0.707	$-3.0dB$	~ -45
100	~ 0.1	$-20.0dB$	~ -90
1000	~ 0.01	$-40.0dB$	~ -90

Example 20 *The frequency response function (FRF) of a second order underdamped system*

Assume that the transfer function of the system is:

$$G(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The FRF of the system is obtained by substituting $s = j\omega$

$$G(j\omega) = \frac{\omega_n^2}{\omega_n^2 - \omega^2 + j2\zeta\omega_n\omega} = \omega_n^2 \frac{(\omega_n^2 - \omega^2) - j(2\zeta\omega_n\omega)}{(\omega_n^2 - \omega^2)^2 + (2\zeta\omega_n\omega)^2}$$
$$\text{Magnitude } |G(j\omega)| = \frac{\omega_n^2}{\sqrt{(\omega_n^2 - \omega^2)^2 + (2\zeta\omega_n\omega)^2}} \;;\; \text{Phase } \phi = \tan^{-1}\frac{-2\zeta\omega_n\omega}{\omega_n^2 - \omega^2}$$

Note that the phase must be calculated properly either by considering the location in complex plane (i.e. the imaginary part is negative, and real part is positive if $\omega < \omega_n$ but negative if $\omega > \omega_n$. The phase will be $0 > \phi > -\pi$). However, if the order is higher, the calculators and computer programs can easily make a mistake in calculating the phase.

$$\text{Phase } \phi = \tan^{-1}\frac{-2\zeta\omega_n\omega}{\omega_n^2 - \omega^2} \rightarrow \begin{cases} \text{if } \omega_n > \omega \text{ than } -\frac{\pi}{2} \leqslant \phi \leqslant 0 \\ \text{if } \omega_n < \omega \text{ than } -\frac{\pi}{2} \leqslant \phi \leqslant \pi \end{cases}$$

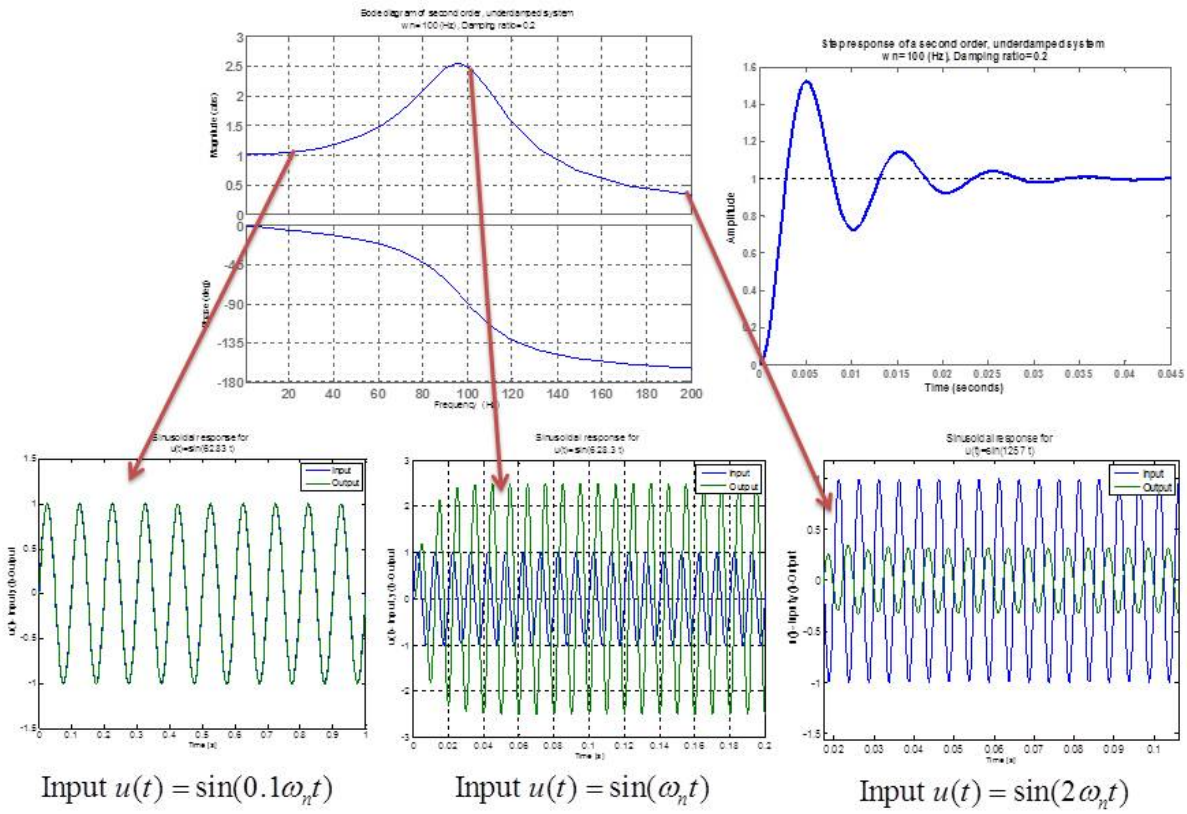


Figure 2.5: Frequency Response Function of a 2nd order underdamped system, and its sunusoidal response at different frequencies.

One can use $\phi = -\frac{\pi}{2} - \tan^{-1}\frac{2\zeta\omega_n\omega}{\omega^2-\omega_n^2}$ when $\omega > \omega_n$ to correct the phase calculation when calculators or computers are used. Alternatively, one needs to use atan2 function in MATLAB.

Step and sinusoidal response of the system at different frequencies are shown in Figure 2.5. Note that the amplitude of the sinusoidal response becomes almost the same at low frequency ($\omega = 0.1\omega_n$), large at the resonant ($\omega = \omega_n$) with $\pi/2$ delay, and quite small at the high frequency ($\omega = 2\omega_n$) with almost $-\pi$ delay.

Chapter 3

Continuous and Discrete Time Control Systems

A general diagram of a computer control system is shown in Figure 3.1. The machine (i.e. drive, motor, amplifier, sensor) is the plant to be controlled by a difference equation executed in discrete time intervals (T) in the computer. The machine's transfer function is modeled in continuous time (Laplace) domain, but the controller's transfer function is modeled in discrete time (z domain) at constant control time intervals of (T). The transfer function of analog devices, the physical system components such as amplifier-servo motor-tachogenerator which make up the analog velocity control system, are modeled in Laplace Domain. The position control law is executed in a computer algorithm which is executed at constant control interval of $T[sec]$ interval or in discrete time domain.

There are two approaches in designing a computer controller: Either the controller must be designed in continuos (Laplace) time domain, and converted into discrete time domain to be implemented in the computer; or the continuous system must be converted into discrete time domain, and the controller must be designed also in discrete time domain.

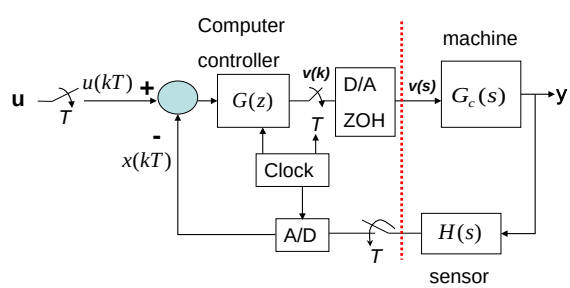


Figure 3.1: Architecture of a computer control application.

3.1 z Transforms

Continuous signals are sampled by computers at discrete time intervals T through analog to digital converters (A/D). Consider a continuous function $x(t)$ which is sampled and held by the device (3.2). Sampling without holding it for a duration of sampling period T can not be realized physically, hence sample and hold must be treated together. However, in order to obtain the mathematical model of the sample and hold device, the sampled signal is treated as $x^*(t)$, which is only a mathematical representation and does not have physical equivalent. The sampled and held data ($\widetilde{x}(t)$) can be mathematically expressed as :

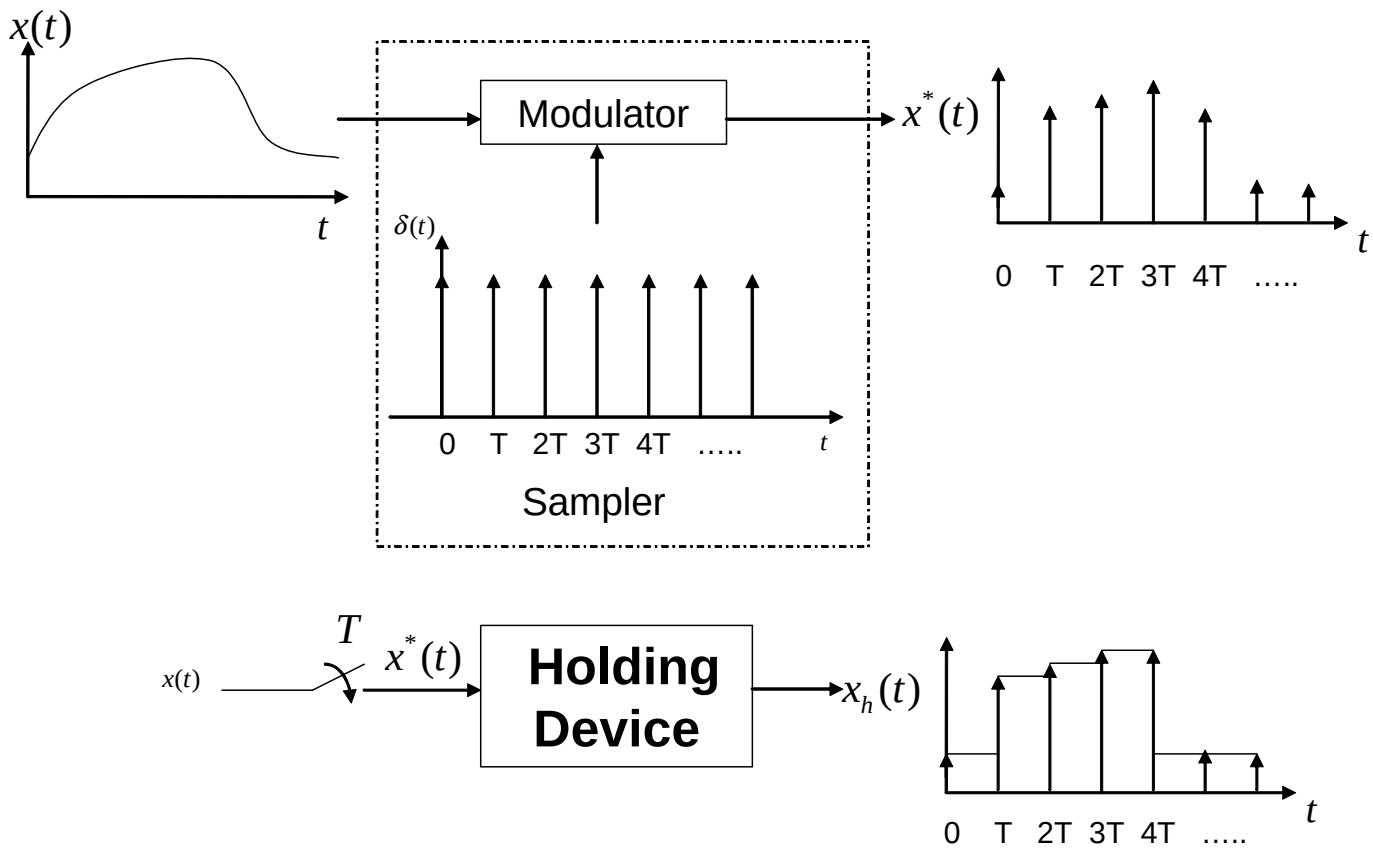


Figure 3.2: Sample and hold mechanism in digital systems.

$$\widetilde{x}(t) = x(0)\left[u(t) - u(t-T)\right] + x(T)\left[u(t-T) - u(t-2T)\right] + \cdots + x(kT)\left[u(t-kT) - u(t-(k+1)T)\right] + \cdots \quad (3.1)$$

where $u(t) - u(t-T)$ is a pulse function with unit amplitude and time width of T, and it represents holding of the sampled data in the device. Taking the Laplace transform of $\widetilde{x}(t)$ leads to,

$$\mathcal{L}\widetilde{x}(t) = x(0)\left[\frac{1}{s} - e^{-sT}\frac{1}{s}\right] + x(T)\left[e^{-sT}\frac{1}{s} - e^{-2sT}\frac{1}{s}\right] + \cdots + x(kT)\left[e^{-skT}\frac{1}{s} - e^{-s(k+1)T}\frac{1}{s}\right] + \cdots \quad (3.2)$$

$$\widetilde{X}(s) = x(0)\frac{1}{s}\left[1 - e^{-sT}\right] + x(T)\frac{1}{s}e^{-sT}\left[1 - e^{-sT}\right] + \cdots + x(kT)\frac{1}{s}e^{-skT}\left[1 - e^{-sT}\right] + \cdots \quad (3.3)$$

$$\widetilde{X}(s) = \underbrace{\frac{1 - e^{-sT}}{s}}_{G_{zoh}(s)}\underbrace{\sum_{k=0}^{\infty} x(kT)e^{-skT}}_{\widetilde{X}^*(s)} = G_{zoh}(s)\widetilde{X}^*(s) \quad (3.4)$$

$\widetilde{X}^*(s)$ does not have any physical meaning without accompanied by the transfer function ($\frac{1-e^{-sT}}{s}$) of the holding device. The inverse Laplace transform of $\widetilde{X}^*(s)$ is:

$$x^*(t) = \mathcal{L}^{-1}\left(\sum_{k=0}^{\infty} x(kT)e^{-skT}\right) = \sum_{k=0}^{+\infty}\left[x(kT) * \delta(t-kT)\right] \quad (3.5)$$

where $\delta(t-kT)$ is the unit impulse function valid at $t=kT$. $x^*(t)$ is a train of impulse functions with infinite amplitude and weight of $x(kT)$ at $t=kT$.

The z transformation is defined as,

$$X^*(s) = \mathcal{L}\left[x^*(t)\right] = x(z) = \mathcal{L}\left[\sum_{k=0}^{+\infty}\left[x(kT) * \delta(t-kT)\right]\right]$$

$$x(z) = \sum_{k=0}^{+\infty}\left[x(kT)e^{-skT}\right] \tag{3.6}$$

By defining $z^{-1} = e^{-sT}$ or $z = e^{sT}$,

$$z = e^{sT}\ ,\ \ln z = sT,\ s = \frac{\ln z}{T} \tag{3.7}$$

$$x(z) = Z[x(t)] = \sum_{k=0}^{+\infty}\left[x(kT)z^{-k}\right] \tag{3.8}$$

Because z transformation considers only the values of the signal at the sampling instants, $X(z)$ gives information on the continuous variable $x(t)$ only at the sampling instants $k = 0, T, 2T, ...$

Example 21 *z transform of a unit step function ($x = 1$ for $t \geq 0$, $\mathcal{L}[1] = \frac{1}{s}$)*

$$Z[1(t)] = \sum_{k=0}^{+\infty}\left[1(kT)z^{-k}\right] = 1 + z^{-1} + z^{-2} + z^{-3} + \cdot\cdot = \frac{z}{z-1} \tag{3.9}$$

Example 22 *z transform of $x = e^{-at}$ for $t \geq 0$ ($\mathcal{L}[e^{-at}] = \frac{1}{s+a}$)*

$$Z[e^{-at}] = \sum_{k=0}^{+\infty}\left[e^{-akT}z^{-k}\right] = 1 + e^{-aT}z^{-1} + e^{-a2T}z^{-2} + e^{-a3T}z^{-3} + \cdot\cdot = \frac{z}{z-e^{-aT}} \tag{3.10}$$

Example 23 *The z transform of a unit ramp t ($\mathcal{L}[t] = \frac{1}{s^2}$) function is evaluated as:*

$$Z\left[t = kT\right] = \sum_{0}^{+\infty}\left[(kT)\cdot z^{-k}\right] = 0\frac{T}{z^0} + 1\frac{T}{z^1} + 2\frac{T}{z^2} + \cdot\cdot + + \infty\frac{T}{z^{+\infty}} = \frac{Tz}{(z-1)^2}$$

Example 24 *z transform of $x = \sin\omega t$ for $t \geq 0$ ($\mathcal{L}[e^{-at}] = \frac{\omega}{s^2+\omega^2}$)*

$$\begin{aligned} Z[\sin\omega t] &= Z\left[\frac{e^{j\omega t} - e^{-j\omega t}}{2j}\right] = \frac{1}{2j}\left[\frac{z}{z-e^{j\omega T}} - \frac{z}{z-e^{-j\omega T}}\right] \\ &= \frac{1}{2j}\frac{z(e^{j\omega T} - e^{-j\omega T})}{z^2 - z(e^{j\omega T} + e^{-j\omega T}) + 1} = \frac{z\sin\omega T}{z^2 - 2z\cos\omega T + 1} \end{aligned} \tag{3.11}$$

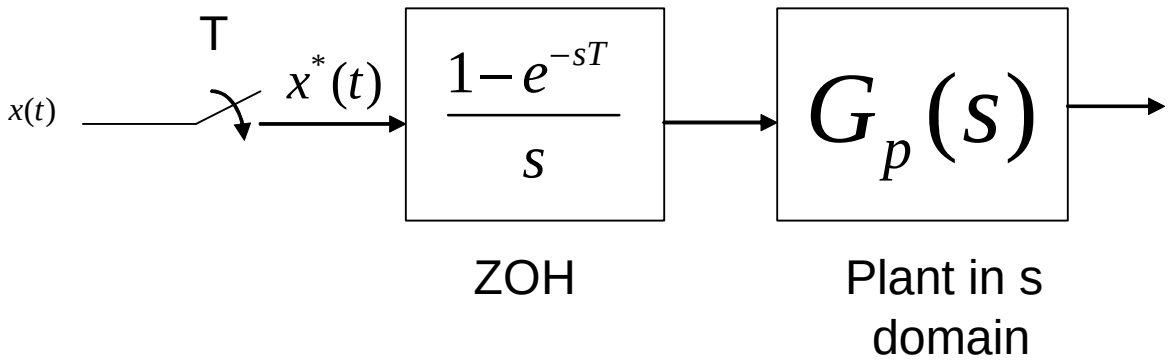


Figure 3.3: Zero Order Hold (ZOH) equivalent of a continuous system $G_p(s)$.

3.2 Zero Order Hold Equivalent of Continuous Systems

The input is passed to a physical system through D/A converter which also samples and holds the discrete commands. The transfer function of sample and hold device was obtained as $(1 - e^{-sT})/s$, whose output is fed into the physical system ($G_p(s)$) as input (3.3). The combined transfer function of ZOH and plant $G_p(s)$ leads to Zero Order Hold (ZOH) equivalent of the plant in z domain:

$$\begin{aligned} ZOH[G_p(s)] = G_p(z) &= Z\left[\frac{1-e^{-sT}}{s}G_p(s)\right] \\ &= Z\left[\frac{G_p(s)}{s} - e^{-sT}\frac{G_p(s)}{s}\right] = Z\left[\frac{G_p(s)}{s}\right] - z^{-1}Z\left[\frac{G_p(s)}{s}\right] \qquad (3.12) \\ G_p(z) &= (1-z^{-1})Z\left[\frac{G_p(s)}{s}\right] \qquad (3.13) \end{aligned}$$

Care must be taken in evaluating the discrete equivalent of continuous systems. If Euler, Forward or Tustin's approximation is used in transforming $G_p(s)$ into its discrete equivalent $G_p(z)$, the transfer function of the zero order hold device must be dropped from the system. The approximate discrete transformations are used to solve differential equations by approximating the derivative terms using Euler, Forward or Tustin's methods which do not lead exact values of time varying states at the designated time, while ZOH does.

Example 25 *Open loop block diagram of physical machine drive with a digital to analog converter is shown in Fig. 3.4.*

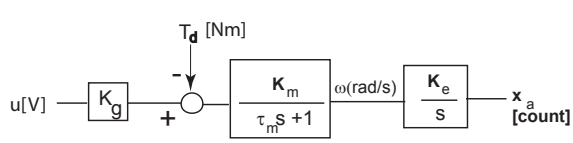


Figure 3.4: Zero Order Hold (ZOH) model of the physical machine drive.

The open loop transfer function of the system is :

$$
\begin{aligned}
G_o(s) &= \frac{K_d K_v K_e}{(\tau_v s+1)s} = \frac{K_d K_v K_e}{\tau_v}\frac{1}{s(s+1/\tau_v)} \\
&= K_0 \frac{1}{s(s+1/\tau_v)}
\end{aligned} \tag{3.14}
$$

The zero order hold equivalent of the system :

$$
G_o(z) = (1-z^{-1})Z\frac{G_o(s)}{s} = (1-z^{-1})K_0 Z\left(\frac{1}{s^2(s+1/\tau_v)}\right) \tag{3.15}
$$

It is convenient to represent the Laplace domain transfer function by its partial expansion, which leads to its easier transformation in z domain. The partial fraction expansion of $G_o(s)/s$ is:

$$
\begin{aligned}
\frac{1}{s^2(s+1/\tau_v)} &= \frac{C_1}{s^2} + \frac{C_2}{s} + \frac{C_3}{s+1/\tau_v} \\
C_1 &= \lim_{s\to 0} s^2 \frac{1}{s^2(s+1/\tau_v)} = \tau_v \\
C_2 &= \lim_{s\to 0}\left\{\frac{1}{1!}\frac{d}{ds}\left(s^2\frac{1}{s^2(s+1/\tau_v)}\right)\right\} = \lim_{s\to 0}\frac{-1}{(s+1/\tau_v)^2} = -\tau_v^2 \\
C_3 &= \lim_{s\to -1/\tau_v}\left\{(s+1/\tau_v)\frac{1}{s^2(s+1/\tau_v)}\right\} = \tau_v^2 \\
\frac{G_o(s)}{s} &= \frac{1}{s^2}\tau_v - \frac{1}{s}\tau_v^2 + \frac{\tau_v^2}{s+1/\tau_v} = \tau_v\left(\frac{1}{s^2} - \frac{\tau_v}{s} + \frac{\tau_v}{s+1/\tau_v}\right)
\end{aligned} \tag{3.16}
$$

By taking the z transform of each element in $G_o(s)/s$,

$$
\begin{aligned}
Z\left(\frac{1}{s^2}\right) &= \frac{Tz^{-1}}{(1-z^{-1})^2} \\
Z\left(-\frac{\tau_v}{s}\right) &= -\frac{\tau_v}{1-z^{-1}} \\
Z\left(\frac{\tau_v}{s+1/\tau_v}\right) &= \frac{\tau_v}{1-e^{-T/\tau_v}z^{-1}}
\end{aligned}
$$

The zero order hold (ZOH) equivalent of $G_o(s)$ is:

$$
\begin{aligned}
G_0(z) &= (1-z^{-1})Z\frac{G_o(s)}{s} \\
&= (1-z^{-1})K_0\tau_v\left(\frac{Tz^{-1}}{(1-z^{-1})^2} - \frac{\tau_v}{1-z^{-1}} + \frac{\tau_v}{1-e^{-T/\tau_v}z^{-1}}\right) \\
&= K_0\tau_v \frac{z^{-2}\left[\tau_v(1-e^{-\frac{T}{\tau_v}}) - Te^{-\frac{T}{\tau_v}}\right] + z^{-1}\left[T - \tau_v(1-e^{-\frac{T}{\tau_v}})\right]}{z^{-2}e^{-\frac{T}{\tau_v}} - z^{-1}(1+e^{-\frac{T}{\tau_v}}) + 1} \\
G_0(z) &= \frac{x_a(k)}{v(k)} = \frac{B(z)}{A(z)} = \frac{z^{-2}b_1 + z^{-1}b_0}{z^{-2}a_0 + z^{-1}a_1 + 1} = \frac{b_0 z + b_1}{z^2 + a_1 z + a_0}
\end{aligned} \tag{3.17}
$$

where

$$
\begin{aligned}
b_1 &= K_d K_v K_e \left[\tau_v(1-e^{-\frac{T}{\tau_v}}) - Te^{-\frac{T}{\tau_v}}\right] \\
b_0 &= K_d K_v K_e \left[T - \tau_v(1-e^{-\frac{T}{\tau_v}})\right] \\
a_0 &= e^{-\frac{T}{\tau_v}} \\
a_1 &= -(1+e^{-\frac{T}{\tau_v}})
\end{aligned}
$$

The response of the open loop loop system $(G_0(z)^{-1})$ can be simulated in discrete time domain. Let the open loop transfer function of the system is:

$$
G_0(z) = \frac{x_a(k)}{u(k)} = \frac{z^{-2}b_1 + z^{-1}b_0}{z^{-2}a_0 + z^{-1}a_1 + 1} = \frac{z^{-1}\left(b_0 + z^{-1}b_1\right)}{z^{-2}a_0 + z^{-1}a_1 + 1}
$$

.Note that the system has one delay (i.e. dead time) because the numerator has z^{-1} as a common factor. If it was z^{-2}, then the delay would have been 2.

$$
\begin{aligned}
G_0(z) &= \frac{x_a(k)}{u(k)} = \frac{z^{-2}b_1 + z^{-1}b_0}{z^{-2}a_0 + z^{-1}a_1 + 1} = \frac{z^{-1}\left(b_0 + z^{-1}b_1\right)}{z^{-2}a_0 + z^{-1}a_1 + 1} \\
\left[z^{-2}a_0 + z^{-1}a_1 + 1\right] x_a(k) &= \left[z^{-2}b_1 + z^{-1}b_0\right] u(k) \\
x_a(k) &= -a_0 x_a(k-2) - a_1 x_a(k-1) + b_1 u(k-2) + b_0 u(k-1)
\end{aligned}
$$

Since we need to have the values of $x_a(k-2)$ and $v(k-2)$, previous two values of the states, it is necessarry to use inittial conditions for $k = 0, -1, -2$.Let's start from $k = 1$ $x_a(k < 1) = 0$
$v(k < 0) = 0$, and $v(k \geq 1) = U$ (Step input).

k	$x_a(k) = -a_0 x_a(k-2) - a_1 x_a(k-1) + b_1 u(k-2) + b_0 u(k-1)$	$x_a(k)$
1	$-a_0 x_a(-1) - a_1 x_a(0) + b_1 u(-1) + b_0 u(0)$	$b_0 U$
2	$-a_0 x_a(0) - a_1 x_a(1) + b_1 u(0) + b_0 u(1)$	$-a_1 x_a(1) + b_1 V_0 + b_0 U$
3	$-a_0 x_a(1) - a_1 x_a(2) + b_1 u(1) + b_0 u(2)$	$-a_0 x_a(1) - a_1 x_a(2) + b_1 U + b_0 U$
4	$-a_0 x_a(2) - a_1 x_a(3) + b_1 u(2) + b_0 u(3)$	$-a_0 x_a(2) - a_1 x_a(3) + b_1 U + b_0 U$

Example 26 *Discrete equivalent of a closed loop system with non-unity continuous feedback transfer function*

Consider the block diagram given in Figure 3.1 where the feedback transfer function $H(s)$ is not unity. Typically $H(s)$ may represent the transfer function of a feedback sensor and its associated signal conditioning circuit. The discrete time closed loop transfer function of the system needs to be evaluated with care as follows. Using the notation as $ZOH[H(s)G_c(s)] = G^*_{CH}(z)$ and $ZOH[G_c(s)] = G^*_c(z)$, the error at discrete time interval k is given as:

$$
\begin{aligned}
e(k) &= u(k) - x(k) \rightarrow x(k) = ZOH\left[H(s)G_c(s)\right] v(k) = G^*_{CH}(z)v(k) \\
v(k) &= G(z)e(k) \ \rightarrow e(k) = u(k) - G^*_{CH}(z)v(k) = u(k) - G(z)\cdot G^*_{CH}(z)e(k) \\
\left[1 + G(z)\cdot G_p(z)\right] e(k) &= u(k) \rightarrow e(k) = \frac{1}{1 + G(z)\cdot G^*_{CH}(z)} u(k) \\
y(k) &= ZOH[G_c(s)]v(k) = G^*_c(z)G(z)e(k) = G^*_c(z)G(z)\frac{1}{1 + G(z)\cdot G^*_{CH}(z)} u(k) \\
y(k) &= \frac{G(z)\cdot G^*_c(z)}{1 + G(z)\cdot G^*_{CH}(z)} u(k) \rightarrow G_{cl}(s) = \frac{y(k)}{u(k)} = \frac{G(z)\cdot G^*_c(z)}{1 + G(z)\cdot G^*_{CH}(z)}
\end{aligned}
$$

3.3 Approximate Transformations between Continuous (s) and Discrete Time Domain Systems

Although the control system must be designed either in z domain if the controller is executed in the computer at discrete time intervals, or in s domain if the controller is implemented in an analog circuit. However, the engineer may have more familiarity usually with s domain even if the controller is executed by a computer in discrete time intervals (T). In such cases, the system is analyzed and controller is designed in s domain, and the controller's transfer function is transformed to z domain for real time software implementation using the following approximate digital integral/derivative methods.

Lets consider a sample example shown in Figure 3.4, where a first order velocity loop ($K_v/\tau_v s+1$) and integrator with encoder gain (K_e/s) are given to represent velocity loop and position measurement system. Digital to Analog (D/A) converter gain is K_d, sample and hold circuit dynamics is modeled by a "Zero Order Hold" system ($1-e^{-sT}/s$), and the input voltage (V_c) is supplied by the D/A circuit in the computer.

Lets consider the unit step response of the first order velocity loop

$$\tau_v \frac{d\Omega(t)}{dt} + \Omega(t) = K_v V_c(t) \rightarrow \tau_v s\Omega(s) + \Omega(s) = K_v V_c(s) \tag{3.18}$$

$$G_c(s) = \frac{\Omega(s)}{V_c(s)} = \frac{K_v}{\tau_v s+1} \tag{3.19}$$

$$\Omega(s) = G_c(s)V_c(s) = \frac{K_v}{(\tau_v s+1)}\frac{1}{s} = \frac{K_v}{\tau_v}\left(\frac{A}{s} + \frac{B}{s+1/\tau_v}\right)$$

$$A = \lim_{s\rightarrow 0} s\frac{1}{s(s+1/\tau_v)} = \tau_v$$

$$B = \lim_{s\rightarrow -1/\tau_v} (s+1/\tau_v)\frac{1}{s(s+1/\tau_v)} = -\tau_v$$

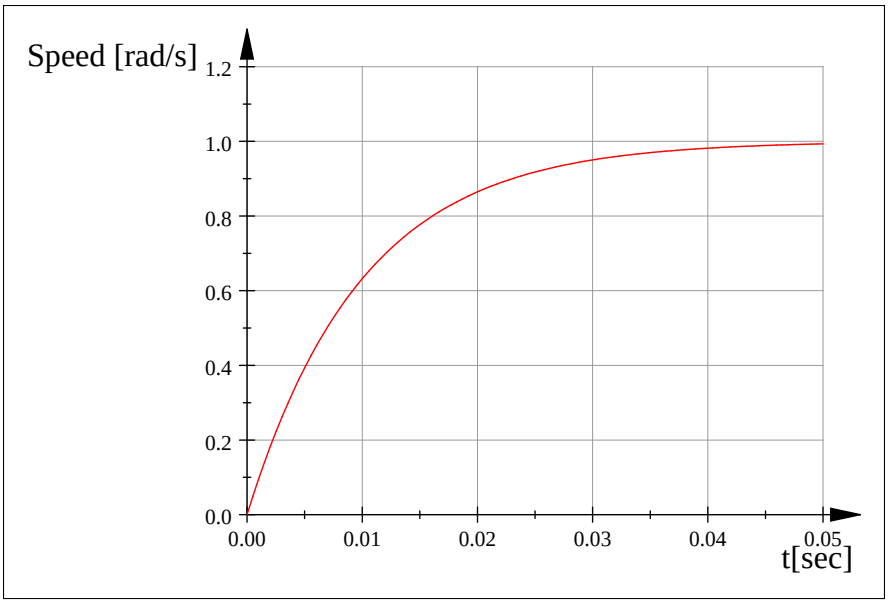
$$\Omega(s) = K_v\left(\frac{1}{s} - \frac{1}{s+1/\tau_v}\right) \rightarrow \Omega(t) = \mathcal{L}^{-1}\Omega(s)$$

$$\Omega(t) = K_v\mathcal{L}^{-1}\left(\frac{1}{s} - \frac{1}{s+1/\tau_v}\right) = K_v\left(1-e^{-t/\tau_v}\right) \tag{3.20}$$

which is the exact solution of the velocity response in continuos time domain.

Example 27 *Let* $K_v = 1[(rad/s)/V]$, $\tau_v = 0.01[\text{sec}]$ *and* $V_c = 1\ [V]$.

$$\Omega(t) = K_v\left(1-e^{-t/\tau_v}\right) = \left(1-e^{-100t}\right)$$



Unit step response of a first order system in continuous time domain.

Step response of the first order system (3.20) is shown in Figure 3.5. If the angular velocity $\Omega(t)$ is simulated at T[sec] intervals by the computer, the values of the velocity in between the sampling intervals are lost but the values at $t = kT$ can be predicted exactly using Eq. 3.20. If the Eq.3.18 needs to be solved at recursively at sampling intervals T for any given input voltage V_c, the continuous time domain equation must be transformed to discrete time by using one of the approximation methods as follows.

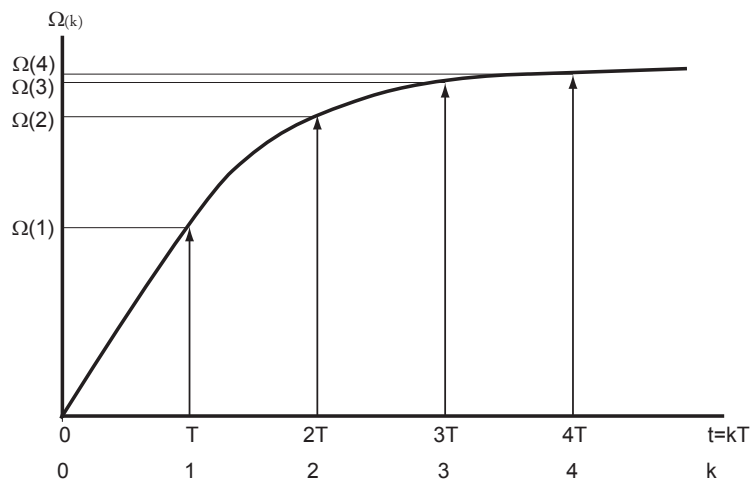


Figure 3.5: Step response of a first order system ats its sampled values at discrete time intervals.

Let's start with the original first order differential equation:

$$\tau_v \frac{d\Omega(t)}{dt} + \Omega(t) = K_v V_c(t) \tag{3.21}$$

which has the following transfer function in continuos time domain:

$$G_c(s) = \frac{\Omega(s)}{V_c(s)} = \frac{K_v}{\tau_v s + 1} \tag{3.22}$$

If the input $V_c(t)$ can be transformed into Laplace domain, the response $\Omega(t)$ can be solved exactly by taking inverse Laplace transform in continuous time domain as $\Omega(t) = \mathcal{L}^{-1}\Omega(s) = \mathcal{L}^{-1}\left[G_c(s)V_c(s)\right]$. However, if $V_c(t)$ can not be represented by a continuous function, which will be the case when machine is expected to follow varying paths, the differential equation needs to be solved at discrete intervals (T) through digital integration using an approximation as follows.

By taking the integrals of the both parts of system's differential equation :

$$\int_0^t \left(\frac{d\Omega(t)}{dt}\right) dt = -\frac{1}{\tau_v}\int_0^t \Omega(t)dt + \frac{K_v}{\tau_v}\int_0^t V_c(t)dt$$

and considering the discrete sampling periods, $t = kT$, $(t - T) = (k-1)T$ where k is the sampling interval counter and T is the sampling period, the following expression is obtained.

$$\begin{aligned}
\int_0^{kT} \left(\frac{d\Omega(t)}{dt}\right) dt &= \Omega(kT) - \Omega(0) = -\frac{1}{\tau_v}\int_0^{kT} \Omega(t)dt + \frac{K_v}{\tau_v}\int_0^{kT} V_c(t)dt \\
\int_0^{(k-1)T} \left(\frac{d\Omega(t)}{dt}\right) dt &= \Omega[(k-1)T] - \Omega(0) = -\frac{1}{\tau_v}\int_0^{(k-1)T} \Omega(t)dt + \frac{K_v}{\tau_v}\int_0^{(k-1)T} V_c(t)dt
\end{aligned} \tag{3.23}$$

By eliminating $\Omega(0)$ from Eq. 3.23, the difference equation for the velocity Ω is found,

$$\Omega(kT) - \Omega[(k-1)T] = -\frac{1}{\tau_v}\int_{(k-1)T}^{kT} \Omega(t)dt + \frac{K_v}{\tau_v}\int_{(k-1)T}^{kT} V_c(t)dt \tag{3.24}$$

Since we don't know $\Omega(t)$ in between the samples (kT) and $[(k-1)T]$, we can not take the continuous integral, therefore the integral must be approximated. One of the simplest approximation is Euler or Backward integration method:

$$\begin{aligned}
\int_{(k-1)T}^{kT} \Omega(t)dt &\simeq \Omega(kT)T \\
\int_{(k-1)T}^{kT} V_c(t)dt &\simeq V_c(kT)T
\end{aligned}$$

By substituting the Euler approximation into Eq. 3.24,

$$\Omega(kT) - \Omega[(k-1)T] = -\frac{1}{\tau_v}\Omega(kT)T + \frac{K_v}{\tau_v}V_c(kT)T$$

By using the standard digital control convention $\Omega(kT) = \Omega(k)$, $\Omega[(k-1)T] = \Omega(k-1) = z^{-1}\Omega(k)$

$$\begin{aligned}
&\rightarrow \Omega(k) - z^{-1}\Omega(k) = -\frac{1}{\tau_v}\Omega(k)T + \frac{K_v}{\tau_v}V_c(k)T \\
&\rightarrow \left[1 + \frac{T}{\tau_v} - z^{-1}\right]\ \Omega(k) = \frac{K_v T}{\tau_v}\ V_c(k) \\
G(z^{-1}) &= \frac{\Omega(k)}{V_c(k)} = \frac{K_v T}{\tau_v}\frac{1}{\left(\frac{\tau_v + T}{\tau_v}\right) - z^{-1}} = \frac{K}{1 - az^{-1}} \\
G(z) &= \frac{Kz}{z-a} \quad \text{where } K = \frac{K_v T}{\tau_v + T},\ a = \frac{\tau_v}{\tau_v + T}
\end{aligned} \tag{3.25}$$

where $G(z)$ represents approximated discrete transfer function of continuous system $G_c(s)$ using Euler or backward approximation.

Example 28 *Lets evaluate the unit step response of the same first order system in discrete time domain using Euler approximation.*$K_v = 1[(rad/s)/V], \tau_v = 0.01$[sec], *and* $V_c = 1[V]$. *Let the sampling period is* $T = 0.005$[sec]. *The corresponding parameters in z domain are:* $K = \frac{K_v T}{\tau_v + T} = \frac{0.005}{0.005+0.01} = 0.33333$, $a = \frac{\tau_v}{\tau_v + T} = \frac{0.01}{0.01+0.005} = 0.66667$. *The resulting transfer function in z domain becomes:*

$$G(z) = \frac{\Omega(k)}{V_c(k)} = \frac{K}{1 - az^{-1}} = \frac{0.33333}{1 - 0.66667z^{-1}}$$
$$\left[1 - 0.66667z^{-1}\right]\Omega(k) = 0.33333V_c(k)$$
$$\Omega(k) = 0.66667\Omega(k-1) + 0.33333V_c(k)$$

The velocity can be solved at discrete time intervals with $\Omega(0) = 0$:

k	$t = kT$[sec]	$V_c(k)$	$\Omega(k) = 0.66667\Omega(k-1) + 0.33333V_c(k)$	$\Omega(k)$
0	0	0	0	0
1	0.005	1	$0.66667 * 0 + 0.33333$	0.33333
2	0.01	1	$0.66667 * 0.33333 + 0.33333$	0.555 55
3	0.015	1	$0.66667 * 0.555\,55 + 0.33333$	0.703 70
4	0.02	1	$0.66667 * 0.703\,70 + 0.33333$	0.802 47
5	0.025	1	$0.66667 * 0.802\,47 + 0.33333$	0.868 31
6	0.03	1	$0.66667 * 0.868\,31 + 0.33333$	0.912 21
7	0.035	1	$0.66667 * 0.912\,21 + 0.33333$	0.941 47
8	0.040	1	$0.66667 * 0.941\,47 + 0.33333$	0.960 98
9	0.045	1	$0.66667 * 0.960\,98 + 0.33333$	0.973 99
10	0.050	1	$0.66667 * 0.973\,99 + 0.33333$	0.982 66

It can be seen from 3.6 that there is a difference between the exact (continuous time domain) and discrete time system which uses Euler approximation.

Euler transformation can be carried out by substituting $s \simeq \frac{1-z^{-1}}{T} = \frac{z-1}{zT}$in any transfer function which is represented in Laplace domain. It is common to use the following transformation techniques in practice:

Method	s	z	Integral approximations
Euler (Backward)	$\frac{z-1}{zT} = \frac{1-z^{-1}}{T}$	$\frac{1}{1-Ts}$	$\Omega(k) \cdot T$
Forward	$\frac{z-1}{T} = \frac{1-z^{-1}}{z^{-1}T}$	$1 + Ts$	$\Omega(k-1) \cdot T$
Tustin (Trapezoidal)	$\frac{2(z-1)}{T(z+1)} = \frac{2(1-z^{-1})}{T(1+z^{-1})}$	$\frac{2+sT}{2-sT}$	$\frac{T}{2}\left[\Omega(k-1) \cdot T + \Omega(k)\right]$

The continuos velocity function $\Omega(t)$ is converted into discrete points from the differential equation using different approximations as illustrated in Figure 3.7. Note that smaller the sampling interval (T) is, closer the transformation approximation will be, which should be kept in mind when switching from discrete to continuos time domain or vice versa. However, regardless of the method used, these are "approximate" digital integration techniques which do not follow the exact solution as shown in Figure 3.6. However, the zero order hold (ZOH), which will be covered next, is used in digital control engineering, and it follows the exact response at discrete intervals as shown in 3.3.

Example 29 ***A sample MATLAB code to analyze the response of a first order system***

```
% Matlab Code for a first order system
% Prof. YUSUF ALTINTAS
% Objective : Show the step response of the system using inverse Laplace
% transform, Euler approximation and Zero Order Hold Response.
% September 2014
clc; %clear the memory
clear all;
close all;
```

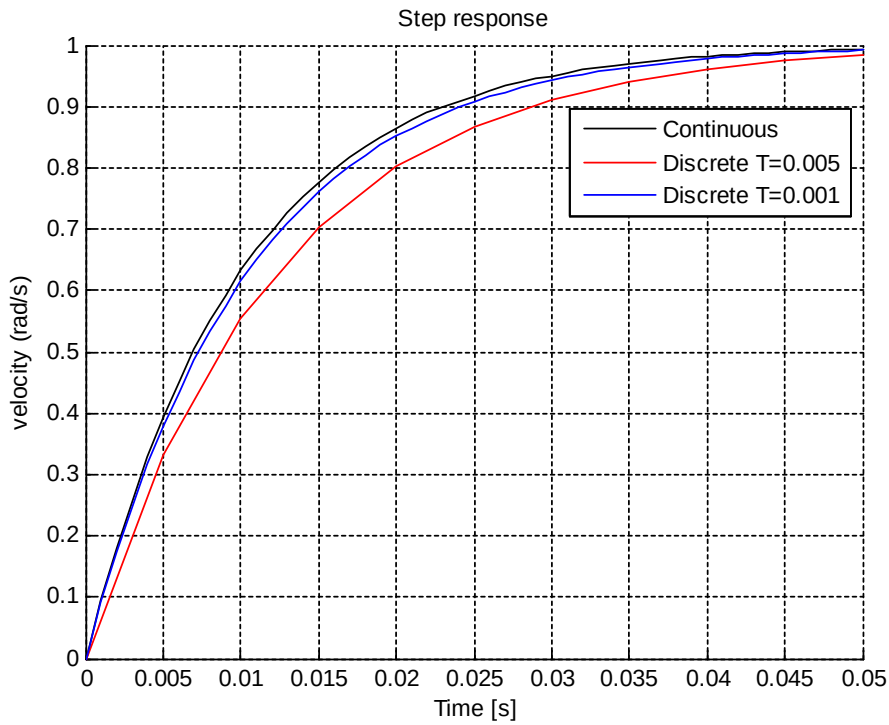


Figure 3.6: Step response of a first order system with exact solution using inverse Laplace tarnsform (Continuous); using Euler approximation with sampling time $T = 0.005[s]$ and $T = 0.001[s]$

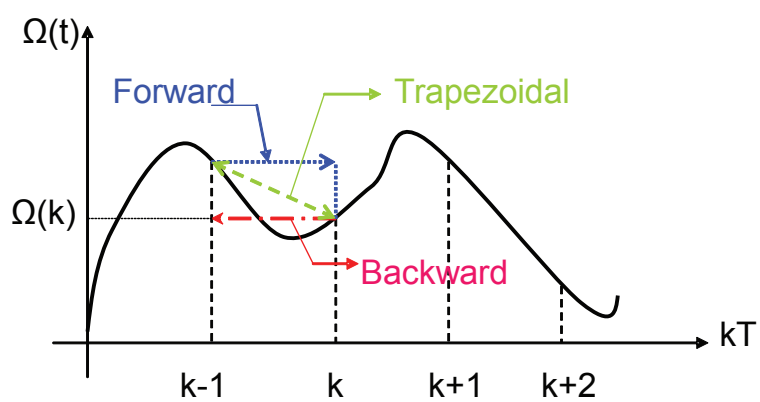
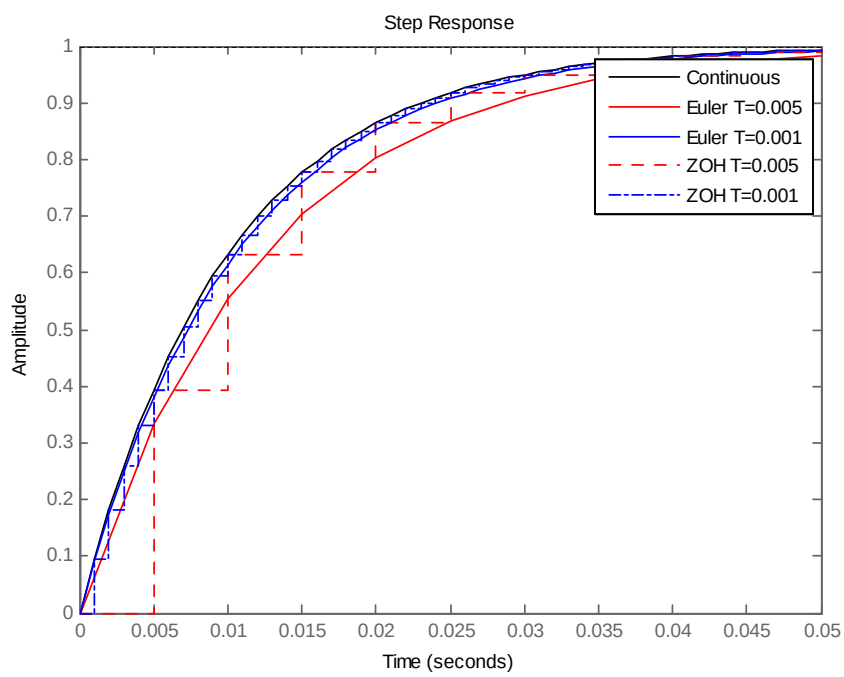


Figure 3.7: Discrete approximation of continuous systems with Euler (backward), forward and trapezoidal (Tustin) approximations.

```
% Parameters of a sample machine. The input is Vc [V] and output is
% velocity w(rad/s)as an example
% Dif. eqn tau*dw(t)/dt+w(t)=Kv*Vc(t)
Kv=1 ; % gain [(rad/s)/V]
tau=0.01 ; % [sec] time constant
%% Transfer Function of drive in s domain
% Modeling of the drive in s domain%
%Gm(s)=Kv/(tau s + 1)
Gv=tf([Kv],[tau 1]);
Vc=1 ; %Amplitude of the input
%in matlab step response is step(Gm) for Vc=1 [V] or v=Vc*step(Gm);
%now lets use our own solution. Simulate the exact solution it in 0.001 sec intervals
t = 0:0.001:0.050;
figure(100); clf;
hold on;
speedc=Kv*Vc*(1-exp(-t/tau)); % velocity response
plot(t,speedc,'k');
axis([0 0.05 0 1]);grid;
title(sprintf('Step response'));
xlabel(' Time [s]'); ylabel('velocity (rad/s)');
% Lets use Euler aproximation G(z)=K/(1-az^(-1))
% First euler approximation
T1=0.005 ; % sampling time is 5 ms
K1=Kv*T1/(tau+T1) ; a1=tau/(tau+T1) ;Vc=1; speed1(1)=0;
% If we use Zero order hold equivalent
G1zoh= c2d(Gv,T1,'zoh');
for k=2:1:11;
t1(k)=(k-1)*T1;
speed1(k)=a1*speed1(k-1)+K1*Vc;
end;
plot(t1,speed1,'r');
legend('Discrete T=0.005');
% Second approximation
T2=0.001 ; % sampling time is 1 ms
K2=Kv*T2/(tau+T2) ; a2=tau/(tau+T2) ;Vc=1; speed2(1)=0;
% If we use Zero order hold equivalent
G2zoh= c2d(Gv,T2,'zoh');
for k=2:1:51;
t2(k)=(k-1)*T2;
speed2(k)=a2*speed2(k-1)+K2*Vc;
end;
plot(t2,speed2,'b');
step(G1zoh,'r–',G2zoh,'b-.',0.05);
% Obtain the step response using Zero Order Hold equivalent
legend('Continuous','Euler T=0.005', 'Euler T=0.001','ZOH T=0.005','ZOH T=0.001');
hold off;
% Bode plot (Frequency Repsonse of the first order system)
w=0;
for k=1:1:500;
w=w+1;
wc(k)=w;
MG(k)=Kv/sqrt((w*tau)^2+1); %magnitude in linear scale
Phase(k)=-(180/pi)*atan2(tau*w,1); %phase in degrees
end;
```

```
figure(200); clf;
hold on;grid on;
subplot(2,1,1); plot(wc,MG)
axis([0 500 0 1.2]);
title(sprintf('FRF diagram of a first order system -Linear Scale '));
xlabel(' Frequency [rads]'); ylabel('Magnitude Gm');
subplot(2,1,2);plot(wc,Phase);
xlabel(' Frequency [rads]'); ylabel('Phase (deg)');
axis([0 500 -90 0]);
drawnow;
hold off;
%Bode plot with MATLAB command in Logarithmic scale
figure(300);clf;
bode(Gv) ; % It will plot the FRF (Bode) plot of GV in log scale
% The frequency is in rad/s
% END
```



Comparison of step response of a first order system with Zero Order Hold and Euler approximation with different sampling time periods.

Example 30 *Response of a single degree of freedom system under step load was solved in continuous time domain using Laplace transforms in Eq.2.29 as*

$$x(t) = \frac{F_0}{k}\left[1 - e^{-\zeta\omega_n t}\frac{1}{\sqrt{1-\zeta^2}}\sin\left(\omega_{\mathrm{d}} t + \phi\right)\right] \rightarrow \phi = \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta} \tag{3.26}$$

The response can be evalauted at discrete time intervals (T) by transforming the transfer function of the system from Laplace to z domain with Euler approximation $s \approx (1 - z^{-1})/T$:

$$
\begin{aligned}
\Phi(s = \frac{1-z^{-1}}{T}) &= \frac{\omega_{\mathrm{n}}^2}{k} \left. \frac{1}{s^2 + 2\zeta\omega_{\mathrm{n}} s + \omega_{\mathrm{n}}^2} \right|_{s=\frac{1-z^{-1}}{T}} \\
\Phi(z^{-1}) &= \frac{\omega_{\mathrm{n}}^2}{k} \frac{1}{\left(\frac{1-z^{-1}}{T}\right)^2 + 2\zeta\omega_{\mathrm{n}} \left(\frac{1-z^{-1}}{T}\right) + \omega_{\mathrm{n}}^2} \\
&= \frac{\omega_{\mathrm{n}}^2}{k} \frac{T^2}{z^{-2} - 2\left(\zeta\omega_{\mathrm{n}} T + 1\right) z^{-1} + \left(1 + 2\zeta\omega_{\mathrm{n}} T + T^2\omega_{\mathrm{n}}^2\right)} \\
\Phi(z^{-1}) &= \frac{b_0}{z^{-2} + a_1 z^{-1} + a_0} \\
\text{where } b_0 &= \frac{T^2\omega_{\mathrm{n}}^2}{k}, \quad a_1 = -2\left(\zeta\omega_{\mathrm{n}} T + 1\right), a_0 = 1 + 2\zeta\omega_{\mathrm{n}} T + T^2\omega_{\mathrm{n}}^2
\end{aligned}
$$

The vibration of the machine with a single degree of freedom dynamics can be evaluated at discrete time intervals as ($z^{-1}x(k) = x(k-1)$),

$$
\begin{aligned}
\Phi(z^{-1}) &= \frac{b_0}{z^{-2} + a_1 z^{-1} + a_0} = \frac{x(k)}{F(k)} \\
\left(z^{-2} + a_1 z^{-1} + a_0\right) x(k) &= b_0 F(k) \\
x(k-2) + a_1 x(k-1) + a_0 x(k) &= b_0 F(k) \\
x(k) &= \frac{1}{a_0}\left[-x(k-2) - a_1 x(k-1)\right] + \frac{b_0}{a_0} F(k)
\end{aligned}
$$

where $x(k)$, is the vibration of the structure when the machine is loaded with forces $F(k)$ at time intervals $t = kT \rightarrow k = 0, 1, 2, ..., k$. The difference equation can be solved recursively at each time interval k.

3.4 Summary

Remember that the Laplace transform of a function $f(t)$ was,

$$
\begin{aligned}
F(s) = \mathcal{L}(t) &= \int_0^{+\infty} f(t) e^{-st} dt \\
\mathcal{L}\frac{df(t)}{dt} &= sF(s) \text{ with zero initial conditions.}
\end{aligned}
$$

In discrete systems, a similar procedure is used for z transforms. The z transform of a discrete function ($f(k)$) is evaluated by:

$$
Z\left[f(k)\right] = \sum_0^{+\infty} f(kT) z^{-k}
$$

where $f(k)$ is the discrete equivalent of continuous function $f(t)$. Note that $z^{-1}f(k) = f(k-1)$, hence z^{-1} acts as a delay of one sampling period (T) on functions $f(k)$. If we consider the same delay in a continuous time domain function, i.e. $f(t-T)$. The Laplace transform of $\mathcal{L}(t-T) = e^{-sT} f(t)$. From the analogy, the z^{-1} operator corresponds to :

$$
z^{-1} = e^{-sT} \leftrightarrows z = e^{sT}
$$

Zero Order Hold equivalent of a system $G_p(s)$ is obtained by

$$
G_p(z) = (1 - z^{-1}) Z\left[\frac{G_p(s)}{s}\right]
$$

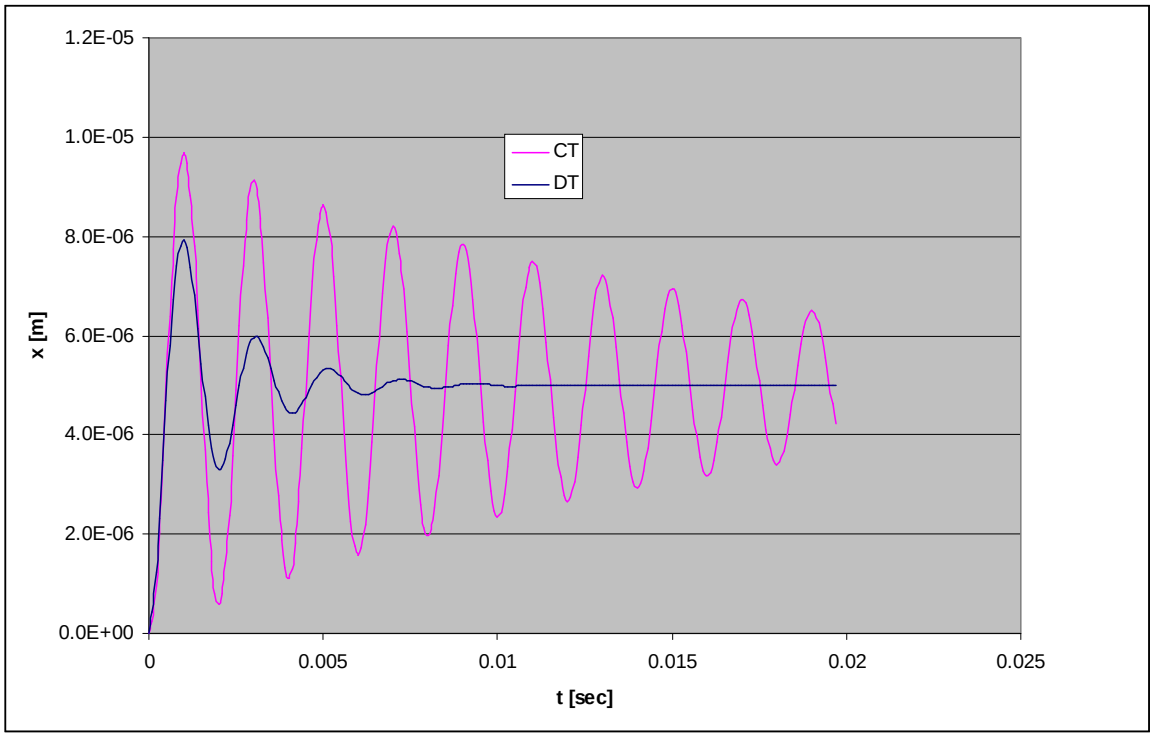


Figure 3.8: Step response of an underdamped system with continuous time (CT) and discrete time (DT) solutions.

Theorem 31 *Final Value Theorem*

$$\textit{Continuous time domain} \lim_{t\to+\infty} f(t) = \lim_{s\to+0} sF(s)$$
$$\textit{Discrete time domain} \lim_{t\to+\infty} f(t) = \lim_{z\to1}(1-z^{-1})F(z)$$

Proof of final value theorem in z domain:

$$\begin{aligned} F(z) &= \lim_{N\to\infty}\sum_{k=0}^{N} f(k)z^{-k} \to z^{-1}F(z) = \lim_{N\to\infty}\sum_{k=0}^{N} f(k-1)z^{-k} \\ (1-z^{-1})F(z) &= \lim_{N\to\infty}\sum_{k=0}^{N}\left[f(k)z^{-k} - f(k-1)z^{-k}\right] \end{aligned}$$

With $f(k) = 0$ when $k < 0$ and taking the limit as $z \to 1$,

$$\lim_{z\to1}(1-z^{-1})F(z) = \lim_{N\to\infty}\left[f(N) - f(-1)\right] = \lim_{N\to\infty} f(N)$$

Theorem 32 *Initial Value Theorem*

$$\textit{Continuous time domain} \lim_{t\to+0} f(t) = f(0) = \lim_{s\to+\infty} sF(s)$$
$$\textit{Discrete time domain} \lim_{t\to+0} f(t) = f(0) = \lim_{z\to+\infty} F(z)$$

Table 3.1: Table of Laplace and z Transforms [1]

$x(t)$	$X(s)$	$x(kT)$	$X(z)$
Impulse	—	$\delta(k) = \left\{ \begin{smallmatrix} 1 & k=0 \\ 0 & k \neq 1 \end{smallmatrix} \right\}$	1
Delay	—	$\delta(n-k) = \left\{ \begin{smallmatrix} 1 & k=n \\ 0 & k \neq n \end{smallmatrix} \right\}$	z^{-k}
1	$\frac{1}{s}$	1	$\frac{1}{1-z^{-1}}$
e^{-at}	$\frac{1}{s+a}$	e^{-akT}	$\frac{1}{1-e^{-aT}z^{-1}}$
t	$\frac{1}{s^2}$	kT	$\frac{Tz^{-1}}{(1-z^{-1})^2}$
t^2	$\frac{2}{s^3}$	$(kT)^2$	$\frac{T^2z^{-1}(1+z^{-1})}{(1-z^{-1})^3}$
t^3	$\frac{6}{s^4}$	$(kT)^3$	$\frac{T^3z^{-1}(1+4z^{-1}+z^{-2})}{(1-z^{-1})^4}$
$1-e^{-at}$	$\frac{a}{s(s+a)}$	$1-e^{-akT}$	$\frac{z^{-1}(1-e^{-aT})}{(1-z^{-1})(1-e^{-aT}z^{-1})}$
$e^{-at}-e^{-bt}$	$\frac{b-a}{(s+a)(s+b)}$	$e^{-akT}-e^{-bkT}$	$\frac{z^{-1}(e^{-aT}-e^{-bT})}{(1-e^{-aT}z^{-1})(1-e^{-bT}z^{-1})}$
te^{-at}	$\frac{1}{(s+a)^2}$	kTe^{-akT}	$\frac{Te^{-aT}z^{-1}}{(1-e^{-aT}z^{-1})^2}$
$(1-at)e^{-at}$	$\frac{s}{(s+a)^2}$	$(1-akT)e^{-akT}$	$\frac{1-(1+aT)e^{-aT}z^{-1}}{(1-e^{-aT}z^{-1})^2}$
t^2e^{-at}	$\frac{2}{(s+a)^3}$	$(kT)^2e^{-akT}$	$\frac{T^2e^{-aT}(1+e^{-aT}z^{-1})z^{-1}}{(1-e^{-aT}z^{-1})^3}$
$\sin\omega t$	$\frac{\omega}{s^2+\omega^2}$	$\sin\omega kT$	$\frac{z^{-1}\sin\omega T}{1-2z^{-1}\cos\omega T+z^{-2}}$
$\cos\omega t$	$\frac{s}{s^2+\omega^2}$	$\cos\omega kT$	$\frac{1-z^{-1}\cos\omega T}{1-2z^{-1}\cos\omega T+z^{-2}}$
$e^{-at}\sin\omega t$	$\frac{\omega}{(s+a)^2+\omega^2}$	$e^{-akT}\sin\omega kT$	$\frac{e^{-aT}z^{-1}\sin\omega T}{1-2e^{-aT}z^{-1}\cos\omega T+e^{-2aT}z^{-2}}$
$e^{-at}\cos\omega t$	$\frac{s+a}{(s+a)^2+\omega^2}$	$e^{-akT}\cos\omega kT$	$\frac{1-e^{-aT}z^{-1}\cos\omega T}{1-2e^{-aT}z^{-1}\cos\omega T+e^{-2aT}z^{-2}}$
		a^k	$\frac{1}{1-az^{-1}}$
		$a^{k-1}, k=1,2,3\ldots$	$\frac{z^{-1}}{1-az^{-1}}$
		ka^{k-1}	$\frac{z^{-1}}{(1-az^{-1})^2}$
		$a^k\cos k\pi$	$\frac{1}{1+az^{-1}}$

Chapter 4

Closed Loop Control System

4.1 Transfer Function of Closed Loop Systems

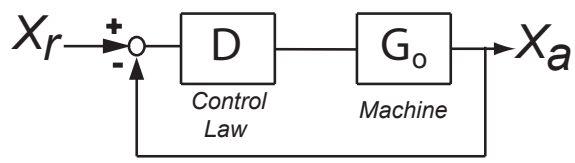


Figure 4.1: A closed loop system with a unity feedback.

If the open loop transfer function of the system is $G_0(s)$, the transfer function of the control law is $D(s)$, and the system has unity feedback. Lets assume that the input is $x_r(t)$, the output is $x_a(t)$.The closed loop transfer function ($G_{cl}(s)$) of the system with unity feedback is evaluated in s domain as follows:

$$G_{cl}(s) = \frac{x_a(s)}{x_r(s)} = \frac{D(s)G_0(s)}{1 + D(s)G_0(s)} \tag{4.1}$$

If the system is represented in discrete time domain (z domain), the closed loop transfer function becomes:

$$G_{cl}(z) = \frac{x_a(z)}{x_r(z)} = \frac{D(z)G_0(z)}{1 + D(z)G_0(z)} \tag{4.2}$$

4.2 Time Domain Response of Closed Loop Control Systems

Consider a dynamic system with a transfer function represented in continuos (Laplace) domain as:

$$\begin{aligned} G(s) = \frac{x(s)}{u(s)} &= \frac{b_m s^m + b_{m-1}s^{m-1} + \cdots + b_1 s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1 s + a_0} \\ &= K\frac{(s - z_m)(s - z_{m-1}) + \cdots + (s - z_1)}{(s - p_n)(s - p_{n-1}) + \cdots + (s - p_1)} \end{aligned} \tag{4.3}$$

where the order of denominator is greater than the order of numerator ($n > m$) for a *proper system*. The system has m zeros and n poles. The system response (output) is $x(s)$ for a given excitation (input) $u(s)$. The transfer function may represent open loop or closed loop dynamics of the system depending on the subject of analysis.

The time domain response of a continuous system is evaluated by specifying the input in Laplace domain, multiplying it with the transfer function followed by the inverse Laplace Transformation of the output.

$$x_a(s) = G_{cl}(s)x_r(s) = \frac{D(s)G_0(s)}{1 + D(s)G_0(s)}x_r(s) \tag{4.4}$$

$$x_a(t) = \mathcal{L}^{-1}x_a(s) = \mathcal{L}^{-1}\left[\frac{D(s)G_0(s)}{1 + D(s)G_0(s)}x_r(s)\right] \tag{4.5}$$

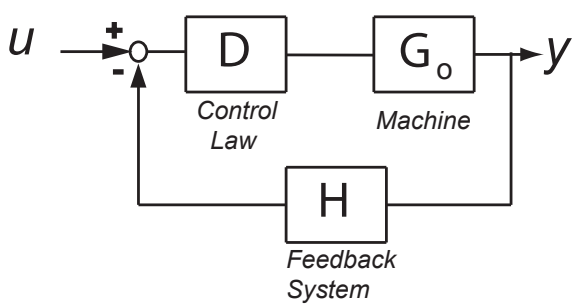
Similarly, the discrete time response of the system is evaluated as:

$$x_a(k) = G_{cl}(z)x_r(k) = \frac{D(z)G_0(z)}{1 + D(z)G_0(z)}x_r(z) \tag{4.6}$$

If the response is analyzed when the output (response) is still approaching to the desired input, the system is called to be in transient state, and the analysis is called transient response analysis. If the system settles down at a fixed trend, it is called stead state response which is identified by analyzing the response when time goes to infinity ($t \to \infty$ or $s \to 0$, $z \to 1$).

Example 33 *Let the closed loop transfer function of the plant is a second order, underdamped system* $G_{cl}(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$, *and the input is step function with an amplitude* $x_r(t) = U$.

Figure 4.2: Closed loop system with feedback (H) dynamics.



$$
\begin{aligned}
x_r(s) &= \mathcal{L}(U) = \frac{U}{s} \rightarrow x_a(s) = G_{cl}(s)x_r(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}\frac{U}{s} \\
x_a(t) &= \mathcal{L}^{-1}x_a(s) = \mathcal{L}^{-1}U\left[\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}\frac{1}{s}\right] \\
x_a(s) &= U\left[\frac{C_1}{s}+\frac{C_2}{s-p_1}+\frac{C_2^*}{s-p_1^*}\right] \rightarrow p_1 = -\zeta\omega_n + j\omega_d, \quad p_1^* = -\zeta\omega_n - j\omega_d \\
p_1 - p_1^* &= -\zeta\omega_n + j\omega_d - (-\zeta\omega_n - j\omega_d) = j2\omega_d \\
p_1^* - p_1 &= -\zeta\omega_n - j\omega_d - (-\zeta\omega_n + j\omega_d) = -j2\omega_d \\
C_1 &= \lim_{s\to 0} s\left[\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}\frac{1}{s}\right] = 1 \\
C_2 &= \lim_{s\to p_1}(s-p_1)\left[\frac{\omega_n^2}{(s-p_1)(s-p_1^*)}\frac{1}{s}\right] = \frac{\omega_n^2}{p_1-p_1^*}\frac{1}{p_1} = \frac{\omega_n^2}{j2\omega_d}\frac{1}{-\zeta\omega_n+j\omega_d} \\
&= \frac{\omega_n^2}{-j2\zeta\omega_d\omega_n - 2\omega_d^2} = \frac{\omega_n^2}{-j2\zeta\omega_n^2\sqrt{1-\zeta^2} - 2\omega_n^2(\sqrt{1-\zeta^2})^2} = \frac{1}{2\sqrt{1-\zeta^2}}\frac{1}{-\sqrt{1-\zeta^2}-j\zeta} \\
C_2^* &= \lim_{s\to p_1^*}(s-p_1^*)\left[\frac{\omega_n^2}{(s-p_1)(s-p_1^*)}\frac{1}{s}\right] = \frac{\omega_n^2}{p_1^*-p_1}\frac{1}{p_1^*} = \frac{\omega_n^2}{-j2\omega_d}\frac{1}{-\zeta\omega_n-j\omega_d} \\
&= \frac{\omega_n^2}{+j2\zeta\omega_d\omega_n - 2\omega_d^2} = \frac{1}{2\sqrt{1-\zeta^2}}\frac{1}{-\sqrt{1-\zeta^2}+j\zeta} \\
x_a(s) &= U\left[\frac{1}{s}+\frac{1}{2\sqrt{1-\zeta^2}}\left(\frac{1}{-\sqrt{1-\zeta^2}-j\zeta}\frac{1}{s-p_1}+\frac{1}{-\sqrt{1-\zeta^2}+j\zeta}\frac{1}{s-p_1^*}\right)\right]
\end{aligned}
$$

By taking the inverse Laplace transform of response $x_a(s)$,

$$
\begin{aligned}
x_a(t) &= \mathcal{L}^{-1}x_a(s) = U\left[1+\frac{1}{2\sqrt{1-\zeta^2}}\left(\frac{1}{-\sqrt{1-\zeta^2}-j\zeta}e^{+p_1 t}+\frac{1}{-\sqrt{1-\zeta^2}+j\zeta}e^{+p_1^* t}\right)\right] \\
&= U\left[1+\frac{e^{-\zeta\omega_n t}}{2\sqrt{1-\zeta^2}}\left(\frac{1}{-\sqrt{1-\zeta^2}-j\zeta}e^{+j\omega_d t}+\frac{1}{-\sqrt{1-\zeta^2}+j\zeta}e^{-j\omega_d t}\right)\right] \\
&= U\left[1+\frac{e^{-\zeta\omega_n t}}{2\sqrt{1-\zeta^2}}\left(\frac{\left(-\sqrt{1-\zeta^2}+j\zeta\right)e^{+j\omega_d t}+\left(-\sqrt{1-\zeta^2}-j\zeta\right)e^{-j\omega_d t}}{\left(-\sqrt{1-\zeta^2}-j\zeta\right)\left(-\sqrt{1-\zeta^2}+j\zeta\right)}\right)\right] \\
&= U\left[1+\frac{e^{-\zeta\omega_n t}}{2\sqrt{1-\zeta^2}}\left(\frac{\left(-\sqrt{1-\zeta^2}+j\zeta\right)(\cos\omega_d t + j\sin\omega_d t)+\left(-\sqrt{1-\zeta^2}-j\zeta\right)(\cos\omega_d t - j\sin\omega_d t)}{(1-\zeta^2+\zeta^2)}\right)\right] \\
&= U\left[1+\frac{e^{-\zeta\omega_n t}}{2\sqrt{1-\zeta^2}}\left\{-\sqrt{1-\zeta^2}\cos\omega_d t + j\zeta\cos\omega_d t - j\sqrt{1-\zeta^2}\sin\omega_d t - \zeta\sin\omega_d t + \right.\right. \\
&\qquad \left.\left. -\sqrt{1-\zeta^2\cos\omega_d t} - j\zeta\cos\omega_d t + \sqrt{1-\zeta^2}\sin\omega_d t - \zeta\sin\omega_d t\right\}\right] \\
x_a(t) &= U\left[1+\frac{e^{-\zeta\omega_n t}}{2\sqrt{1-\zeta^2}}\left(-2\sqrt{1-\zeta^2}\cos\omega_d t - 2\zeta\sin\omega_d t\right)\right] \\
&= U\left[1-e^{-\zeta\omega_{\mathrm{n}} t}\left(\cos\omega_{\mathrm{d}} t + \frac{\zeta}{\sqrt{1-\zeta^2}}\cdot\sin\omega_{\mathrm{d}} t\right)\right], \omega_{\mathrm{d}} = \omega_{\mathrm{n}}\sqrt{1-\zeta^2} \qquad (4.7)
\end{aligned}
$$

We can further simplify the expression by using the trigonometric properties $\sin(\phi + b) = \sin\phi\cos b +$

$\cos\phi \sin b$ and by assigning $\sin\phi \equiv \sqrt{1-\zeta^2}, \cos\phi \equiv \zeta$ the following identity can be expressed $\tan\phi = \frac{\sqrt{1-\zeta^2}}{\zeta} \rightarrow \phi = \tan^{-1}(\frac{\sqrt{1-\zeta^2}}{\zeta})$. The step response can be reorganized as

$$\begin{aligned} x_a(t) &= U\left[1 - \frac{e^{-\zeta\omega_{\mathrm{n}}t}}{\sqrt{1-\zeta^2}}\left(\sqrt{1-\zeta^2}\cos\omega_{\mathrm{d}}t + \zeta\cdot\sin\omega_{\mathrm{d}}t\right)\right] \\ &= U\left[1 - \frac{e^{-\zeta\omega_{\mathrm{n}}t}}{\sqrt{1-\zeta^2}}\cdot\sin(\omega_{\mathrm{d}}t+\phi)\right], \phi = \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta} \end{aligned}$$

where the system oscillates at its damped natural frequency ω_{d}. The unit step response of a second order underdamped system is shown in Figure 4.3. The system is described by its rise time (t_r) where the output reaches to the step input level, peak time (t_p) where the response reaches to maximum overshoot, time constant (τ) where the response reaches to 63% of the step input level, and settling time (t_s) where the response reaches to step input level within %4.

The **rise time** can be evaluated by $x_a(t \rightarrow t_r) = U$. Since $e^{-\zeta\omega_{\mathrm{n}}t_r} \neq 0$,

$$\begin{aligned} \sin(\omega_{\mathrm{d}}t_r + \phi) = 0 &\rightarrow \omega_{\mathrm{d}}t + \phi = 0, \pm\pi, \pm 2\pi \\ \omega_{\mathrm{d}}t_r + \phi = \pi &\rightarrow t_r = \frac{\pi-\phi}{\omega_{\mathrm{d}}}, \phi = \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta} \end{aligned} \tag{4.8}$$

The **peak time** is determined when the output reaches the first maximum overshoot,

$$\begin{aligned} \frac{dx_a(t)}{dt} &= \zeta\omega_{\mathrm{n}}e^{-\zeta\omega_{\mathrm{n}}t_p}\left[\cos\omega_{\mathrm{d}}t_p + \frac{\zeta}{\sqrt{1-\zeta^2}}\cdot\sin\omega_{\mathrm{d}}t_p\right] - e^{-\zeta\omega_{\mathrm{n}}t_p}\left[-\omega_{\mathrm{d}}\sin\omega_{\mathrm{d}}t_p + \frac{\zeta\omega_{\mathrm{d}}}{\sqrt{1-\zeta^2}}\cdot\cos\omega_{\mathrm{d}}t_p\right] = 0 \\ &= \omega_{\mathrm{n}}e^{-\zeta\omega_{\mathrm{n}}t_p}\cos\omega_{\mathrm{d}}t_p\left[\zeta - \frac{\zeta\sqrt{1-\zeta^2}}{\sqrt{1-\zeta^2}}\right] + \omega_{\mathrm{n}}\sin\omega_{\mathrm{d}}t_p\left[\frac{\zeta^2}{\sqrt{1-\zeta^2}} + \sqrt{1-\zeta^2}\right] \\ &= \omega_{\mathrm{n}}e^{-\zeta\omega_{\mathrm{n}}t_p}\cos\omega_{\mathrm{d}}t_p\left[\frac{\zeta\sqrt{1-\zeta^2} - \zeta\sqrt{1-\zeta^2}}{\sqrt{1-\zeta^2}}\right] + \omega_{\mathrm{n}}\sin\omega_{\mathrm{d}}t_p\left[\frac{\zeta^2+1-\zeta^2}{\sqrt{1-\zeta^2}}\right] \qquad (4.9) \\ &= 0 + \frac{\omega_{\mathrm{n}}}{\sqrt{1-\zeta^2}}\sin\omega_{\mathrm{d}}t_p = 0 \qquad (4.10) \end{aligned}$$

The corresponding peak time (t_p) becomes,

$$\sin\omega_{\mathrm{d}}t_p = 0 \rightarrow \omega_{\mathrm{d}}t_p = 0, \pi, 2\pi, .. \qquad \rightarrow t_p = \frac{\pi}{\omega_{\mathrm{d}}} \tag{4.11}$$

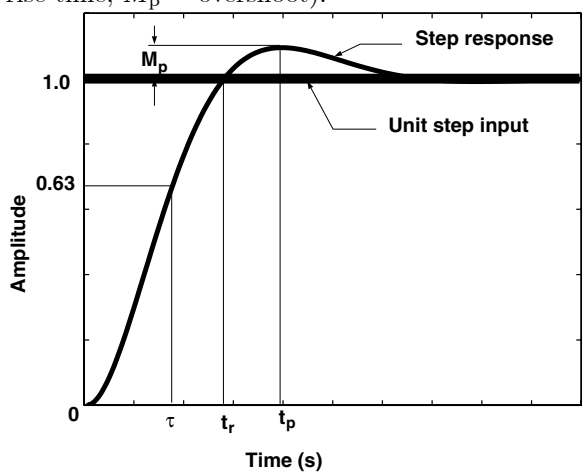
By substituting peak time (t_p) into the step response $x_a(t \rightarrow t_p)$ and $t_p = \frac{\pi}{\omega_{\mathrm{d}}}$, the maximum overshoot (M_p) can be evaluated as,

$$M_p = x_a(t) - U = -Ue^{-\zeta\omega_{\mathrm{n}}\frac{\pi}{\omega_{\mathrm{d}}}}\left(\cos\pi + \frac{\zeta}{\sqrt{1-\zeta^2}}\cdot\sin\pi\right) = Ue^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{4.12}$$

If the overshoot amplitude is given for a unit step input ($U = 1$), the damping can be evaluated from the following equation,

$$\ln M_p = -\frac{\zeta\pi}{\sqrt{1-\zeta^2}} \tag{4.13}$$

Figure 4.3: Step response of a second-order underdamped system (τ = time constant, $t_{\rm p}$ = peak time, $t_{\rm r}$ = rise time, $M_{\rm p}$ = overshoot).



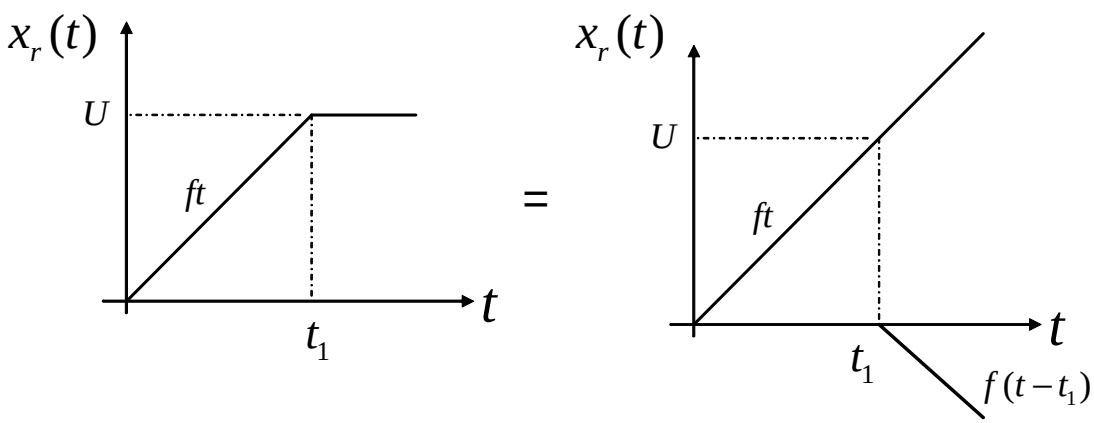


Figure 4.4: Ramp input followed by a constant excitation.

Example 34 *If the closed loop transfer function of the underdamped plant is* $G_{cl}(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$, *and the position input is* $x_r(t) = ft$ *where* f *is the feed in (counts/s), the response of the system is evaluated as:.*

$$x_r(s) = \mathcal{L}(ft) = \frac{f}{s^2} \rightarrow x_a(s) = G_{cl}(s)x_r(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}\frac{f}{s^2}$$

$$x_a(t) = \mathcal{L}^{-1}x_a(s) = \mathcal{L}^{-1}\left[\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}\frac{f}{s^2}\right]$$

$$= f\left[t - \frac{2\zeta}{\omega_n} + \frac{e^{-\zeta\omega_n t}}{\omega_n\sqrt{1-\zeta^2}}\sin\left(\omega_n\sqrt{1-\zeta^2}t+\theta\right)\right], \theta = \cos^{-1}(2\zeta^2-1)$$

Example 35 *Some input commands may have discontinuity but they can still be modeled in Laplace domain. Let the input be:*

$$x_r(t) = \left.\begin{array}{cc} 0 & t<0 \\ ft & 0\le t\le t_1 \\ U & t>t_1 \end{array}\right\}$$

where $f = U/t_1$. The discontinuous input is represented by two ramp inputs where the second one is applied in the opposite direction after time t_1

$$x_r(t) = ft - f < t - t_1 > \tag{4.14}$$

$$x_r(s) = \frac{f}{s^2} - e^{-st_1}\frac{f}{s^2} = \frac{f}{s^2}(1 - e^{-st_1}) \tag{4.15}$$

The time response of $G_{cl}(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$ can be found as:

$$y(s) = G_{cl}(s)x_r(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\frac{f}{s^2}(1 - e^{-st_1}) \tag{4.16}$$

$$y(t) = \mathcal{L}^{-1}y(s) \tag{4.17}$$

Example 36 *Let the closed loop transfer function of the system is* $G_{cl}(z) = \frac{z^{-2}b_1+z^{-1}b_0}{z^{-2}a_0+z^{-1}a_1+1} = \frac{b_0 z+b_1}{z^2+a_1 z+a_0}$, *and the position input is* $x_r(t) = ft$ *where* f *is the feed in (counts/s).*

$$\begin{aligned}
x_a(k) = G_{cl}(z)x_r(k) &= \frac{z^{-2}b_1 + z^{-1}b_0}{z^{-2}a_0 + z^{-1}a_1 + 1}x_r(k) \\
\left(z^{-2}a_0 + z^{-1}a_1 + 1\right)x_a(k) &= \left(z^{-2}b_1 + z^{-1}b_0\right)x_r(k) \\
a_0x_a(k-2) + a_1x_a(k-1) + x_a(k) &= b_1x_r(k-2) + b_0x_r(k-1) \\
x_a(k) &= -a_0x_a(k-2) - a_1x_a(k-1) + b_1x_r(k-2) + b_0x_r(k-1) \\
\textit{where} \qquad k &= 1, 2, ..., k \quad \textit{are control counters at each sampling period } (T)
\end{aligned}$$

Note that if the Zero Order Hold equivalent of the continuous system is used to obtain the z domain equivalent of the system, $(1 - e^{-sT})/s$ *is already included in the transformation process and should not be added again to the remaining calculations and models. Also, if one uses Euler, Tustin or Forward Approximation instead of Zero Order Hold, again the zero order hold must be dropped from the block diagram. The Euler, Tustin or Forward methods represent the continuous to discrete transformation already, and its reuse will lead to incorrect transfer functions.*

4.2.1 Steady State Error of the Closed Loop Control System

Steady state response of a system can be evaluated by applying final value theorem as follows:

$$x_{a,ss} = \lim_{s\to 0} s \cdot x_a(s) = \lim_{s\to 0} s \cdot G_{cl}(s)x_r(s)$$

Steady state error of a closed system to a given input is found as follows. The steady state position error e is:

$$e_{ss} = e(t) = x_r(t) - x_a(t) \quad \text{when } t \to +\infty$$

From the closed loop transfer function block diagram of the system with a unit feedback is expressed in Laplace domain as,

$$\begin{aligned}
e(s) &= x_r(s) - x_a(s) = x_r(s) - G_{cl}(s)x_r(s) = [1 - G_{cl}(s)]\, x_r(s) \\
e(s) &= \left[1 - \frac{D(s)G_o(s)}{1 + D(s)G_o(s)}\right] x_r(s) = \left[\frac{1}{1 + D(s)G_o(s)}\right] x_r(s)
\end{aligned}$$

The steady state error of the system can be evaluated from the final value theorem as ($t \to \infty \Leftrightarrow s \to 0$):

$$e_{ss} = \lim_{s\to 0} s \cdot e(s) = \lim_{s\to 0} s \cdot \left[\frac{1}{1 + D(s)G_o(s)}\right] x_r(s)$$

The steady error of the system in discrete time domain is found using similar approach, except it is evaluated in z domain. Using the z transform property: $t = +\infty \rightarrow s = 0 \rightarrow z = 1$ since $z = e^{sT}$

$$e_{ss} = \lim_{t\to\infty} e^*(t) = \lim_{k\to\infty} e^*(kT) = \lim_{z\to 1}(1 - z^{-1})E(z) \text{ Final value theorem}$$
$$e_{ss} = \lim_{z\to 1}(1 - z^{-1})\left[\frac{1}{1 + D(z)G_o(z)}\right] x_r(z)$$

Example 37 *Steady state error of the system to a step response;*

$$x_r(t) = U \qquad \begin{cases} 0 & t < 0 \\ U & t \geq 0 \end{cases} \rightarrow x_r(s) = \frac{U}{s} \qquad x_r(z) = U\frac{1}{1 - z^{-1}} = U\frac{z}{z - 1}$$
$$e_{ss} = \lim_{s\to 0} s \cdot \left[\frac{1}{1 + D(s)G_o(s)}\right]\frac{U}{s} = \frac{U}{1 + D(0)G_o(0)}$$
$$e_{ss} = \lim_{z\to 1}(1 - z^{-1})\left[\frac{1}{1 + D(z)G_o(z)}\right] U\frac{1}{1 - z^{-1}} = \lim_{z\to 1}\left[\frac{1}{1 + D(z)G_o(z)}\right] U$$

where $D(0)G_o(0)$ in s domain or $D(1)G_o(1)$ in z domain is the gain of the open loop transfer function of the system with a unit feedback. The practical interpretation is that the higher the open loop gain is, smaller the steady state error response will be for a step input to the closed loop systems.

Example 38 *Steady state error of the system to a ramp response (i.e. following or tracking error of a feed drive when it travels at constant feed speed f_c);*

$$x_r(t) = f_c t \rightarrow \qquad x_r(s) = \frac{f_c}{s^2} \qquad x_r(z) = f_c\frac{Tz^{-1}}{(1 - z^{-1})^2} = f_c\frac{Tz}{(z - 1)^2}$$

In Laplace Domain

$$e_{ss} = \lim_{s\to 0} s \cdot \left[\frac{1}{1 + D(s)G_o(s)}\right]\frac{f_c}{s^2} = \lim_{s\to 0}\frac{f_c}{s + sD(s)G_o(s)} = \lim_{s\to 0}\frac{f_c}{sD(s)G_o(s)}$$

In z Domain

$$e_{ss} = \lim_{z\to 1}(1 - z^{-1})\left[\frac{1}{1 + D(z)G_o(z)}\right] f_c\frac{Tz^{-1}}{(1 - z^{-1})^2}$$
$$= \lim_{z\to 1}\left[\frac{f_c T}{[(z - 1) + (z - 1)D(z)G_o(z)]}\right] = \lim_{z\to 1}\left[\frac{f_c T}{(z - 1)D(z)G_o(z)}\right]$$

The actual position of the drive will lag e_{ss} units behind the command to the drive, which is inversely proportional to the derivative of the open loop transfer function.

Example 39 *The steady state error of a second order underdamped system to a ramp input:*

$$e(s) = x_r(s) - x_a(s) = x_r(s) - G_{cl}(s)x_r(s) = [1 - G_{cl}(s)]\, x_r(s)$$
$$e(s) = \left[1 - \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right]\frac{f_c}{s^2} = \left[\frac{s^2 + 2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right]\frac{f_c}{s^2}$$
$$e_{ss} = \lim_{s\to 0} s\left[\frac{s^2 + 2\zeta\omega_n s}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right]\frac{f_c}{s^2} = f_c\lim_{s\to 0}\left[\frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2}\right] = f_c\frac{2\zeta}{\omega_n}$$

The same result can be obtained from the difference between the input and the steady output of the system,

$$e_{ss} = x_r(t) - x_a(t) = \lim_{t\to\infty}\left\{f_c t - f_c\left[t - \frac{2\zeta}{\omega_n} + \frac{e^{-\zeta\omega_n t}}{\omega_n\sqrt{1 - \zeta^2}}\sin\left(\omega_n\sqrt{1 - \zeta^2}t + \theta\right)\right]\right\} \tag{4.18}$$
$$e_{ss} = f_c\frac{2\zeta}{\omega_n} \tag{4.19}$$

4.3 Discrete Identification of Dynamic Systems

The transfer function of systems can be identified either using impulse, step, and frequency response techniques in the continuous or discrete time domain. However, it is not always trivial to identify the transfer function of systems using standard transient or frequency response measurement techniques. There are also statistical methods which treat the system as a black box, and identify the dynamics by observing only the input and output time history. The statistical identification methods are usually classified as "Time Series Identification", and it has both off-line and on-line versions.

4.3.1 Off-Line Least Squares Estimation

When the process is to be controlled by a computer, it may be easier to identify the transfer function parameters of the system by collecting input and output data at uniform discrete time intervals that are equal to the computer control period.

Consider a general transfer function with d samples delay, with orders nb, and na in the numerator and denominator, respectively (such a system is called ARIMA(na, d, nb) in the literature):

$$G_{\mathrm{o}}(z^{-1}) = \frac{z^{-d}(b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{nb} z^{-nb})}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{na} z^{-na}}. \tag{4.20}$$

The dead time can be easily observed by applying step, pulse, or harmonic input to the system and then measuring the time delay between the input and output on a digital scope. It is usually difficult to guess the order of a transfer function (i.e., d, na, nb) if we don't have any engineering feeling about the process. In such cases, one has to resort to stochastic order identification techniques, which are not covered here. In motion control of mechanical systems however, the orders of drives are usually low, up to about second or third order if we neglect the dynamic characteristics of the structure. Furthermore, a good engineer has to have the basic understanding of the drive design and control process. It is customary to start with a low order transfer function assumption and increase the order if the identification does not yield good output estimates with the measurements.

Let us assume a simple process with the following discrete transfer function:

$$G(z^{-1}) = \frac{y(k)}{u(k)} = \frac{z^{-2}(b_0 + b_1 z^{-1})}{1 + a_1 z^{-1}}, \tag{4.21}$$

which is a first-order process whose output (y) has a two sample period delay in responding to an input (u). By expanding the transfer function, we get

$$(1 + a_1 z^{-1}) y(k) = [z^{-2}(b_0 + b_1 z^{-1})] u(k),$$

$$y(k) = -a_1 y(k-1) + b_0 u(k-2) + b_1 u(k-3), \tag{4.22}$$

or

$$y(k) = \{-y(k-1) \quad u(k-2) \quad u(k-3)\}\{a_1 \quad b_0 \quad b_1\}^T = \phi(k)^T \theta, \tag{4.23}$$

where k is the sampling counter, $\phi(k)^T = \{-y(k-1)\ \ u(k-2)\ \ u(k-3)\}$ is the regression or measurement vector, and $\theta = \{a_1\ \ b_0\ \ b_1\}^T$ is the unknown parameter vector. The three unknown parameters are *estimated* from the N measurements, which are greater than the number of unknowns (i.e., $N \gg 3$). If $\hat{\theta} = \{\hat{a}_1\ \ \hat{b}_0\ \ \hat{b}_1\}$ are the estimated parameters, the error between the estimated and actual measurement at interval k is

$$\epsilon(k) = y(k) - \hat{y}(k) = y(k) - \phi(k)^T \hat{\theta}. \tag{4.24}$$

Assume that a pulse or random input containing a wide range of frequencies is given to the system, and N number of inputs and outputs are collected at fixed time intervals. The following set of expressions can

be written for each measurement:

$$\begin{Bmatrix} \epsilon(3) \\ \epsilon(4) \\ \vdots \\ \epsilon(k) \\ \vdots \\ \epsilon(N) \end{Bmatrix} = \begin{Bmatrix} y(3) \\ y(4) \\ \vdots \\ y(k) \\ \vdots \\ y(N) \end{Bmatrix} - \begin{bmatrix} \{-y(2) & u(1) & u(0)\} \\ \{-y(3) & u(2) & u(1)\} \\ \vdots & \vdots & \vdots \\ \{-y(k-1) & u(k-2) & u(k-3)\} \\ \vdots & \vdots & \vdots \\ \{-y(N-1) & u(N-2) & u(N-3)\} \end{bmatrix} \begin{Bmatrix} a_1 \\ b_0 \\ b_1 \end{Bmatrix}$$

$$= \begin{Bmatrix} y(3) \\ y(4) \\ \vdots \\ y(k) \\ \vdots \\ y(N) \end{Bmatrix} - \begin{bmatrix} \phi^T(3) \\ \phi^T(4) \\ \vdots \\ \phi^T(k) \\ \vdots \\ \phi^T(N) \end{bmatrix} \begin{Bmatrix} a_1 \\ b_0 \\ b_1 \end{Bmatrix}$$

or, in matrix form,

$$\epsilon = Y - \Phi\theta, \tag{4.25}$$

where

$$\begin{aligned} Y &= \{y(3) \quad y(4) \quad \cdots y(k) \quad \cdots y(N)\}^T, \\ \Phi &= \{\phi(3) \quad \phi(4) \quad \cdots \phi(k) \quad \cdots \phi(N)\}^T \end{aligned}$$

The least squares identification method is based on minimizing the sum of the squares of errors evaluated from all measurements. The sum of the squares of errors, or *cost function*, to be minimized is

$$J(\theta, N) = \sum_{k=3}^{N} \epsilon(k)^2 = (Y - \Phi\theta)^T (Y - \Phi\theta) = (Y^T - \theta^T \Phi^T)(Y - \Phi\theta). \tag{4.26}$$

By expanding the expression further, we have

$$J(\theta, N) = Y^T Y - Y^T \Phi\theta - \theta^T \Phi^T Y - \theta^T \Phi^T \Phi\theta.$$

The parameters are obtained by minimizing the error or cost function, that is ($\frac{\partial}{\partial\theta}(\theta^T \Phi^T \Phi\theta) = (\theta^T \Phi^T \Phi)^T + \Phi^T \Phi\theta = 2\Phi^T \Phi\theta$),

$$\begin{aligned} \frac{\partial J(\theta, N)}{\partial\theta} &= +0 - Y^T \Phi - Y^T \Phi + (\theta^T \Phi^T \Phi)^T + \Phi^T \Phi\theta \\ &= -2\Phi^T Y + 2\Phi^T \Phi\theta = 0, \end{aligned}$$

which leads to the estimates of parameters from N measurements:

$$\theta = [\Phi^T \Phi]^{-1} \Phi^T Y. \tag{4.27}$$

The least squares technique presented here is a useful tool for identifying the parameters of a discrete transfer function with any order. The matrix multiplication and inversion can be evaluated by either writing a short computer code or by using popular software tools.

4.3.2 Recursive Parameter Estimation Algorithm

The dynamics of some machines and processes change with time. When the control parameter is tuned to work only for a constant, time invariant process, the system can become unstable if the process (i.e. the parameters of the differential equation) vary with time. The parameters of the system can be estimated in real time, and the control parameters can be adaptively varied at each sample time. Such a control law is called as "adaptive control" and based on recursive identification of time varying process parameters.

An on-line, recursive, regularized constant-trace algorithm is given below:

$$\hat{\theta}(k) = \hat{\theta}(k-1) + a(k)k_0(k)[y(k) - \phi^T(k)\hat{\theta}(k-1)], \tag{4.28}$$

$$k_0(k) = P(k-1)\phi(k)(1 + \phi^T(k)P(k-1)\phi(k))^{-1}, \tag{4.29}$$

$$\bar{P}(k) = \bar{P}(k-1) - a(t)k_0(k)\phi^T(k)P(k-1), \tag{4.30}$$

$$P(k) = c_1 \frac{\bar{P}(k)}{\mathrm{tr}(\bar{P}(k))} + c_2 I, \tag{4.31}$$

$$a(t) = \begin{cases} \bar{a} & \text{if } |y(k) - \phi^T(k)\hat{\theta}(k-1)| > 2\delta, \\ 0 & \text{otherwise,} \end{cases} \tag{4.32}$$

where $\hat{\theta}(k) = [\hat{a}_1 \ \hat{b}_0 \ \hat{b}_1]^T$ is the estimated parameter vector, $\vec{k}_0(k)$ is the estimation gain, $\phi(k) = [-y(k-1) \ u(k-2) \ u(k-3)]^T$ is the regression vector or observation vector, $\bar{P}(k)$ is the auxiliary covariance matrix, $\mathrm{tr}(\bar{P}(k)) = \sum_{i=1}^{5} \bar{P}_{ii}$ is the trace of the auxiliary covariance matrix, and $P(k)$ is the covariance matrix. Here $c_1 > 0$, $c_2 \geq 0$, and δ is an estimate of the magnitude of the tolerable fluctuation of the output of the process or noise.

4.4 Dynamics of a Position Control Servo System

A diagram of a typical three-axis CNC machining center is shown in Figure 4.5 as an example of computer controlled mechatronics system. The CNC machining center consists of mechanical, power electronic, and CNC units. The mechanical unit consists of beds, columns, spindle assembly, and feed drive mechanisms. Spindle and feed drive motors and their servo-amplifiers, high voltage power supply unit, and limit switches are part of the power electronics group. The CNC consists of a computer unit and position and velocity sensors for each drive mechanism. The operator enters the NC program to the CNC unit. The CNC computer processes the data and generates discrete numerical position commands for each feed drive and velocity command for the spindle drive. The numerical commands are converted into signal voltage (± 5 V or ± 10 V) by the CNC unit and sent to servo-amplifiers, which process and amplify them to the high voltage levels required by the drive motors. As the drives move, sensors measure their velocities and positions. The CNC periodically executes digital control laws that maintain the feed speed and tool path at programmed rate with sensor feedback measurements.

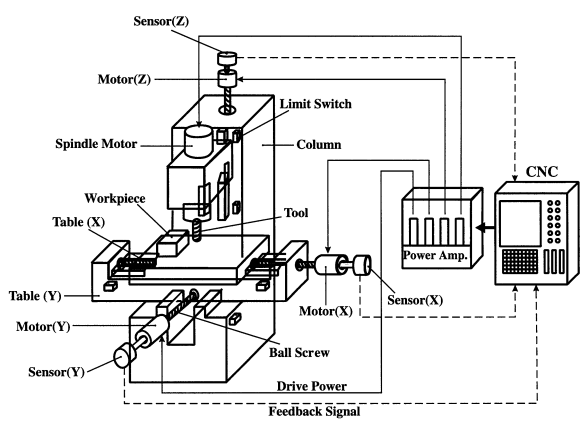


Figure 4.5: Functional diagram of a 3-axis CNC machining center.

The dynamics of open or closed loop systems can be modeled in transfer function or state space model forms. The concepts are explained here by using a real, ball screw driven positioning system shown in Figure

4.6. The system has two parts: physical components and the computer which controls the system behavior. The physical components are modeled in continuous time domain and it consists of table, ball-nut, lead-screw, motor shaft, sensors and amplifier circuit dynamics. The position control algorithm resides in the computer, and the commands are passed to the operational amplifier through Digital to Analog converter.

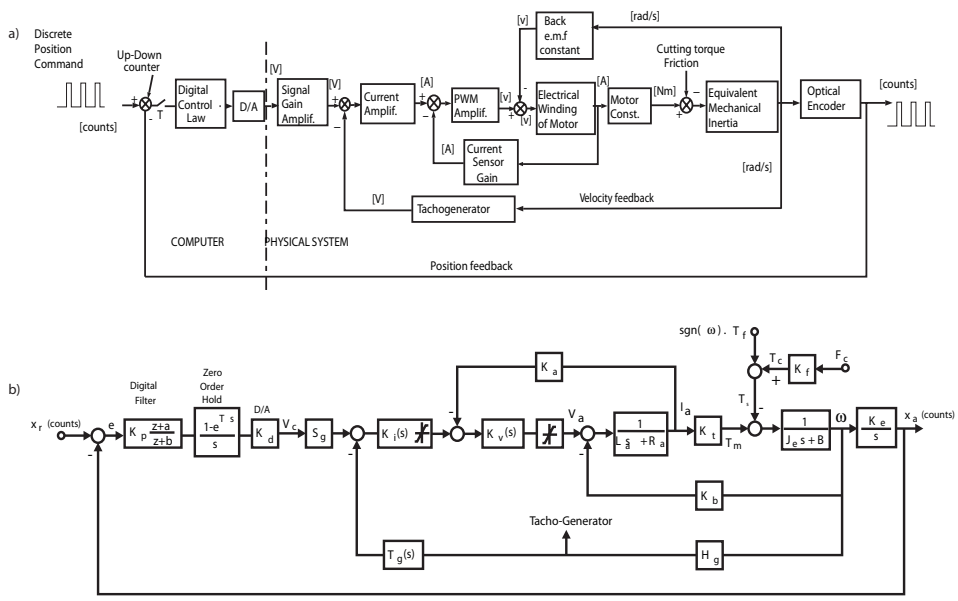


Figure 4.6: Physical and mathematical block diagram of a sample feed drive system.

Let us take one of the feed drives in a machine tool as an example. The feed drive has the following mechanical components: the table with a workpiece, the nut, the ball lead screw, the torque reduction gear set, and the servomotor (see Fig. 4.7). Because of their efficient torque delivery capability at various speeds, the most common servomotors used in the feed drives are direct current (dc) motors. However, alternating current (ac) servomotors have also gained popularity owing to their improved performance. The electrical components of a servomotor system comprise the servomotor amplifier, velocity and position feedback transducers, a digital computer, and a digital to analog converter circuit.

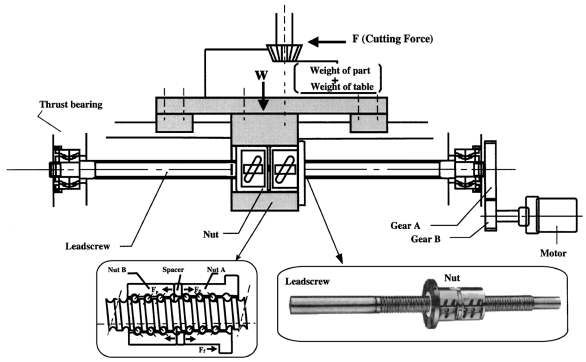


Figure 4.7: Feed drive mechanism with a ball screw drive.

4.4.1 Mechanical Components and Torque Requirements

The feed drive motor has to overcome both the static and dynamic loads in the machine tool. The sources of the static loads are the friction losses in the guideways and bearings and the cutting forces acting in the feeding direction of the table. The motor must deliver a high enough dynamic torque to accelerate the table, workpiece, and leadscrew assembly for a short period of time until the drive reaches the desired steady state speed. The dynamic torque is given as *peak torque* or *peak current* delivery with a period of 2–3 seconds by the servomotor manufacturers. The motors must have a sufficiently high continuous torque delivery range and a sufficient peak torque and delivery period to overcome the static and dynamic loads respectively. Estimation of static and dynamic motor loads are briefly introduced in the following paragraphs.

Static Loads. There are three sources of static loads: the friction in the slideways, frictional losses in the feed drive bearings, and cutting forces.

The friction in the guideways depends on the type of contact between the sliding table and the stationary guideway. High friction coefficients are found at plain lubricated guideways where the metal to metal table–guideway surface contact area is large. The metal contact surface area is reduced in hydrostatic and hydrodynamic guideways where pressurized lubricant is injected between the table and the guideway. The friction coefficient is probably smallest in the guideway designs where roller bearings are used at the guideway–table assembly (see Fig. 4.7). The ball screw's mean diameter is d_p, pitch length is h_p, and screw pitch angle is β.

The force (F_a) acting along the axis of the lead screw is transmitted as a torque (T_a) as follows (4.8):

$$\tan\beta = \frac{h_p}{\pi d_p} = \frac{F_t}{F_a} \rightarrow F_t = F_a \frac{h_p}{\pi d_p}$$

where F_t is the tangential force acting on the rotating screw which creates the torque T_a :

$$T_a = F_t \frac{d_p}{2} = F_a \frac{h_p}{\pi d_p} \frac{d_p}{2} = F_a \frac{h_p}{2\pi}$$

Transmitted torque can also be evaluated from the equilibrium of work done by the linear and rotational motions during one revolution of the screw as follows:

$$F_a h_p \equiv T_a 2\pi \rightarrow T_a = F_a \frac{h_p}{2\pi}$$

The torque reflected on the feed drive motor due to friction (T_{gf}) in the guideways can be estimated as

$$T_{\mathrm{gf}} = \frac{h_{\mathrm{p}}}{2\pi}\, \mu_{\mathrm{gf}}\, [(m_{\mathrm{t}} + m_{\mathrm{w}})g + F_z], \tag{4.33}$$

where μ_{gf} is the friction coefficient on the guideways, m_{t} is the table mass, m_{w} is the workpiece mass, F_z is the normal cutting force on the table, h_{p} is the leadscrew pitch length, and g is the gravitational acceleration ($9.81\,\mathrm{m/s^2}$).

The friction coefficient for plain guideways typically ranges from 0.05 to 0.1 and the vertical cutting force (F_z) can be taken as 10% of the maximum resultant cutting force in a typical vertical milling machine tool.

Axial thrust bearings are used at both ends of the leadscrew in order to absorb the feed forces and also to guide the screw radially at the same time. Axial thrust bearings are pre-loaded in tension to offset backlash produced by the thermal expansion of the leadscrew due to friction in the feed drive assembly. In addition to preload, the axial thrust bearings are loaded by the feed forces. The feed forces can be estimated using the cutting mechanics relationships given in the metal cutting chapter. The torque lost in the bearings and preload is estimated as

$$T_{\mathrm{lf}} = \mu_{\mathrm{b}}\, \frac{d_{\mathrm{b}}}{2}\, (F_{\mathrm{f}} + F_{\mathrm{p}}), \tag{4.34}$$

where μ_{b} is the friction coefficient on bearings (typically around 0.005), d_{b} is the mean bearing diameter, F_{f} is the maximum feed force on the table, and F_{p} is the preload force.

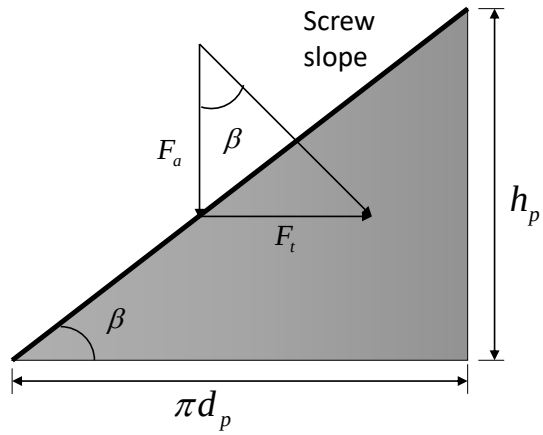


Figure 4.8: Force transmission in ball screw drives.

The torque reflected on the leadscrew shaft due to the cutting forces in the feed direction is given as

$$T_{\mathrm{f}} = \frac{h_{\mathrm{p}}}{2\pi} F_{\mathrm{f}}. \tag{4.35}$$

The total static disturbance load reflected on the leadscrew shaft (T_{s}) is found by summing the three torque values calculated in Eqs. (4.33), (4.34), and (4.35):

$$T_{\mathrm{s}} = T_{\mathrm{gf}} + T_{l\mathrm{f}} + T_{\mathrm{f}}. \tag{4.36}$$

In the cases where the static torque (T_{s}) is too large, a gear reduction can be applied between the motor shaft and the leadscrew. The gear reduction ratio (r_{g}) is defined as

$$r_{\mathrm{g}} = \frac{z_l}{z_m} = \frac{n_m}{n_l}, \tag{4.37}$$

where z_m is the number of teeth on the motor's gear, z_l is the number of teeth on the feed screw's gear, n_m is the motor's angular velocity (rev/min), and n_l is the feed screw's angular velocity (rev/min).

To reduce the speed, we must have $z_l > z_m$, which gives a gear reduction ratio larger than one (i.e., $r_{\mathrm{g}} > 1$). The reduced reflected torque on the motor's shaft (T_{sr}) is found as

$$T_{\mathrm{sr}} = \frac{T_{\mathrm{s}}}{r_{\mathrm{g}}}. \tag{4.38}$$

The CNC designer must select a dc motor that has a larger continuous torque delivery capacity than the static torque reflected on the motor's shaft.

Dynamic Loads. Machine tools require high acceleration torque during speed changes. The reflected inertia on the motor's shaft consists of the inertia of the table, the workpiece, the leadscrew, the gears, and the motor's shaft. Assume that the table with mass m moved one pitch length (h_p) with a linear table velocity of v and corresponding angular velocity (ω) of screw rotating inertia J. The equilibrium of linear and rotational kinetic energies can be expressed as:

$$\frac{1}{2}mv^2 \equiv \frac{1}{2}J\omega^2$$

The machine moves h_p linear and 2π angular distance in one revolution period of δt seconds, hence $v = h_p/\delta t$ and $\omega = 2\pi/\delta t$. The energy equilibrium leads to the inertia reflected by linearly moved mass (m) as a rotating inertia (J) as:

$$\frac{1}{2}m\left(\frac{h_p}{\delta t}\right)^2 \equiv \frac{1}{2}J\left(\frac{2\pi}{\delta t}\right)^2 \rightarrow J = m\left(\frac{h_p}{2\pi}\right)^2$$

The moment of inertia of the table (m_t) and workpiece (m_w) reflected on the leadscrew shaft is

$$J_\mathrm{tw} = (m_\mathrm{t} + m_\mathrm{w})\left(\frac{h_\mathrm{p}}{2\pi}\right)^2. \tag{4.39}$$

The moment of inertia of the leadscrew with a pitch diameter of d_p is

$$J_l = \frac{1}{2}\, m_l \left(\frac{d_\mathrm{p}}{2}\right)^2, \tag{4.40}$$

where m_l is the mass of the lead screw shaft. The total inertia reflected on the motor's shaft is

$$J_\mathrm{e} = \frac{J_\mathrm{tw} + J_l}{r_\mathrm{g}^2} + J_m, \tag{4.41}$$

where J_m is the inertia of the motor's shaft and $r_\mathrm{g} \geq 1$ is the gear reduction ratio between the feed screw and motor speeds.

There is another friction torque in the drive system that is proportional to the velocity, namely the viscous friction torque. The total dynamic torque required to accelerate the inertia J_e and to overcome viscous friction and the static loads is given as

$$T_\mathrm{d} = J_\mathrm{e}\,\frac{d\omega}{dt} + B\omega + T_\mathrm{sr}, \tag{4.42}$$

where ω is the angular velocity of the motor and B is the viscous friction coefficient. Note that since the cutting is carried out at low feeds, the contribution of cutting forces to the static torque (T_s) does not have to be considered in Eq. (4.42). The peak torque value delivered by the motor must be larger than the dynamic torque calculated from Eq. (4.42). If a gear reduction is used between the motor's shaft and leadscrew, the dynamic torque reflected on the motor's shaft is reduced; see Eqs. (4.41) and (4.42).

4.4.2 Feedback Devices

There are two basic feedback devices in the feed drive control system: position and velocity feedback transducers. Tachometers are used as the velocity transducers and encoders are used as the position feedback transducers in general.

Tachometers

A tachometer is a small permanent magnet dc motor mounted directly on the rear of the servomotor's shaft. The tachometer produces a voltage proportional to the actual velocity of the motor shaft. It has a factory set constant and an adjustable gain that enables tuning of the velocity feedback loop. The transfer function between the actual motor velocity and the tachometer circuit output is given as

$$\frac{V_\mathrm{t}(s)}{\omega(s)} = T_\mathrm{g} \cdot H_\mathrm{g}, \tag{4.43}$$

where $V_\mathrm{t}(s)$ is the output voltage of the tachometer circuit, $\omega(s)$ is the actual angular velocity of the motor shaft, H_g is the tachometer constant, T_g is the adjustable tachometer gain, and s is the Laplace operator.

Encoders

Encoders are used as digital position measurement transducers in servo drives. The encoder is based on the principle of emitting light with photodiodes. The encoder, which can be a disk or linear scale type, contains bands of dark and transparent segments. Light is transmitted from one side of the transparent band to the other side where there is a photodiode receiver. The photodiode gives a logic signal (i.e., a binary code) depending on the number of dark and transparent bands detected at incremental positions of the encoder. Linear encoders are used to measure the actual position of the table, whereas rotary shaft encoders are used to measure the angular position of the motor's shaft. In precision machines, a linear encoder may be used for more accurate measurement of the actual position of the table. However, if there is an unmodeled backlash in the feed drive system, the table-mounted linear encoder may produce a *limit cycle* in the position control servo. Since disk encoders are mounted directly on the motor shaft, they do not feel the backlash and therefore they do not produce a nonlinear limit cycle. Manufacturers of servomotors usually provide tachometer and shaft encoders installed on the rear of the motor's shaft in the factory. Encoder is provided with a *line density* and the type of sensing decoder used. For example, a shaft encoder with 1,000 line density and quadrature sensing decoder circuit gives 4,000 *counts* or impulses per revolution of the shaft. Encoders are simply represented as a gain K_e (counts/rad or counts/mm) in position control loop analysis.

Example 40 *A 1,000 line encoder with quadrature sensing decoder is used as a position feedback transducer in a feed drive control system. The motor is directly connected to the leadscrew, which has a pitch length of 5.08 mm. The encoder gain is calculated as*

$$K_e = \frac{4 \cdot 1000}{2\pi} \quad (counts/rad)$$

or

$$K_e = \frac{5.08}{4000} = 0.00127\,\text{mm}\,(0.00005\,\text{in})/count.$$

Thus, one count signal sent by the encoder corresponds to 0.00127 mm linear movement of the table. Henceforth, count *is used to represent the basic length unit of the position control servo system.*

4.4.3 Electrical Drives

The feed drives can be powered by either electrical motors (step, dc, or ac motors) or hydraulic motors. The type of motor used depends very much on the torque delivery and time response requirements of the machine tool or robot drive.

Hydraulic motors are used when a wide torque range and rapid response are required from a drive system. Rotary hydraulic motors are used in heavy duty industrial robots, lathes, and milling machines. Linear hydraulic motors with piston-displacement units are used in grinding machines, planers, and shapers where reciprocating motion is required. The disadvantages of hydraulic drives include leakage, low efficiency, the sensitivity to dirt in the supply oil, and high maintenance costs.

The use of alternating current (ac) motors is also common in CNC machines. The speed of the ac motor is controlled by varying the frequency of the supply voltage. The fundamental problem has been to design the low cost inverters used to vary the frequency of the supply voltage. However, recent microprocessor technology has enabled the calculation of firing frequencies used in the inverter.

The most common motors used in the feed drives are direct current (dc) motors since they allow a wide range of operating speeds with the sufficiently large torque delivery required by machine tools and robots. In the following section, analysis of a permanent magnet dc motor is explained to provide a detailed understanding of the control of dc motors. However, the analysis and modeling procedures are similar for both dc and ac servomotors.

4.4.4 Permanent Magnet Armature-Controlled dc Motors

An electrical diagram of a permanent magnet dc motor is shown in Figure 4.9. The speed of the dc motor is controlled by feeding a dc voltage V_a to the armature (rotor) of the motor. The dc voltage produces a variable dc current I_a on the armature, which in turn creates a mag5netic field between the rotor and stationary

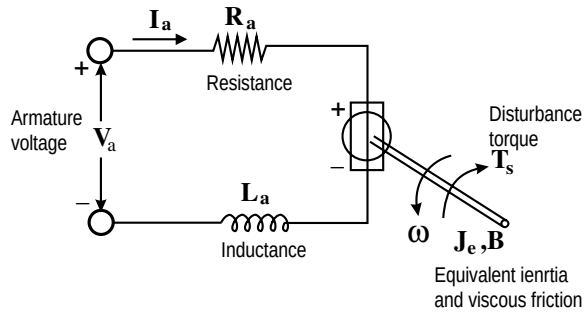


Figure 4.9: Electrical diagram of a dc motor.

stator. Note that the current drawn by the armature can not exceed the maximum current supply capacity of the power amplifier. The current limit is treated as a nonlinearity in the dc motor control system. It is evident that peak current will be drawn for a short period during acceleration and braking of the motor. Peak current and its duration time are given by the amplifier manufacturers. Magnetic flux is constant while the armature voltage is variable in armature-controlled dc motors. The magnetic field produces a torque, which is used for rotating the rotor connected to the motor's shaft. In dc motors, a back electro-motor voltage (e.m.v.) V_{b}, which is proportional to the rotor speed, is produced in the rotor circuit.

The following fundamental dynamic equations govern the motion of dc motors.

The applied armature voltage V_{a} is derived by applying Kirchoff's law to the motor's circuit:

$$V_{\mathrm{a}}(t) = R_{\mathrm{a}} I_{\mathrm{a}}(t) + L_{\mathrm{a}} \frac{dI_{\mathrm{a}}(t)}{dt} + K_{\mathrm{b}} \omega(t), \tag{4.44}$$

where ω is the angular velocity (rad/s), I_{a} is the armature current (A), R_{a} is the armature resistance (Ω), L_{a} is the armature inductance (H), K_{b} is the motor's voltage (e.m.f.) constant (V/(rad/s)), and t is the time (s).

The magnetic field produces a useful motor torque T_m, which is proportional to the armature current I_{a}:

$$T_m(t) = K_{\mathrm{t}}\, I_{\mathrm{a}}(t). \tag{4.45}$$

The useful torque produced by the motor is spent in accelerating the reflected inertia on the motor's shaft, overcoming the friction in the motor's bearings and guideways, and resisting against feed cutting forces and friction loads reflected as a disturbance torque on the motor's shaft. Thus

$$T_m(t) = J_{\mathrm{e}} \frac{d\omega(t)}{dt} + B\, \omega(t) + T_{\mathrm{s}}(t), \tag{4.46}$$

where J_{e} is the reflected inertia on the motor shaft, B is the equivalent friction (viscous damping), and T_{s} is the static disturbance torque reflected on the motor shaft.

Note that the disturbance torque consists of cutting and friction components. The Coulomb friction torque opposes the velocity (ω). The direction of the velocity is considered in the block diagram (Fig. 4.6) using the sign function sgn. In the Eq. (4.46) the viscous friction is assumed to be proportional to the speed. The behavior of the friction torque varies depending on the type of guideways used in the machine tool. The viscous friction torque is linearly proportional to the velocity, and the proportionality constant is B.

Note that Coulomb type sticking friction is usually dominant in feed drives. Coulomb friction demands constant friction torque (or current) and is independent of the feeding velocity variation. The Coulomb friction torque and viscous friction coefficient can be identified from Eq. (4.46) by measuring the armature current at various steady state feeding velocities (i.e., $d\omega/dt = 0$). The dc component of the resulting curve gives the constant current drawn by Coulomb friction, and the slope of the linear component is the viscous damping coefficient (B). To avoid nonlinearity in the system, Coulomb friction is disregarded and an average viscous damping value is assumed.

Taking the Laplace transforms of Eqs. (4.44), (4.45), and (4.46) gives

$$
\begin{aligned}
I_{\mathrm{a}}(s) &= \frac{V_{\mathrm{a}}(s) - K_{\mathrm{b}}\,\omega(s)}{L_{\mathrm{a}}\,s + R_{\mathrm{a}}}, \\
T_m(s) &= K_{\mathrm{t}}\,I_{\mathrm{a}}(s), \\
\omega(s) &= \frac{T_m(s) - T_{\mathrm{s}}(s)}{J_{\mathrm{e}}\,s + B}.
\end{aligned}
\tag{4.47}
$$

These equations have physical interpretations. The transfer function of the armature current is derived using the error voltage as an input. The error voltage is the difference between the supplied reference armature voltage V_{a} and back e.m.v., which acts like a feedback signal. The resulting current produces a useful torque $T_m(s)$. A portion of this torque is spent in overcoming the disturbance torque $T_{\mathrm{s}}(s)$. The remaining torque is used to accelerate the inertia and overcome the viscous friction torque, which is proportional to the velocity. The two important design parameters are

$$
\begin{aligned}
\tau_{\mathrm{e}} &= \frac{L_{\mathrm{a}}}{R_{\mathrm{a}}}, \\
\tau_m &= \frac{J_{\mathrm{e}}}{B},
\end{aligned}
\tag{4.48}
$$

where τ_m is the mechanical time constant and τ_{e} is the electrical time constant of the dc motor. Both time constants are provided by the motor's manufacturer by considering only the inertia of the motor's shaft and friction in the motor's shaft bearings. The mechanical time constant of the dc motor obviously increases when the reflected inertia of the table and leadscrew assembly are taken into account.

4.4.5 dc Motor Power Amplifiers and Velocity Control Loop

The armature voltage is supplied by a power amplifier. The power amplifier receives a large constant dc voltage from a transformer, which converts ac line voltage to the desired dc voltage level. The power amplifier illustrated in this text is a pulse width modulated (PWM), current-controlled amplifier. However, the operation of other types of power amplifiers are quite similar.

The block diagram of a complete velocity control loop is shown in Figure 4.6. The power amplifier receives a velocity command signal V_{c} from the digital to analog converter output of the digital controller. The velocity command signal is first buffered with an adjustable differential preamplifier represented by a gain S_{g}. The output of S_{g} is compared with an actual velocity signal measured by the tachometer feedback unit. The resulting velocity error signal (in volts) is converted to a demand current by the current amplifier, which has a gain of K_{I}. Most of the current amplifiers use an armature current feedback signal to improve the dynamic response of the motor. The feedback current signal is pulled from a current sense coupler and compared with the demand current. The PWM circuit generates a varying saw-tooth shape dc voltage at certain frequency. The PWM frequency is usually above 10 khz. Low frequency PWM switching signals (up to 6 khz) generate irritating audible noise. The current error signal is modulated by the PWM circuit, which is modeled as a gain K_{v}. The resulting dc voltage becomes an ON–OFF type rectangular waveform. The average voltage level of the waveform (dc value) is used as armature voltage V_{a} in the calculations. The complete block diagram of the amplifier, motor, and tachometer feedback unit are shown in Figure 4.6a.

The block diagram is organized with temporary state variables to illustrate the derivation of the velocity loop transfer function. The following relationships can be expressed from the block diagram using temporary states, V_1, V_2, and V_3:

$$
\begin{aligned}
V_1(s) &= S_{\mathrm{g}}V_{\mathrm{c}}(s) - T_{\mathrm{g}}H_{\mathrm{g}}\omega(s), \\
V_2(s) &= K_I V_1(s) - K_{\mathrm{a}}I_{\mathrm{a}}(s) \\
&= K_{\mathrm{I}}S_{\mathrm{g}}V_{\mathrm{c}}(s) - K_{\mathrm{I}}T_{\mathrm{g}}H_{\mathrm{g}}\omega(s) - K_{\mathrm{a}}I_{\mathrm{a}}(s), \\
V_3(s) &= K_{\mathrm{v}}V_2(s) - K_{\mathrm{b}}\omega(s) \\
&= K_{\mathrm{v}}K_{\mathrm{I}}S_{\mathrm{g}}V_{\mathrm{c}}(s) - (K_{\mathrm{v}}K_{\mathrm{I}}T_{\mathrm{g}}H_{\mathrm{g}} + K_{\mathrm{b}})\omega(s) - K_{\mathrm{v}}K_{\mathrm{a}}I_{\mathrm{a}}(s).
\end{aligned}
$$

The transfer function between the current and state V_3 is

$$I_{\mathrm{a}}(s) = \frac{V_3(s)}{L_{\mathrm{a}}s + R_{\mathrm{a}}}.$$

Substituting the value of V_3 into the current expression yields

$$I_{\mathrm{a}}(s) = \frac{K_{\mathrm{v}}K_{\mathrm{I}}S_{\mathrm{g}}}{L_{\mathrm{a}}s + R_{\mathrm{a}} + K_{\mathrm{v}}K_{\mathrm{a}}}V_{\mathrm{c}}(s) - \frac{K_{\mathrm{v}}K_{\mathrm{I}}T_{\mathrm{g}}H_{\mathrm{g}} + K_{\mathrm{b}}}{L_{\mathrm{a}}s + R_{\mathrm{a}} + K_{\mathrm{v}}K_{\mathrm{a}}}\omega(s). \tag{4.49}$$

The motor's mechanical transfer function (see Eq. 4.47) is

$$\omega(s) = \frac{T_m(s) - T_{\mathrm{s}}(s)}{J_{\mathrm{e}}s + B}$$

or

$$\omega(s) = \frac{K_{\mathrm{t}}}{J_{\mathrm{e}}s + B}I_{\mathrm{a}}(s) - \frac{1}{J_{\mathrm{e}}s + B}T_{\mathrm{s}}(s). \tag{4.50}$$

Substituting the current expression (4.49) into Eq. (4.50) yields the transfer function between the output velocity ω and the velocity command input voltage V_{c} and the disturbance torque T_{d}:

$$\omega(s) = \frac{K_1}{s^2 + K_2\, s + K_3}V_{\mathrm{c}}(s) - \frac{(1/J_{\mathrm{e}})[s + (R_{\mathrm{a}} + K_{\mathrm{v}}K_{\mathrm{a}})/L_{\mathrm{a}}]}{s^2 + K_2\, s + K_3}T_{\mathrm{s}}(s), \tag{4.51}$$

where

$$K_1 = \frac{K_{\mathrm{t}}S_{\mathrm{g}}K_{\mathrm{I}}K_{\mathrm{v}}}{L_{\mathrm{a}}J_{\mathrm{e}}},$$

$$K_2 = \frac{B}{J_{\mathrm{E}}} + \frac{R_{\mathrm{a}} + K_{\mathrm{v}}K_{\mathrm{a}}}{L_{\mathrm{a}}},$$

$$K_3 = \frac{B\,(R_{\mathrm{a}} + K_{\mathrm{v}}K_{\mathrm{a}}) + K_{\mathrm{t}}(K_{\mathrm{b}} + H_{\mathrm{g}}T_{\mathrm{g}}K_{\mathrm{v}}K_{\mathrm{I}})}{J_{\mathrm{e}}\,L_{\mathrm{a}}}$$

The feed drive servo velocity controller is designed to have a fast rise time with zero overshoot at step changes in the velocity. Let us analyze the velocity loop as a function of the velocity command input voltage V_{c}. The transfer function (Eq. 4.51) can be expressed as

$$G_v(s) = \frac{\omega(s)}{V_{\mathrm{c}}(s)} = \frac{K_1}{s^2 + 2\xi\omega_{\mathrm{n}}s + \omega_{\mathrm{n}}^2}. \tag{4.52}$$

Here the natural frequency (w_{n}) and damping ratio (ξ) of the velocity loop are defined as follows:

$$\begin{aligned} \omega_{\mathrm{n}} &= \sqrt{K_3} \quad [\mathrm{rad/s}], \\ \xi &= \frac{K_2}{2\sqrt{K_3}} < 1, \end{aligned} \tag{4.53}$$

where $K_1, K_2 > 0$. The step response of this underdamped velocity servo is expressed as

$$\omega(t) = V_{\mathrm{c}} \cdot \frac{K_1}{K_3}\left[1 - e^{-\xi\omega_{\mathrm{n}}t}\left(\cos\omega_{\mathrm{d}}t + \frac{\xi}{\sqrt{1-\xi^2}}\sin\omega_{\mathrm{d}}t\right)\right] \tag{4.54}$$

$$= V_{\mathrm{c}} \cdot \frac{K_1}{K_3}\left[1 - \frac{e^{-\xi\omega_{\mathrm{n}}t}}{\sqrt{1-\xi^2}} \cdot \sin(\omega_{\mathrm{d}}t + \phi)\right] \tag{4.55}$$

where damped natural frequency ω_{d} and phase shift ϕ are defined as

$$\omega_{\mathrm{d}} = \omega_{\mathrm{n}}\sqrt{1-\xi^2},$$

$$\phi = \tan^{-1}\left(\frac{\sqrt{1-\xi^2}}{\xi}\right).$$

The variable gains of the amplifier (i.e., $S_{\mathrm{g}}, T_{\mathrm{g}}, K_{\mathrm{I}}, K_{\mathrm{v}}$) are tuned to have a desired velocity loop gain and step response characteristics as shown in Figure 4.10.
The steady state, desired velocity is

$$\omega_{ss} = \omega(t \to \infty) = V_{\mathrm{c}} \cdot \frac{K_1}{K_3} \left[1 - \frac{e^{-\xi\omega_{\mathrm{n}} t}}{\sqrt{1-\xi^2}} \cdot \sin(\omega_{\mathrm{d}} t + \phi) \right] \Bigg|_{t\to\infty} = V_{\mathrm{c}} \cdot \frac{K_1}{K_3}$$

The steady state velocity can be evaluated in Laplace domain using *final value* theory,

$$\begin{aligned} \omega_{ss} &= \lim_{s\to 0} s\omega(s) = \lim_{s\to 0} sG_v(s)\frac{V_c}{s} \\ &= \lim_{s\to 0} s\frac{K_1}{s^2 + 2\xi\omega_{\mathrm{n}} s + \omega_{\mathrm{n}}^2}\frac{V_c}{s} = V_c \cdot \frac{K_1}{\omega_{\mathrm{n}}^2} = V_c \cdot \frac{K_1}{K_3} \end{aligned}$$

The static gain of the velocity loop is therefore K_1/K_3.

When a unit step input (i.e., $V_{\mathrm{c}} = 1\,\mathrm{V}$) is applied on the amplifier input port, the maximum response of the velocity loop occurs at time t_{p} where the derivative of the velocity is zero (i.e., $d\omega(t)/dt = 0$). Also, the servo will first hit the desired velocity at the rise time t_r where $\omega(t_r) = \omega_{ss}$,

$$V_{\mathrm{c}} \cdot \frac{K_1}{K_3} \left[1 - e^{-\xi\omega_{\mathrm{n}} t_r} \left(\cos\omega_{\mathrm{d}} t_r + \frac{\xi}{\sqrt{1-\xi^2}} \sin\omega_{\mathrm{d}} t_r \right) \right] = V_c \cdot \frac{K_1}{K_3} = \omega_{ss}$$

Since $V_c \cdot \frac{K_1}{K_3} \neq 0$ and $e^{-\xi\omega_{\mathrm{n}} t} \neq 0$, the following function must be zero at $t = t_r$,

$$\cos\omega_{\mathrm{d}} t_r + \frac{\xi}{\sqrt{1-\xi^2}} \sin\omega_{\mathrm{d}} t_r = 0 \to \tan\omega_{\mathrm{d}} t_r = -\frac{\sqrt{1-\xi^2}}{\xi}$$

which leads to:

$$t_r = \frac{1}{\omega_{\mathrm{d}}} \tan^{-1}\left(-\frac{\sqrt{1-\xi^2}}{\xi} \right)$$

The servo will reach to its first peak where the first overshoot occurs at time t_p which can be evaluated from derivative of the response, i.e. $\frac{d\omega(t)}{dt} = 0$ at $t = t_p$. The following expression can be obtained from the time derivative of Eq. (4.54):

$$\begin{aligned} \frac{d\omega(t)}{dt} &= \frac{d}{dt}\left[1 - \frac{e^{-\xi\omega_{\mathrm{n}} t}}{\sqrt{1-\xi^2}} \cdot \sin(\omega_{\mathrm{d}} t + \phi) \right] = \frac{\omega_d e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \cos(\omega_{\mathrm{d}} t + \phi) - \frac{\xi\omega_n e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \sin(\omega_{\mathrm{d}} t + \phi) = 0 \\ &= \omega_d \cos(\omega_{\mathrm{d}} t + \phi) - \xi\omega_n \sin(\omega_{\mathrm{d}} t + \phi) = \sqrt{1-\xi^2} \cos(\omega_{\mathrm{d}} t + \phi) - \xi \sin(\omega_{\mathrm{d}} t + \phi) = 0 \\ &= \sqrt{1-\xi^2} \cos\omega_{\mathrm{d}} t \cos\phi - \sqrt{1-\xi^2} \sin\omega_{\mathrm{d}} t \sin\phi - \xi \sin\omega_{\mathrm{d}} t \cos\phi - \xi \cos\omega_{\mathrm{d}} t \sin\phi \end{aligned}$$

Substituting the following relationship

$$\tan\phi = \frac{\sin\phi}{\cos\phi} = \frac{\sqrt{1-\xi^2}}{\xi} \to \xi \sin\phi = \sqrt{1-\xi^2} \cos\phi$$

into the derivative expression $d\omega(t)/dt$

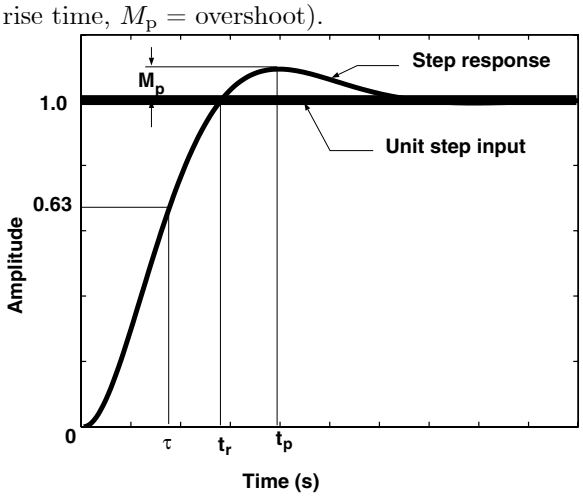
$$\begin{aligned} \xi \sin\phi \cos\omega_{\mathrm{d}} t - \sqrt{1-\xi^2} \sin\phi \sin\omega_{\mathrm{d}} t - \xi \sin\omega_{\mathrm{d}} t \cos\phi - \xi \sin\phi \cos\omega_{\mathrm{d}} t &= 0 \\ \sin\omega_{\mathrm{d}} t \left(\sqrt{1-\xi^2} \sin\phi + \xi \cos\phi \right) &= 0 \\ \sin\omega_{\mathrm{d}} t_p = 0 \to \omega_{\mathrm{d}} t_p &= 0, \pi, 2\pi, .. \end{aligned}$$

$$\sin\omega_{\rm d}t_p = 0 \rightarrow \omega_{\rm d}t_p = 0, \pi, 2\pi, .. \rightarrow t_{\rm p} = \frac{\pi}{\omega_{\rm d}}. \tag{4.56}$$

Alternatively, the peak time can be evaluated directly in Laplace domain as follows

$$\begin{aligned}
\mathcal{L}\frac{d\omega(t)}{dt} &= s\left[\frac{K_1}{s^2+2\xi\omega_{\rm n}s+\omega_{\rm n}^2}\frac{V_c}{s}\right] = V_c\frac{K_1}{s^2+2\xi\omega_{\rm n}s+\omega_{\rm n}^2} = \frac{V_cK_1}{\omega_{\rm n}^2}\frac{\omega_{\rm n}^2}{s^2+2\xi\omega_{\rm n}s+\omega_{\rm n}^2}\\
\frac{d\omega(t)}{dt} &= \frac{V_cK_1}{\omega_{\rm n}^2}\mathcal{L}^{-1}\frac{\omega_{\rm n}^2}{s^2+2\xi\omega_{\rm n}s+\omega_{\rm n}^2} = \frac{V_cK_1}{\omega_{\rm n}^2}\frac{\omega_{\rm n}}{\sqrt{1-\xi^2}}e^{-\xi\omega_{\rm n}t}\sin\omega_{\rm d}t = 0\\
\sin\omega_{\rm d}t_p &= 0 \rightarrow \omega_{\rm d}t_p = 0, \pi, 2\pi,
\end{aligned}$$

Figure 4.10: Step response of a second-order underdamped system (τ = time constant, $t_{\rm p}$ = peak time, $t_{\rm r}$ = rise time, $M_{\rm p}$ = overshoot).



By substituting peak time (t_p) into the step response $\omega(t \rightarrow t_p)$ and $t_p = \frac{\pi}{\omega_{\rm d}}$, the maximum overshoot (M_p) can be evaluated as,

$$M_p = \omega(t_p) - V_c\frac{K_1}{K_3} = V_c\cdot\frac{K_1}{K_3}\left[1 - e^{-\xi\omega_{\rm n}t_p}\left(\cos\omega_{\rm d}t_p + \frac{\xi}{\sqrt{1-\xi^2}}\sin\omega_{\rm d}t_p\right)\right] - V_c\frac{K_1}{K_3} \tag{4.57}$$

$$= -V_c\cdot\frac{K_1}{K_3}e^{-\zeta\omega_{\rm n}\frac{\pi}{\omega_{\rm d}}}\left(\cos\pi + \frac{\zeta}{\sqrt{1-\zeta^2}}\cdot\sin\pi\right) = V_c\cdot\frac{K_1}{K_3}e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \tag{4.58}$$

If the overshoot amplitude is given for a unit step input ($V_c\cdot\frac{K_1}{K_3} = 1$), the damping can be evaluated from the following equation,

$$\ln M_p = -\frac{\zeta\pi}{\sqrt{1-\zeta^2}} \tag{4.59}$$

Typical design values for a feed drive servo may be a damping ratio of $\xi = 0.707$ and a peak time of $t_{\rm p} = 10\,{\rm ms}$. The natural frequency $\omega_{\rm n}$ can be estimated from Eq. (4.56):

$$\begin{aligned}
\omega_{\rm n} &= \frac{\pi}{t_{\rm p}\sqrt{1-\xi^2}}\\
&= 554\ {\rm rad/s} = 88{\rm Hz}.
\end{aligned}$$

The corresponding servo parameters are identified by substituting the values of ξ and $\omega_{\rm n}$ into Eq. (4.53). It is fairly obvious that one can not demand a higher natural frequency than the maximum capacities of the motor and amplifier gains can provide.

4.4.6 Position Control Loop

The position loop consists of an up–down counter, an encoder, a digital compensation filter, and a digital to analog converter.

Up–Down Counter

The up–down counter register receives both command and measured position counts. The position commands increment while the encoder feedback counts decrement the contents of the up–down counter. The instantaneous content of the counter represents the accumulated or integrated position error within digital servo control interval T. The block diagram representation of the up–down counter is

$$\frac{X_{\mathrm{a}}(s)}{\omega(s)} = \frac{K_{\mathrm{e}}}{s}. \tag{4.60}$$

4.4.7 Digital Compensation Filter

The contents of the up–down counter, the position error, is sampled every T seconds. The discrete position error $E(k)$ is the difference between the reference and actual position of the table:

$$E(k) = X_{\mathrm{r}}(k) - X_{\mathrm{a}}(k), \tag{4.61}$$

where $X_{\mathrm{r}}(k)$ is the discrete reference position, $X_{\mathrm{a}}(k)$ is the discrete actual position, and z is the discrete forward time shift operator.

The error is passed from a digital filter, which has a typical transfer function form of

$$D(z) = K_{\mathrm{p}}\,\frac{z+a}{z+b}, \tag{4.62}$$

where K_{p} is the position control filter gain, a is the filter's zero, and b is the filter's pole.

The digital filter is programmable and resides in the servo motion control computer. The filter's parameters are tuned to provide a desired transient response of the position control system.

4.4.8 Digital to Analog (D/A) Converter

The motion control computer sends the digital filter's output to a D/A converter circuit of the motion control computer. The D/A converter is modeled as a zero order hold (ZOH) and a gain K_{d}. The D/A gain is found as

$$K_{\mathrm{d}} = \frac{\text{voltage range of the D/A chip}}{2^{nb}}, \tag{4.63}$$

where nb is the number of bits employed by the D/A chip in converting a binary number to analog voltage. For example; a 12 bit D/A converter chip with $\pm 10\,\mathrm{V}$ voltage range has a gain of

$$K_{\mathrm{d}} = \frac{20\,\mathrm{V}}{2^{12}} = 0.00488\ (\mathrm{V/count}).$$

4.5 Transfer Function of the Position Loop

The block diagram of the complete position control system can be organized as shown in Figure 4.6. The system has continuous and discrete components. The continuous part of the system's transfer function is represented in Laplace domain as

$$G_{\mathrm{c}}(s) = \frac{K_1}{s^2 + K_2\,s + K_3}\,\frac{K_{\mathrm{e}}}{s} = \frac{K_1}{s^2 + 2\xi\omega_{\mathrm{n}}\,s + \omega_{\mathrm{n}}^2}\,\frac{K_{\mathrm{e}}}{s} \tag{4.64}$$

The velocity control signal V_{c} of the digital motion control unit is applied to the power amplifier at T second intervals via a D/A converter with a gain of K_{d}. The zero-order hold equivalent of the $G_{\mathrm{c}}(s)$ for one millisecond sampling interval is

$$G_c(z) = K_d \left(1 - z^{-1}\right) Z \left[\frac{G_c(s)}{s}\right]$$

which, after taking the z transform, becomes

$$G_{\mathrm{c}}(z) = \frac{K_{\mathrm{d}} K_1 K_{\mathrm{e}}}{K_3} \frac{b_2 z^2 + b_1 z + b_0}{(z-1)(z^2 + a_1 z + a_0)}, \tag{4.65}$$

where the forward operator is equivalent to $z = e^{sT}$. The derived expressions for the parameters of $G_{\mathrm{c}}(z)$ are given as

$$\begin{aligned}
b_2 &= T - \frac{1}{\omega_{\mathrm{d}}}\, e^{-\xi\omega_{\mathrm{n}}T}\, \sin(\omega_{\mathrm{d}}T) \\
&- \frac{K_2}{K_3} \left\{1 - e^{-\xi\omega_{\mathrm{n}}T} \left[\frac{\xi\omega_{\mathrm{n}}}{\omega_d}\, \sin(\omega_{\mathrm{d}}T) + \cos(\omega_{\mathrm{d}})\right]\right\}, \\
b_1 &= 2\, e^{-\xi\omega_{\mathrm{n}}T} \left[\frac{\sin(\omega_{\mathrm{d}}T)}{\omega_{\mathrm{d}}} - T\, \cos(\omega_{\mathrm{d}}T)\right] \\
&+ \frac{K_2}{K_3}(1 - e^{-2\xi\omega_{\mathrm{n}}T}) - 2\, e^{-\xi\omega_{\mathrm{n}}T}\, \sin(\omega_{\mathrm{d}}T)\, \frac{K_2}{K_3}\, \frac{\xi\omega_{\mathrm{n}}}{\omega_{\mathrm{d}}}, \\
b_0 &= Te^{-2\xi\omega_{\mathrm{n}}T} - \frac{1}{\omega_{\mathrm{d}}}\, e^{-\xi\omega_{\mathrm{n}}T}\, \sin(\omega_{\mathrm{d}}T) \\
&+ \frac{K_2}{K_3} \left\{e^{-2\xi\omega_{\mathrm{n}}T} + e^{-\xi\omega_{\mathrm{n}}T} \left[\frac{\xi\omega_{\mathrm{n}}}{\omega_{\mathrm{d}}}\, \sin(\omega_{\mathrm{d}}T) - \cos(\omega_{\mathrm{d}}T)\right]\right\}, \\
a_1 &= -2\, e^{-\xi\omega_{\mathrm{n}}T}\, \cos(\omega_{\mathrm{d}}T), \\
a_0 &= e^{-2\xi\omega_{\mathrm{n}}T}.
\end{aligned}$$

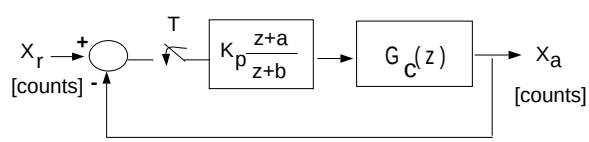


Figure 4.11: Discrete time position control loop.

The reduced block diagram of the equivalent discrete position control system is shown in Figure 4.11. The closed-loop transfer function of the complete feed drive control system is derived as follows:

$$G_{\mathrm{cl}}(z) = \frac{X_{\mathrm{a}}(k)}{X_k(z)} = \frac{D(z)\, G_{\mathrm{c}}(z)}{1 + D(z) G_{\mathrm{c}}(z)},$$

or

$$G_{\mathrm{cl}}(z) = K_{\mathrm{cl}}\, \frac{z^3 + \beta_2 z^2 + \beta_1 z + \beta_0}{z^4 + \alpha_3 z^3 + \alpha_2 z^2 + \alpha_1 z + \alpha_0}. \tag{4.66}$$

The parameters of the closed-loop transfer function $G_{\mathrm{cl}}(z)$ are given as follows:

$$
\begin{aligned}
\beta_2 &= \frac{b_1 + a\, b_2}{b_2}, \\
\beta_1 &= \frac{b_0 + a\, b_1}{b_2}, \\
\beta_0 &= \frac{a\, b_0}{b_2}, \\
\alpha_3 &= \frac{K_1\, K_{\mathrm{e}}\, K_{\mathrm{d}}\, K_{\mathrm{p}}}{K_3}\, b_2 + b + a_1 - 1, \\
\alpha_2 &= \frac{K_1\, K_{\mathrm{e}}\, K_{\mathrm{d}}\, K_{\mathrm{p}}}{K_3}\, (b_1 + a\, b_2) + b(a_1 - 1) + a_0 - a_1, \\
\alpha_1 &= \frac{K_1\, K_{\mathrm{e}}\, K_{\mathrm{d}}\, K_{\mathrm{p}}}{K_3} (b_0 + a\, b_1) + b(a_0 - a_1) - a_0, \\
\alpha_0 &= \frac{K_1\, K_{\mathrm{e}}\, K_{\mathrm{d}}\, K_{\mathrm{p}}}{K_3} \cdot a \cdot b_0 - b a_0, \\
K_{\mathrm{cl}} &= \frac{K_1\, K_{\mathrm{e}}\, K_{\mathrm{d}}\, K_{\mathrm{p}}\, b_2}{K_3}.
\end{aligned}
$$

The transfer function (Eq. 4.66) can be expressed using the backward time shift operator z^{-1} by multiplying both numerator and denominator by z^{-4}. This gives

$$
G_{\mathrm{cl}}(z^{-1}) = K_{\mathrm{cl}}\, \frac{z^{-1} + \beta_2 z^{-2} + \beta_1 z^{-3} + \beta_0 z^{-4}}{1 + \alpha_3 z^{-1} + \alpha_2 z^{-2} + \alpha_1 z^{-3} + \alpha_0 z^{-4}} \tag{4.67}
$$

Note that the z^{-1} backward time shift operator acts on the discrete signal as

$$
z^{-1} x(kT) = x[(k-1)T],
$$

where T is the discrete sampling time interval and $x(kT)$ is the discrete value of x at sampling interval number k.

For any given discrete time input $X_{\mathrm{r}}(kT)$, the position response $X_{\mathrm{a}}(kT)$ can be found by rearranging the transfer function $G_{\mathrm{cl}}(z^{-1})$:

$$
\begin{aligned}
X_{\mathrm{a}}(k) = &-(\alpha_3 z^{-1} + \alpha_2 z^{-2} + \alpha_1 z^{-3} + \alpha_0 z^{-4}) X_{\mathrm{a}}(z^{-1}) \\
&+ K_{\mathrm{cl}} \cdot (z^{-1} + \beta_2 z^{-2} + \beta_1 z^{-3} + \beta_0 z^{-4}) X_{\mathrm{r}}(z^{-1}).
\end{aligned} \tag{4.68}
$$

The discrete time response of the position servo can be simulated by taking the inverse z^{-1} transform of Eq. (4.68). The resulting difference equation contains the discrete history of input and the past history of output position values.

Note that the position control loop can be analyzed in the continuous time (s) domain as an alternative to the discrete time (z) domain analysis. However, this approach requires approximating the digital filter ($D(z)$) in the s domain. One of the widely used approximations is $z = (1 + sT/2)/(1 - sT/2)$, which is called Tustin's bilinear transformation. The corresponding digital filter becomes

$$
D(s) = K'_{\mathrm{p}} \frac{s + a'}{s + b'}, \tag{4.69}
$$

where

$$
K'_{\mathrm{p}} = K_{\mathrm{p}} \frac{1 - a}{1 - b}\,, \quad a' = \frac{2}{T} \frac{1 + a}{1 - a}\,, \quad b' = \frac{2}{T} \frac{1 + b}{1 - b}
$$

or

$$
K_{\mathrm{p}} = K'_{\mathrm{p}} \frac{1 - b'}{1 - a'}\,, \quad a = \frac{Ta' - 2}{Ta' + 2}\,, \quad b = \frac{Tb' - 2}{Tb' + 2}.
$$

In this case, the closed-loop transfer function of the position loop in Laplace domain becomes

$$
G_{\mathrm{cl}}(s) = \frac{D(s) K_{\mathrm{d}} G_{\mathrm{c}}(s)}{1 + D(s) K_{\mathrm{d}} G_{\mathrm{c}}(s)}. \tag{4.70}
$$

4.6 State Space Model of Dynamic Systems

Dynamic systems can be either modeled using transfer functions expressed by the division of output over input in continuous (Laplace) or discrete (z) time domain as explained in the previous sections. Alternatively, the same system dynamics can be expressed by a set of first order coupled differential equations in continuous or difference equations in discrete time domain. The resulting solution will be identical since the same system dynamics are expressed in a different form only.

State space model of the system in continuous time domain is expressed by a set of first order differential equations as follows:

$$\left.\begin{array}{lll} \text{State Equations:} & \left\{\dot{x}(t)\right\} = & [A]\{x(t)\} + [B]\{u(t)\} \\ \text{Output Equations:} & \{y(t)\} = & [C]\{x(t)\} + [D]\{u(t)\} \end{array}\right\} \tag{4.71}$$

where

$\{x\}$	:	(n x 1) State vector containing system variables
$\{y\}$	:	(q x 1) Output vector containing variables to be observed
$\{u\}$	:	(p x 1) Input vector containing external inputs to the system
$[A]$	:	(n x n) State transition matrix containing system dynamics
$[B]$	:	(n x p) Input Transition matrix which relate the input to the system dynamics
$[C]$	:	(q x n) Output transition matrix
$[D]$	:	(q x p) Input transmission matrix

When $[A]$ has non -zero off diagonal terms, the first order differential equations will be coupled and can be solved using eigenvalue solution. If we neglect the input terms $(u(t))$, the homogeneous state equation can be solved as:

$$\begin{aligned} \frac{dx(t)}{dt} &= Ax(t), \rightarrow sX(s) - x(0) = AX(s) \\ x(t) &= \mathcal{L}^{-1}\left[(sI-A)^{-1}\right]x(0), \rightarrow \mathcal{L}^{-1}\left[(sI-A)^{-1}\right] = e^{At} \\ \phi(t) &= e^{At} = I + At + \frac{1}{2!}A^2t^2 + \frac{1}{3!}A^3t^3 + \cdots \end{aligned}$$

Note that the solution can be proven easily by : $x(t) = e^{At}x(0) \rightarrow \frac{dx(t)}{dt} = Ae^{At}x(0) = Ax(t)$. The characteristic equation of the system is independent of any input variable, and it is given by $\det\left|(sI-A)^{-1}\right|$ and its roots give the poles (p_i) of the system.

$$\det\left|(sI-A)^{-1}\right| = 0 = (s-p_1)(s-p_2)\cdots(s-p_n)$$

Hence the denominator of output/input transfer function model is identical to the characteristic equation $\det\left|(sI-A)^{-1}\right|$ in state space model. Now, taking the Laplace transform of state space equations (Eq. 4.71) :

$$\begin{aligned} sX(s) - x(0) &= AX(s) + BU(s) \\ X(s) &= (sI-A)^{-1}x(0) + (sI-A)^{-1}BU(s) \\ x(t) &= \mathcal{L}^{-1}\left[(sI-A)^{-1}\right]x(0) + \mathcal{L}^{-1}\left\{(sI-A)^{-1}\left[BU(s)\right]\right\} \\ x(t) &= e^{At}x(0) + \int_0^t \left[e^{A(t-\tau)}Bu(\tau)\right]d\tau \end{aligned}$$

Discrete equivalent of State Space solution can be directly expressed as follows:

$$\phi(T) = e^{AT} = I + AT + \frac{1}{2!}A^2T^2 + \frac{1}{3!}A^3T^3 + \cdot\cdot$$
$$H(T) = \left[\int_0^T e^{A\tau}d\tau\right] \cdot B$$
$$x(k+1) = \phi(T)x(k) + H(T) \cdot u(k) \tag{4.72}$$

where k is the discrete time interval (T) counter. If the sampling period T is small enough, the first two or three terms in Taylor series expansion of e^{AT} is sufficient.

Proof. Equation 4.72 can be proven as follows. Let $t = kT$ where k is the discrete sampling interval counters. The system is sampled at a constant time interval T and input is fed to a zero hold device. Let $kT < \tau < (k+1)T$ and initial condition of the state $x(0) = x(t = t_0)$

$$x(t) = e^{A(t-t_0)}x(0) + \int_{t_0}^{t} \left[e^{A(t-\tau)}Bu(\tau)\right] d\tau$$

Since the state at the sampling intervals need to be evaluated, let $t_0 = kT$ and $t = (k+1)T$, and $u(\tau) = u(kT)$

$$x(kT) = e^{AT}x(kT) + \int_{kT}^{(k+1)T} \left[e^{A[(k+1)T-\tau]}Bu(kT)\right] d\tau$$

Let $\lambda = (k+1)T - \tau \rightarrow d\tau = -d\lambda$.The integral boundaries become ($\tau = kT \rightarrow \lambda = T$); ($\tau = (k+1)T \rightarrow \lambda = 0$).By rewriting the integral equation with a new variable,

$$x(kT) = e^{AT}x(kT) - \int_T^0 \left[e^{A\lambda}B\right] d\lambda \cdot u(kT) = e^{AT}x(kT) + \int_0^T \left[e^{A\lambda}B\right] d\lambda \cdot u(kT)$$
$$\phi(T) = e^{AT}, \qquad H(T) = \int_0^T \left[e^{A\lambda}B\right] d\lambda$$

■

Example 41 *State space model of a mass, spring, damper system*

Consider a single degree of freedom structure with a mass (m), spring constant (k), and damping constant (c) with an external force, i.e. input, ($F(t)$) applied to the mass. The vibration is $x(t)$. The equation of motion is:

$$m\frac{d^2x(t)}{dt^2} = F(t) - c\frac{dx(t)}{dt} - kx(t) \rightarrow \frac{d^2x(t)}{dt^2} = \frac{1}{m}F(t) - \frac{c}{m}\frac{dx(t)}{dt} - \frac{k}{m}x(t)$$

Or using a conventional vibration structure ($\omega_n^2 = k/m$, $1/m = \omega_n^2/k$, $2\zeta\omega_n = c/m$), the equation of motion is expressed as:

$$\frac{d^2x(t)}{dt^2} = \frac{\omega_n^2}{k}F(t) - 2\zeta\omega_n\frac{dx(t)}{dt} - \omega_n^2x(t) \tag{4.73}$$

Laplace transform of Eq. 4.73 with zero initial conditions leads to the following transfer function with a characteristic equation $s^2 + 2\zeta\omega_n s + \omega_n^2$.,

$$\begin{aligned} s^2x(s) &= \frac{\omega_n^2}{k}F(t) - 2\zeta\omega_n sx(s) - \omega_n^2x(s) \\ \frac{x(s)}{F(s)} &= \frac{\omega_n^2/k}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{aligned}$$

The equation of motion (Eq. 4.73), which is described by a second order differential equation, can be expressed by two states as:

$$\begin{aligned} x_1(t) &= x(t) && \text{vibration} \\ x_2(t) &= \frac{dx_1(t)}{dt} = \frac{dx(t)}{dt} && \text{vibration velocity} \end{aligned}$$

By substituting states x_1 and x_2 into Eq. 4.73

$$\begin{aligned} \frac{dx_1(t)}{dt} &= x_2(t) \\ \frac{dx_2(t)}{dt} &= -\omega_n^2 x_1(t) - 2\zeta\omega_n x_2(t) + \frac{\omega_n^2}{k} F(t) \end{aligned}$$

In a matrix form:

$$\left\{ \begin{array}{c} \frac{dx_1(t)}{dt} \\ \frac{dx_1(t)}{dt} \end{array} \right\} = \left[\begin{array}{cc} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{array} \right] \left\{ \begin{array}{c} x_1(t) \\ x_2(t) \end{array} \right\} + \left\{ \begin{array}{c} 0 \\ \frac{\omega_n^2}{k} \end{array} \right\} F(t) \tag{4.74}$$

$$\left\{ \dot{x} \right\} = [A]\{x(t)\} + \{B\} F(t) \tag{4.75}$$

where

$$[A] = \left[\begin{array}{cc} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{array} \right] \ , \ \{B\} = \left\{ \begin{array}{c} 0 \\ \frac{\omega_n^2}{k} \end{array} \right\} \ , \ \{x(t)\} = \left\{ \begin{array}{c} x_1(t) \\ x_2(t) \end{array} \right\}$$

By taking Laplace transform of both sides of equation,

$$\begin{aligned} s\{x(s)\} - \{x(0)\} &= [A]\{x(s)\} + [B]F(s) \\ (s[I] - [A])\{x(s)\} &= \{x(0)\} + [B]F(s) \\ \{x(s)\} &= (s[I] - [A])^{-1}\{x(0)\} + (s[I] - [A])^{-1}[B]F(s) \end{aligned}$$

where the inverse is:

$$\begin{aligned} s[I] - [A] &= s\left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] - \left[\begin{array}{cc} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{array} \right] = \left[\begin{array}{cc} s & -1 \\ \omega_n^2 & s + 2\zeta\omega_n \end{array} \right] \\ (s[I] - [A])^{-1} &= \frac{adj\,(s[I] - [A])}{\det(s[I] - [A])} = \frac{\left[\begin{array}{cc} s + 2\zeta\omega_n & -\omega_n^2 \\ 1 & s \end{array} \right]^T}{(s(s + 2\zeta\omega_n) + \omega_n^2)} = \frac{\left[\begin{array}{cc} s + 2\zeta\omega_n & 1 \\ -\omega_n^2 & s \end{array} \right]}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ (s[I] - [A])^{-1} &= \left[\begin{array}{cc} \frac{s+2\zeta\omega_n}{s^2+2\zeta\omega_n s+\omega_n^2} & \frac{1}{s^2+2\zeta\omega_n s+\omega_n^2} \\ \frac{-\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} & \frac{s}{s^2+2\zeta\omega_n s+\omega_n^2} \end{array} \right] \end{aligned}$$

The continuous time domain response of the system can be evaluated as follows:

$$\{x(t)\} = \mathcal{L}^{-1}\left[(sI - A)^{-1}\right]\{x(0)\} + \mathcal{L}^{-1}\left\{(sI - A)^{-1}[B]F(s)\right\}$$

For example, if the initial displacement is $x(0) = x_0 = x_1(0)$ and the initial velocity is zero $x_2(0) = 0$,

$$\{x(0)\} = \left\{ \begin{array}{c} x_0 \\ 0 \end{array} \right\} \rightarrow F(s) = \frac{F_0}{s}$$

The inverse Laplace transform would lead to the solution of both vibration ($x(t)$) and vibration velocity $\dot{x}(t)$.

Note that the $\det(s[I]-[A]) = s^2 + 2\zeta\omega_n s + \omega_n^2$ which is the characteristic equation in both transfer function of state space domains. The roots of characteristic equation ($p_1 = -\zeta\omega_n + j\omega_d, p_2 = -\zeta\omega_n - j\omega_d$) will lead to the transient response of the system which will diminish over time (i.e. steady state), i.e. $e^{p_1 t} = e^{-\zeta\omega_n t + j\omega_d d}$.

Example 42 *State Space Model of Feed Drive Control Systems*

A state space method is used to compare the analytically modelled drive system with experimental measurements.

The continuous part of the system consists of the velocity control loop (Eq. 4.64) and the up–down counter (Eq. 4.60). Three *states* – the armature current I_a, the angular velocity ω, and the actual position X_a – are derived as follows. Remember that the continuous part of the motion control system was derived as follows:

$$\begin{aligned}
\text{Armature voltage } V_\mathrm{a}(t) &= R_\mathrm{a} I_\mathrm{a}(t) + L_\mathrm{a}\frac{dI_\mathrm{a}(t)}{dt} + K_\mathrm{b}\omega(t) \\
\text{Motor Torque } T_m(t) &= K_\mathrm{a} I_\mathrm{a}(t) = J_\mathrm{e}\frac{d\omega(t)}{dt} + B\,\omega(t) + T_\mathrm{s}(t) \\
\text{Actual position of the table } \frac{dX_\mathrm{a}(t)}{dt} &= K_\mathrm{e}\cdot\omega(t)
\end{aligned}$$

From Figure 4.6, the armature voltage can be expressed as

$$V_\mathrm{a} = K_\mathrm{v}[K_\mathrm{I}(S_\mathrm{g}V_\mathrm{c} - T_\mathrm{g}H_\mathrm{g}\omega) - K_\mathrm{a}I_\mathrm{a}]. \tag{4.76}$$

Substituting Eq. (4.44) into armature voltage equation (4.76) yields

$$\frac{dI_\mathrm{a}}{dt} = -\frac{K_\mathrm{v}K_\mathrm{a} + R_\mathrm{a}}{L_\mathrm{a}} I_\mathrm{a} - \frac{K_\mathrm{b} + K_\mathrm{v}K_\mathrm{I}T_\mathrm{g}H_\mathrm{g}}{L_\mathrm{a}}\omega + \frac{K_\mathrm{v}K_\mathrm{I}S_\mathrm{g}}{L_\mathrm{a}}\, V_\mathrm{c}. \tag{4.77}$$

By substituting current into motor torque as $T_m(t) = K_t I_\mathrm{a}(t)$ gives

$$\frac{d\omega}{dt} = \frac{K_\mathrm{t}}{J_\mathrm{e}}\, I_\mathrm{a} - \frac{B}{J_\mathrm{e}}\,\omega - \frac{1}{J_\mathrm{e}}\, T_\mathrm{s}. \tag{4.78}$$

The inverse Laplace transform of the transfer function of the up–down counter and the encoder (Eq. 4.60) gives

$$\frac{dX_\mathrm{a}(t)}{dt} = K_\mathrm{e}\cdot\omega(t). \tag{4.79}$$

The state equations (4.77–4.79) can be organized in a standard state space form:

$$\dot{x_\mathrm{c}}(t) = A_\mathrm{c}\, x_\mathrm{c}(t) + B_\mathrm{c}\, u_\mathrm{c}(t), \tag{4.80}$$

where the *state vector* ($x_\mathrm{c}(t)$) and the *input vector* ($u_\mathrm{c}(t)$) are defined as

$$x_\mathrm{c}(t) = \begin{bmatrix} I_\mathrm{a}(t) \\ \omega(t) \\ X_\mathrm{a}(t) \end{bmatrix}, \quad u_\mathrm{c}(t) = \begin{bmatrix} V_\mathrm{c}(t) \\ T_\mathrm{s}(t) \end{bmatrix}$$

and A_{rmc} and B_{c} are constant matrices:

$$A_{\mathrm{c}} = \begin{bmatrix} -\dfrac{K_{\mathrm{v}}K_{\mathrm{a}} + R_{\mathrm{a}}}{L_{\mathrm{a}}} & -\dfrac{K_{\mathrm{b}} + K_{\mathrm{v}}K_{\mathrm{I}}T_{\mathrm{g}}H_{\mathrm{g}}}{L_{\mathrm{a}}} & 0 \\ \dfrac{K_{\mathrm{t}}}{J_{\mathrm{e}}} & -\dfrac{B}{J_{\mathrm{e}}} & 0 \\ 0 & K_{\mathrm{e}} & 0 \end{bmatrix},$$

$$B_{\mathrm{c}} = \begin{bmatrix} \dfrac{K_{\mathrm{v}}K_{\mathrm{I}}S_{\mathrm{g}}}{L_{\mathrm{a}}} & 0 \\ 0 & -\dfrac{1}{J_{\mathrm{e}}} \\ 0 & 0 \end{bmatrix}.$$

The state equation (4.80) represents the continuous part of the feed drive servo system, which has the following discrete equivalent solution for an observation interval of T :

$$x_{\mathrm{c}}(k+1) = \Phi(T)\, x_{\mathrm{c}}(k) + H(T)\, u_{\mathrm{c}}(k), \tag{4.81}$$

where the state and input vectors at sampling interval k are defined as

$$x_{\mathrm{c}}(k) = \begin{bmatrix} I_{\mathrm{a}}(k) \\ \omega(k) \\ X_{\mathrm{a}}(k) \end{bmatrix}, \quad u_{\mathrm{c}}(k) = \begin{bmatrix} V_{\mathrm{c}}(k) \\ T_{\mathrm{s}}(k) \end{bmatrix},$$

$$\Phi(T) = e^{A_{\mathrm{c}}T} = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \\ \phi_{31} & \phi_{32} & \phi_{33} \end{bmatrix},$$

$$H(T) = \int_0^T e^{A_{\mathrm{c}}t} dt \cdot B_{\mathrm{c}} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{31} & h_{32} \end{bmatrix}.$$

The matrix $\Phi(T)$ is computed from the eigenvalues of the A_{c} matrix or its Taylor series expansion for the discrete-time equivalent of the continuous-time system. Because the sampling interval (T) is small, the first three terms of the following Taylor series approximation may be sufficient for most applications:

$$\Phi(T) = e^{A_{\mathrm{c}}T} = [I] + [A]T + [A]^2\frac{T^2}{2!} + \cdots. \tag{4.82}$$

The discrete-time components of the position control loop consists of digital filter $D(z)$ and D/A converter gain K_{d}. The velocity command signal can be expressed in the z domain as

$$V_{\mathrm{c}}(k) = K_{\mathrm{p}}\frac{z+a}{z+b} \cdot K_{\mathrm{d}} \cdot [X_{\mathrm{r}}(k) - X_{\mathrm{a}}(k)]. \tag{4.83}$$

By noting that $(z+a)/(z+b) = 1 + (a-b)/(z+b)$,the equation can be rearranged as

$$V_{\mathrm{c}}(k) = K_{\mathrm{p}}\, K_{\mathrm{d}} \left(1 + \frac{a-b}{z+b}\right)[X_{\mathrm{r}}(k) - X_{\mathrm{a}}(k)] = K_{\mathrm{p}}\, K_{\mathrm{d}}\, [X_{\mathrm{r}}(k) - X_{\mathrm{a}}(k)] + K_{\mathrm{p}}\, K_{\mathrm{d}}\, \frac{a-b}{z+b}\, [X_{\mathrm{r}}(k) - X_{\mathrm{a}}(k)] \tag{4.84}$$

$$= K_{\mathrm{p}}\, K_{\mathrm{d}}\, [X_{\mathrm{r}}(k) - X_{\mathrm{a}}(k)] + V_{\mathrm{d}}(k), \tag{4.85}$$

where

$$V_{\mathrm{d}}(k) = K_{\mathrm{p}}\, K_{\mathrm{d}}\, \frac{a-b}{z+b}\, [X_{\mathrm{r}}(k) - X_{\mathrm{a}}(k)]. \tag{4.86}$$

$$(z+b)\, V_{\mathrm{d}}(k) = K_{\mathrm{p}}\, K_{\mathrm{d}}\, (a-b)\, [X_{\mathrm{r}}(k) - X_{\mathrm{a}}(k)] \tag{4.87}$$

The new variables V_c and V_d can be treated as a fourth *state* respectively. After rearranging Eqs. (4.86) and (4.83) taking their inverse z transforms, we obtain the following discrete-time state equations:

$$\begin{aligned} V_\mathrm{d}(k+1) &= -\,b\,V_\mathrm{d}(k) + K_\mathrm{p}\,K_\mathrm{d}\,(a-b)\,[X_\mathrm{r}(k) - X_\mathrm{a}(k)], \\ V_\mathrm{c}(k) &= K_\mathrm{p}\,K_\mathrm{d}\,[X_\mathrm{r}(k) - X_\mathrm{a}(k)] + V_\mathrm{d}(k). \end{aligned} \tag{4.88}$$

The discrete-time state equation (4.88) can be combined with the state equation (4.81), which represents the discrete-time equivalent of the continuous part of the feed drive servo. An algebraic rearrangement yields the following complete state equations for the feed drive servo:

$$\begin{aligned} x(k+1) &= G(T)\cdot x(k) + \Gamma(T)\cdot u(k), \\ y(k) &= C_\mathrm{s}\cdot x(k) + D_\mathrm{s}\cdot u(k), \end{aligned} \tag{4.89}$$

where the state, input, and output vectors are defined respectively as

$$x(k) = \begin{bmatrix} V_\mathrm{d}(k) \\ I_\mathrm{a}(k) \\ \omega(k) \\ X_\mathrm{a}(k) \end{bmatrix}, u(k) = \begin{bmatrix} X_\mathrm{r}(k) \\ T_\mathrm{s}(k) \end{bmatrix}, y(k) = \begin{bmatrix} V_\mathrm{c}(k) \\ I_\mathrm{a}(k) \\ \omega(k) \\ X_\mathrm{a}(k) \end{bmatrix}.$$

The state matrix $G(T)$, input matrix $\Gamma(T)$, output matrix $C_\mathrm{s}(T)$, and transmission matrix $D_\mathrm{s}(T)$ are defined in order as follows:

$$G(T) = \begin{bmatrix} -b & 0 & 0 & -K_\mathrm{p}K_\mathrm{d}(a-b) \\ h_{11} & \phi_{11} & \phi_{12} & \phi_{13} - h_{11}K_\mathrm{p}K_\mathrm{d} \\ h_{21} & \phi_{21} & \phi_{22} & \phi_{23} - h_{21}K_\mathrm{p}K_\mathrm{d} \\ h_{31} & \phi_{31} & \phi_{32} & \phi_{33} - h_{31}K_\mathrm{p}K_\mathrm{d} \end{bmatrix},$$

$$\Gamma(T) = \begin{bmatrix} K_\mathrm{p}K_\mathrm{d}(a-b) & 0 \\ h_{11}K_\mathrm{p}K_\mathrm{d} & h_{12} \\ h_{21}K_\mathrm{p}K_\mathrm{d} & h_{22} \\ h_{31}K_\mathrm{p}K_\mathrm{d} & h_{32} \end{bmatrix},$$

$$C_\mathrm{s} = \begin{bmatrix} 1 & 0 & 0 & -K_\mathrm{p}K_\mathrm{d} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, D_\mathrm{s} = \begin{bmatrix} K_\mathrm{p}K_\mathrm{d} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}.$$

The output vector $y(k)$ gives access to four useful dynamic parameters in the feed drive servo, namely the velocity command input to the amplifier (V_c), armature current (I_a), the angular velocity (ω), and the position of the table (X_a) for a given position command (X_r) and applied cutting torque (T_s . Other states in the control system can be easily found by multiplying them with the appropriate gains according to the block diagram shown in Figure 4.6.

Example 43 *Develop the state space model of a simple drive system shown in Figure 4.12.*

The system receives position command (x_r) and measured (i.e. actual) position from encoder (x_a) in counts. The position error [$e = x_r - x_a$] is multiplied by a proportional controller gain (K_p) and sent to motor as a torque command through DAQ and amplifier circuits in the form of voltage [$u[V]$]. The disturbance (i.e. friction and external load) takes some of the torque (T_d), and the remaining is used to rotate the ball screw at angular velocity (Ω[rad/s)) which has a first order transfer function ($K_v/(s+a)$). The velocity is converted to encoder units [count] with a gain K_e and summed (i.e. integrated) by a counter ($1/s$).

The continuous part of the system dynamics need to be represented by a set of first order differential equations. There are two states: velocity (Ω) and actual position (x_a) which change as a function of time.

$$\begin{aligned} \frac{\Omega(s)}{u(s) - T_d(s)} &= \frac{K_v}{s+a} \rightarrow (s+a)\,\Omega(s) = K_v\,[u(s) - T_d(s)] \\ \frac{d\Omega(t)}{dt} &= -a\Omega(t) + K_v u(t) - K_v T_d(t) \end{aligned}$$

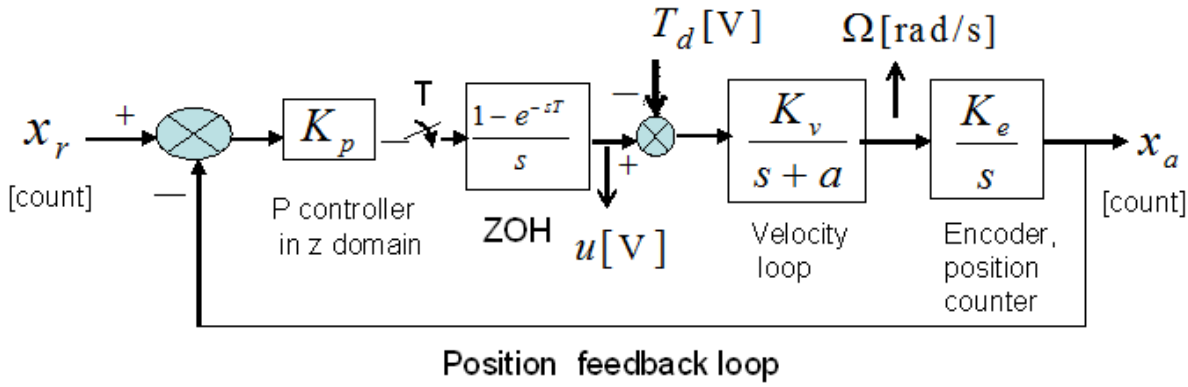


Figure 4.12: Blcok diagram of a simple position control system.

$$\frac{x_a(s)}{\Omega(s)} = \frac{K_e}{s} \rightarrow s x_a(s) = K_e \Omega(s)$$
$$\frac{dx_a(t)}{dt} = K_e \Omega(t)$$

The state space matrix need to be formed by separating states (Ω, x_a) and inputs (u, T_d) with the derivatives of states on the left hand side:

$$\underbrace{\left\{ \begin{array}{c} \frac{d\Omega(t)}{dt} \\ \frac{dx_a(t)}{dt} \end{array} \right\}}_{\dot{X}_c(t)} = \underbrace{\left[\begin{array}{cc} -a & 0 \\ K_e & 0 \end{array} \right]}_{A} \underbrace{\left\{ \begin{array}{c} \Omega(t) \\ x_a(t) \end{array} \right\}}_{X_c(t)} + \underbrace{\left[\begin{array}{cc} K_v & -K_v \\ 0 & 0 \end{array} \right]}_{B} \underbrace{\left\{ \begin{array}{c} u(t) \\ T_d(t) \end{array} \right\}}_{U(t)}$$
$$\dot{X}_c(t) = AX_c(t) + BU(t)$$

The transition matrices can either be evaluated analytically using inverse transforms, or numerically by Taylor Series expansion as follows. Noting that $e^{At} = L^{-1}\left(s[I] - [A]\right)^{-1}$.

$$\Phi(t) = e^{At} = L^{-1}\left(s[I] - [A]\right)^{-1} = L^{-1}\left(s \left[\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right] - \left[\begin{array}{cc} -a & 0 \\ K_e & 0 \end{array} \right]\right)^{-1}$$
$$= L^{-1}\left(\left[\begin{array}{cc} s+a & 0 \\ -K_e & s \end{array} \right]^{-1}\right) = L^{-1}\left(\left[\begin{array}{cc} \frac{1}{a+s} & 0 \\ \frac{1}{s}\frac{K_e}{a+s} & \frac{1}{s} \end{array} \right]\right)$$
$$\Phi(t) = e^{At} = \left[\begin{array}{cc} e^{-at} & 0 \\ \frac{K_e}{a}\left(1 - e^{-at}\right) & 1 \end{array} \right]$$

or in discrete time domain

$$\Phi(T) = e^{AT} = \left[\begin{array}{cc} e^{-aT} & 0 \\ \frac{K_e}{a}\left(1 - e^{-aT}\right) & 1 \end{array} \right]$$

Note that $e^{-aT} = 1 - aT + \frac{1}{2!}(-aT)^2 + \frac{1}{3!}(-aT)^3 + ...$ if we take the first two terms of the Taylor Series , $e^{-aT} \cong 1 - aT$

$$\Phi(T) = e^{AT} = \left[\begin{array}{cc} e^{-aT} & 0 \\ \frac{K_e}{a}\left(1 - e^{-aT}\right) & 1 \end{array} \right] \simeq \left[\begin{array}{cc} 1 - aT & 0 \\ \frac{K_e}{a}\left(1 - 1 + aT\right) & 1 \end{array} \right] = \left[\begin{array}{cc} 1 - aT & 0 \\ K_e T & 1 \end{array} \right]$$

Alternatively, if we use Taylor Series expansion directly, the state transition matrices in discrete time domain can be expressed as:

$$\Phi(T) = e^{AT} = [I] + [A]T + \frac{1}{2!}[AT]^2 + ..$$
$$\Phi(T) \simeq \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} -a & 0 \\ K_e & 0 \end{bmatrix} T = \begin{bmatrix} 1-aT & 0 \\ K_eT & 1 \end{bmatrix}$$

$$\begin{aligned} H(T) &= \int_0^T e^{A_c t} dt \cdot B_c \simeq \begin{bmatrix} \int_0^T (1-at)dt & 0 \\ \int_0^T K_e t dt & \int_0^T dt \end{bmatrix} \cdot \begin{bmatrix} K_v & -K_v \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} T - \frac{1}{2}aT^2 & 0 \\ K_e \frac{T^2}{2} & T \end{bmatrix} \begin{bmatrix} K_v & -K_v \\ 0 & 0 \end{bmatrix} \\ H(T) &= \begin{bmatrix} K_vT(1-\frac{1}{2}aT) & -K_vT(1-\frac{1}{2}aT) \\ K_vK_e\frac{T^2}{2} & -K_vK_e\frac{T^2}{2} \end{bmatrix} \end{aligned}$$

The state space model of the machine in discrete time domain becomes

$$\underbrace{\begin{Bmatrix} \Omega(k+1) \\ x_a(k+1) \end{Bmatrix}}_{X_c(k+1)} = \underbrace{\begin{bmatrix} 1-aT & 0 \\ K_eT & 1 \end{bmatrix}}_{\Phi_c(T)} \underbrace{\begin{Bmatrix} \Omega(k) \\ x_a(k) \end{Bmatrix}}_{X_c(k)} + \underbrace{\begin{bmatrix} K_vT(1-\frac{1}{2}aT) & -K_vT(1-\frac{1}{2}aT) \\ K_vK_e\frac{T^2}{2} & -K_vK_e\frac{T^2}{2} \end{bmatrix}}_{H_c(T)} \underbrace{\begin{Bmatrix} u(k) \\ T_d(k) \end{Bmatrix}}_{U_c(k)}$$
$$X_c(k+1) = \Phi(T)X_c(k) + H_c(T)U_c(k)$$

Note that ZOH is already considered when the continuous system is discretized here in an approximate way, hence it must not be re-considered again.

The discrete part of the machine is just the proportional controller:

$$u(k) = K_p\left[x_r(k) - x_a(k)\right] = K_p x_r(k) - K_p x_a(k)$$

One may add output vector that contains the desired states to be monitored as follows

$$\underbrace{\begin{Bmatrix} u(k) \\ \Omega(k) \\ x_a(k) \end{Bmatrix}}_{Y(k)} = \underbrace{\begin{bmatrix} 0 & -K_p \\ 1 & 0 \\ 0 & 1 \end{bmatrix}}_{C} \underbrace{\begin{Bmatrix} \Omega(k) \\ x_a(k) \end{Bmatrix}}_{X(k)} + \underbrace{\begin{bmatrix} K_p & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}}_{D} \underbrace{\begin{Bmatrix} x_r(k) \\ T_d(k) \end{Bmatrix}}_{U(k)}$$
$$Y(k) = CX(k) + Du(k)$$

The final state space model becomes

$$X_c(k+1) = \Phi(T)X_c(k) + H_c(T)U_c(k)$$
$$Y(k) = CX(k) + Du(k)$$

Alternatively, one can substitute $u(k) = K_p x_r(k) - K_p x_a(k)$ in $X_c(k+1) = \Phi(T)X_c(k) + H_c(T)U_c(k)$

$$\begin{aligned} \Omega(k+1) &= (1-aT)\Omega(k) + K_vT(1-\frac{1}{2}aT)u(k) - K_vT(1-\frac{1}{2}aT)T_d(k) \\ &= (1-aT)\Omega(k) + K_vT(1-\frac{1}{2}aT)K_px_r(k) - K_vT(1-\frac{1}{2}aT)K_px_a(k) - K_vT(1-\frac{1}{2}aT)T_d(k) \\ \Omega(k+1) &= (1-aT)\Omega(k) - \left[K_vT(1-\frac{1}{2}aT)K_p\right]x_a(k) + \left[K_pK_vT(1-\frac{1}{2}aT)\right]x_r(k) - \left[K_vT(1-\frac{1}{2}aT)\right]T_d(k) \end{aligned}$$

$$
\begin{aligned}
x_a(k+1) &= [K_eT]\,\Omega(k) + x_a(k) + \left[K_vK_e\frac{T^2}{2}\right]u(k) - \left[K_vK_e\frac{T^2}{2}\right]T_d(k) \\
&= [K_eT]\,\Omega(k) + x_a(k) + \left[K_vK_e\frac{T^2}{2}\right]K_px_r(k) - \left[K_vK_e\frac{T^2}{2}\right]K_px_a(k) - \left[K_vK_e\frac{T^2}{2}\right]T_d(k) \\
x_a(k+1) &= [K_eT]\,\Omega(k) + \left[1 - K_pK_vK_e\frac{T^2}{2}\right]x_a(k) + \left[K_pK_vK_e\frac{T^2}{2}\right]x_r(k) - \left[K_vK_e\frac{T^2}{2}\right]T_d(k)
\end{aligned}
$$

The discrete time domain state space model of the entire closed loop system becomes

$$
\underbrace{\left\{\begin{array}{c}\Omega(k+1)\\ x_a(k+1)\end{array}\right\}}_{X(k+1)} = \underbrace{\begin{bmatrix}(1-aT) & -\left[K_vT(1-\frac{aT}{2})K_p\right]\\ K_eT & \left[1-K_pK_vK_e\frac{T^2}{2}\right]\end{bmatrix}}_{\Phi(T)}\underbrace{\left\{\begin{array}{c}\Omega(k)\\ x_a(k)\end{array}\right\}}_{X(k)} + \underbrace{\begin{bmatrix}\left[K_pK_vT(1-\frac{aT}{2})\right] & -\left[K_vT(1-\frac{aT}{2})\right]\\ \left[K_pK_vK_e\frac{T^2}{2}\right] & -K_vK_e\frac{T^2}{2}\end{bmatrix}}_{H(T)}\underbrace{\left\{\begin{array}{c}x_r(k)\\ T_d(k)\end{array}\right\}}_{U(k)}
$$

$$
X(k+1) = \Phi(T)X(k) + H(T)U(k)
$$

One may add output vector that contains the desired states to be monitored as follows

$$
\underbrace{\left\{\begin{array}{c}u(k)\\ \Omega(k)\\ x_a(k)\end{array}\right\}}_{y(k)} = \underbrace{\begin{bmatrix}0 & -K_p\\ 1 & 0\\ 0 & 1\end{bmatrix}}_{C}\underbrace{\left\{\begin{array}{c}\Omega(k)\\ x_a(k)\end{array}\right\}}_{X(k)} + \underbrace{\begin{bmatrix}K_p & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}}_{D(T)}\underbrace{\left\{\begin{array}{c}x_r(k)\\ T_d(k)\end{array}\right\}}_{U(k)}
$$

The system can be analyzed in z domain as well. The zero order hold equivalent of the continuous system is :

$$
\begin{aligned}
G_c(z) &= (1-z^{-1})Z\frac{G_c(s)}{s} = (1-z^{-1})Z\frac{K_v}{s(s+a)}\frac{K_e}{s} = \frac{K_eK_v}{a}(1-z^{-1})Z\left(\frac{1}{s^2} - \frac{1}{as} + \frac{1}{a}\frac{1}{a+s}\right) \\
&= \frac{K_eK_v}{a}(1-z^{-1})\left(\frac{Tz^{-1}}{(1-z^{-1})^2} - \frac{1}{a}\frac{1}{1-z^{-1}} + \frac{1}{a}\frac{1}{1-e^{-aT}z^{-1}}\right) \\
&= \frac{K_eK_v}{a}\left(\frac{Tz^{-1}}{1-z^{-1}} - \frac{1}{a} + \frac{1}{a}\frac{1-z^{-1}}{1-e^{-aT}z^{-1}}\right) \\
&= \frac{K_eK_v}{a}\left(\frac{aTz^{-1}(1-e^{-aT}z^{-1}) - (1-e^{-aT}z^{-1})(1-z^{-1}) + (1-z^{-1})^2}{a(1-e^{-aT}z^{-1})(1-z^{-1})}\right) \\
&= \frac{K_eK_v}{a^2}\left[\frac{aTz^{-1} - aTe^{-aT}z^{-2} - 1 + z^{-1} + e^{-aT}z^{-1} + e^{-aT}z^{-2} + 1 - 2z^{-2} + z^{-2}}{(1-e^{-aT}z^{-1})(1-z^{-1})}\right] \\
&= \frac{K_eK_v}{a^2}z^{-1}\left[\frac{(aT+1+e^{-aT}) + z^{-1}(-aTe^{-aT} + e^{-aT} - 1)}{(1-e^{-aT}z^{-1})(1-z^{-1})}\right]
\end{aligned}
$$

The closed loop transfer function of the position loop becomes:

$$
\begin{aligned}
G_{cl}(z) &= \frac{x_a}{x_r} = \frac{K_pG_c(z)}{1+K_pG_c(z)} = \frac{K_p\frac{K_eK_v}{a^2}z^{-1}\left[\frac{(aT+1+e^{-aT})+z^{-1}(-aTe^{-aT}+e^{-aT}-1)}{(1-e^{-aT}z^{-1})(1-z^{-1})}\right]}{1+K_p\frac{K_eK_v}{a^2}z^{-1}\left[\frac{(aT+1+e^{-aT})+z^{-1}(-aTe^{-aT}+e^{-aT}-1)}{(1-e^{-aT}z^{-1})(1-z^{-1})}\right]} \\
&= \frac{z^{-1}\left[(aT+1+e^{-aT}) + z^{-1}(-aTe^{-aT} + e^{-aT} - 1)\right]}{\frac{a^2}{K_pK_eK_v} + z^{-1}\left[(aT+1+e^{-aT}) + z^{-1}(-aTe^{-aT} + e^{-aT} - 1)\right]} = \frac{z^{-1}(b_0 + b_1z^{-1})}{a_0 + a_1z^{-1} + a_2z^{-2}}
\end{aligned}
$$

The output $x_a(k)$ can be simulated as a function of past inputs and outputs recursively as:

$$
\begin{aligned}
\left(a_0 + a_1z^{-1} + a_2z^{-2}\right)x_a(k) &= z^{-1}(b_0 + b_1z^{-1}) \\
x_a(k) &= \frac{1}{a_0}\left[-a_1x_a(k-1) - a_2x_a(k-2) + b_0x_r(k-1) + b_1x_r(k-2)\right]
\end{aligned}
$$

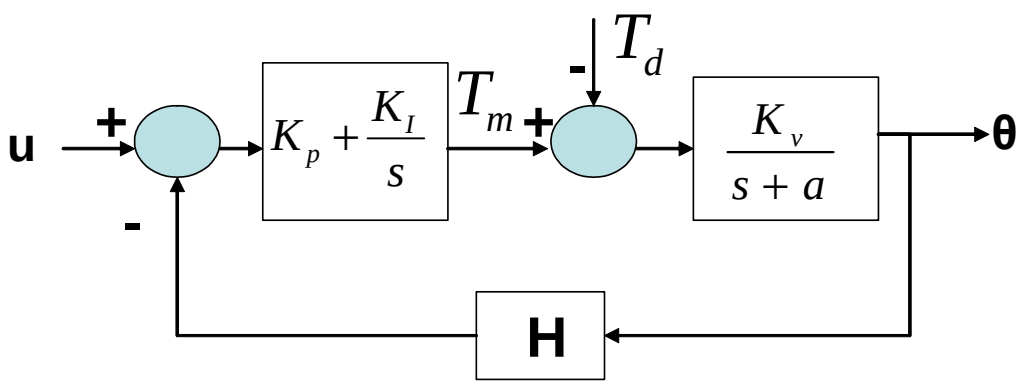
The tracking error between the output (x_a) and input (x_r) can be expressed as:

$$
\begin{aligned}
e(k) &= x_r(k) - x_{ar}(k) = [1 - G_{cl}(z)]\, x_r(k) = \left[1 - \frac{K_p G_c(z)}{1 + K_p G_c(z)}\right] x_r(k) = \left[\frac{1}{1 + K_p G_c(z)}\right] x_r(k) \\
e(k) &= \left[\frac{1}{1 + K_p \frac{K_e K_v}{a^2} z^{-1} \left[\frac{(aT+1+e^{-aT}) + z^{-1}(-aTe^{-aT} + e^{-aT} - 1)}{(1 - e^{-aT} z^{-1})(1 - z^{-1})}\right]}\right] x_r(k) \\
&= \left[\frac{(1 - e^{-aT} z^{-1})(1 - z^{-1})}{(1 - e^{-aT} z^{-1})(1 - z^{-1}) + K_p \frac{K_e K_v}{a^2} z^{-1} \left[aT + 1 + e^{-aT}) + z^{-1}(-aTe^{-aT} + e^{-aT} - 1)\right]}\right] x_r(k)
\end{aligned}
$$

If a ramp input is given to the system, $x_r(z) = Z[f_c t] = f_c \frac{Tz^{-1}}{(1-z^{-1})^2}$, the steady state tracking error is:

$$
\begin{aligned}
e_{ss} &= \lim_{z \to 1} (1 - z^{-1}) \left[\frac{1}{1 + K_p G_c(z)}\right] x_r(z) \\
&= \lim_{z \to 1} (1 - z^{-1}) \left[\frac{(1 - e^{-aT} z^{-1})(1 - z^{-1})}{(1 - e^{-aT} z^{-1})(1 - z^{-1}) + K_p \frac{K_e K_v}{a^2} z^{-1} \left[aT + 1 + e^{-aT}) + z^{-1}(-aTe^{-aT} + e^{-aT} - 1)\right]}\right] f_c \frac{Tz^{-1}}{(1 - z^{-1})^2} \\
e_{ss} &= \lim_{z \to 1} \left[\frac{(1 - e^{-aT} z^{-1})}{K_p \frac{K_e K_v}{a^2} z^{-1} \left[aT + 1 + e^{-aT}) + z^{-1}(-aTe^{-aT} + e^{-aT} - 1)\right]}\right] f_c T z^{-1} = \frac{f_c T (1 - e^{-aT})}{K_p \frac{K_e K_v}{a^2} \left[aT + 1 + e^{-aT}) + (-aTe^{-aT}\right.}
\end{aligned}
$$

Example 44 *Develop the state space model of the following closed loop system defined in continuous (Laplace) domain.*



Block diagram of a PI controller applied on a first order system.

The integral term K_I/s brings some confusion in the modeling as follows:

$$
\begin{aligned}
(u - H\theta)\left(K_p + \frac{K_I}{s}\right) = T_m \rightarrow K_p u - K_p H\theta + \frac{K_I}{s} u - \frac{K_I}{s} H\theta = T_m \\
sT_m = K_p s u - K_p H s\theta + K_I u - K_I H\theta \\
\frac{dT_m}{dt} = -K_p H \frac{d\theta}{dt} - K_I H\theta + K_p \frac{du}{dt} + K_I u
\end{aligned}
$$

Since we have derivative terms at each side, we are not able to express the state space equations easily. However, proportional - integral (PI) controllers are commonly used in industry, and they need to be modeled. Lets re-organize the variables as follows. The error is defined as $e = u - H\theta$.

From the PI controller's transfer function:

$$T_m(s) = \left(K_p + \frac{K_I}{s}\right) e(s) = K_p e(s) + K_I \frac{e(s)}{s} \leftarrow e(s) = u(s) - H\theta(s)$$

$$\text{Let } \frac{e(s)}{s} = x_1(s) \rightarrow s x_1(s) = e(s),\ \frac{dx_1}{dt} = e(t), x_1(t) = \mathcal{L}^{-1}(\frac{e(s)}{s}) = \int e(t)$$

$$\frac{dx_1}{dt} = e(t) = u(t) - H\theta(t)$$

$$T_m(s) = K_p e(s) + K_I \frac{e(s)}{s} = K_p \left[u(s) - H\theta(s)\right] + K_I x_1(s)$$

$$T_m(t) = +K_I x_1(t) - K_p H\theta(t) + K_p u(t)$$

From the first order transfer function of the plant,

$$(s+a)\theta = K_v \left(T_m - T_d\right) \rightarrow \frac{d\theta}{dt} = -a\theta + K_v T_m - K_v T_d$$

$$\frac{d\theta}{dt} = -a\theta + K_v \left(K_I x_1 - K_p H\theta + K_p u\right) - K_v T_d$$

$$\frac{d\theta}{dt} = +K_v K_I x_1 - \left(a + K_v K_p H\right)\theta + K_v K_p u - K_v T_d$$

Since the two sets of first order differential equations are derived from the system dynamics model, it is possible to express them in a state space model as follows. Lets define the states as follows:

$$x_1(t) = \int e(t) \rightarrow \frac{dx_1}{dt} = e(t) = u - H\theta$$

$$x_2(t) = \theta(t)$$

By substituting states into the two first order differential equations:

$$\frac{dx_1}{dt} = -Hx_2 + u$$

$$\frac{dx_2}{dt} = K_v K_I x_1 - \left(a + K_v K_p H\right) x_2 + K_v K_p u - K_v T_d$$

or

$$\left\{ \begin{array}{c} \dot{x}_1 \\ \dot{x}_2 \end{array} \right\} = \left[\begin{array}{cc} 0 & -H \\ K_v K_I & -\left(a + K_v K_p H\right) \end{array} \right] \left\{ \begin{array}{c} x_1 \\ x_2 \end{array} \right\} + \left[\begin{array}{cc} 1 & 0 \\ K_v K_p & -K_v \end{array} \right] \left\{ \begin{array}{c} u \\ T_d \end{array} \right\}$$

or

$$\left\{\dot{x}\right\} = [A]\{x\} + [B]\{U\}$$

The output equations can be expressed as ($T_m = y_1$, $\theta = y_2$)

$$y_1 = T_m = K_I x_1 - K_p H\theta + K_p u = K_I x_1 - K_p H x_2 + K_p u$$

$$y_2 = \theta = x_2$$

$$\left\{ \begin{array}{c} T_m \\ \theta \end{array} \right\} = \left\{ \begin{array}{c} y_1 \\ y_2 \end{array} \right\} = \left[\begin{array}{cc} K_1 & -K_p H \\ 0 & 1 \end{array} \right] \left\{ \begin{array}{c} x_1 \\ x_2 \end{array} \right\} + \left[\begin{array}{cc} K_p & 0 \\ 0 & 0 \end{array} \right] \left\{ \begin{array}{c} u \\ T_d \end{array} \right\}$$

$$\{y\} = [C]\{x\} + [D]\{U\}$$

The state space model of the closed loop system is represented by the following state equations:

$$\left\{\dot{x}\right\} = [A]\{x\} + [B]\{U\}$$
$$\{y\} = [C]\{x\} + [D]\{U\}$$

The discrete time equivalent of the system is obtained by solving the state space equations at discrete time interval T:

$$\{x(k+1)\} = [\phi(T)]\{x(k)\} + [H(T)]\{U(k)\}$$
$$\{y(k)\} = [C]\{x(k)\} + [D]\{U(k)\}$$

If we take only first two terms in the Taylor Series expansion of e^{At}, the state transition ($[\phi(T)]$) and input transition ($[H(T)]$) matrices can be solved as,

$$\begin{aligned}
\phi(T) &= e^{AT} = I + AT \\
H(T) &= \left[\int_0^T e^{A\tau} d\tau\right] \cdot B = \left(\int_0^T \left\{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} a_{11}\tau & a_{12}\tau \\ a_{21}\tau & a_{22}\tau \end{bmatrix}\right\} d\tau\right) \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\
&= \left(\int_0^T \begin{bmatrix} 1 + a_{11}\tau & a_{12}\tau \\ a_{21}\tau & 1 + a_{22}\tau \end{bmatrix} d\tau\right) \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\
&= \begin{bmatrix} T + \frac{1}{2}T^2 a_{11} & \frac{1}{2}T^2 a_{12} \\ \frac{1}{2}T^2 a_{21} & T + \frac{1}{2}T^2 a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{2}T^2 a_{12} b_{21} + b_{11}\left(T + \frac{1}{2}T^2 a_{11}\right) & \frac{1}{2}T^2 a_{12} b_{22} + b_{12}\left(T + \frac{1}{2}T^2 a_{11}\right) \\ \frac{1}{2}T^2 a_{21} b_{11} + b_{21}\left(T + \frac{1}{2}T^2 a_{22}\right) & \frac{1}{2}T^2 a_{21} b_{12} + b_{22}\left(T + \frac{1}{2}T^2 a_{22}\right) \end{bmatrix}
\end{aligned}$$

Note that e^{At} can be solved analytically as well, although Taylor series expansion is easier to use when the sampling interval is small.

Algorithm 45 *If $[A]$ is an invertible square matrix [6] ,*

$$\int_0^T e^{[A]\tau} d\tau = [A]^{-1} e^{[A]\tau}\Big|_0^T = [A]^{-1} e^{[A]T} - [A]^{-1}[I]$$

If the matrix is not square , $[A]$ its not invertible, hence Taylor Series expansion can be unavoidable to evaluate the integration. However, $e^{[A]T}$ still needs to be solved either by opening into Taylor series or analytically using Cayley-Hamilton theorem as follows. A square matrix $A(n \times n)$ can be expressed as a polynomial of degree $n-1$ in A as:

$$f(A) = a_{n-1}A^{n-1} + a_{n-2}A^{n-2} + ... + a_2 A^2 + a_1 A + a_0 I$$

where I is the unity matrix. The coefficients of polynomials $a_{n-1}, a_{n-2}, .., a_{2,} a_1, a_0$ are determined from the eigenvalues of the system λ:

$$r(\lambda) = a_{n-1}\lambda^{n-1} + a_{n-2}\lambda^{n-2} + ... + a_2\lambda^2 + a_1\lambda + a_0$$

For each distinct eigenvalue λ_i of matrix A,

$$f(\lambda_i) = r(\lambda_i)$$

or if there are k repeating eigenvalues,

$$
\begin{aligned}
f'(\lambda)|_{\lambda=\lambda_i} &= r'(\lambda)|_{\lambda=\lambda_i} \\
f''(\lambda)|_{\lambda=\lambda_i} &= r^{'}(\lambda)\Big|_{\lambda=\lambda_i} \\
&\cdots \\
f^{(k-1)}(\lambda)\Big|_{\lambda=\lambda_i} &= r^{(k-1)}(\lambda)\Big|_{\lambda=\lambda_i}
\end{aligned}
$$

which leads to the solution of parameters $a_{n-1}, a_{n-2}, .., a_{2,}a_1, a_0$ which can be substituted in $f(A)$.

Example 46 *Lets find* e^{At} *for* $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$. *Let* $B = At = \begin{bmatrix} 0 & t \\ -t & 0 \end{bmatrix}$ *.Since the matrix* B *has a size of* 2×2*, the polynomial which represents the matrix is first order,*

$$
e^B = a_1 B + a_0 I = \begin{bmatrix} 0 & a_1 t \\ -a_1 t & 0 \end{bmatrix} + \begin{bmatrix} a_0 & 0 \\ 0 & a_0 \end{bmatrix} = \begin{bmatrix} a_0 & a_1 t \\ -a_1 t & a_0 \end{bmatrix}
$$

The eigenvalues of B can be found as :

$$
\begin{aligned}
\left| \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & t \\ -t & 0 \end{bmatrix} \right| = \begin{vmatrix} \lambda & -t \\ t & \lambda \end{vmatrix} &= \lambda^2 + t^2 \\
\lambda = \sqrt{-t^2}, \rightarrow \lambda_1 &= -it, \quad \lambda_1 = it \\
f(\lambda) = e^{\lambda} = \ r(\lambda) &= a_1 \lambda + a_0 \\
e^{\lambda_1} &= a_1 \lambda_1 + a_0, \quad e^{-it} = -a_1 it + a_0 \\
e^{\lambda_2} &= a_1 \lambda_2 + a_0, \quad e^{it} = a_1 it + a_0
\end{aligned}
$$

Two parameters (a_1, a_0) can be found as :

$$
\begin{aligned}
e^{it} + e^{-it} = 2a_0 \rightarrow a_0 &= \frac{1}{2}(\cos t + i \sin t + \cos t - i \sin t) = \cos t \\
e^{it} - e^{-it} = -i2a_1 t \rightarrow a_1 &= \frac{-1}{2it}(\cos t + i \sin t - \cos t + i \sin t) = \frac{\sin t}{t} \\
e^B = e^{[A]t} &= \begin{bmatrix} a_0 & a_1 t \\ -a_1 t & a_0 \end{bmatrix} = \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix}
\end{aligned}
$$

If we take the integral of $e^{[A]t}$,

$$
\int_0^t e^{[A]t} dt = \int_0^t \begin{bmatrix} \cos t & \sin t \\ -\sin t & \cos t \end{bmatrix} dt = \begin{bmatrix} \sin t & -\cos t \\ \cos t & \sin t \end{bmatrix} \Bigg|_0^t = \begin{bmatrix} \sin t & -\cos t + 1 \\ \cos t - 1 & \sin t \end{bmatrix}
$$

If we take the integral only at sampling interval $T = 0.01s$,

$$
\int_0^T e^{[A]t} dt = \begin{bmatrix} \sin T & -\cos T + 1 \\ \cos T - 1 & \sin T \end{bmatrix} = \begin{bmatrix} 9.9998 \times 10^{-3} & 5.0000 \times 10^{-5} \\ -5.0000 \times 10^{-5} & 9.9998 \times 10^{-3} \end{bmatrix}
$$

Let us compare the evaluation of $e^{[A]T}$ using analytical solution Taylor expansion methods using $T = 0.002\,\mathrm{sec}$.

$$
\begin{aligned}
\text{Analytical } e^{[A]T} &= \begin{bmatrix} \cos T & \sin T \\ -\sin T & \cos T \end{bmatrix} = \begin{bmatrix} 1.00000 & 0.002 \\ -0.002 & 1.00000 \end{bmatrix} \\
\text{Taylor Series } e^{[A]T} &= [I] + [A]T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} T = \begin{bmatrix} 1.0 & 0.002 \\ -0.002 & 1.0 \end{bmatrix}
\end{aligned}
$$

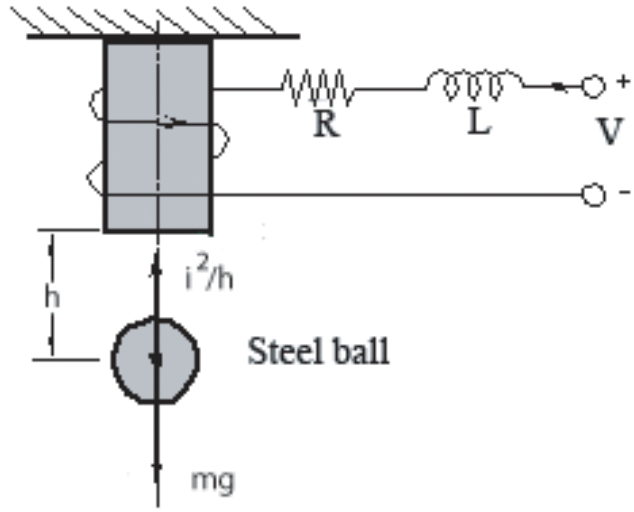


Figure 4.13: Magnetically lavitated spherical ball.

However, as T becomes larger, more terms in the Taylor Series expansion need to be taken into account in order to improve the accuracy.

Example 47 *State Space Model of a Nonlinear system [5]*

Consider a magnetically levitated a spherical ball with a mass (m) whose center needs to be suspended (h) mm away from the magnetic head. The magnetic system is energized by a electrical network having inductance of L and resistance of R. The voltage and current across the electrical network is V and i, respectively. The system dynamics can be represented by the following equations:

$$\begin{array}{rcl} \text{Electrical system} & : & V(t) = L\frac{di(t)}{dt} + Ri(t) \\ \text{Equation of motion for the sphere} & : & m\frac{d^2h(t)}{dt} = \sum F = mg - \frac{i^2(t)}{h(t)} \end{array} \tag{4.90}$$

While the electrical system dynamics is linear, the terms $(i^2, \frac{1}{h})$ makes the equation of the motion non-linear. In order to design a control system, it is desirable to linearize the system around a nominal operating point, i.e. $h_0 = 0.01m$. The system dynamic parameters are given as $m = 0.05kg, L = 0.01H, R = 1\Omega, g = 9.81m/s^2$.

Let the system states are expressed as:

$$\begin{array}{rll} x_1(t) = & h(t) & \\ x_2(t) = & \frac{dx_1}{dt} & = \frac{dh(t)}{dt} \\ x_3(t) = & i(t) & \end{array} \tag{4.91}$$

The derivatives of the states can be expressed as:

$$\begin{array}{rcll} \frac{dx_1}{dt} = & x_2(t) & = \frac{dh(t)}{dt} \\ \frac{dx_2}{dt} = & g - \frac{1}{m}\frac{x_3^2(t)}{x_1(t)} = f_2(t) & = \frac{d^2h}{dt} = g - \frac{1}{m}\frac{i^2(t)}{h(t)} \\ \frac{dx_3}{dt} = & -\frac{R}{L}x_3(t) + \frac{1}{L}V(t) = f_3(t) & = i(t) = -\frac{R}{L}i(t) + \frac{1}{L}V(t) \end{array} \tag{4.92}$$

The current corresponding to the equilibrium position of the sphere is $(h_0 = 0.01m)$:

$$g - \frac{1}{m}\frac{i_0^2}{h_0} = 0 \rightarrow i_0 = \sqrt{mgh_0} = \sqrt{0.05 \cdot 9.81 \cdot 0.01} = 0.07A \tag{4.93}$$

The system can be linearized at its equilibrium position $(h_0 = 0.01m, i_0 = 0.07A)$ as:

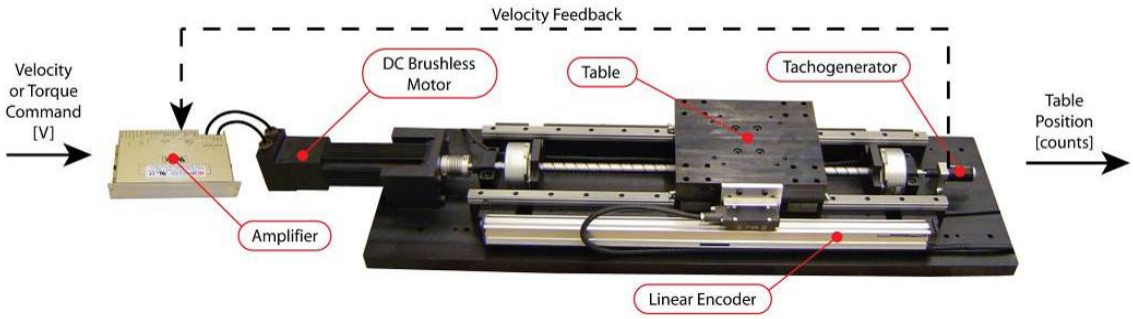


Figure 4.14: A ball screw servo system used in the laboratory.

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\partial f_2(t)}{\partial x_1} & \frac{\partial f_2(t)}{\partial x_2} & \frac{\partial f_2(t)}{\partial x_3} \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\partial f_3(t)}{\partial V} \end{bmatrix} V(t) \tag{4.94}$$

The partial derivatives are evaluated at the equilibrium of the sphere for the linearization of the system:

$$\begin{aligned} \left.\frac{\partial f_2(t)}{\partial x_1}\right|_{x_1=h_0} &= \frac{\partial}{\partial x_1}\left(g - \frac{1}{m}\frac{x_3^2(t)}{x_1(t)}\right)\Big|_{x_1=h_0} = \frac{1}{m}\frac{i_0^2}{h_0^2} = \frac{1}{m}\frac{mgh_0}{h_0^2} = \frac{g}{h_0} \\ \left.\frac{\partial f_2(t)}{\partial x_2}\right|_{x_2=x_{20}} &= 0 \\ \left.\frac{\partial f_2(t)}{\partial x_3}\right|_{x_3=x_{30}} &= \frac{\partial f}{\partial x_3}(g - \frac{1}{m}\frac{x_3^2}{x_1})\Big|_{x_3=x_{30}} = -\frac{2x_{30}}{mx_{10}} = -2\frac{i_0}{mh_0} \\ &= -2\sqrt{\frac{mgh_0}{m^2h_0^2}} = -2\sqrt{\frac{g}{mh_0}} \\ \left.\frac{\partial f_3(t)}{\partial V}\right|_{V=V_0} &= \frac{1}{L} \end{aligned} \tag{4.95}$$

The linearized equations are expressed as:

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{Bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{g}{h_0} & 0 & -2\sqrt{\frac{g}{mh_0}} \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \end{Bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} V(t) \tag{4.96}$$

which is valid only at when the ball position is around $h_0 = 0.01m$. The system now can be solved in state space as in the case of a linear case.

Example 48 *A single axis ball screw driven table is given below. Develop the state space model of the system, and simulate the step and ramp input response.*

Ball Screw table parameters

Encoder Gain	$K_e(count/rad/s)$	Current amplifier proportional gain	$K_{vp}(V/V)$
Equivalent inertia	$J_e(kgm^2)$	Current amplifier integral gain	$K_{vi}(V/Vs)$
Equivalent viscous damping	$B(kgm^2/s)$	Current command	$i_a(V)$
Disturbance Torque	$T_d(Nm)$	Velocity amplifier proportional gain	$K_{ip}(V/V)$
Motor Torque	$T_m(Nm)$	Velocity amplifier integral gain	$K_{ii}(V/Vs)$
Motor inductance	$L(H)$	Current Sensor gain	$K_a(V/A)$
Motor Resistance	$R(\Omega)$	Velocity amplifier gain	$S_g(V/V)$
Back electromotor constant	$K_b(V/rad/s)$	Motor torque constant	$K_t(Nm/A)$
Angular velocity of ball screw	$\omega(rad/s)$	Transfer function of controller	$G_c(s)$
Velocity feedback gain	$T_g(V/count/s)$	Position command	$x_r(counts)$
		Measured Position	$x_a(counts)$

Table of Transfer Function Blocks

Encoder	$\omega(s)\frac{K_e}{s} = x_a(s)$
Mechanical System	$(T_m - T_d)\frac{1}{J_e s + B} = \omega(s)$
Electrical Winding of the Motor	$(V_i - K_b\omega)\frac{1}{Ls+R} = i(s)$
Current PI Amplifier	$(i_a - K_a i)\left(K_{vp} + \frac{K_{vi}}{s}\right) = V_i(s)$
Velocity PI amplifier	$(S_g V_c - \omega)\left(K_{ip} + \frac{K_{ii}}{s}\right) = i_a(s)$
Position Error Controller	$(x_r - x_a)G_c(s) = V_c(s)$

Motor electrical winding and mechanical system can be modeled as :

$$J\frac{d\omega}{dt} + B\omega = K_t i \rightarrow \frac{d\omega}{dt} = -\frac{B}{J}\omega + \frac{K_t}{J}i - \frac{1}{J}T_d \tag{4.97}$$

$$L\frac{di}{dt} + Ri = V_i - K_b\omega \rightarrow \frac{di}{dt} = -\frac{K_b}{L}\omega - \frac{R}{L}i + \frac{1}{L}V_i \tag{4.98}$$

PI Current Loop

$$V_i = (i_a - K_a i)\left(K_{vp} + \frac{K_{vi}}{s}\right) = K_{vp}(i_a - K_a i) + K_{iv}\underbrace{\frac{1}{s}(i_a - K_a i)}_{z_1} \tag{4.99}$$

Auxiliary state $$z_1 = \frac{1}{s}(i_a - K_a i) \rightarrow \frac{dz_1}{dt} = i_a - K_a i \tag{4.100}$$

$$V_i = K_{pv}(i_a - K_a i) + K_{iv}z_1 \tag{4.101}$$

PI Velocity Loop

$$i_a = (S_g V_c - \omega)\left(K_{ip} + \frac{K_{ii}}{s}\right) = K_{ip}(S_g V_c - \omega) + K_{ii}\underbrace{\frac{1}{s}(S_g V_c - \omega)}_{z_2} \tag{4.102}$$

Auxiliary state $$z_2 = \frac{1}{s}(S_g V_c - \omega) \rightarrow \frac{dz_2}{dt} = S_g V_c - \omega \tag{4.103}$$

$$i_a = K_{ip}(S_g V_c - \omega) + K_{ii}z_2 \tag{4.104}$$

Re-writing the equation for $\frac{dz_1}{dt}$ and V_i by substituting i_a

$$\frac{dz_1}{dt} = i_a - K_a i = K_{ip}(S_g V_c - \omega) + K_{ii}z_2 - K_a i \tag{4.105}$$

$$V_i = K_{vp}(i_a - K_a i) + K_{vi}z_1 = K_{vp}(K_{ip}(S_g V_c - \omega) + K_{ii}z_2 - K_a i) + K_{vi}z_1 \tag{4.106}$$

$$V_i = K_{vp}K_{ip}S_g V_c - K_{vp}K_{ip}\omega + K_{vp}K_{ii}z_2 - K_{vp}K_a i + K_{vi}z_1 \tag{4.107}$$

By substituting V_i into $\frac{di}{dt}$

$$\frac{di}{dt} = -\frac{K_b}{L}\omega - \frac{R}{L}i + \frac{1}{L}V_i \tag{4.108}$$

$$= -\frac{K_b}{L}\omega - \frac{R}{L}i + \frac{K_{vp}K_{ip}S_g}{L}V_c - \frac{K_{vp}K_{ip}}{L}\omega + \frac{K_{vp}K_{ii}}{L}z_2 - \frac{K_{vp}K_a}{L}i + \frac{K_{vi}}{L}z_1 \tag{4.109}$$

$$= \left(-\frac{K_b + K_{vp}K_{ip}}{L}\right)\omega - \frac{R + K_{vp}K_a}{L}i + \frac{K_{vi}}{L}z_1 + \frac{K_{vp}K_{ii}}{L}z_2 + \frac{K_{vp}K_{ip}S_g}{L}V_c \tag{4.110}$$

The states are combined as :

$$\left\{\dot{x}\right\} = \left\{\begin{array}{c} \frac{d\omega}{dt} \\ \frac{di}{dt} \\ \frac{dz_1}{dt} \\ \frac{dz_2}{dt} \\ \frac{dz_a}{dt} \end{array}\right\} = \left[\begin{array}{ccccc} -\frac{B}{J} & \frac{K_t}{J} & 0 & 0 & 0 \\ -\frac{K_b+K_{vp}K_{ip}}{L} & -\frac{R+K_{vp}K_a}{L} & \frac{K_{vi}}{L} & \frac{K_{vp}K_{ii}}{L} & 0 \\ -K_{ip} & -K_a & 0 & K_{ii} & 0 \\ -1 & 0 & 0 & 0 & 0 \\ K_e & 0 & 0 & 0 & 0 \end{array}\right] \left\{\begin{array}{c} \omega \\ i \\ z_1 \\ z_2 \\ x_a \end{array}\right\} + \left[\begin{array}{cc} 0 & -\frac{1}{J} \\ \frac{K_{pv}K_{pi}S_g}{L} & 0 \\ K_{ip}s_g & 0 \\ S_g & 0 \\ 0 & 0 \end{array}\right] \left\{\begin{array}{c} V_c \\ T_d \end{array}\right\} \tag{4.111}$$

Since we are interested in physical variables such as velocity (ω), current (i), equivalent current command (i_a), motor voltage (V_i)

$$\left\{\begin{array}{c} \omega \\ i \\ i_a \\ V_i \\ x_a \end{array}\right\} = \left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -K_{ip} & 0 & 0 & K_{ii} & 0 \\ -K_{vp}K_{ip} & -K_{vp}K_a & K_{vi} & K_{vp}K_{ii} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right] \left\{\begin{array}{c} \omega \\ i \\ z_1 \\ z_2 \\ x_a \end{array}\right\} + \left[\begin{array}{cc} 0 & 0 \\ 0 & 0 \\ K_{ip}S_g & 0 \\ K_{vp}K_{ip}S_g & 0 \\ 0 & 0 \end{array}\right] \left\{\begin{array}{c} V_c \\ T_d \end{array}\right\} \tag{4.112}$$

or in standard state space notation :

$$\left.\begin{array}{c} \left\{\dot{x}\right\} = [A]\{x\} + [B]\{u\} \\ \{y\} = [C]\{x\} + [D]\{u\} \end{array}\right\} \tag{4.113}$$

The state space model of the machine indicated above represents the continuous part of the physical system. If the system needs to be controlled in discrete time domain, its zero order hold equivalent must be considered which leads to state space model of the system in z domain. The digital controller then can be combined to achieve a full state space model of the closed loop system in discrete time domain.

Chapter 5

Stability Analysis of Continuous and Discrete Time Closed Loop Control Systems

5.1 Introduction

Consider a system shown in the Figure 5.1.

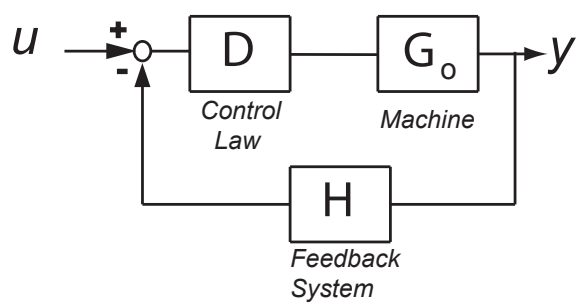


Figure 5.1: Close loop system with feedback (H) dynamics.

$s-$**Domain:**The closed loop transfer function of a system in s domain can be expressed as:

$$G_{cl}(s) = \frac{y(s)}{u(s)} = \frac{G(s)}{1 + G(s)H(s)} \tag{5.1}$$

where $G(s) = D(s)G_0(s)$ is the forward, and $H(s)$ is the feedback loop transfer function. The loop transfer function of the system is $D(s)G(s)H(s)$.The system type is defined by the number of poles the forward loop transfer function has at the origin ($s = 0$),

$$G(s) = D(s)G_0(s) = \frac{K(s-b_1)(s-b_2)\cdots(s-b_m)}{s^k(s-a_1)(s-a_2)\cdots(s-a_n)}$$

where the system is called type k. The steady state error of the system is highly dependent on its type for a given input function. Systems with higher type orders produce less errors due its error integration ($1/s$) ability. The stability of the system is determined by the system poles ($p_1, p_2, p_{3,}.., p_n$) which are the roots of the characteristic equation,

$$1 + G(s)H(s) = (s - p_1)(s - p_2)...(s - p_n) = 0 \tag{5.2}$$

The poles can be mapped on a complex s plane as shown in Figure 5.2. A pole can be expressed as $p_n = \sigma_n + j\gamma_n$, which has real (σ_n) and complex (γ_n) parts. The poles will affect the system's response by showing up in the output as $y(t) = Y_1 e^{p_1 t} + \cdots + Y_n e^{p_n t} = Y_1 e^{\sigma_1 t}\sin(\gamma_1 t + \phi_1) + \cdots + Y_n e^{\sigma_n t}\sin(\gamma_n t + \phi_n)$, regardless of any input type. The input will bring additional time response component to the output, but it has no effect on the stability of the system.

Example 49 *Consider a first order system with a transfer function $G(s)$, with step input $u(s) = U/s$ and output $y(s)$:*

$$G(s) = \frac{K}{s-a}; u(s) = \frac{U}{s} \rightarrow y(s) = G(s)u(s) = \frac{K}{s-a}\frac{U}{s} = KU\frac{1}{s(s-a)}$$
$$y(s) = G(s)u(s) = KU\frac{1}{s(s-a)} = KU\left[\frac{C_1}{s} + \frac{C_2}{s-a}\right]$$
$$C_1 = \lim_{s=0} s\frac{1}{s(s-a)} = -\frac{1}{a} \quad ; \quad C_2 = \lim_{s=a}(s-a)\frac{1}{s(s-a)} = \frac{1}{a}$$
$$y(s) = G(s)u(s) = KU\frac{1}{s(s-a)} = KU\frac{1}{a}\left[-\frac{1}{s} + \frac{1}{s-a}\right]$$
$$y(t) = \mathcal{L}^{-1}y(s) = \mathcal{L}^{-1}\frac{KU}{a}\left[-\frac{1}{s} + \frac{1}{s-a}\right] = \frac{KU}{a}\left[-1 + e^{at}\right]$$

Note that the characteristic equation of the system is $s-a$, hence the pole is $p = a$. The step input brings only -1 times gain KU/a but the pole brings the time varying exponential term e^{at}.

$$\text{if } a > 0 \text{ when } t \rightarrow \infty \text{ than } e^{at} \rightarrow \infty \text{ hence } y(t \rightarrow \infty) = +\infty \qquad \text{Unstable system}$$
$$\text{if } a < 0 \text{ when } t \rightarrow \infty \text{ than } e^{-\infty} \rightarrow 0 \text{ hence } y(t \rightarrow \infty) = \frac{KU}{|a|} \qquad \text{Stable system}$$

Hence, the system is stable if the poles of its characteristic equation have negative real parts, regardless of the input. Regardless of the input, the system is unstable if the poles have positive real parts.

5.1.1 $z-$Domain

Similar to s domain, the closed loop transfer function of a system can be expressed as:

$$G_{cl}(z) = \frac{y(k)}{u(k)} = \frac{G(z)}{1 + G(z)H(z)} \tag{5.3}$$

The characteristic equation of the system with poles (z_n) are:

$$1 + G(z)H(z) = (z - z_1)(z - z_2)...(z - z_n) = 0 \tag{5.4}$$

Remembering that the relationship between s and z operators as $z = e^{sT}$, the poles in z domain becomes

$$z_1 = e^{p_1 T}, z_2 = e^{p_2 T}, ..., z_n = e^{p_n T} \tag{5.5}$$

Example 50 *Consider the same first order system*

$$G(s) = \frac{K}{s-a} \rightarrow \ s - a = 0 \ \text{ hence pole } p = a$$

The pole is transformed to z plane as $p_z = e^{pT} = e^{aT}$. Let a first order transfer function in z domain is:

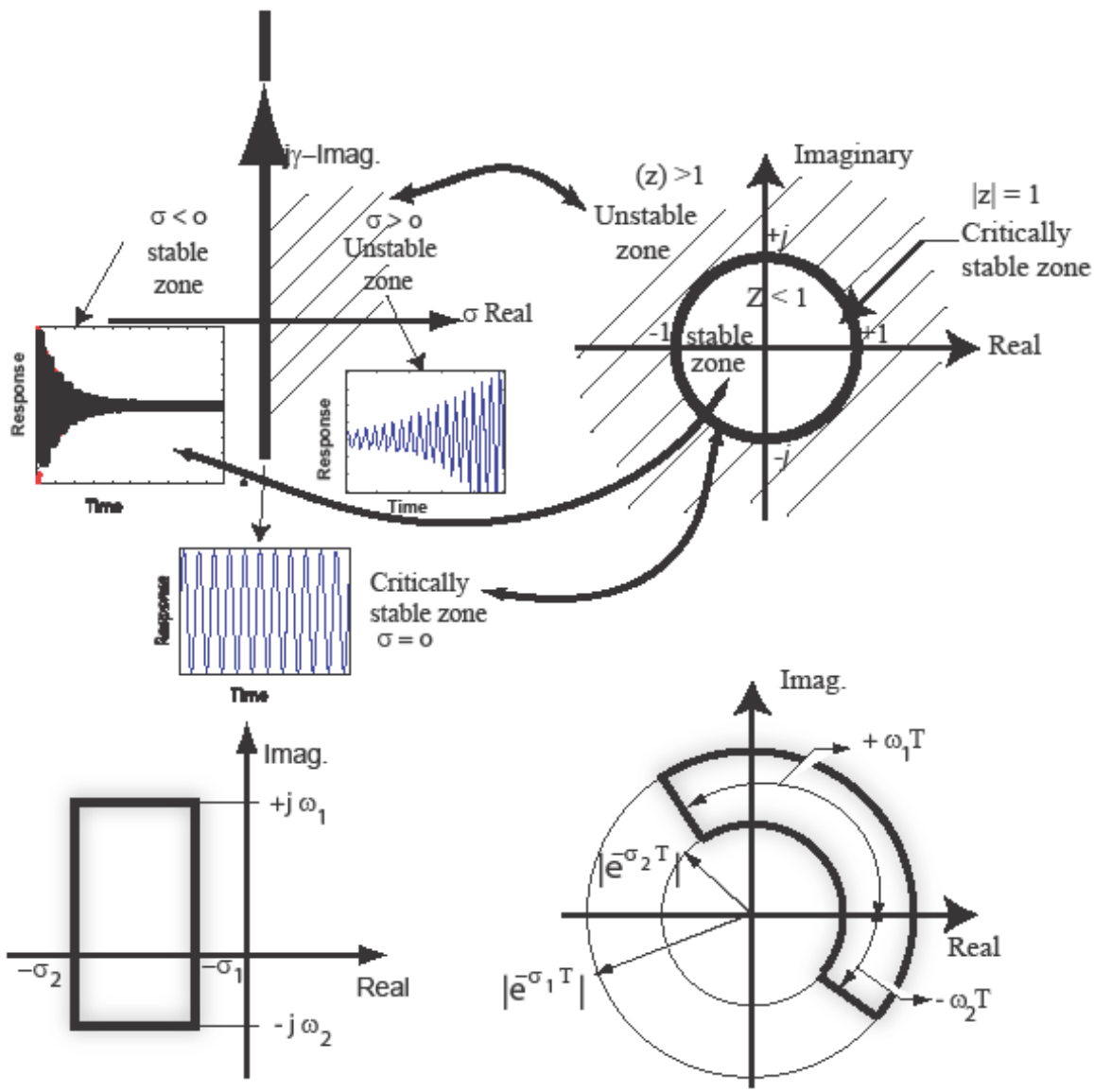


Figure 5.2: Mapping of poles in s and z planes.

$$G(z) = \frac{y(k)}{u(k)} = \frac{K}{z - p_z} = \frac{K}{z - e^{aT}}$$

- If $a > 0$ then $e^{aT} > 1$ which corresponds to unstable case like in s domain.
- If $a < 0$ then $e^{aT} < 1$ which corresponds to stable case like in s domain

If we remember that every pole in z domain, will have a corresponding impulse response. A system with transfer function $K/(z - p_z)$ will give an impulse response in z domain as :

$$G(z) = \frac{y(z)}{u(z)} = \frac{K}{z - e^{aT}} \rightarrow y(z) = \frac{K}{z - p_z} \rightarrow y(k) \rightarrow Kp_z^k$$

Hence if $p_z > 1$, the impulse response will have increasing amplitude as k grows, and the system becomes unstable.If $p_z < 1$,the impulse response will have decreasing amplitude (p_z^k) as k grows and the system will be stable.

Let the input $u(k) = U$ Step input

$$\begin{aligned}
G(z) &= \frac{y(k)}{u(k)} = \frac{K}{z - p_z} u(k) \rightarrow (z - p_z)y(k) = Ku(k) \\
y(k+1) &= +p_z y(k) + KU = e^{aT} y(k) + KU \\
\text{if } a &> 0 \text{ then } e^{aT} > 1 \text{ and } k \rightarrow +\infty \text{ and } e^{aT} y(k) \rightarrow +\infty \text{ hence the output } y(+\infty\ ) \rightarrow +\infty \\
\text{if } a &< 0 \text{ then } e^{aT} < 1 \text{ and } k \rightarrow +\infty \text{ and } e^{aT} y(k) \rightarrow 0 \text{ hence the output } y(+\infty\ ) \rightarrow KU
\end{aligned}$$

If $a > 0$ than $e^{aT} > 1$,as k increases, than the response $y(k+1)$ will keep increasing beyond the desired steady state response KU, hence the system will be unstable. if $a < 0$ than $e^{aT} < 1$,as k increases, than the response $y(k+1)$ will keep decreasing and the system will converge to the desired KU, hence the system will be stable.

5.2 Stability maps in s and z planes

Let a closed loop transfer function of the system in s and z domains be:

$$\begin{aligned}
G_{cl}(s) &= \frac{G(s)}{1 + G(s)H(s)} = \frac{s - b}{s - p_n} \text{ with pole } s = p_n \text{ and let } p_n = \sigma_n + j\gamma_n \\
G_{cl}(z) &= \frac{G(z)}{1 + G(z)H(z)} = \frac{z - c}{z - z_n} \text{with pole } z = z_n \text{ and let } z_n = e^{p_n T} = e^{\sigma_n T} \cdot e^{j\gamma_n T}
\end{aligned}$$

Consider a pole $p_n = \sigma_n + j\gamma_n$ in s plane (i.e. one of the roots of characteristic equation of the closed loop control system) in s domain and map the same pole to z domain as:

$$z_n = e^{p_n T} = e^{(\sigma_n + j\gamma_n)T} = e^{\sigma_n T} \cdot e^{j\gamma_n T} = e^{\sigma_n T}[\cos(\gamma_n T) + j\sin(\gamma_n T)]$$

The second part of the function will have identical values regardless of $\gamma_n T$ if $2\pi k$ intervals are added to the imaginary part, i.e.

$$e^{\sigma_n T}[\cos T(\gamma_n + k\frac{2\pi}{T}) + j\sin T(\gamma_n + k\frac{2\pi}{T})] \rightarrow k = 0, 1, 2, ..., +\infty$$

As a result, for every pole defined in s domain, we can have $+\infty$ number of corresponding poles in z domain. The poles will be apart from each other at amount of control frequency $(\omega_s = \frac{2\pi}{T})$. It is important to select a sampling frequency ω_s which is at least twice faster than the dominant pole of the system. The ideal

control frequency would be few times higher than the minimum value in order to minimize the phase delay in the control system. Consider a pole within the unit circle, i.e. $z_1 = e^{(\sigma+j\gamma)T} = e^{\sigma T}[\cos(\gamma T) + j\sin(\gamma T)] = a + jb$. The pole in polar coordinates can be represented by its magnitude $|z_1| = \sqrt{a^2+b^2}$, and the phase from the positive real axis in counter-clockwise direction is $\gamma T = \tan^{-1}\frac{b}{a}$. The location of the pole is on a circle which has radius of $|z_1|$, and it is γT[rad] counterclockwise away from the positive real axis.

Consider a pole $p_n = \sigma_n + j\gamma_n$ in s domain, and its equivalent in z domain as:

$$\begin{aligned} p_n &= \sigma_n + j\gamma_n \\ z_n &= e^{p_n T} = e^{\sigma_n T}\cdot e^{j\gamma_n T} = e^{\sigma T}[\cos(\gamma_n T) + j\sin(\gamma_n T)]. \\ |z_n| &= e^{\sigma_n T}, \quad \angle z_n = \gamma_n T \end{aligned}$$

Hence, the system is stable if the poles of its characteristic equation have negative real parts, regardless of the input. Regardless of the input, the system is unstable if the poles have positive real parts.

Condition 51 (Unstable Pole, $\sigma_n > 0$) *The system has a positive real pole which leads to output containing exponentially growing time response, i.e.*

$$y(t) = Ye^{\sigma_n t}\sin(\gamma_n t + \phi_d) \qquad t \to +\infty \quad y(t) = +\infty$$

hence the system is unstable regardless of its imaginary part γ_n. The pole in z domain becomes $z_n = e^{p_n T} = e^{\sigma_n T}\cdot e^{j\gamma_n T}$. The magnitude of $\left|e^{j\gamma_n T}\right| = \sqrt{\cos^2(\gamma_n T) + \sin^2(\gamma_n T)} = 1$. Since $\sigma_n > 0$, $e^{\sigma_n T} > 1$, which leads to a pole with a magnitude always greater than unity, $|z_n| > 1$.

$$\begin{aligned} z_n &= e^{p_n T} = e^{\sigma_n T}\cdot e^{j\gamma_n T} \\ \left|e^{j\gamma_n T}\right| &= \sqrt{\cos^2(\gamma_n T) + \sin^2(\gamma_n T)} = 1. \\ \sigma_n &> 0, \quad e^{\sigma_n T} > 1 \quad \to |z_n| > 1 \end{aligned}$$

In summary, unstable poles will lie on the right hand side of the s plane, or outside the unit circle in z plane.

Condition 52 (Critically stable pole, $\sigma_n = 0$) *The pole does not have a real part, hence the pole is on the imaginary axis which represents critical stability. The system will have indefinitely oscillating response regardless of the input, i.e. $y(t) = Ye^{j\gamma t} = Y\sin(\gamma t + \phi_d)$ where the γ corresponds to oscillation frequency. The same pole lies on the unit circle of z plane.*

$$\sigma_n = 0 \ \to z_n = e^{\sigma_n T}\cdot e^{j\gamma_n T} = 1\cdot e^{j\gamma_n T} \to |z_n| = \left|e^{j\gamma_n T}\right| = 1 \ , \angle z_n = \gamma_n T$$

Condition 53 (Stable Pole, $\sigma_n < 0$) *The pole has a negative real part, and the system is stable. The system will have an output with a component determined by the pole, i.e. $y(t) = Ye^{\sigma_n t}e^{j\gamma_n t} = Ye^{\sigma_n t}\sin(\gamma_n t + \phi_d)$. Since $\sigma_n < 0$, $t \to +\infty$ $y(t) = 0$, hence the influence of the pole on the steady state output response of the system disappears. The same pole lies within the unit circle of z plane, i.e. $z_n = e^{\sigma_n T}\cdot e^{j\gamma_n T}, \sigma_n < 0 \to e^{\sigma_n T} < 1$. Hence $|z_n| = \left|e^{\sigma_n T}\right| < 1$,which means that any pole inside the unit circle is stable. If the pole does not have any imaginary part, the system will not exhibit any oscillatory (i.e. sinusoidal) response.*

$$\begin{aligned} y(t) &= Ye^{\sigma_n t}e^{j\gamma_n t} = Ye^{\sigma_n t}\sin(\gamma_n t + \phi_d) \\ \sigma_n &< 0, \qquad y(t \to +\infty) = Ye^{-|\sigma_n| t}\sin(\gamma_n t + \phi_d) = 0 \\ z_n &= e^{\sigma_n T}\cdot e^{j\gamma_n T}, \sigma_n < 0 \to e^{\sigma_n T} < 1 \to |z_n| = \left|e^{\sigma_n T}\right| < 1 \end{aligned}$$

The control systems must be designed to be stable, hence the real part of all of its closed loop poles must be negative in s plane and remain within the unit circle in z plane. If we concentrate on the stable

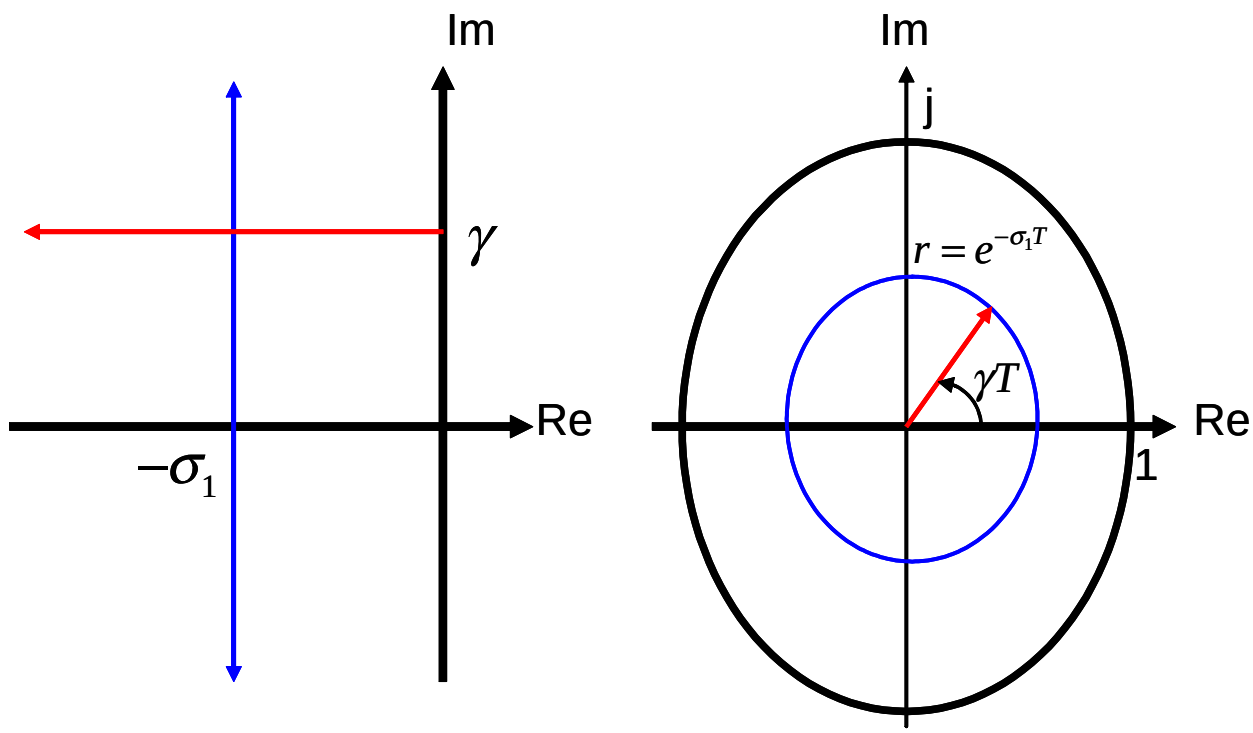


Figure 5.3: Constant attenuation (-$\sigma_1, e^{-\sigma_1 T}$) and constant frequency ($j\gamma, \measuredangle\gamma T$) loci in s and z planes, respectively.

zones of s plane (i.e. left hand side of the imaginary axis where the real part is negative) or in z plane (i.e. within the unit circle), the following properties must be remembered, see Figure 5.3. A vertical line with a constant negative real value ($-\sigma$=constant) in s plane corresponds to a circle with a radius ($e^{-\sigma T}$) less than unity in z plane, and they represent *constant attenuation loci.* A horizontal line with a constant imaginary part ($j\gamma$ =constant) *in* s plane, which corresponds to a straight line with a phase angle of $\measuredangle\gamma T$ measured counter clockwise from the real axis in z plane, and they represent *constant frequency loci.*

Example 54 *Consider a region of poles in s plane, which is bounded by* $(-\sigma_1, -\sigma_2)$ *in real axis, and* $(j\omega_1, -j\omega_2)$ *in imaginary axis as shown in Figure 5.2 The poles in z domain are mapped as* $z_1 = e^{s_1 T} = e^{-\sigma_1 T} e^{j\omega_1 T} = |z_1| \langle \omega_1 T$ *,* $z_2 = e^{s_2 T} = e^{-\sigma_2 T} e^{-j\omega 2T} = |z_2| \langle -\omega_2 T$ *. The poles stay between the two circles, which have radius of* $e^{s_1 T}, e^{-\sigma_2 T}$ *and bounded by a clockwise phase of* $\langle \omega_1 T$ *, and counterclockwise of phase* $\langle -\omega_2 T$ *.*

5.3 Summary: Stability of Closed Loop Control Systems

Consider the closed loop control system shown in Figure 5.1.

$$\text{Closed Loop Transfer Function } \; G_{cl} = \frac{y}{u} = \frac{DG_0}{1 + DG_0H}$$

$$\text{Characteristic Equation:} 1 + DG_0H$$

$$\text{Poles of the closed loop system in } s \text{ domain:} 1 + D(s)G_0(s)H(s) = (s - p_1)(s - p_2)...(s - p_n) = 0$$

$$\text{Poles of the closed loop system in } z \text{ domain:} 1 + D(z)G_0(z)H(z) = (z - z_1)(z - z_2)...(z - z_n) = 0$$

$$\text{Mapping from } s \text{ to } z \text{ plane } (p_n = \sigma_n + j\gamma_n)\text{: } z = e^{\,sT} = e^{p_nT} = e^{\sigma_nT} \cdot e^{j\gamma_nT}$$

$$\text{Unstable } \sigma_n > 0\text{; Critically stable } \sigma_n = 0 \text{ ; Stable } \sigma_n < 0$$

Example 55 *Sample stability case in s domain- Closed loop transfer function with two poles*

$$G_{cl}(s) = \frac{y(s)}{u(s)} = \frac{K}{(s-a)(s-b)} \rightarrow u(s) = \frac{U}{s} \;\; \text{Step input}$$

$$y(s) = \frac{K}{(s-a)(s-b)}\frac{U}{s} = \frac{C_1}{s} + \frac{C_2}{s-a} + \frac{C_3}{s-b}$$

$$C_1 = \lim_{s=0} s\frac{K}{(s-a)(s-b)}\frac{U}{s} = \frac{KU}{ab}$$

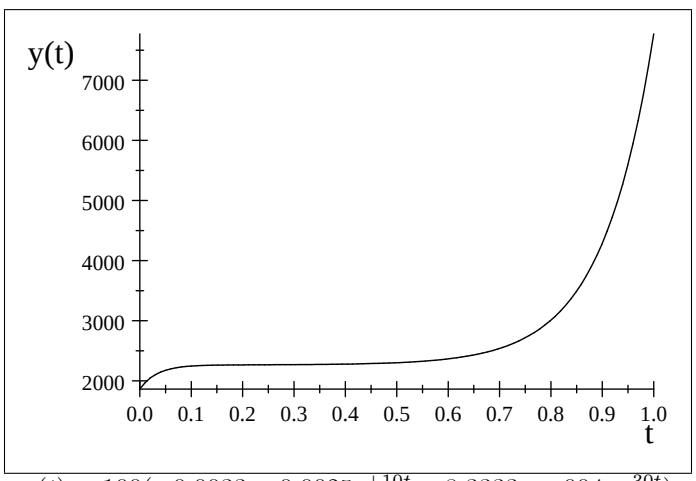
$$C_2 = \lim_{s=a}(s-a)\frac{K}{(s-a)(s-b)}\frac{U}{s} = \frac{KU}{a(a-b)}$$

$$C_3 = \lim_{s=b}(s-b)\frac{K}{(s-a)(s-b)}\frac{U}{s} = \frac{-KU}{b(a-b)}$$

$$y(t) = \mathcal{L}^{-1}y(s) = C_1 + C_2e^{at} + C_3e^{bt}$$

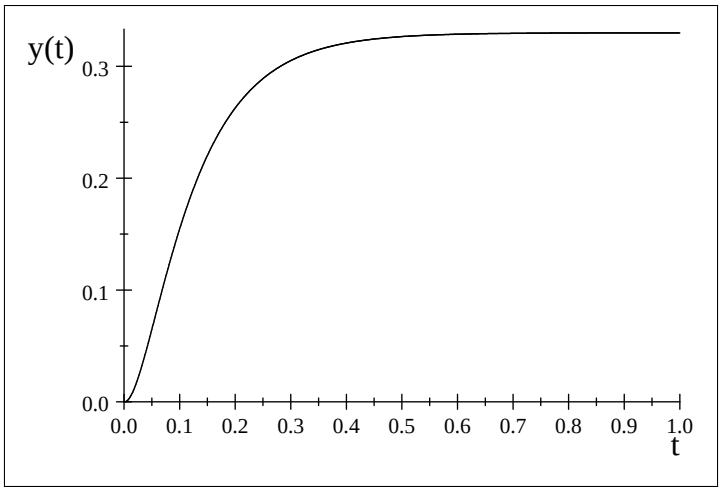
Let $K = 50$ and $U = 2$

Case 56 *Unstable condition $a > 0$ or $b > 0$ Let $a = 10$ and $b = -30$*



$$y(t) = 100(-0.0033 + 0.0025e^{+10t} + 8.3333e - 004e^{-30t})$$

Case 57 *Stable condition $a < 0$ and $b < 0$ Let $a = -10$ and $b = -30$*



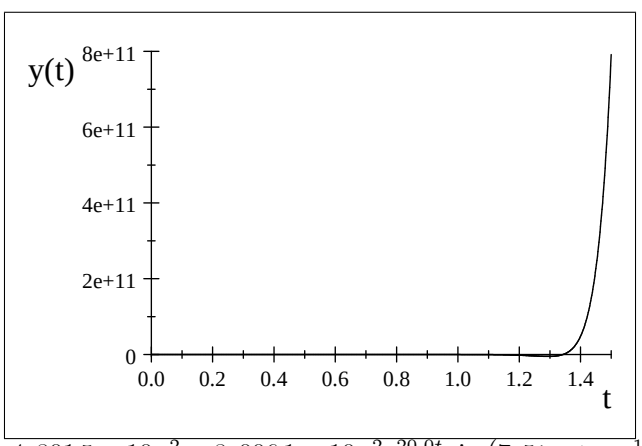
Stable system

Example 58 ***Complex Poles***

$$
\begin{aligned}
y(s) &= \frac{\omega_d}{s^2 + 2\zeta\omega_n s + \omega_n^2}\frac{U}{s} \\
&= \frac{\omega_d}{(s-p)(s-p^*)}\frac{U}{s} \rightarrow p, p^* = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} \rightarrow \zeta < 1 \\
\text{Response } y(t) &= C_1 + C_2 e^{pt} + C_3 e^{p^* t} = C_1 + C_2 e^{(-\zeta\omega_n + j\omega_d)t} + C_3 e^{(-\zeta\omega_n - j\omega_d)t} = \\
\text{Let } U &= 1, \omega_n = 12.5 \rightarrow \omega_d = 7.5
\end{aligned}
$$

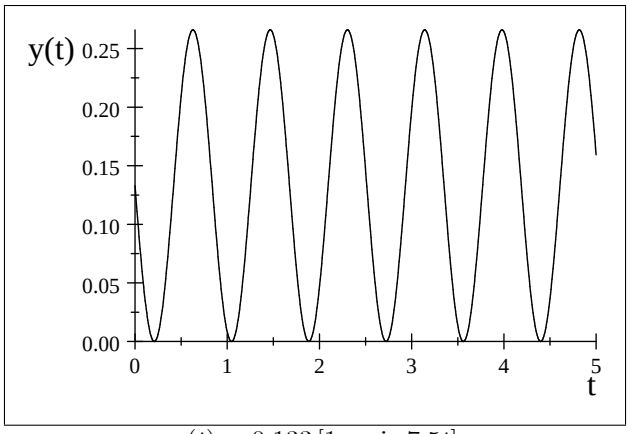
	Poles	
Unstable condition	$\zeta < 0, \zeta = -0.8$	$\rightarrow\ p, p^* = 10 \pm j7.5$
Critically stable condition	$\zeta = 0,$	$\rightarrow\ p, p^* = \pm j12.5$
Stable condition	$\zeta > 0, \zeta = +0.8$	$\rightarrow\ p, p^* = -10 \pm j7.5$

Case 59 ***Unstable condition- Unstable Poles with positive real parts*** $\zeta = -0.8 \rightarrow p, p^* = 10 \pm j7.5$



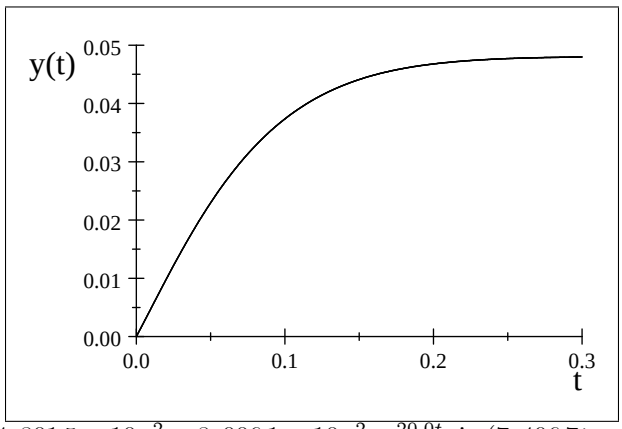
$$y(t) = 4.801\,5 \times 10^{-2} - 8.006\,1 \times 10^{-2} e^{20.0t} \sin\left(7.5t - \tan^{-1} 0.75\right)$$

Case 60 ***Critically stable condition - Critically stable poles*** $\zeta = 0,\ p, p^* = \pm j12.5$



$y(t) = 0.133\,[1 - \sin 7.5t]$

Case 61 ***Stable condition- Stable and underdamped poles with negative real parts*** $\zeta > 0, \zeta = +0.8 \rightarrow p, p^* = -10 \pm j7.5$



$y(t) = 4.801\,5 \times 10^{-2} - 8.006\,1 \times 10^{-2} e^{-20.0t} \sin(7.496\,7t + 0.643\,5)$

5.4 Root Locus in Laplace (s) and Discrete (z) Domain

5.4.1 Introduction

Let the complex root of a characteristic equation for a second order under-damped system be $p_n = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2} = -\zeta\omega_n + j\omega_d$ where $\zeta < 1$ and $\omega_d = \omega_n\sqrt{1-\zeta^2}$. The mapping of poles between s and z planes are given in Figure 5.4.

Any line which has constant damping ratio is called the phase (ψ) loci of the pole, and it is measured counter-clockwise from the real axis in s plane,

$$\tan\psi = \frac{\omega_d}{-\zeta\omega_n} = \frac{\omega_n\sqrt{1-\zeta^2}}{-\zeta\omega_n} = \frac{\sqrt{1-\zeta^2}}{-\zeta}$$

The root locus of the pole can be plotted as a function of varying natural frequency while keeping the damping ratio constant along the line with a phase angle of (ψ) as shown in Figure 5.4.The damping increases as the pole moves away from the imaginary axis in the real negative direction. The damping coefficient decreases as the phase (ψ) approaches to the imaginary axis and the overshoot of the oscillatory step response increases. The system is over damped ($\zeta > 1$) if the pole does not have a complex part, hence the system response does not contain any oscillation.

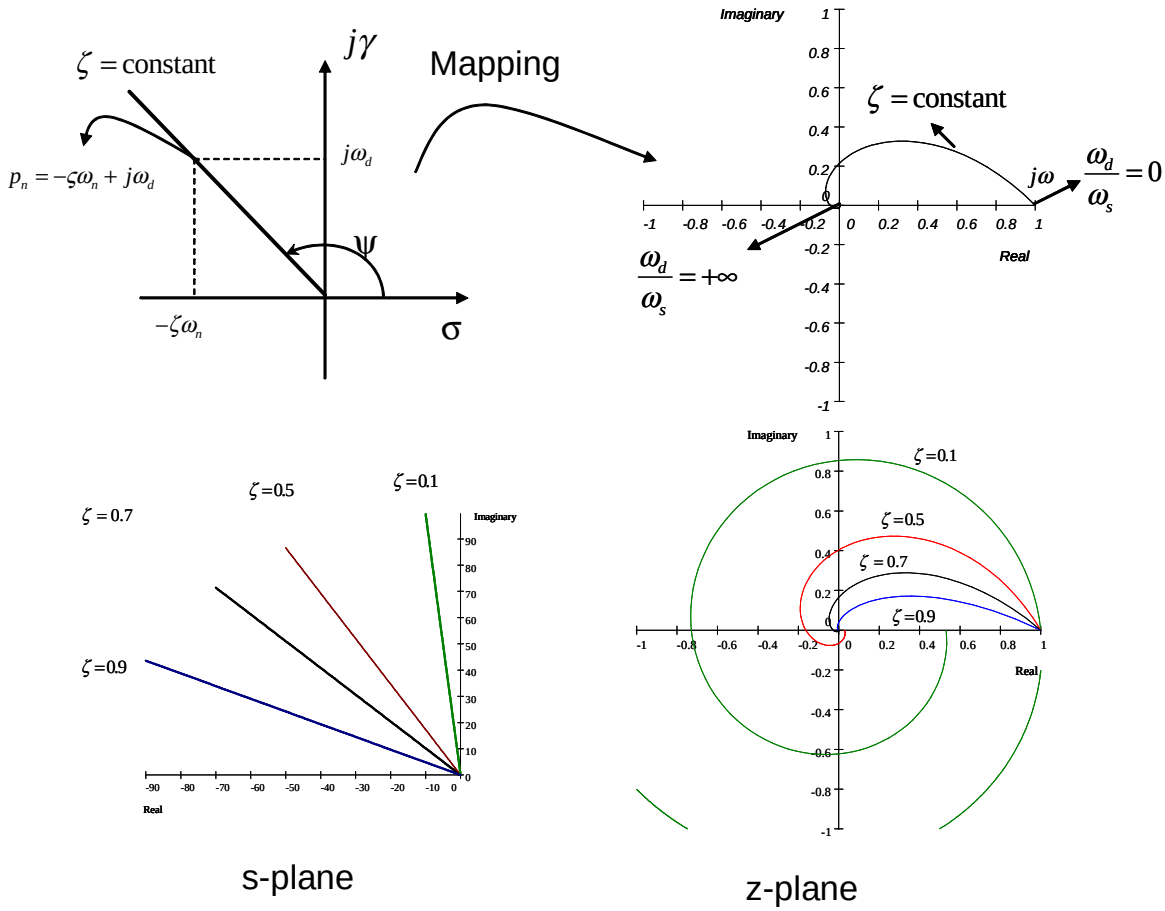


Figure 5.4: Mapping between s and z planes. Constant damping loci in s and z planes.

The root in the continuous time domain can be mapped to z domain as $z = e^{sT} = e^{-\zeta\omega_n T}e^{j\omega_d T}$. If the sampling interval is T, the sampling frequency is $\omega_s = 2\pi/T$. The root in z domain can be re-written by substituting $\omega_n = \omega_d/\sqrt{1-\zeta^2}$ and $T = 2\pi/\omega_s$ as:

$$z = e^{-\zeta\omega_n T}e^{j\omega_d T} = \exp\left(\frac{-\zeta\omega_d}{\sqrt{1-\zeta^2}}\frac{2\pi}{\omega_s} + j\omega_d\frac{2\pi}{\omega_s}\right) \tag{5.6}$$

$$= \exp\left(\frac{-2\pi\zeta}{\sqrt{1-\zeta^2}}\frac{\omega_d}{\omega_s} + j2\pi\frac{\omega_d}{\omega_s}\right) = \exp\left(\frac{-2\pi\zeta}{\sqrt{1-\zeta^2}}\frac{\omega_d}{\omega_s}\right)\exp\left(j2\pi\frac{\omega_d}{\omega_s}\right) \tag{5.7}$$

$$= \exp\left(\frac{-2\pi\zeta}{\sqrt{1-\zeta^2}}\frac{\omega_d}{\omega_s}\right)\left[\cos\left(2\pi\frac{\omega_d}{\omega_s}\right) + j\sin\left(j2\pi\frac{\omega_d}{\omega_s}\right)\right] \tag{5.8}$$

$$|z| = \exp\left(\frac{-2\pi\zeta}{\sqrt{1-\zeta^2}}\frac{\omega_d}{\omega_s}\right), \psi = 2\pi\frac{\omega_d}{\omega_s} \rightarrow z = |z|\angle 2\pi\frac{\omega_d}{\omega_s} \tag{5.9}$$

where the magnitude of the root is $|z| = \exp\left(\frac{-2\pi\zeta}{\sqrt{1-\zeta^2}}\frac{\omega_d}{\omega_s}\right)$ and the phase is $\psi = 2\pi\frac{\omega_d}{\omega_s}$ which are used in locating the pole on the z plane. Since the control interval T is constant, the sampling frequency $\omega_s = 2\pi/T$ is constant as well. One can fix either the damping ratio (ζ) or the ratio of damped natural frequency to sampling frequency to plot the variation of the poles in either s or z plane which is called *root locus* of the system, see Figure 5.4.

Example 62 *Let $\zeta = 0.707$ and $x = \frac{\omega_d}{\omega_s} = 0, 0.1, 0.2, ..., 1$.*

$$z = \exp\left(\frac{-2\pi 0.7}{\sqrt{1-0.7^2}}x + j2\pi x\right) \tag{5.10}$$

$$= \cos(2\pi x)\cdot\exp\left(\frac{-2\pi 0.7}{\sqrt{1-0.7^2}}x\right) + j\sin(2\pi x)\cdot\exp\left(\frac{-2\pi 0.7}{\sqrt{1-0.7^2}}x\right) \tag{5.11}$$

which can be plotted as a function of $x = \frac{\omega_d}{\omega_s}$ by allowing it to change between $(0, +1)$.

$$\textit{Root Locus Graph}\ \left(\cos(2\pi x)\cdot e^{\frac{-2\pi 0.7}{\sqrt{1-0.7^2}}x}, \sin(2\pi x)\, e^{\frac{-2\pi 0.7}{\sqrt{1-0.7^2}}x}\right)$$

which is shown in Figure 5.4. The damping ratios can be changed within an acceptable range at fixed increments, which leads to a series of root locus plots of the poles.

5.4.2 Root Locus

Consider a closed loop system with the following transfer function:

$$G_{cl}(s) = \frac{KG}{1+KGH} \tag{5.12}$$

where the characteristic equation is $1+KGH = 0$ where K is the proportional gain of the control system. For each value of K, we have a set of poles for the characteristic equation. Some values of the K will lead to unstable poles (i.e. positive real part in s plane, or larger than unity magnitude in z domain). By changing the values of $K = 0$ to ∞, we can scan all the roots of the characteristic equation. The root locus routine in MATLAB allows the prediction of poles for a each K. The computation steps in root locus method is summarized as follows:

- Express the characteristic equation as $1 + KGH = 0$ where K is the total gain of the system which needs to be identified by the control designer.

- G is the forward and H is the feedback transfer functions in zero and pole representation, and KGH is the loop transfer function of the system. Let $H = 1$ for simplicity.

$$\begin{aligned}
\text{Loop transfer function } KG &= K\frac{\prod_{i=1}^{m}(s-z_i)}{\prod_{i=1}^{n}(s-p_i)} \\
1+KG &= 1+K\frac{(s-z_1)(s-z_2)..(s-z_m)}{(s-p_1)(s-p_2)...(s-p_n)} = 1+K\frac{\prod_{i=1}^{m}(s-z_i)}{\prod_{i=1}^{n}(s-p_i)} = \frac{\prod_{i=1}^{n}(s-p_i)+K\prod_{i=1}^{m}(s-z_i)}{\prod_{i=1}^{n}(s-p_i)}
\end{aligned}$$

Closed loop transfer function becomes:

$$G_{cl}(s) = \frac{KG}{1+KG} = \frac{K\prod_{i=1}^{m}(s-z_i)}{\prod_{i=1}^{n}(s-p_i)+K\prod_{i=1}^{m}(s-z_i)} \tag{5.13}$$

- Find the open loop zeros (z_i) and open loop poles (p_i) of the system on the s plane.
- The locus of the roots of the characteristic equation $1 + KGH = 0$ starts at the open loop poles (p_i) and ends at the open loop zeros (z_i) as the gain K is increased from zero to infinity ($K = 0 \rightarrow +\infty$). The root locus will have as many branches as the number of roots in the characteristic equation.
- Root loci on the real axis are identified by open loop poles and zeros which lie on the real axis.
- Determine the asymptotes of the root loci. The root loci for every large value of z must be asymptotic to the straight lines whose angles are given as follows:

$$\textit{Angle of Asymptote } \theta = \frac{\pm 180^0(2N+1)}{n-m}, \qquad N = 0, 1, 2, ..$$

$$\textit{Location of Asymptote on real axis } \alpha = \frac{\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} z_i}{n-m}$$

where $N = 0$ corresponds to the asymptote with the smallest angle with the real axis, n is the number of poles and m is the number of zeros of open loop transfer function. p_i and z_i are the values of poles and zeros of open loop transfer function.

Further details of root locus can be found in any standard control textbook, such as 'Digital Control" by Ogata [1].

Example 63 *Let the loop transfer function of a system be $GH = 1/[s(s+1)]$ and $H = 1$,hence the open loop poles are at $s_1 = p_1 = 0$ and $s_2 = p_2 = -1$. The open loop zero does not exist. i.e. at the infinity. The closed loop transfer function of the system is :*

$$G_{cl}(s) = \frac{KG(s)}{1+KG(s)} = \frac{\frac{K}{s(s+1)}}{1+\frac{K}{s(s+1)}} = \frac{K}{s(s+1)+K} = \frac{K}{s^2+s+K} \tag{5.14}$$

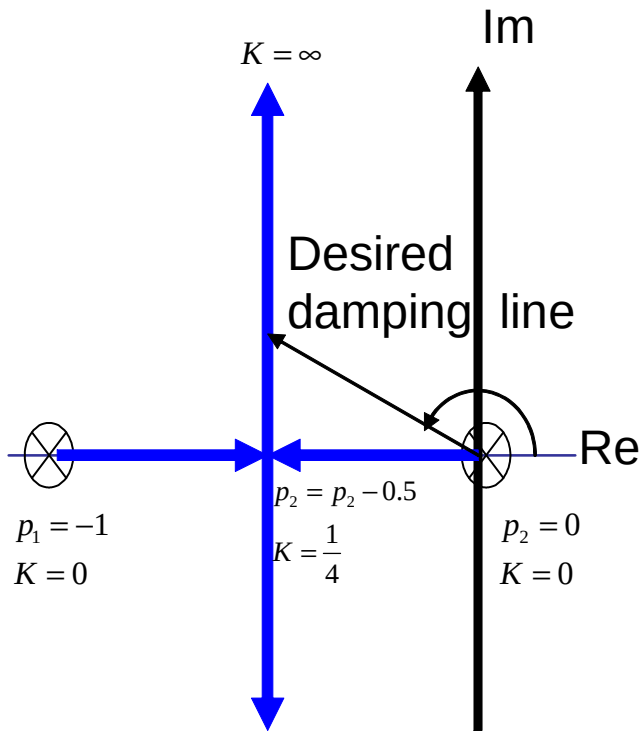


Figure 5.5: Root locus of $s^2 + s + K = 0 \rightarrow p_1, p_2 = -\frac{1}{2} \pm \frac{1}{2}\sqrt{1-4K}$The gain K can be identified by specifying the damping ratio.

Characteristic equation:

$$s(s+1) + K = (s - p_1)(s - p_2) = 0 \tag{5.15}$$

$$p_1 = -\frac{1}{2} + \frac{1}{2}\sqrt{1-4K}, \quad p_2 = -\frac{1}{2} - \frac{1}{2}\sqrt{1-4K} \tag{5.16}$$

K	$p_1 = -0.5 - \frac{1}{2}\sqrt{1-4K}$	$p_2 = -0.5 + \frac{1}{2}\sqrt{1-4K}$	$\left\lvert \frac{K}{s(s+1)} \right\rvert$	ψ_2	ψ_1
0	-1	0	0	0	-180
$\frac{1}{4}$	-0.5	-0.5	1	-180	-180
1	$-0.5 - j0.866\,03$	$-0.5 + j0.866\,03$	1	-120	-240
2	$-0.5 - j1.322\,9$	$-0.5 + j1.322\,9$	1	-110	-250
10	$-0.5 - j3.122\,5$	$-0.5 - j3.122\,5$	1	-99	-261

(5.17)

The root locus of the system is shown in Figure 5.5. When $K = 0$, the poles of the closed and open loops are the same ($p_1 = 0, p_2 = -1$). The break away point is when $K = 1/4$ where the poles approach from -1 towards right and from 0 towards left and booth meet at $p_1 = p_2 = -1/2$ where the system is critically damped ($\zeta = 1$). As K increased towards ∞, the real part remains the same (-1/2), but the imaginary part of the first pole $p_1$moves towards $+\infty$, the imaginary part of the second pole p_2 moves towards $-\infty$ and the system is under damped.

If a proportional controller with gain K needs to be designed with a performance criteria of $M_p = 2\%$ overshoot, the corresponding damping ratio is :

$$\ln M_p = \ln 0.02 = -\frac{\zeta\pi}{\sqrt{1-\zeta^2}} \rightarrow (-3.912)^2(1-\zeta^2) = \zeta^2\pi^2 \tag{5.18}$$

$$15.304 = (\pi^2 + 15.304)\zeta^2 \rightarrow \zeta = \sqrt{15.304/\left(\pi^2 + 15.304\right)} = 0.78 \tag{5.19}$$

By equating the actual and desired characteristic equations:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + s + K \quad ; \quad 2\zeta\omega_n = 1 \quad ; \omega_n^2 = K \tag{5.20}$$

$$\omega_n = \frac{1}{2\zeta} = \frac{1}{2 \times 0.78} = 0.64103 \quad ; \quad K = 0.64103^2 = 0.41092 \tag{5.21}$$

Note that any point on the root loci satisfies the following condition:

$$\langle G(s)H(s) = \left\langle \frac{K}{s(s+1)} = \left\langle \frac{1}{s} + \left\langle \frac{1}{s+1} = -\langle s - \langle (s+1) = \pm 180(2k+1) \quad k = 0, 1, 2, ...\right.\right.\right.$$

$$|G(s)H(s)| = 1 \rightarrow \left| \frac{K}{s(s+1)} \right|_{s=p_1,p_2} = 1$$

The asymptotes can be found as follows:

$$Angle\ \theta = \frac{\pm 180^0(2N+1)}{n-m} = \frac{\pm 180^0(0+1)}{2-0} = \pm 90^0, \qquad N = 0, n = 2, m = 0..$$

$$\text{Real axis intercept} \quad \alpha = \frac{\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} z_i}{n-m} = \frac{(0-1)-0}{2-0} = -0.5$$

MATLAB has a convenient tool box to identify roots of polynomials as well as plotting the root locus of a characteristic equation. The characteristic equation can be expressed in the following form by collecting the terms with and without gain K :

$$1 + KG(s)H(s) = Q(s) + KR(s) = 0 \tag{5.22}$$

When $K = 0$, the roots of the characteristic equation are determined by $Q(s)$ only. When $K = \infty$, $Q(s)$ becomes negligible hence the $R(s)$ determines the roots. In short, the roots of the characteristic equation starts at $K = 0 \rightarrow Q(s) = 0$, and ends at $K = \infty \rightarrow R(s) = 0$.

Example 64 *Open loop transfer function of the system is* $KG(s) = K\frac{R(s)}{Q(s)} = K\frac{s+4}{s(s+1)(s+2)}$. *The closed loop transfer function of the system with unity feedback* ($H = 1$) *becomes:*

$$G_{cl}(s) = \frac{KG(s)}{1+KG(s)} = \frac{K\frac{R(s)}{Q(s)}}{1+K\frac{R(s)}{Q(s)}} = \frac{KR(s)}{Q(s)+KR(s)}$$

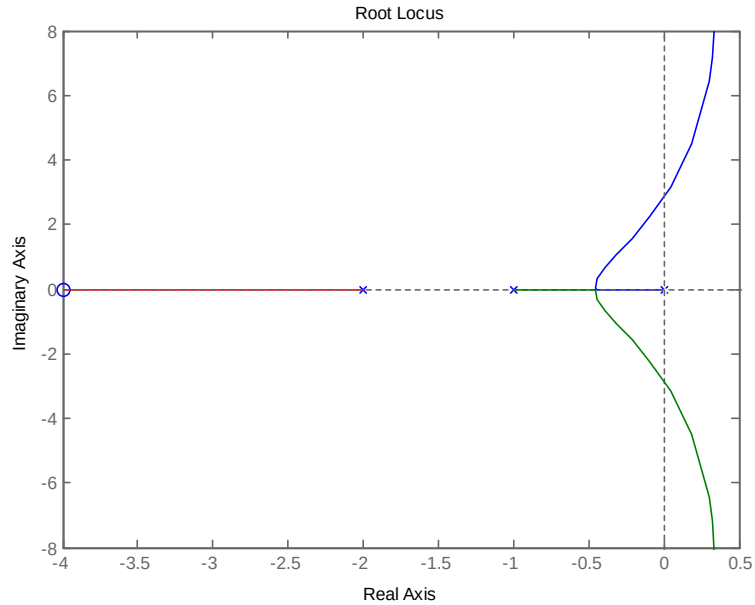
$$= \frac{K\frac{s+4}{s(s+1)(s+2)}}{1+K\frac{s+4}{s(s+1)(s+2)}} = \frac{K(s+4)}{s(s+1)(s+2)+K(s+4)} = \frac{KR(s)}{Q(s)+KR(s)}$$

The characteristic equation of the closed loop transfer function becomes:

$$1 + KGH(s) = s(s+1)(s+2) + K(s+4) = Q(s) + KR(s) = 0$$

$$Q(s) = s(s+1)(s+2) = s^3 + 3s^2 + 2s, \quad R(s) = (s+4)$$

where the roots of $Q(s)$ and $R(s)$ are found from *roots(Q) function of MATLAB. The root loci starts from the roots of* $\{K = 0 \rightarrow Q(s) = [0, -1, -2]\ \}$ and ends at $\{K = \infty \rightarrow R(s) = [-4]\ \}$. Since the characteristic



equation is third order, the root loci has to split into 3 branches. The MATLAB code and corresponding root locus are given below:

```
% Example Root Locus
% Polynomials of characteristic equation
R=[1 4]  ;        % Numerator
Q=[ 1 3 2 0] ; % Denominator
G=tf(R,Q) ;    % Polynomial style transfer function
Gpz=zpk(G);    % Zero Pole style transfer function
% Poles: s (s+2) (s+1)  , zero: (s+4)
roots(Q) ; % Gives the roots of Q which are 0,-1,-2
rlocus(R,Q) ; % plots the root locus diagram as the gain is increased K = 0 → ∞
```

5.5 Nyquist Stability Criterion

The close loop transfer function of a system can be expressed as:

$$G_{cl}(s) = \frac{KG}{1 + KGH} \tag{5.23}$$

where the denominator can be expanded as:

$$1 + KGH = 1 + K\frac{(s - z_1)(s - z_2)..(s - z_m)}{(s - p_1)(s - p_2)...(s + -p_n)} = 1 + K\frac{\prod_{i=1}^{m}(s - z_i)}{\prod_{i=1}^{n}(s - p_i)} = 0 \tag{5.24}$$

where z_i are the zeros and p_i are the poles of the open loop system. The poles of the closed loop system, which determine the stability, are found from the roots of the characteristic equation $(1 + KGH = 0)$. Any pole would satisfy the condition that $KGH = -1$. If any pole is on the right hand side of the complex

plane in s domain or outside unit circle in z domain, it is unstable. Nyquist theory dictates that if the open loop frequency response of the system (KGH), that is as the excitation frequency varies from zero to infinity $(0 \to \infty)$, encircles -1 in the complex plane, the system becomes unstable. The margin of stability is measured by checking how far the gain K can be increased until the frequency response curve approaches to -1 in complex plane.

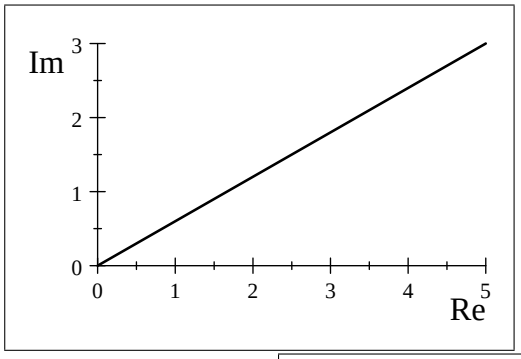
It is essential to introduce vector representation of complex numbers and conformal mapping before the Nyquist stability theory is understood clearly. A brief review of the background material and Nyquist stability theory will be introduced here, but the students are encouraged to study more details which can be found in any control text book [3],[5],[4],[1].

5.5.1 Vector Representation of Poles and Zeros

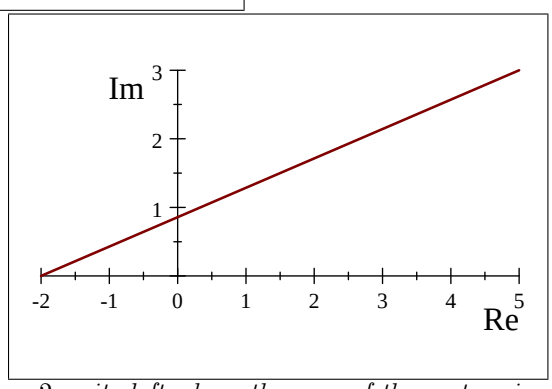
Consider a complex number $s = \sigma + j\omega$ and a function $F(s) = s + a$. If we substitute $s = \sigma + j\omega$ into $F(s)$,

$$s = \sigma + j\omega \to F(s) = s + a = (\sigma + a) + j\omega \tag{5.25}$$

Example 65 $s = 5 + j3$‘



When mapped on $F(s) = s + 2 = (5 + 2) + j3$



The vector is simply translated to the left by $a = 2$ *units left where the zero of the system is* $s = -2$.

$F(s) = s + 2|_{s=5+j3}$ is a complex function which can be represented by a vector drawn from its zero $(s = -2)$ to the point $s = 5 + 3j$

The purpose of this exercise is to show that the stability of the system can be investigated indirectly by checking the $F(s)$ contour, as opposed to checking the roots of the characteristic equation in s plane which may be difficult.

Consider a general function with multiple zeros and poles as where each s is a complex number:

$$F(s) = \frac{(s - z_1)(s - z_2)..(s - z_m)}{(s - p_1)(s - p_2)...(s - p_n)} = \frac{\prod_{i=1}^{m}(s - z_i)}{\prod_{j=1}^{n}(s - p_j)} \tag{5.26}$$

The magnitude and phase of the vector $F(s)$ can be found as :

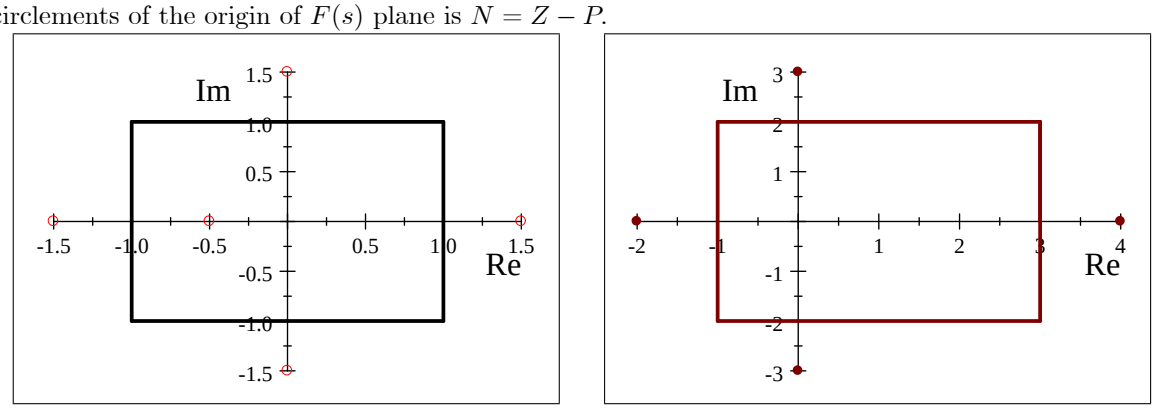
$$F(s) = |F(s)| \angle F(s)$$
$$|F(s)| = \frac{\prod_{i=1}^{m} |(s - z_i)|}{\prod_{j=1}^{n} |(s - p_j)|}, \quad \phi_f = \sum_{i=1}^{m} \angle(s - z_i) - \sum_{j=1}^{n} \angle(s - p_j) \tag{5.27}$$

The complex function $F(s)$ can be plotted by its vector whose amplitude and phase angle (+ from real axis in CCW direction) can be evaluated. The frequency response of $F(s)$ can be plotted by substituting $F(s \to j\omega)$. When the magnitude and phase are plotted separately as a function of frequency (ω), it is called Bode plot. If it is plotted in polar coordinates with real and imaginary parts as the frequency is varied from $-+\infty \leq \omega \leq +\infty$, it is called Nyquist diagram of $F(s)$. The Nyquist diagram can be plotted by the vector of $F(j\omega)$ which is expressed by its magnitude and phase.

5.5.2 Conformal Mapping

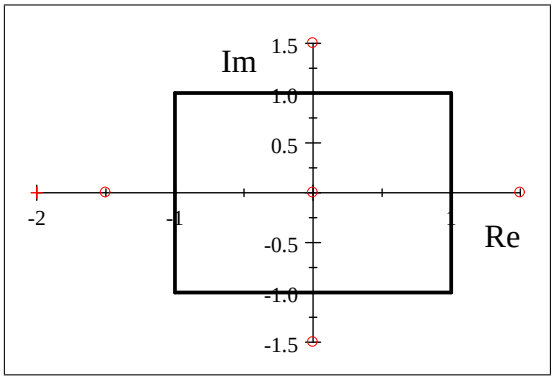
Lets consider s plane with $s = \sigma + j\omega$ which will be mapped to $F(s) = 2s + 1 = 2(\sigma + 1) + j2\omega = u + jv$. The root of $F(s)$ is $z_i = -1/2$. Let s changes within a contour bounded by

$s \in [(1, 0); (1, -j); (0, -j); (-1, -j); (-1, 0); (-1, j); (j, 0); (1, j)]$ in clockwise direction. The contour in s plane encircles the zero ($z_i = -1/2$) completely once. If the number of zero is Z, and pole is P within the closed contour in s plane, the difference for this application is $Z - P = 1 - 0 = 1$. When s values are mapped into $F(s) = 2s + 1$,the corresponding contour is also similar but it encircles the origin of $F(s)$ once, which is similar to the difference between the number of zeros and poles in s plane. Hence, the number of encirclements of the origin of $F(s)$ plane is $N = Z - P$.

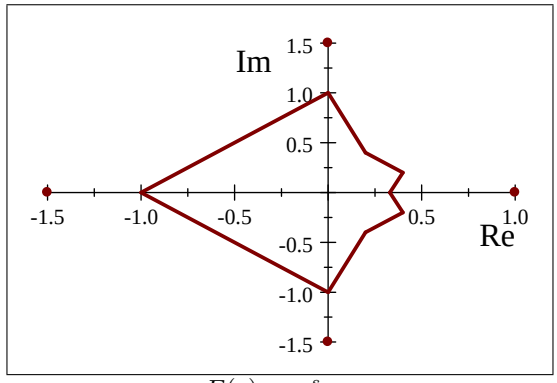


Example 66 *Consider a case with a zero and pole* $F(s) = \frac{s}{s+2} \to s = \sigma + j\omega \quad (-1 \leq \sigma \leq 1, -1 \leq \omega \leq 1)$.

If we consider the s plane, we have a zero at $z = 0$, *and pole at* $p = -2$. *The s plane contour encircles only the zero, hence* $Z = 1, P = 0$. *The expected encirclement of the origin of* $F(s)$ *must be ONE* ($N = Z - P = 1 - 0 = 1$), *and* $F(s)$ *encircles the origin only ONCE as shown by the contours plotted below.*



$s = \sigma + j\omega$



$F(s) = \frac{s}{s+2}$

The students are encouraged to look at the animated contour maps given in by Professor Mastascusa of Bucknell University, PA .

Theorem 67 *Cauchy's Theorem-Principle of Argument: "If a contour Γ_s in s plane encircles Z zeros and P poles of $F(s)$, and does not pass through any of the poles and zeros of $F(s)$ as the traversal is in the clockwise direction along the contour, the corresponding contour Γ_F in $F(s)$ plane encircles the origin of the $F(s)$ plane $N = Z - P$ times" [3].*

$$\begin{aligned} F(s) &= \frac{(s-z_1)(s-z_2)}{(s-p_1)(s-p_2)} = |F(s)|\angle F(s) \\ &= \frac{|s-z_1|\,|s-z_2|}{|s-p_1|\,|s-p_2|},\ \phi_f = +\angle(s-z_1) + \angle(s-z_1) - \angle(s-p_1) - \angle(s-p_2) \end{aligned} \tag{5.28}$$

If s is varied to have a complete contour in s plane to encircle the zeros and poles in clockwise direction, it is evident that the number of encirclement of $F(s)$ origin will be :

$$2\pi N = 2\pi Z - 2\pi P \rightarrow N = Z - P \tag{5.29}$$

5.5.3 Nyquist Stability Law

The closed loop transfer function of the system is given by :

$$G_{cl}(s) = \frac{KG}{1+KGH} = \frac{KG}{1+G_0(s)} \rightarrow G_0(s) = KGH = K\frac{B(s)}{A(s)} \tag{5.30}$$

The characteristic equation becomes:

$$F(s) = 1 + KG_0(s) = 1 + K\frac{B(s)}{A(s)} = \frac{A(s)+KB(s)}{A(s)} = \frac{\prod_{i=1}^{m}(s-z_i)}{\prod_{j=1}^{n}(s-p_j)} \tag{5.31}$$

Hence, the poles of open loop system ($G_0(s)$) is the same as the poles of the characteristic equation $F(s)$ of the closed loop system. However, we are interested about the poles of the closed loop system which correspond to zeros ($A(s) + B(s)$) of the characteristic equation to determine the stability of the system. In short, the zeros of $F(s)$ must not lie on the right hand side of the s plane. Therefore we need to check

whether any of the zeros of $F(s)$ lie on the right hand side of the s plane. We can reorganize the equation as follows:

$$F'(s) = F(s) - 1 = KG_0(s) = K\frac{B(s)}{A(s)} \tag{5.32}$$

Now, we need to check only the open loop transfer function $G_0(s) = \frac{N(s)}{D(s)}$ which has known zeros and poles and it is easy to plot Nyquist diagram of $G_0(s = j\omega) = \frac{B(s=\omega)}{A(s=\omega)}$, while $F(s) = 1 + K\frac{B(s)}{A(s)} = \frac{A(s)+KB(s)}{A(s)}$'s zeros are more difficult to find. By shifting the origin from zero to -1, and applying Cauchy's theorem, we can check whether $G_0(s)$ encircles -1 in clockwise while $s = j\omega$ and -$\infty \leq \omega \leq +\infty$.The number of zeros (Z) of $F(s)$ which lie on the right, unstable part of the s plane is :

$$N = Z - P \rightarrow Z = N + P \tag{5.33}$$

where

- N is the number of encirclements of -1 by the open loop frequency response of the system $KG_0(j\omega) = KG(j\omega)H(j\omega)$,which can be calculated from the Nyquist plot of open loop transfer function.
- P is the number of unstable open loop poles $G(s)H(s)$
- Z is the number of zeros of $1 + KGH$ on the right hand s plane which are also the unstable poles of the closed loop system, and they are calculated from easier evaluation of $N + P$.

The clockwise encirclement is counted as positive, and counterclockwise encirclement is subtracted from N. Since it is easy to see the number of unstable poles $G_0(s)$ of the open loop system, we can simply calculate the number of unstable zeros (Z) of the characteristic equation which actually corresponds to the number of unstable poles in a closed loop system. We need to also note that most physical systems do not have unstable poles in their open loop system, hence usually $P = 0$.[3], [5], [1]

Note that the discussions in this course will deal only with minimum phase systems, which are most common in mechatronics systems.

Minimum Phase Systems : Systems whose open loop zeros and poles are on the left hand of s plane.

Non-Minimum Phase Systems: Systems which have zeros or poles on the right hand of s plane.

Example 68 *The open loop transfer function of a closed loop system is given as* $KG(s)H(s) = KG_0(s) = K\frac{1}{s(s+a)(s+b)}$. *Note that* $KG_0(s)$ *now corresponds to* $F'(s) = F(s) - 1$, *and we need to check the encirclement of* $(-1, 0)$ *of* $KG_0(j\omega)$.*Plot the Nyquist diagram and identify the maximum value of gain* K *before the system becomes unstable. Consider* $G_0(j\omega)$ *without the gain* K, *which we need to identify and tune according to the performance required from the closed loop control system.*

$$G_0(s) = \frac{1}{s(s+a)(s+b)}$$

Frequency Response Function (FRF) of the system can be represented by its magnitude and phase as:

$$\begin{aligned} G_0(j\omega) &= \frac{1}{j\omega(j\omega+a)(j\omega+b)} \\ &= \frac{1}{|\omega|\sqrt{\omega^2+a^2}\sqrt{\omega^2+b^2}}\left(\angle - \frac{\pi}{2} - \tan^{-1}\frac{\omega}{a} - \tan^{-1}\frac{\omega}{b}\right) \end{aligned}$$

Alternatively we can express the system using polar coordinates, or Nyquist Plane, by separating real and imaginary parts as a function of frequency:

$$\begin{aligned}
G_0(j\omega) &= \frac{1}{j\omega(j\omega+a)(j\omega+b)} = \frac{1}{-\omega^2(a+b)+j\omega(ab-\omega^2)} \\
&= \frac{-\omega^2(a+b)-j\omega(ab-\omega^2)}{\omega^4(a+b)^2+\omega^2(ab-\omega^2)^2} = \frac{-\omega(a+b)-j(ab-\omega^2)}{\omega\left[\omega^2(a+b)^2+(ab-\omega^2)^2\right]} \\
&= -\frac{a+b}{\omega^2(a+b)^2+(ab-\omega^2)^2} - j\frac{ab-\omega^2}{\omega\left[\omega^2(a+b)^2+(ab-\omega^2)^2\right]}
\end{aligned}$$

However, one has to be careful in calculating the phase from real and imaginary parts, especially when the phase $\phi > 2\pi$ or $\phi < -2\pi$.

Lets start plotting the Nyquist diagram of the system:

- If $\omega = 0 \rightarrow \Re[G_0(\omega)] = -\frac{a+b}{a^2b^2}, \Im[G_0(\omega)] = -\infty; |G_0(\omega)| = \infty,\ \phi = -\frac{\pi}{2}$
- If $\omega = \infty \rightarrow \Re[G_0(\omega)] = 0, \Im[G_0(\omega)] = 0; |G_0(\omega)| = 0,\ \phi = -\frac{\pi}{2} - \frac{\pi}{2} - \frac{\pi}{2} = -3\frac{\pi}{2} = -270\,\text{deg}$ (Which means that the Nyquist diagram will start from $-\frac{\pi}{2}$ and pass through third quadrant (cross negative real axis) and reach to the origin.
- Real part crossing, which is called **Phase Crossing**, can be calculated by forcing the imaginary part to be zero. The Phase Crossing frequency (ω_p) is : $(ab-\omega_p^2) = 0, \omega_p = \sqrt{ab}$. The real part crossing can be calculated by substituting $\omega_p = \sqrt{ab}$ into the real part
  $\text{Re}\left[G_0(j\omega_p)\right] = -\frac{a+b}{ab(a+b)^2+(ab-ab)^2} = -\frac{1}{ab(a+b)}$
- If $\text{Re}\left[G_0(j\sqrt{ab})\right] < -1$, the Nyquist plot encircles the $-1, 0$ point once, and the closed loop system is unstable since there is no additional term in the open loop transfer function which may lead to counterclockwise encirclement. This means that the characteristic equation $1 + G(s)H(s) = 0$ has roots on the right hand side of the s plane.
- If $\text{Re}\left[G_0(j\sqrt{ab})\right] > -1$, the system is stable since the crossing is on the right hand side of -1, and the poles of the closed system is on the left hand side of the s plane. How much do we increase the gain K until the closed loop system becomes critically stable?

  $$K_{\text{lim}}\,\text{Re}\left[G_0(j\sqrt{ab})\right] = -1 \rightarrow K_{\text{lim}} = -\frac{1}{\text{Re}\left[G_0(j\sqrt{ab})\right]}$$

  The inverse of the open loop gain at the zero crossing is called gain margin: *Gain Margin* $K_{\text{lim}} = -1/\,\text{Re}\left[G_0(j\sqrt{ab})\right]$. That means when the gain K is increased to $K_{\text{lim}} = -1/\,\text{Re}\left[G_0(j\sqrt{ab})\right]$, the system becomes critically stable. The gain must be designed to be less than K_{lim}, $K < K_{\text{lim}}$ to leave some margin before the system becomes critically stable. Phase margin corresponds to the phase of the open loop transfer function when the circle with a unit radius intersects the Nyquist plot in the third quadrant $\text{Re}\left[G_0(j\sqrt{ab})\right] < 0, \text{Im}\left[G_0(j\sqrt{ab})\right] < 0$.
- When the system is critically stable, i.e. poles are on the imaginary axis of the s plane, $\text{Re}\left[KG_0(j\sqrt{ab})\right] = -1$.

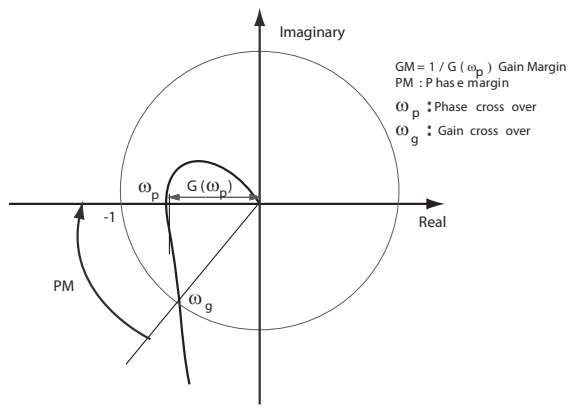


Figure 5.6: Gain and Phase margins on Nyquist plot.

5.6 Bode Plot

Frequency Response of the open loop system can be graphically represented by its magnitude and phase as shown in Figure 5.7. It is customary to represent the Bode plot in logarithmic scale where $1dB = \log_{10} 10$ and $20 \log_{10} 10 = 20dB$.

$$G_0(j\omega) = \left.\frac{1}{s(s+a)(s+b)}\right|_{s=j\omega} = \frac{1}{j\omega(j\omega+a)(j\omega+b)}$$

$$|G_0(j\omega)| = 20 \log \left[\frac{1}{|\omega| \sqrt{\omega^2+a^2}\sqrt{\omega^2+b^2}}\right] = -20 \log \omega - 20 \log \sqrt{\omega^2+a^2} - 20 \log \sqrt{\omega^2+b^2}$$

$$\phi = -\frac{\pi}{2} - \tan^{-1}\frac{\omega}{a} - \tan^{-1}\frac{\omega}{b}$$

Gain Cross Over Frequency (ω_g): The frequency, where the magnitude becomes 0 dB ($20 \log_{10} 1 = 0dB$) in logarithmic or unity in linear scale.

Phase Margin (PM): The difference between the phase at the gain cross over frequency and -180 degrees, $PM = +180 - \phi(\omega_g)$ [deg]. If the phase is below -180[deg] at the gain cross over frequency, the system is already unstable. Larger the phase margin is, further the system is away from the instability.

Phase Cross Over Frequency (ω_p): The frequency, where the phase becomes -180[deg].

Gain Margin (GM): The difference between the gain at the phase crossing frequency (ω_p) and 0 dB, or the magnitude at ω_p.The gain margin indicates amount of gain increase before making the system critically stable. If the magnitude is positive and above $0dB$, the system has a negative gain margin, and it is already unstable.

Summary 69 *Nyquist stability criteria allows us to asses the relative stability of the systems from the frequency response plots (Bode or Nyquist diagrams) of the open loop system alone, instead of finding the roots of characteristic equation of the closed loop system. When the Nyquist plot crosses* -1, *the phase lag will be exactly* -180 *degrees. This corresponds to a unity gain (or* $0dB$ *in log scale) and* -180*degree phase in Bode plot. At this point, the gain and phase margins are zero, and the closed loop poles have no real part and have only complex parts. This also means that we have no room to increase the gain, otherwise the system will be unstable or the poles of the closed loop system will have positive real parts. If the phase angle is -180 degrees or less, the feedback (output) signal will have opposite sign with the input signal. Instead of subtracting the output from the input, we will be adding the output to the input, which lead to the growth of the error signal continuously. The system will be unstable. However, if we reduce the gain, the phase lag will also be reduced, leading to positive gain and phase margins, hence the system will be pushed to stable region*

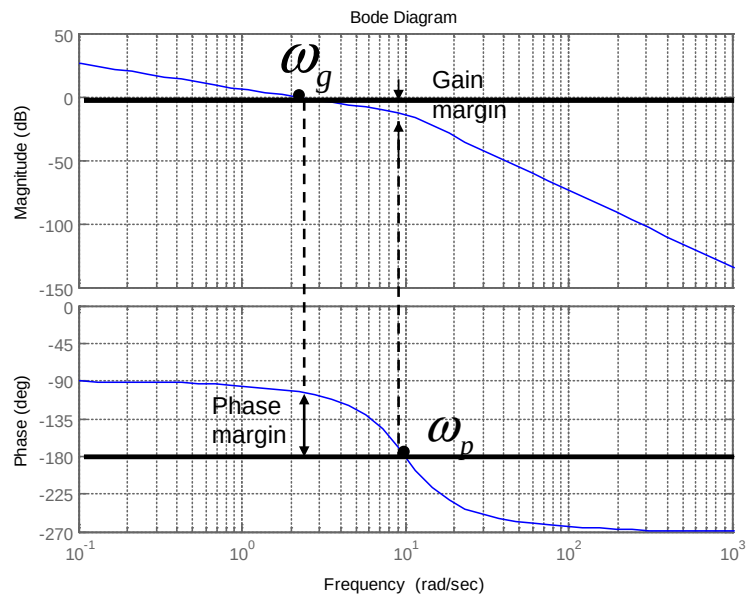


Figure 5.7: Definition of gain and phase margins on a Bode plot.

and the closed loop poles will have negative real parts. In summary, the Nyquist stability theory allows us to assess the relative stability of closed loop system by analyzing only its open loop frequency response, which is much easier than checking the roots of the characteristic equation as the controller gain increases.

Exercise 70 *Loop transfer function of a system is* $G(s)H(s) = KG_0(s) = K\frac{1}{s(s+a)(s+b)}$ *with* $a = 2$, $b = 10$. *Plot the Nyquist diagram and find the gain margin for a unit gain* $K = 1$. *What would be the maximum gain* ($K = ?$) *for a critical stability of the closed loop system?*

$$
\begin{aligned}
G_0(j\omega) &= -\frac{a+b}{\omega^2(a+b)^2 + (ab-\omega^2)^2} - j\frac{(ab-\omega^2)}{\omega\left[\omega^2(a+b)^2 + (ab-\omega^2)^2\right]} \\
&= -\frac{12}{144\omega^2 + (20-\omega^2)^2} - j\frac{20-\omega^2}{\omega\left[144\omega^2 + (20-\omega^2)^2\right]}
\end{aligned}
$$

- Real axis crossing frequency is the *phase cross over frequency*: $\omega_p = \sqrt{ab} = \sqrt{20} = 4.472\,1(rad/s)$.
- At the *phase cross over point of* $\mathrm{Re}\left[G_0(\omega_p = \sqrt{ab})\right] = -\frac{1}{ab(a+b)} = -\frac{1}{20(12)} = -0.004167166$
- The characteristic equation at phase crossing frequency $\omega = \omega_p$ is:

$$
K_{\lim}G_0(\omega_p) + 1 = 0 \rightarrow K_{\lim}\,\mathrm{Re}\left[G_0(\omega_p\right] = -1 \rightarrow K_{\lim} = \frac{-1}{\mathrm{Re}\left[G_0(\omega_p = \sqrt{ab}\right]}
$$

$$
K_{\lim} = \frac{-1}{-0.004167166} = 239.97 \text{ or } \; GM = 20\log 239.97 = 47.603\ dB
$$

where the gain margin is $K_{\lim} = 239.97$ in linear or $GM = 20\log 239.97 = 47.603\ dB$ in logarithmic scale. The maximum gain for critical stability is equal to gain margin of the system: $K = 10^{(47.603/20)} = 239.97$ which is the gain margin in linear scale. It means that when the gain of the closed loop system

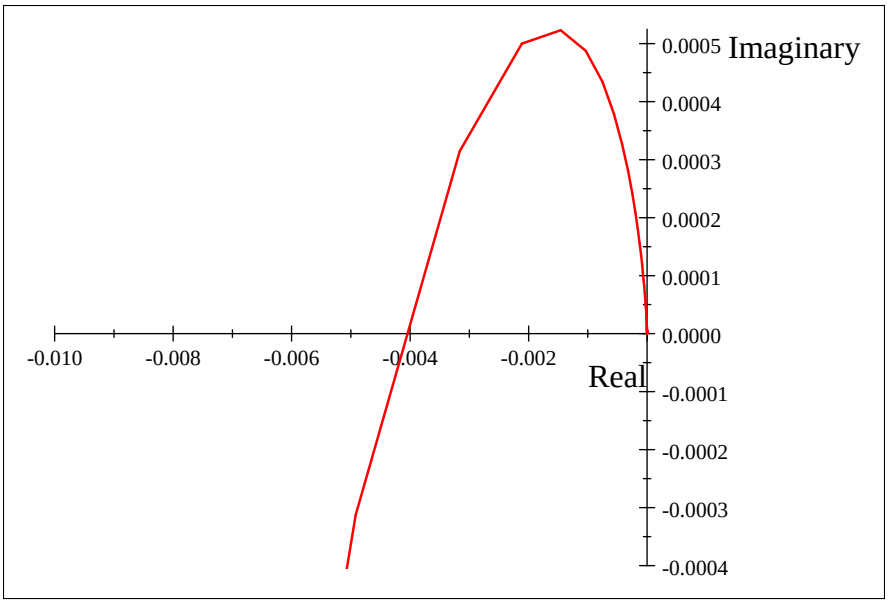


Figure 5.8: Nyquist plot of $\frac{1}{s(s+2)(s+10)}$ with gain $K = 1$

is increased from 1 to 239.97, the system will be critically stable and Nyquist plot will go through $-1, 0$ point. If we increase the gain further, the system will be unstable and Nyquist plot will encircle $-1, 0$ and the phase crossing point will be on the left of $-1, 0$. Lets repeat the Nyquist plot with $K = 300$ by simply multiplying the real and imaginary parts by 300.

Lets repeat the process using Bode plot. When the loop transfer function is normalized with gain $(K = 1)$:

$$\begin{aligned} G_0(j\omega) &= \frac{1}{|\omega|\sqrt{\omega^2+a^2}\sqrt{\omega^2+b^2}}\left(\measuredangle - \frac{\pi}{2} - \tan^{-1}\frac{\omega}{a} - \tan^{-1}\frac{\omega}{b}\right) \\ &= \frac{1}{|\omega|\sqrt{\omega^2+4}\sqrt{\omega^2+100}}\left(\measuredangle - \frac{\pi}{2} - \tan^{-1}\frac{\omega}{2} - \tan^{-1}\frac{\omega}{10}\right) \end{aligned}$$

It is customary to represent the Bode plot () in logarithmic scale where $1dB = \log_{10} 10$ and $20\log_{10} 10 = 20dB$.

$$\begin{aligned} |G_0(j\omega)| &= -20\log\omega - 20\log\sqrt{\omega^2+a^2} - 20\log\sqrt{\omega^2+b^2} \\ \measuredangle G_0(j\omega) &= -\frac{\pi}{2} - \tan^{-1}\frac{\omega}{a} - -\tan^{-1}\frac{\omega}{b} \end{aligned}$$

The Bode plot of the open loop transfer function indicates the same gain and phase margins as calculated from the Nyquist plot. Since the gain cross over is close to zero dB, and the phase margin at this low frequency is almost 90 deg .The phase cross over frequency is at 4.47 rad/s which gives 47.6 dB gain margin.

Example 71 *Open loop transfer function of the system is* $G_p(s) = \frac{K}{(s+2)(s+3)(s+5)}$*.if the gain is K=100, the system is stable. However, if the gain is increased to K=400, the system becomes unstable because Nyquist plot encircles* -1 *once. The second branch of the Nyquist diagram represents the negative change of the frequency, and it is a mirror image of the positive frequency range.*

Example 72 *Consider an open loop transfer function of a ball screw drive system as:*

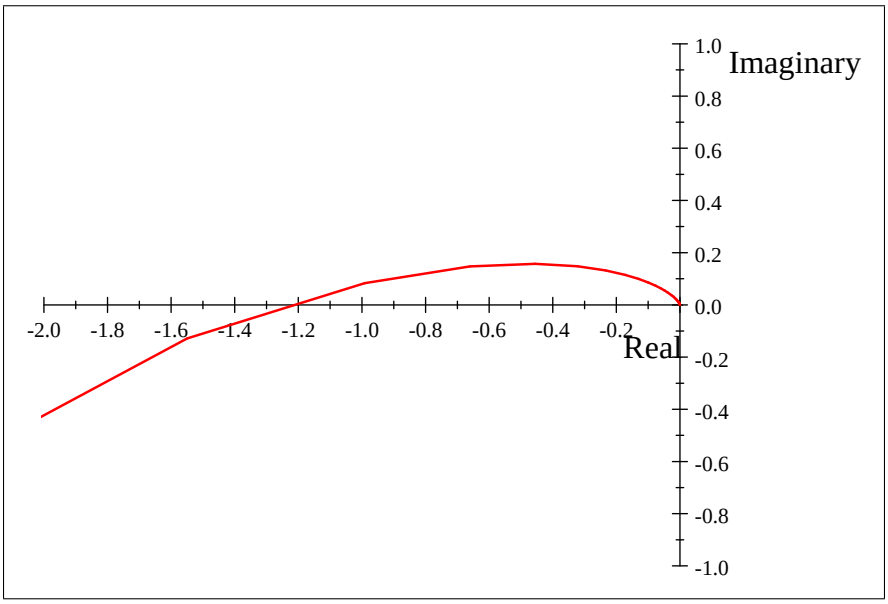


Figure 5.9: Nyquist plot of $\frac{1}{s(s+2)(s+10)}$ with gain $K = 300$

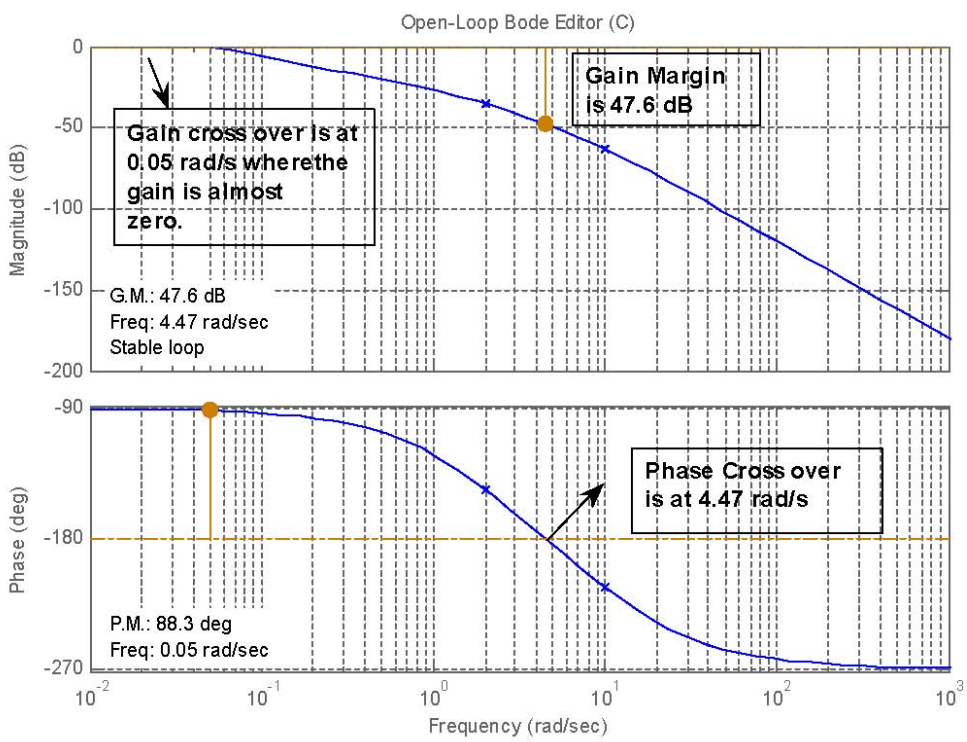
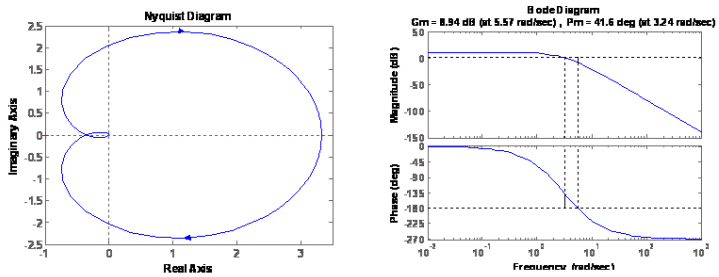
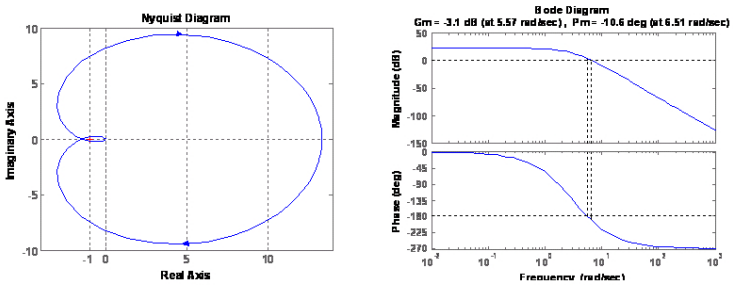


Figure 5.10: Bode plot of loop transfer function $\frac{1}{s(s+2)(s+10)}$. $G_0(j\omega) = -20\log\omega - 20\log\sqrt{\omega^2+2^2} - 20\log\sqrt{\omega^2+10^2}$, , $\measuredangle G_0(j\omega) = -\frac{\pi}{2} - \tan^{-1}\frac{\omega}{2} - -\tan^{-1}\frac{\omega}{10}$



a) Stable ssytem $G(s) = \frac{100}{(s+2)(s+3)(s+5)}$



b) Stable ssytem $G(s) = \frac{400}{(s+2)(s+3)(s+5)}$

Figure 5.11: Nyquist and Bode diagrams of stable (a) and unstable (b) systems when the gain is increased from 100 to 400.

$$G(s) = \frac{41820.3}{s\left(s^2 + 241.8s + 41820.3\right)}$$

The Bode plot of the system with unity controller gain is shown as :

The phase cross over frequency $\omega_p = 204\ rad/s$ which gives $47.6dB$ gain margin. Since the gain is under zero db, the phase margin is close to $90\,\text{deg}$. That means we can add 47.6 dB to the gain, or which is equivalent to proportional gain of $K_p = 240$ (47.6 dB) before it becomes critically stable.

Lets regenerate the Bode plot by increasing the gain from $K_p = 1$ to $K_p = 50$. Now the phase cross over frequency is $\omega_p = 240$ rad/s, which gives $10.7dB$ gain margin. The gain cross over frequency is $\omega_g = 50.9$ rad/s which gives $72.9\,\text{deg}$ phase margin.

Nyquist plot of the system is:

The unit step response of the system when $K_p = 50$.is given below:

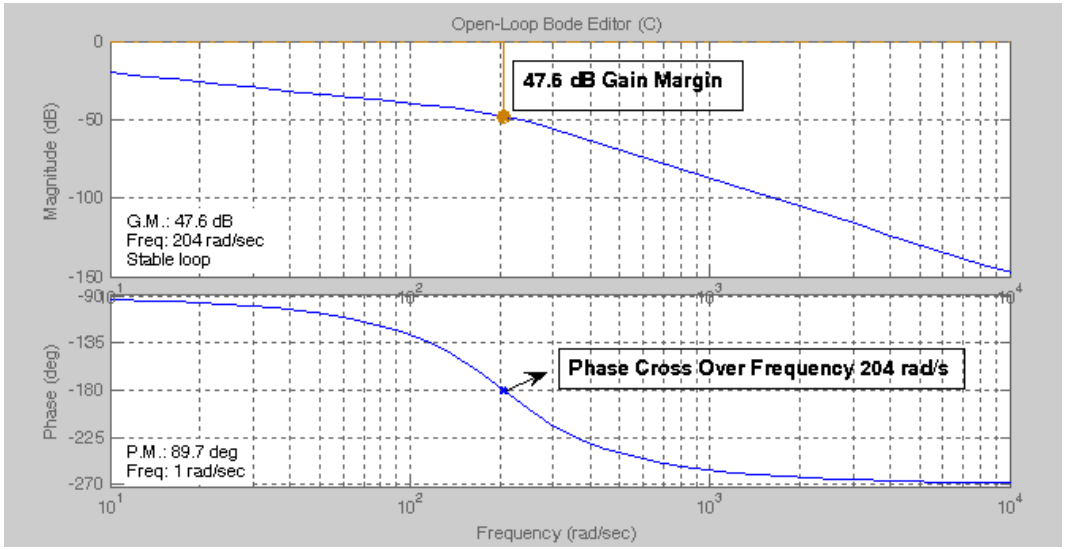


Figure 5.12: Bode plot of loop transfer function $\frac{41820.3}{s(s^2+241.8s+41820.3)}$ with unity gain ($K = 1$)

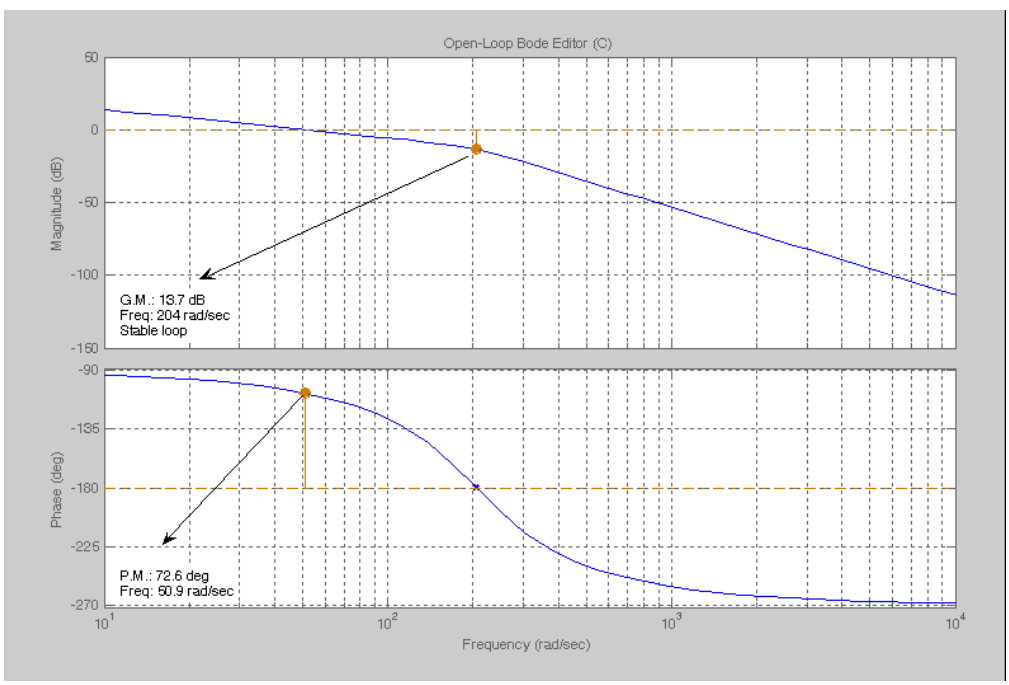
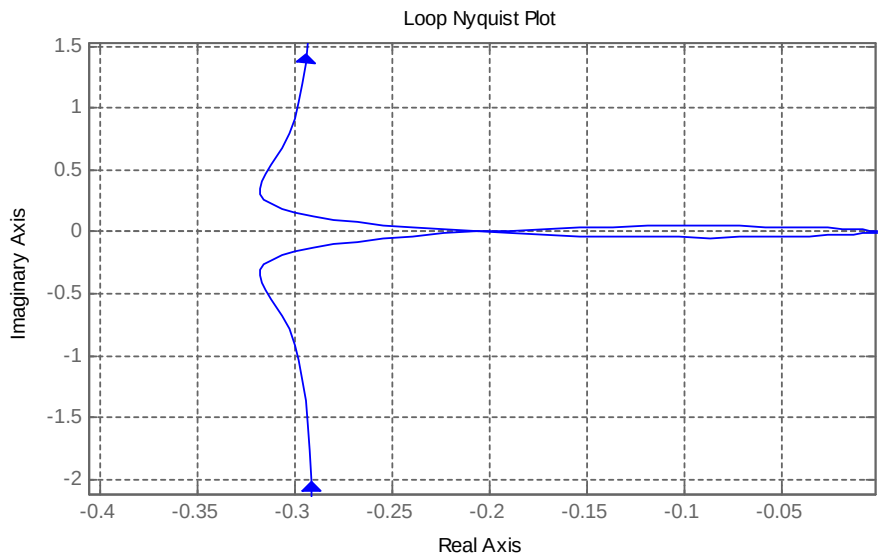
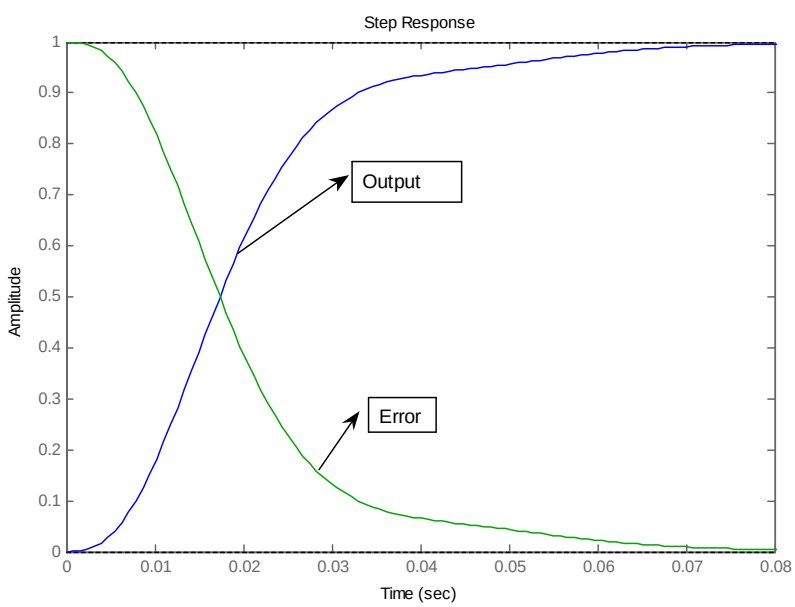


Figure 5.13: Bode plot of loop transfer function $\frac{41820.3}{s(s^2+241.8s+41820.3)}$ with unity gain ($K = 50$)





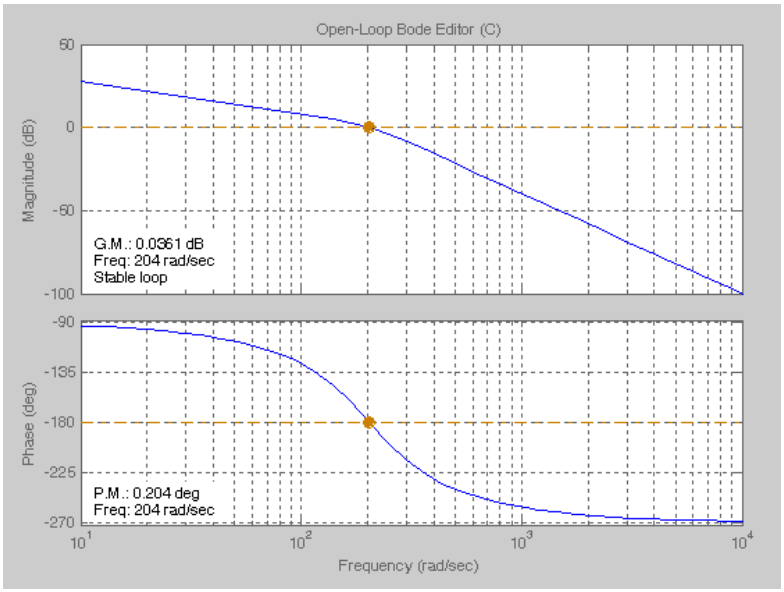
Lets regenerate the Bode plot by increasing the gain from one to $K_p = 240$, which is the total gain margin. As you see below that the gain margin becomes zero, so as the phase margin.

Lets continue to increase the gain to $K_p = 2000$. Note that the phase cross over frequency is still the same at 204 rad/s, but now it gives negative gain margin of $-18.4\ dB$. The gain cross over frequency is now $\omega_g = 444$ rad/s which gives negative phase margin of -55.4 and the closed loop system is unstable with calculated unstable poles of $+127.65 \pm j390.17$and a stable pole -496.

Nyquist plot of the system shown below encircles (-1,0) twice by two symmetric contours, which indicate the presence of two unstable complex conjugate closed loop poles.

Beside reducing the gain, one can add zeros and poles to the controller to compensate only for the phase, or mainly the gain or both by shaping the Bode plot.

5.6.1 Approximate Bode Plots



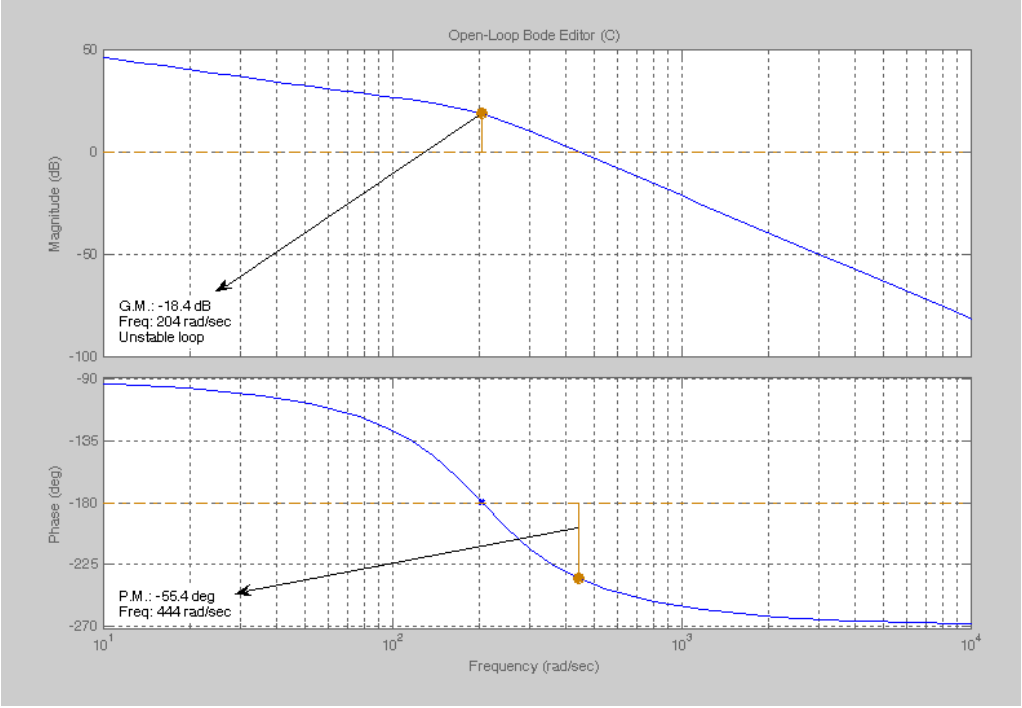
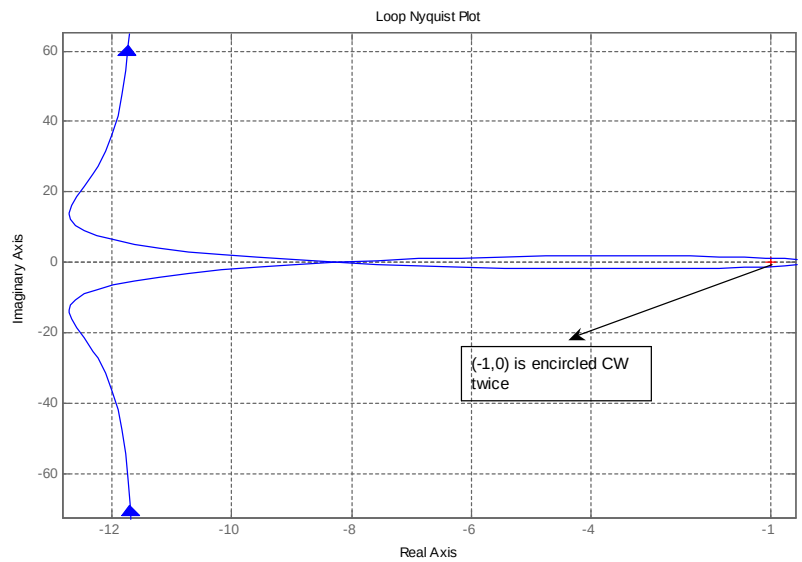


Figure 5.14: Bode plot of loop transfer function $\frac{41820.3}{s(s^2+241.8s+41820.3)}$ with unity gain ($K = 2000$)



Bode plots can be quickly plotted approximately in logarithmic scale by simply increasing the frequency by ten fold at each increment. The following gains, first order and second order systems are used as examples.

Case 73 *Gain (K) has a constant amplitude and phase for all frequencies, Mag=*$20\log K$ *and phase is always zero.*

Case 74 *Integrator* $G(s) = \frac{1}{s} \rightarrow G(j\omega) = \frac{1}{j\omega}$ $\quad |G| = -20\log\omega \quad \measuredangle G(j\omega) = -\pi/2$ *. The amplitude decreases by* $-20db$ *for each* 10 *times increase in the frequency.* $|G| = -20\log 10 = -20dB$, $|G| = -20\log 10^2 = -40dB$ *but the phase remains the same.*

Case 75 *First order lag* $G(s) = 1/(\tau s + 1) \rightarrow G(s) = 1/(+j\tau\omega)$, $|G| = -20\log\left[\sqrt{(\tau\omega)^2 + 1}\right] \approx -20\log(\tau\omega)$, $\phi = -\tan^{-1}(\tau\omega)$. *The system's magnitude drops -20dB per decade.*

ω	$\approx \lvert G\rvert\,[dB]$	ϕ[deg]
$\frac{1}{\tau}$	0	-45
$\frac{10}{\tau}$	-20	-90
$\frac{10^2}{\tau}$	-40	-90

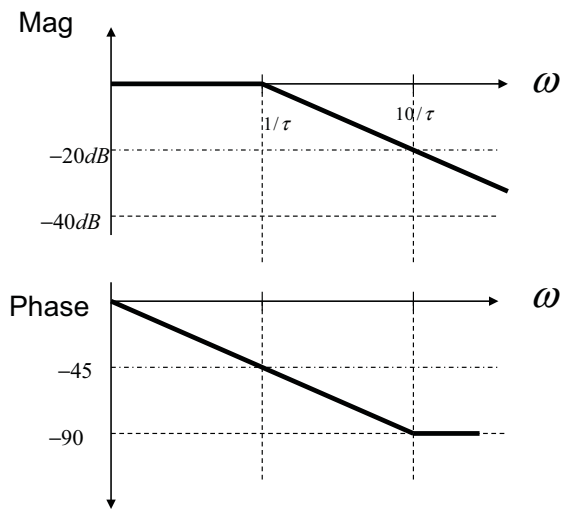


Figure 5.15: Approximate Bode plot of $G(s) = 1/(\tau s + 1)$

Case 76 *A second order lag is given by*

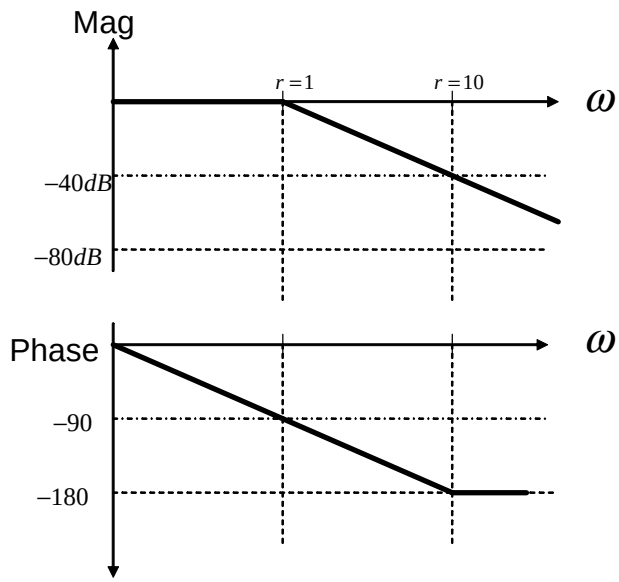
$$
\begin{aligned}
G(s) &= \frac{K}{s^2 + 2\varsigma\omega_n s + \omega_n^2} \\
G(s) &= \frac{K}{(s + \varsigma\omega_n - j\omega_d)(s + \varsigma\omega_n + j\omega_d)} \\
G(j\omega) &= \frac{K}{[\varsigma\omega_n + j\,(\omega - \omega_d)]\,[\varsigma\omega_n + j\,(\omega + \omega_d)]} \\
\phi[\text{deg}] &= -\tan^{-1}\frac{\omega - \omega_d}{\varsigma\omega_n} - \tan^{-1}\frac{\omega + \omega_d}{\varsigma\omega_n} \\
&= -\tan^{-1}\frac{\omega - \omega_n\sqrt{1-\varsigma}}{\varsigma\omega_n} - \tan^{-1}\frac{\omega + \omega_n\sqrt{1-\varsigma}}{\varsigma\omega_n} \\
\phi[\text{deg}] &= -\tan^{-1}\frac{r - \sqrt{1-\varsigma}}{\varsigma} - \tan^{-1}\frac{r + \sqrt{1-\varsigma}}{\varsigma} \\
G(j\omega) &= \frac{K}{\omega_n^2 - \omega^2 + j2\varsigma\omega_n\omega} = \frac{K/\omega_n^2}{1 - r^2 + j2\varsigma r} \rightarrow r = \frac{\omega}{\omega_n} \\
|G|\,[dB] &= +20\log(K/\omega_n^2) - 20\log\sqrt{(1-r^2)^2 + (2\varsigma r)^2} \\
&\approx 20\log(K/\omega_n^2) - 40\log r \quad \textit{for } r \gg 1
\end{aligned}
$$

If the phase is calculated from $\phi[\text{deg}] = \tan^{-1}\frac{-2\varsigma r}{1-r^2}$, one must be careful in calculating it correctly by manually tracking the location of the real and imaginary parts on the complex, geometric plane. The function *atan2* must be used in MATLAB in order to calculate the phase correctly.

r	$\approx \lvert G\rvert\,[dB]$	$\phi[\text{deg}]$
0	$20\log(K/\omega_n^2)$	0
1	$20\log(K/\omega_n^2)$	-90
10	$20\log(K/\omega_n^2) - 40dB$	-180
10^2	$20\log(K/\omega_n^2) - 80dB$	-180

The system attenuates by -40dB per decade increase in the frequency. The minimum phase approaches to -180 degrees.

Example 77 *Low frequency gain is 14db, zero db crossing is at 0.42Hz where the phase is -150* deg*. Hence*



the phase margin is 30 deg*. The 180 degree phase crossing is at 0.6Hz, and the corresponding gain is -6.5 db. hence the gain margin is 6.5db.Gain*

- Margin: "Gain margin is just the amount of gain that you can add to move the zero db crossing to occur at the same frequency as the -180 deg crossing."
- Phase margin: " Phase margin is just the difference between -180o and the actual phase angle of the frequency response function, KG(jw)H(jw), measured at the frequency where the magnitude of the frequency response function, |KG(jw)H(jw)|, is equal to one."

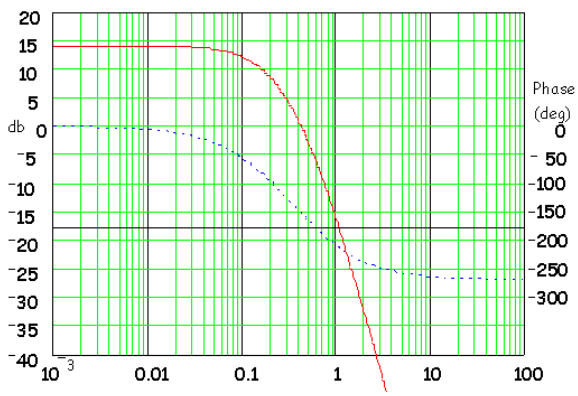


Figure 5.16: Bode plot of a system.

5.7 Frequency Response of a Closed Loop System

Consider a close loop system

$$M(s) = \frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \tag{5.34}$$

The frequency response of the close loop system, i.e. the steady state response of a system when the input $R(t) = R_0 \sin \omega t$, can be obtained by substituting $s \to j\omega$:

$$
\begin{array}{ccc}
M(s)|_{s\to j\omega} = M(j\omega) = & \frac{Y(j\omega)}{R(j\omega)} = & \frac{G(j\omega)}{1+G(j\omega)H(j\omega)} \\
M(j\omega) = & |M(j\omega)|\angle M(j\omega) = & \mathrm{Re}[M(j\omega] + j\,\mathrm{Im}[M(j\omega] \\
\text{Magnitude} & |M(j\omega)| = & \sqrt{(\mathrm{Re}[M(j\omega])^2 + (\mathrm{Im}[M(j\omega])^2} \\
\text{Phase} & \measuredangle M(j\omega) = & \tan^{-1}\frac{\mathrm{Im}[M(j\omega]}{\mathrm{Re}[M(j\omega]}
\end{array}
$$

Chapter 6

Design of Control Laws

6.1 Introduction

The objective of any control law is to modify the dynamic response of a closed loop system to exhibit a desired behavior. The block diagram of a close loop system is given in Figure 6.1, where $G_o(s)$ is the transfer function of the plant, $H(s)$ is the feedback block and $D(s)$ represents the transfer function of the controller.

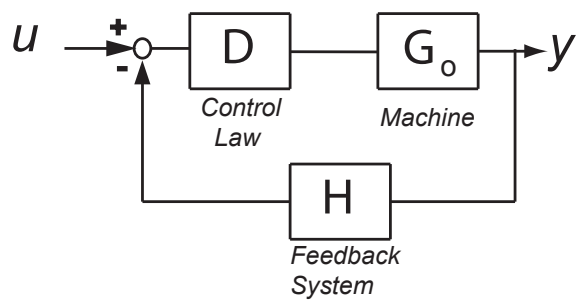


Figure 6.1: Close loop system with feedback (H) dynamics.

For example, if the user gives a step input which demands physically unrealizable infinite acceleration or undesirable sharp reaction, the controller is expected to soften the shape of sharp step input by smoothing it out. The dynamic response of the system is mainly determined by its poles, or the roots of its characteristic equation.

The control system is designed by the following steps

- Identify what to control, and model its open loop transfer function
- Specify the performance of the controller (relative stability, robustness to changing parameters, overshoot, bandwidth, sensitivity)
- Decide on the type of controller configuration (feedback compensation, feedback-feedfoward compensation)
- Tune the parameters of the controller to achieve design specifications

The controller can be designed in frequency or time domain depending on the design specifications and convenience. When gain margin, phase margin, sensitivity and bandwidth are specified, it is more convenient to design the controller in frequency domain using Bode and Nyquist plots. When overshoot or rise time are specified, it is more convenient to design the control law in time domain. While high order system can

be more easily handled in Frequency domain, they may need to be approximated by second order models in order to analyze their step, ramp or parabolic responses.

The performance of the controller can be assessed by remembering the following properties during the design process. Let the pole be $p_n = \sigma_n + j\gamma_n$ which will have an outout response with the following shape $y(y) = Ye^{p_n t} = Ye^{\sigma_n t}e^{j\gamma_n t} = Ye^{\sigma_n t}(\cos\gamma_n t + j\sin\gamma_n t)$

- The system response is dominated by the poles closest to the origin in $s-$ plane.
- Complex conjugate poles lead to an underdamped system with transient oscillations. Real negative poles lead to an overdamped, and pure complex poles indicate a critically damped system.
- Poles at the far left of the s plane lead to fast disappearance of their effect on the transient response, and may be neglected when model order needs to be reduced.
- Further the dominant poles are on the left hand side of the s plane, the higher bandwidth the closed loop system will have. However, the system will require more energy to drive the plant, and physical limits of the actuators and amplifiers may be violated.
- Shorter the rise time is, higher the bandwidth will be. The larger gain and phase margins will improve the damping of the system and reduce the overshoot.

6.2 Proportional Controller Design

6.2.1 Proportional Controller Design in s Domain

Lets consider a first order velocity control system $(K_v/(\tau_v s + 1))$ cascaded with a position encoder and an integrator (K_e/s). The open loop transfer function of a system is given by:

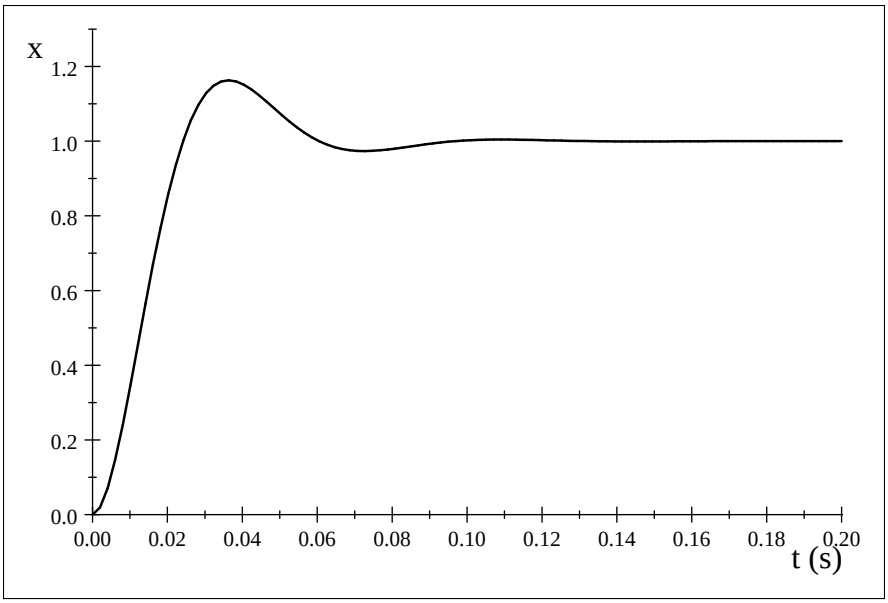
$$G_o(s) = \frac{K_d K_v K_e}{s(\tau_v s + 1)} = \frac{33.1223}{s(0.010s + 1)}$$

where $K_v = 10.65[V/(rad/s)]$, $K_e = 4000/(2\pi) = 636.62[\text{count/rad}], \tau_v = 0.01$. The gain of the D/A converter is $K_d = 20/2^{12} = 0.0048828$. A proportional controller, which gives 1% overshoot is required. If we design in s domain, the zero order hold $(1 - e^{-sT})/s$ needs to be neglected since its purpose is only to connect the continuous open loop system to the digital controller. The closed loop transfer function of the system becomes:

$$\begin{aligned} G_{cl}(s) &= \frac{K_p G_o}{1 + K_p G_0} = \frac{K_p K_d K_v K_e}{s(\tau_v s + 1) + K_p K_d K_v K_e} \\ &= \frac{K_p K_d K_v K_e/\tau_v}{s^2 + s/\tau_v + K_p K_d K_v K_e/\tau_v} \equiv \frac{K}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{aligned}$$

Unit step response of a second order, under-damped system is given by:

$$\begin{aligned} x_a(t) &= 1 - e^{-\zeta\omega_n t}(\cos\omega_d t + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\omega_d t) \\ &= 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}(\sin\omega_d t + \tan^{-1}\frac{\zeta}{\sqrt{1-\zeta^2}}) \end{aligned}$$



Step response of the second order underdamped system.

The transient response of the system can be characterized by the rise time (t_r), maximum percentage overshoot (M_p) and corresponding peak time (t_p), settling time (t_s). The rise time is measured when the response reaches to 90% of its target value. The approximate formula for the rise time is given by:

$$t_r \simeq \frac{1 - 0.4167\zeta + 2.917\zeta^2}{\omega_n} \rightarrow \zeta < 1$$

In order to reach the target,the natural frequency (or bandwidth) of the system must be high. If the system is under-damped, it will overshoot at peak time t_p. The peak value of the step response can be evaluated by :

$$\left.\frac{dx_a(t)}{dt}\right|_{t=t_p} = 0 = \sin(\omega_d t_p)\frac{\omega_n}{\sqrt{1-\zeta^2}}e^{-\zeta\omega_n t_p} = 0 \rightarrow \sin(\omega_d t_p) = 0$$
$$\omega_d t_p = 0, \pi, 2\pi, 3\pi, .. \quad .t_p = \frac{\pi}{\omega_d}$$

The overshoot M_p occurs at time t_p. By substituting $t_p = \frac{\pi}{\omega_d}$into step response, the overshoot can be evaluated by:

$$M_p = x_a(t_p) - 1 = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \rightarrow \zeta = \frac{-\ln M_p}{\sqrt{\pi^2 + (\ln M_p)^2}}$$

The system's transient response is bounded by the exponential term $e^{-\zeta\omega_n t}$. The envelope of the response from bottom part, $1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}$, will force the system to approach to steady state step response level as the time goes to infinity. If the system approaches within x_{ts} of the target, the corresponding settling time can be evaluated from :

$$1 - \frac{e^{-\zeta\omega_n t_{ts}}}{\sqrt{1-\zeta^2}} = x_{ts} \rightarrow \zeta\omega_n t_{ts} = -\ln\left[(1-x_{ts})\sqrt{1-\zeta^2}\right]$$
$$t_{st} = -\frac{1}{\zeta\omega_n}\ln\left[(1-x_{ts})\sqrt{1-\zeta^2}\right]$$

By assigning the damping ratio to the characteristic equation of the closed loop control system ($\tau_v = 0.01, K_d = 0.004\,882\,8, K_v = 10.65, K_e = 4000/(2\pi)$) :

$$
\begin{aligned}
G_{cl}(s) &= \frac{K_p K_d K_v K_e/\tau_v}{s^2 + s/\tau_v + K_d K_p K_v K_e/\tau_v} \equiv \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\
M_p &= 1\% \quad \rightarrow \zeta = \frac{-\ln 0.01}{\sqrt{\pi^2 + (\ln 0.01)^2}} = 0.826 \\
2\zeta\omega_n &= 1/\tau_v, \quad \zeta = 0.826, \quad \tau_v = 0.01 \\
\omega_n &= \frac{1}{2\zeta\tau_v} = \frac{1}{2 \cdot 0.826 \cdot 0.01} = 60.55\ [rad/s] = 9.64Hz \\
\omega_n^2 &= \frac{K_p K_d K_v K_e}{\tau_v} \rightarrow K_p = \frac{\omega_n^2 \tau_v}{K_d K_v K_e} = \frac{60.55^2 \cdot 0.01}{33.1223} = 1.1068\ [V/count] \\
G_{cl}(s) &= \frac{36.663}{s^2 + 100s + 36.663}
\end{aligned}
$$

With the assignment of overshoot value only, we were able to tune the proportional gain in time domain analysis.

Lets repeat the design in Root Locus. The open loop transfer function of the system was:

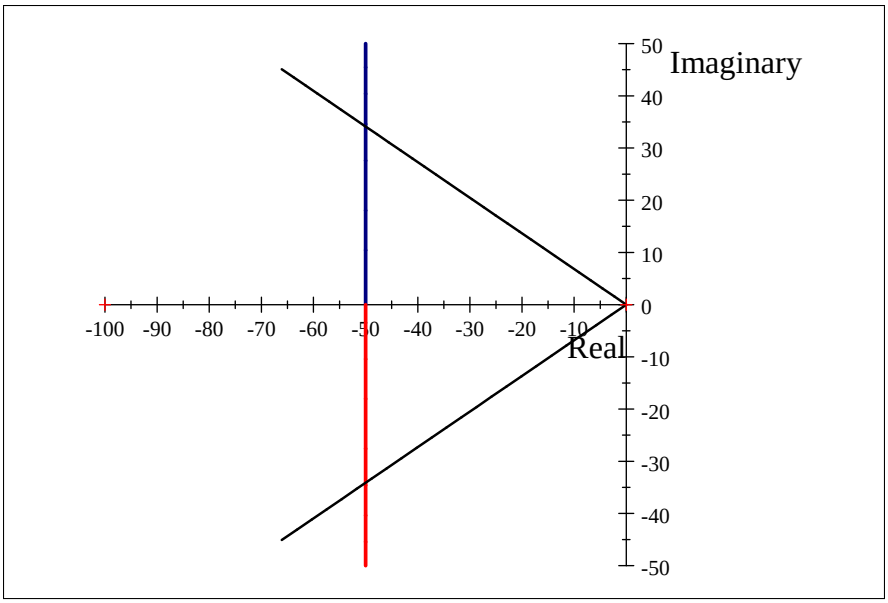
$$
G_o(s) = \frac{K_d K_v K_e}{s(\tau_v s + 1)} = \frac{K_0}{s(\tau_v s + 1)} = \frac{33.1223}{s(0.010s + 1)}
$$

which has a closed loop transfer function of

$$
G_{cl}(s) = \frac{K_p G_o(s)}{1 + K_p G_o(s)}
$$

The characteristic equation of the closed loop system becomes ($K = K_p K_0$):

$$
\begin{aligned}
1 + K_p G_o(s) &= 1 + K_p \frac{K_0}{s(\tau_v s + 1)} = 0 \\
s(\tau_v s + 1) + K &= \tau_v s^2 + s + K = 0 \\
p_1 &= \frac{-1 + \sqrt{1 - 4K\tau_v}}{2\tau}, \quad p_2 = \frac{-1 - \sqrt{1 - 4K\tau_v}}{2\tau} \\
p_1 &= -50 + 50\sqrt{1 - 0.04K}, \quad p_2 = -50 - 50\sqrt{1 - 0.04K}
\end{aligned}
$$



with poles at $p_1 = 0$ and $p_2 = -1/0.01 = -100$. The root locus is plotted by starting from the open loop poles, and keep increasing the gain K. The constant damping loci is plotted as :

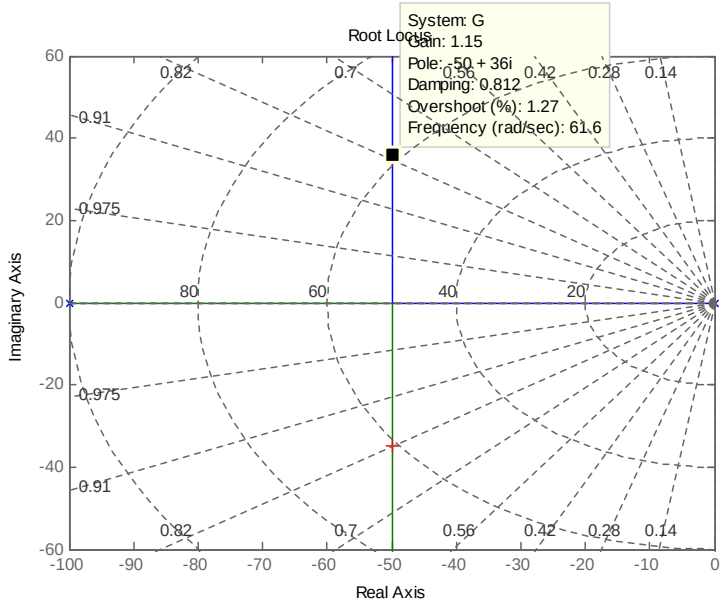
$$\psi = -\tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta} = -\tan^{-1}\frac{\sqrt{1-0.826^2}}{0.826} = -34\,\text{deg} \text{ clockwise from the negative real axis,}$$
$$\text{or } 180 - 34\,\text{deg} \text{ counter clockwise from the positive real axis.}$$

The intersection of the constant damping logi and root locus gives the desired poles of the system.

$$\begin{aligned}
p_1 &= -50 + j50\sqrt{0.04K-1} \quad \rightarrow K > 25 \\
\tan\psi = \tan 34 &= \frac{\sqrt{1-\zeta^2}}{\zeta} = \frac{\text{Im}(p)}{\text{Re}(p)} = \frac{50\sqrt{0.04K-1}}{50} = \sqrt{0.04K-1} = 0.682\,41 \\
K &= (1 + 0.682\,41^2)/0.04 = 36.642 \\
K_p &= K/K_0 = 36.642/33.125 = 1.1062
\end{aligned}$$

Algorithm 78 *MATLAB CODE TO DESIGN P CONTROLLER WITH ROOT LOCUS*

```
% MECH467/MECH590 Computer Aided Manufacturing
% Professor Yusuf Altintas, UBC, Feb. 2007
%m file to demonstrate interactive root locus design
clear;
% open loop tf of the plant in s domain
Kv=10.65; % velocity loop gain [rad/sec/V]
tauV=0.01; % velocity loop time constant [sec]
Ke=4000/(2*pi); % Encoder gain [count/rad]
Kd=20/2^12;
% Open Loop transfer function
G=tf([Kd*Kv*Ke],[tauV 1 0]);
figure(1); clf;
%zoom on;
rlocus(G);                    % display root locus
```



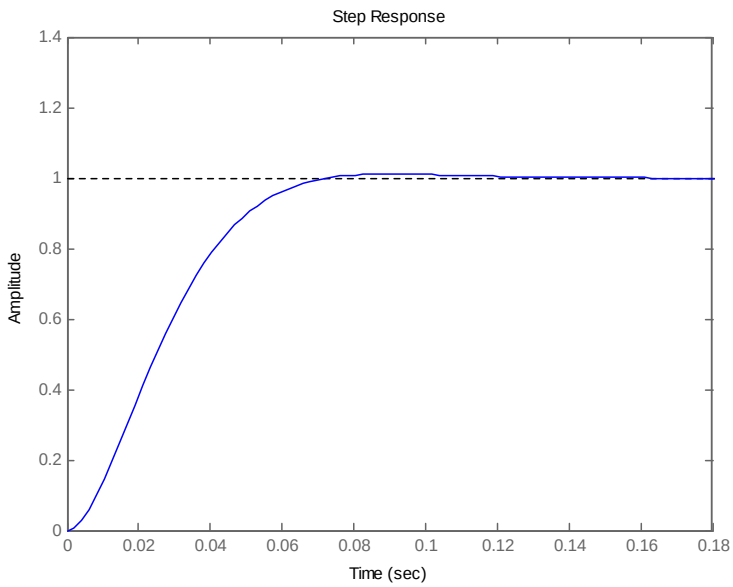

```
sgrid;                          % show grid (as wn vs. zeta)
AXIS([-100 0 -60 60])
%
% Dashed circles represent constant natural frequency loci, or magnitude of poles
% Lines extending from the coordinate center (zero point) represents constant
% damping loci
% Cross (x) points represent poles of the open loop system
%
disp('Select points on the locus with the mouse');
[KComp,poles]=rlocfind(G);                      % user selects feedback gain
disp(['The proportional gain and poles of the designed closed system are- Kcomp (V/Count): ' num2str(KComp)
]);
poles                % Print the designed poles
% Obtain the closed loop transfer function
Gclp=feedback(G*KComp,1);                % closed-loop t.f.
% Generate step response of the system for the selected poles
disp('Step response of the designed, closed loop system is displayed ');
figure(2); clf;
%zoom on;
step(Gclp);                     % show step response
```

Design in Frequency Domain

The open loop transfer function of the system is

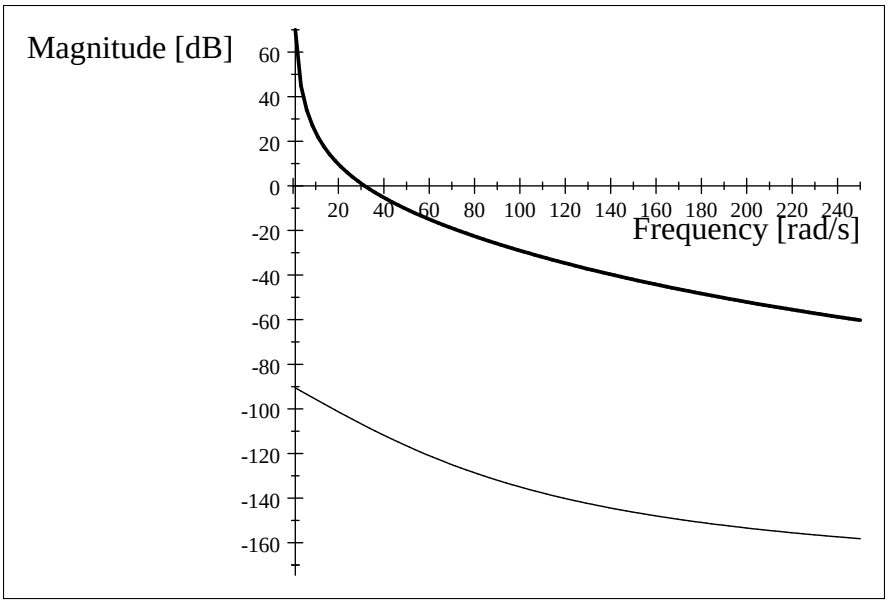
$$G_o(s) = \frac{K_d K_v K_e}{s(\tau_v s + 1)} = \frac{33.1223}{s(0.010s + 1)}$$

with a Frequency Response Function of



$$
\begin{aligned}
G_o(j\omega) &= \frac{33.1223}{j\omega(1+j0.010\omega)} = \frac{33.1223}{\omega\sqrt{1+(0.01\omega)^2}} \quad \measuredangle\left(-\frac{\pi}{2} - \tan^{-1} 0.01\omega\right) \\
&= \frac{-j33.1223(1-j0.010\omega)}{j\omega(1+j0.010\omega)\,(-j)\,(1-j0.010\omega)} = \frac{-33.1223(0.010\omega+j)}{\omega\left[(0.01\omega)^2+1\right]} \\
&= -\frac{0.331223\omega}{\omega\left[(0.01\omega)^2+1\right]} - j\frac{33.1223}{\omega\left[(0.01\omega)^2+1\right]}
\end{aligned}
$$

The system never intersects negative real axis with zero imaginary part, hence it is always stable. The phase lag is always less than π, therefore, the system is stable regardless of the gain as long as it is positive. In summary, the gain margin is infinite. The magnitude becomes 0 dB around 31.7 rad/s which corresponds to -108 degree, which is 72 degree away from -180 hence the phase margin is 72 degrees.



6.2.2 Proportional Controller Design in z Domain

The proportional controller can be designed in z domain as well. The ZOH equivalent of open loop transfer function of the system for a sampling period of $T = 0.001s$ is:

$$
\begin{aligned}
G_0(z) &= (1 - z^{-1})Z\frac{G_o(s)}{s} = (1 - z^{-1})Z\frac{K}{s^2(\tau_v s + 1)} \\
&= (1 - z^{-1})Z\frac{33.1223}{s^2(0.01s + 1)} \\
&= \frac{b_0 z + b_1}{z^2 + a_1 z + a_0} = \frac{0.0016015z + 0.001549}{z^2 - 1.9048z + 0.9048} \\
&= \frac{0.0016015(z + 0.9672)}{(z - 1)(z - 0.9048)} = \frac{b_0(z + b_1/b_0)}{(z - 1)(z - a_0)}
\end{aligned}
$$

where $(\tau_v = 0.01, K = K_v K_e K_d = 33.1223)$

$$
\begin{aligned}
b_1 &= K\left[\tau_v(1 - e^{-\frac{T}{\tau_v}}) - Te^{-\frac{T}{\tau_v}}\right] = 0.001549 \\
b_0 &= K\left[T - \tau_v(1 - e^{-\frac{T}{\tau_v}})\right] = 0.0016015 \\
a_0 &= e^{-\frac{T}{\tau_v}} = 0.9048 \\
a_1 &= -(1 + e^{-\frac{T}{\tau_v}}) = -1.9048
\end{aligned}
$$

The closed loop transfer function of the system becomes:

$$
\begin{aligned}
G_{cl}(s) &= \frac{K_p G_0(z)}{1+K_p G_0(z)} = \frac{K_p b_0(z+b_1/b_0)}{(z-1)(z-a_0)+K_p b_0(z+b_1/b_0)} \\
&= \frac{A(z+b_1/b_0)}{(z-1)(z-a_0)+A(z+b_1/b_0)} \\
&= \frac{A(z+b_1/b_0)}{z^2+(A-a_0-1)z+(Ab_1/b_0+a_0)} \\
&= \frac{A(z+0.9672)}{(z-1)(z-0.9048)+A(z+0.9672)} \\
&= \frac{A(z+0.9672)}{z^2+(A-1.9048)z+0.967\,2A+0.904\,8}
\end{aligned}
$$

where $A = 0.0016015K_p$. The proportional gain can be tuned through Root Locus method. The roots of the characteristic equation(i.e. closed loop poles) are:

When $K_p = 0$, the

$$
\begin{aligned}
\text{Characteristic equation} \quad &: \quad (z-1)(z-a_0)+K_p b_0(z+b_1/b_0) = 0 \\
\text{When } K_p \quad &= \quad 0 \to z_1 = 1, z_2 = a_0
\end{aligned}
$$

When $K_p = \infty$:

$$
\begin{aligned}
0 &= z^2+(A-1.9048)z+0.967\,2A+0.904\,8 \\
p_{1,2} &= \frac{-(A-1.9048) \pm \sqrt{(A-1.9048)^2-4\,(0.967\,2A+0.904\,8)}}{2}
\end{aligned}
$$

The poles will be identical ($p_1 = p_2 = \frac{-(A-1.9048)}{2}$) when the system has critically damped poles ($\zeta = 1$) which lie on the real axis of z plane.

$$
\begin{aligned}
(A-1.9048)^2-4\,(0.967\,2A+0.904\,8) &= 0 \\
A^2-7.678\,4A+9.063\times10^{-3} &= 0 \to A_1 = 1.180\,5\times10^{-3}, A_2 = 7.677\,2 \\
p_1 &= p_2 = \frac{-(A_1-1.9048)}{2} = \frac{-\left(1.1805\times10^{-3}-1.9048\right)}{2} = 0.951\,81 \\
p_1 &= p_2 = \frac{-(A_2-1.9048)}{2} = \frac{-(7.6772-1.9048)}{2} = -2.886\,2
\end{aligned}
$$

: If we select $A = A_2 = 7.677\,2$, the poles become $p_{1,2} = -2.886\,2$ outside unit circle and unstable. If we select $A = A_1 = 1.180\,5\times10^{-3}$, the poles become $p_{1,2} = 0.951\,81$ which are inside the unit circle and stable. We can keep increasing A until it intercepts damping logi $\zeta = 0.826$

Unlike in s domain, the root locus in z domain has a phase and magnitude which depend on the natural frequency and damping ratio together. In order to make sure that we have a pole that lies on the root locus, we need to increase the gain until the root locus intersects the desired damping ratio contour. In short, we need to search the intersection of root locus and the damping contour numerically, which can be done in MATLAB environment.

The z domain root locus solution in MATLAB would lead to $K_p = 1.05$ for a damping ratio of $\zeta = 0.8$

```
% MATLAB CODE TO DESIGN P CONTROLLER WITH ROOT LOCUS
% MECH467
% Professor Yusuf Altintas, UBC, Nov. 2013
%m file to demonstrate interactive root locus design
clear;
```

```
% open loop tf of the plant in s domain
Kv=10.65; % velocity loop gain [rad/sec/V]
tauV=0.01; % velocity loop time constant [sec]
Ke=4000/(2*pi); % Encoder gain [count/rad]
Kd=20/2^12;
K=Kd*Kv*Ke ;
% Open Loop transfer function
G0=tf([K],[tauV 1 0]); % Open loop tf
% Calculated ZOH coefficients of discete transfer function
b1=K*(tauV*(1-exp(-T/tauV))-T*exp(-T/tauV));
b0=K*(T-tauV*(1-exp(-T/tauV)));
a0=exp(-T/tauV);
a1=-(1+exp(-T/tauV));
%
% The following will give the same answer for a0,a1,b0,b1
G0z=c2d(G0,T,'ZOH');        % Zero Order Hold Equivalent command
figure(1); clf; zoom on;
rlocus(G0ZP);                          % display root locus
zgrid;                                 % show grid (as wn vs. zeta)
% Blue circles are zeros and blue x s are the poles of open loop tf
% The blue and red curve is the locus of poles for varying K
% Black dotted curves are constant damping loci.
% Damping is larger close to the real axis%
disp('Select points on the locus');
[KCompz,Ppolesz]=rlocfind(G0ZP);                          % user selects feedback gain
KCompz                % Print corresponding Kcomp for the selected poles
Ppolesz               % Print the designed poles
GclpPz=feedback(G0ZP*KCompz,1);                      % closed-loop t.f.
% Generate step response of the system for the selected poles
figure(2); clf; zoom on;
step(GclpPz);                          % show step response
```

Example 79 *Lead - lag Controller design: If the open loop transfer function has stable poles which are away from the imaginary axis in s plane, or well within the unit circle in z domain, they can be canceled by assigning them to the zeros of the lead lag controller. Let the lead - lag controller be* $D(s) = K_p' \frac{s+b'}{s+a'}$. *The closed loop transfer function becomes :*

$$G_{cl}(s) = \frac{D(s)K_d G_o}{1 + D(s)K_p G_0} = \frac{K_p' \frac{s+b'}{s+a'} K_d \frac{K_v K_e}{s(\tau_v s+1)}}{1 + K_p' \frac{s+b'}{s+a'} K_d \frac{K_v K_e}{s(\tau_v s+1)}} \tag{6.1}$$

By assigning $s + b' \equiv s + 1/\tau_v \rightarrow b' = 1/\tau_v = 10$, *the closed loop transfer function is reduced to:*

$$G_{cl}(s) = \frac{K_p' \frac{s+b'}{s+a'} K_d \frac{K_v K_e/\tau_v}{s(s+1/\tau_v)}}{1 + K_p' \frac{s+b'}{s+a'} K_d \frac{K_v K_e/\tau_v}{s(s+1/\tau_v)}} \tag{6.2}$$

$$= \frac{K_p' K_d K_v K_e/\tau_v}{s(s+a') + K_p' K_d K_v K_e/\tau_v} = \frac{K_p' K_d K_v K_e/\tau_v}{s^2 + sa' + K_p' K_d K_v K_e/\tau_v} \tag{6.3}$$

By equating the desired characteristic equation to the actual one:

$$s^2 + sa' + K_p' K_d K_v K_e/\tau_v \equiv s^2 + 2\zeta\omega_n s + \omega_n^2$$

Let the gain of the plant $K = \frac{K_d K_v K_e}{\tau_v} = 3310.05$ and its time constant is $\tau_v = 0.1$. The desired natural frequency is $\omega_n = 60.55[rad/s]$

$$s^2 + sa' + K_p' K \equiv s^2 + 2\zeta\omega_n s + \omega_n^2 \tag{6.4}$$

$$a' = 2\zeta\omega_n = 2 \cdot 0.826 \cdot 60.55 = 100.3 \tag{6.5}$$

$$K_p' = \frac{\omega_n^2}{K} = \frac{60.55^2}{3310.05} = 1.1076 \tag{6.6}$$

Example 80 *The controller can be designed similarly in z domain. Let the digital control system has a transfer function of* $D(z) = K_p \frac{z+b}{z+a}$. *The plant parameters were:*

$$G_p(z) = \frac{b_0 z + b_1}{(z-1)(z-a_0)}$$
$$b_1 = 0.001549; b_0 = 0.0016015; a_0 = 0.9048$$

The closed loop transfer function of the system becomes:

$$G_{cl}(s) = \frac{K_p \frac{z+b}{z+a} \frac{b_0(z+b_1/b_0)}{(z-1)(z-a_0)}}{1 + K_p \frac{z+b}{z+a} \frac{b_0(z+b_1/b_0)}{(z-1)(z-a_0)}} = \frac{K_p \frac{z+b}{z+a} \frac{0.0016015(z+0.9672)}{(z-1)(z-0.9048)}}{1 + K_p \frac{z+b}{z+a} \frac{0.0016015(z+0.9672)}{(z-1)(z-0.9048)}} \tag{6.7}$$

Let the desired model has a characteristic equation A_m with poles s_{m1} and s_{ms}

$$A_m(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 \ = (s - s_{m1})(s - s_{m2})$$
$$s_{m1} = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2}, \quad s_{m2} = -\zeta\omega_n - j\omega_n\sqrt{1-\zeta^2}$$

where ζ and ω_n are the damping ratio and natural frequency of the model. The desired poles can be mapped to the z domain by,

$$z_{m1} = e^{s_{m1}T} = e^{-\zeta\omega_n T} \cdot e^{j\omega_n\sqrt{1-\zeta_n^2}T},$$
$$z_{m2} = e^{s_{m2}T} = e^{-\zeta\omega_n T} \cdot e^{-j\omega_n\sqrt{1-\zeta_n^2}T}$$

$$\begin{aligned} A_m(z) &= (z - z_{m1})(z - z_{m2}) = z^2 - z(z_{m1} + z_{m2}) + z_{m1}z_{m2} \\ &= z^2 - ze^{-\zeta\omega T}(e^{j\omega_n\sqrt{1-\zeta^2}T} + e^{-j\omega_n\sqrt{1-\zeta^2}T}) + e^{-2\zeta\omega_n T} \\ &= z^2 - 2e^{-\zeta\omega_n T}cos(\omega_n\sqrt{1-\zeta^2}T)z + e^{-2\zeta\omega_n T} \end{aligned}$$

The desired damping ratio was $\zeta = 0.826$. Let the desired natural frequency is obtained in s domain as $\omega_n = 60.55$ $[rad/s]$ with a damping ratio of $\zeta = 0.826$ from the rise time and overshoot requirements. If we assign the desired poles : $s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2} = -0.826 \cdot 60.55 \pm j60.55\sqrt{1-0.826^2} = -50.014 \pm j34.13$. When the sampling time $T = 0.001[s]$,the two poles can be mapped to the z domain as:

$$z_{m1} = e^{s_1 T} = e^{(-50.014+j34.13)T} = 0.9507 + j0.0325,$$
$$z_{m2} = e^{s_1 T} = e^{(-50.014-j34.13)T} = 0.9507 - j0.0325$$

$$z_{m1}z_{m2} = 0.9048, \ z_{m1} + z_{m2} = 1.9014$$

By letting $z+b \equiv z-a_0 = z-0.9048 \rightarrow b=-0.9048$, the reduced closed loop transfer function becomes:

$$G_{cl}(z) = \frac{K_p \frac{z+b}{z+a} \frac{b_0(z+b_1/b_0)}{(z-1)(z-a_0)}}{1+K_p \frac{z+b}{z+a} \frac{b_0(z+b_1/b_0)}{(z-1)(z-a_0)}} = \frac{K_p b_0(z+b_1/b_0)}{(z-1)(z+a)+K_p b_0(z+b_1/b_0)}$$
$$= \frac{K_p b_0(z+b_1/b_0)}{z^2+az-z-a+K_p b_0 z+K_p b_1} = \frac{K_p b_0(z+b_1/b_0)}{z^2+(a-1+K_p b_0)\,z+(K_p b_1-a)}$$

By forcing the closed loop poles having the desired poles, the characteristic equation of the system must be the same as the desired one:

$$z^2+(a-1+K_p b_0)\,z+(K_p b_1-a) \equiv z^2-z(z_{m1}+z_{m2})+z_{m1}z_{m2}$$
$$(a-1+K_p b_0) = -(z_{m1}+z_{m2})$$
$$K_p b_1 - a = z_{m1}z_{m2} \rightarrow a = K_p b_1 - z_{m1}z_{m2}$$
$$(K_p b_1 - z_{m1}z_{m2} - 1 + K_p b_0) = -(z_{m1}+z_{m2}) \rightarrow K_p = \frac{-(z_{m1}+z_{m2})+z_{m1}z_{m2}+1}{b_0+b_1}$$
$$K_p = \frac{-1.9014+0.9048+1}{0.0016015+0.001549} = 1.0792$$
$$a = K_p b_1 - z_{m1}z_{m2} = 1.0792 \times 0.001549 - 0.9048 = -0.90313$$

:

6.2.3 Proportional Integral Derivative (PID) Controller Design

The transfer function of PID controller (Figure 6.1) is given as :

$$D(s) = K_p + \frac{K_I}{s} + K_d s = \frac{K_d s^2 + K_p s + K_I}{s} \tag{6.8}$$

where the performance trend is affected as shown in Table 6.2.3.

Control Gains	**Steady- State Error**	**Overshoot**	**Rise Time**	**Settling time**
$\mathbf{K}_p$ =**Proportional gain**	Decrease	Increase	Decrease	Small effect
$\mathbf{K}_I$ =**Integral gain**	Eliminates	Increase	Decreases	Increase
$\mathbf{K}_d$ =**Derivative gain**	Small effect	Decrease	Small effect	Decreases

Note that the effects of the gains are not definitive because the three parameters are coupled, and any change in one affects the performance contributed by the other two.

Lets consider a second order, underdamped dynamic system ($G_o(s)$) which needs to be controlled by a PID controller.

$$G_o(s) = \frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2} \tag{6.9}$$

The closed loop transfer function of the system($G_{cl}(s)$) :

$$G_{cl}(s) = \frac{D(s)G_o(s)}{1+D(s)G_o(s)} \tag{6.10}$$

where the feedback has a unit gain ($H(s)=1$).Transfer function of the close lop system varies as :

Controller	**D(s)**	Closed loop TF $G_{cl}(s)$
P controller	K_p	$\omega_n^2 \frac{K_p}{s^2+2\zeta\omega_n s+\omega_n^2(1+K_p)}$
PD Controller	$K_p + K_d s$	$\omega_n^2 \frac{K_d s+K_p}{s^2+\omega_n(2\zeta+K_d\omega_n)s+(K_p+1)\omega_n^2}$
PI Controller	$K_p + \frac{K_I}{s} = \frac{K_p s+K_I}{s}$	$\omega_n^2 \frac{K_p s+K_I}{s^3+2\zeta\omega_n s^2+\omega_n^2(1+K_p)s+K_I\omega_n^2}$
PID Controller	$K_p + \frac{K_I}{s} + K_d s$	$\omega_n^2 \frac{K_d s^2+K_p s+K_I}{s^3+(2\zeta\omega_n+\omega_n^2 K_d)s^2+\omega_n^2(1+K_p)s+K_I\omega_n^2}$

It becomes rather difficult to tune PID controller since it has 3 gains to be identified. The following rule of thumb approach can be used as a guide to tune PID controllers:

- Obtain the open loop transfer function of the system to be controlled.
- Add a proportional control to improve the rise time
- Add a derivative control to improve the overshoot
- Add an integral control to remove the steady state error
- Adjust the gains (K_p, K_d, K_I) until you are satisfied with the response (i.e. step response).

The following MATLAB program may help to understand the effects of different gains in designing P, PI, PD and PID controllers.

```
% MECH 467 Prof. Y. ALtintas
% Example : Proportional Integral Control
% UBC November 26, 2008
clear all; close all
% open loop tf of the plant in s domain
eta=0.1; % damping ratio
wn=100*pi*2; % natural frequency
G=tf([wn^2],[1 2*eta*wn wn^2]);% transfer function of the machine
% Proportional controller (P)
Kp=20 ;
Gp=feedback(Kp*G,1) ;
% Proportional Derivative (PD) Control
Kd=10;
PD=tf([Kd Kp],[1]);
Gpd=feedback(PD*G,1);
% Proportional Integral (PI) Control
Ki=200 ;
PI=tf([Kp Ki],[1 0]);
Gpi=feedback(PI*G,1);
% Proportional Derivative Integral (PID) Control
PID=tf([Kd Kp Ki],[1 0]);
Gpid=feedback(PID*G,1);
% PLot the Step Response of original plant (G), P, PD, PI and PID
% controllers
figure(1);
step(G,'b–',Gp,'c-',Gpd,'g-',Gpi,'r-.',Gpid,'k-');
legend('Plant','P','PD','PI','PID');
axis([0 0.1 0 2]);
% Plot Bode diagrams of original plant (G), P, PD, PI and PID
% controllers
figure(2);
bode(G,'b–',Gp,'c-',Gpd,'g-',Gpi,'r-.',Gpid,'k-');
legend('Plant','P','PD','PI','PID');
```

6.3 Lead-Lag Compensator Design in Frequency Domain

The frequency response provides significant engineering information about the system. The open loop transfer function of a system is shown in Figure 6.2. The higher magnitude at low frequency region, i.e. static gain, decreases the steady state error, attenuates the external disturbance and sensitivity against small changes in the plant parameters. The higher gain cross over frequency improves the response speed, but lower one attenuates the noise. Higher phase margin is translated to an improved stability margin. The lead lag compensators are used to re-shape the frequency response function of the system to achieve a desired response and performance.

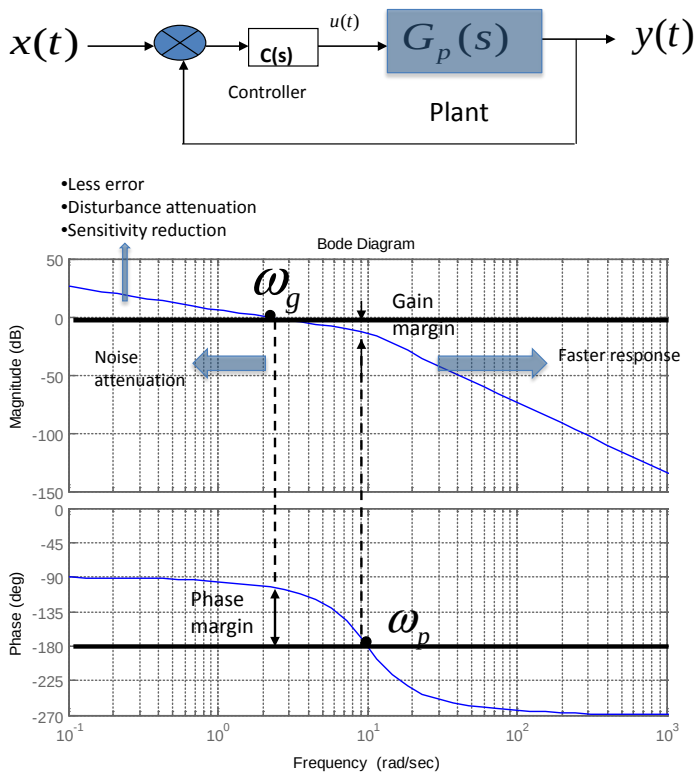


Figure 6.2: Bode plot properties

Consider a plant ($G_p(s)$) needs to be compensated by $C(s)$ with the following transfer function

$$C(s) = \frac{1+\alpha T s}{1+Ts} \begin{cases} \alpha > 1 & \text{Phase lead or high pass filter} \\ \alpha < 1 & \text{Phase lag or low pass filter} \end{cases}$$

$$C(s=j\omega) = \frac{1+j\alpha T\omega}{1+jT\omega} = \frac{\sqrt{1+(T\omega)^2\alpha^2}}{\sqrt{1+(T\omega)^2}} \angle(\tan^{-1}(\alpha T\omega) - \tan^{-1}(T\omega)$$

When $\alpha > 1$, the numerator gives higher phase lead than teh phase lag contributed by the denominator. As a result, it adds phase lead to the system. Alternatively, the transfer function can be represented by its zero and pole form,

$$C(s) = \alpha \frac{s + \frac{1}{\alpha T}}{s + 1/T} = \alpha \frac{s+z}{s+p} \begin{cases} \alpha > 1, z < p & \text{Phase lead or high pass filter} \\ \alpha < 1, p < z & \text{Phase lag or low pass filter} \end{cases}$$

$$C(s = j\omega) = \alpha \frac{j\omega + z}{j\omega + p} = \alpha \frac{\sqrt{\omega^2 + z^2}}{\sqrt{\omega^2 + p^2}} \angle(\tan^{-1} \frac{\omega}{z} - \tan^{-1} \frac{\omega}{p}$$

When $z < p$, than $\tan^{-1} \frac{\omega}{z} > \tan^{-1} \frac{\omega}{p}$ which adds to positive phase (i.e. lead) to the response of the system. Otherwise ($z > p$, than $\tan^{-1} \frac{\omega}{p} > \tan^{-1} \frac{\omega}{z}$, the net added phase becomes negative and we add lag to the system's response.

Example 81 *FRF of a high pass ($z < p$) and low pass ($z > p$) filters are given as follows (6.3)*

$$\text{Phase lead (} z < p) \text{ - High pass filter } C_{hi-pass}(s) = \frac{s + z_1}{s + p_1} = \frac{s + 10}{s + 100}$$

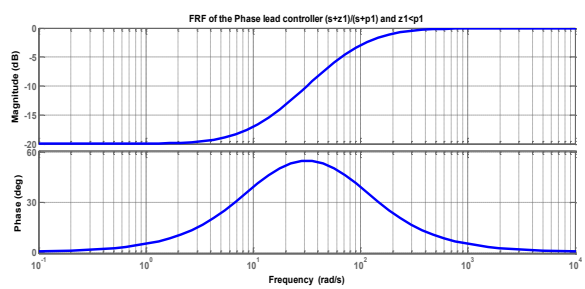
$$\text{Phase lag (} p < z) \text{ - Low pass filter } C_{lo-pass}(s) = \frac{s + z_2}{s + p_2} = \frac{s + 100}{s + 10}$$

Phase lead

$$C(s) = \alpha \frac{s+z}{s+p} \quad \leftarrow z < p$$

High Pass Filter

$$\boldsymbol{C(s) = \frac{s+10}{s+100}}$$



Phase lag

$$C(s) = \alpha \frac{s+z}{s+p} \quad \leftarrow z > p$$

Low Pass Filter

$$\boldsymbol{C(s) = \frac{s+100}{s+10}}$$

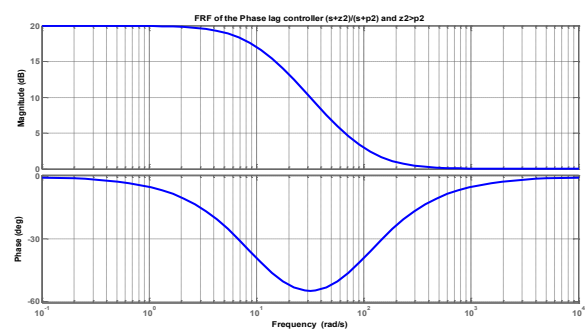
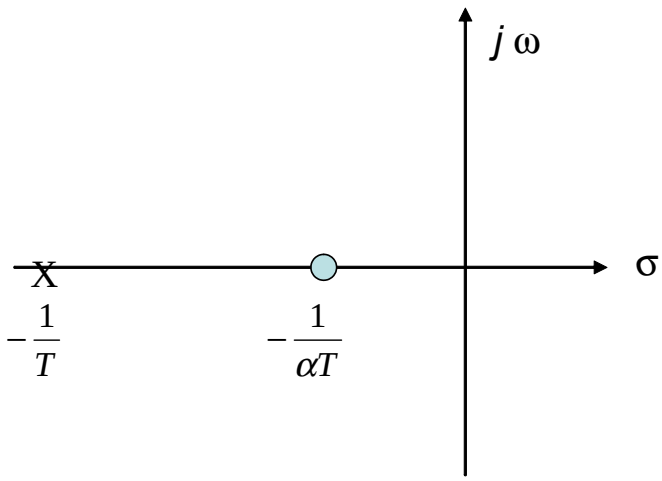
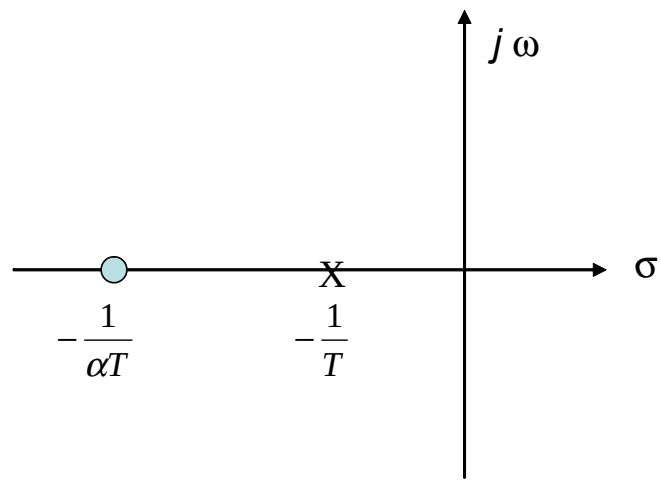


Figure 6.3: Lead - Lag Compensator example.



Lead Compensator α >1



Lag Compensator α < 1

The closed loop transfer function of the system with lead - lag compansator becomes:

$$G_{cl}(s) = \frac{C(s)G_p(s)}{1 + C(s)G_p(s)}$$

and the loop transfer function ($C(s)G(s)$) is

$$C(s) = \frac{1 + \alpha T s}{1 + T s}, \quad C(j\omega) = \frac{1 + j\alpha T\omega}{1 + jT\omega}$$
$$G_o(s) = C(s)G_p(s)$$

Bode or Nyquist plot of $G_o(s)$ is used to determine the gain and phase margins of the closed loop system. If we consider the Bode plot,

$$\begin{aligned}
G_o(j\omega) &= C(j\omega)G_p(j\omega) \\
|G_o(j\omega)| &= 20\log|C(j\omega)| + 20\log|G_p(j\omega)| \\
&= 20\left[\log\sqrt{1 + (\alpha T\omega)^2} - \log\sqrt{1 + (T\omega)^2}\right] + 20\log|G_p(j\omega)| \\
\phi &= \left[\tan^{-1}(\alpha T\omega) - \tan^{-1}(T\omega)\right] + \measuredangle G_p(j\omega)
\end{aligned}$$

- $\alpha > 1$ the system will have net increase in phase (+) and magnitude as the frequency increases (lead compensator)
- $\alpha < 1$ the system will have more phase lag and reduced magnitude as the frequency increases (lag compensator)

6.3.1 Design of Low Pass Filter

A simple low pass (LP) filter design is presented to illustrate the modification of Frequency Response Function of a dynamic system. LP filter allows the passage of signals at the low frequency, but removes the signals at

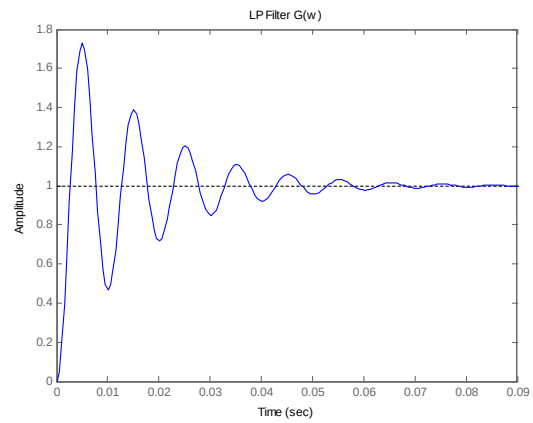


Figure 6.4: Step response of underdamped, second order plant.

the frequency beyond the LP's cut-off frequency. It is usually used to filter high frequency noise or signals from measurements which contain useful information within a desired frequency range.

Lets consider a single degree of freedom, structural dynamic system as the plant:

$$
\begin{aligned}
G_p(s) &= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \rightarrow \zeta \ll 1 \\
G_p(j\omega) &= \frac{\omega_n^2}{\omega_n^2 - \omega^2 + j2\zeta\omega_n\omega} \\
G_p(j\omega) &= \frac{\omega_n^2}{\sqrt{\left(\omega_n^2 - \omega^2\right)^2 + \left(2\zeta\omega_n\omega\right)^2}} \measuredangle \tan^{-1} \frac{-2\zeta\omega_n\omega}{\omega_n^2 - \omega^2}
\end{aligned}
$$

If a step input, static force of $F_0 = 1$,is applied to this underdamped system, it will exhibit the following response:

$$
x(t) = F_0 \left[1 - \frac{e^{-\zeta\omega_{\mathrm{n}} t}}{\sqrt{1-\zeta^2}} \cdot \sin(\omega_{\mathrm{d}} t + \psi) \right], \psi = \tan^{-1} \frac{\sqrt{1-\zeta^2}}{\zeta}
$$

with oscillations at its damped natural frequency $\omega_d = \omega_n\sqrt{1-\zeta^2}$ before it settles down at the static deflection level.

The objective is to minimize the oscillating part ($\sin(\omega_{\mathrm{d}} t)$), or low pass filter it from the response. Lets design a low pass filter with a cut off frequency of $\omega_c = 0.1\omega_n\sqrt{1-\zeta^2}$,

$$
\begin{aligned}
C(s) &= \frac{\omega_c}{s + \omega_c} \\
C(j\omega) &= \frac{\omega_c}{\omega_c + j\omega} = \frac{\omega_c}{\sqrt{\omega_c^2 + \omega^2}} \measuredangle - \tan^{-1} \frac{\omega}{\omega_c}
\end{aligned}
$$

$C(j\omega)$ has a flat gain but starts attenuating its magnitude and increasing its phase lag as the frequency approaches to $\omega = \omega_c$.

$$
\begin{aligned}
|C(j\omega)|_{\omega=\omega_c} &= \frac{\omega_c}{\sqrt{\omega_c^2 + \omega_c^2}} = \frac{1}{\sqrt{2}} = 0.707 \\
\phi &= -\tan^{-1} \frac{\omega_c}{\omega_c} = -\frac{\pi}{4}
\end{aligned}
$$

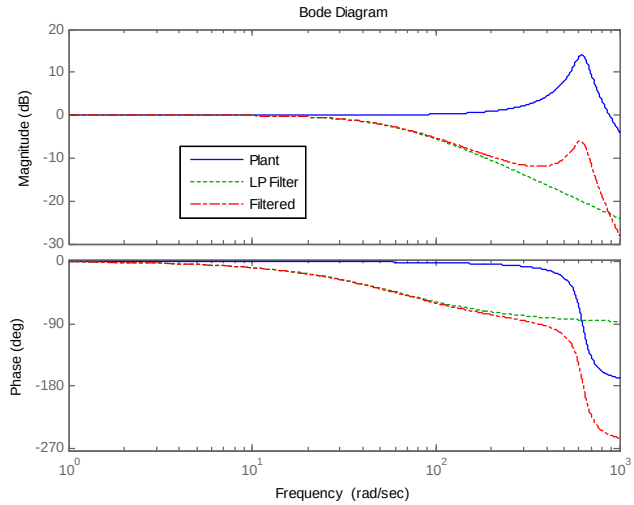


Figure 6.5: Bode diagram of a second order plant $[G(\omega_n = 200\pi, \zeta = 0.1)$, low pass filter $[C(\omega_c = 0.1\omega_n\sqrt{1-\zeta^2})$ and compensated system (CG).

The frequency response of the combined plant and low pass filter becomes,

$$
\begin{aligned}
GC(s) &= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \times \frac{\omega_c}{s + \omega_c} \\
GC(j\omega) &= \frac{\omega_n^2}{\omega_n^2 - \omega^2 + j2\zeta\omega_n\omega} \times \frac{\omega_c}{\omega_c + j\omega} \\
|GC(j\omega)| &= \frac{\omega_n^2}{\sqrt{(\omega_n^2 - \omega^2)^2 + (2\zeta\omega_n\omega)^2}} \times \frac{\omega_c}{\sqrt{\omega_c^2 + \omega^2}} \\
\phi &= \tan^{-1}\frac{-2\zeta\omega_n\omega}{\omega_n^2 - \omega^2} - \tan^{-1}\frac{\omega}{\omega_c}
\end{aligned}
$$

Bode plot of the plant (G_p), low pass filter (C) and combined system (CG_p) are shown in Figure 6.5. The magnitude of the output at resonance frequency is dropped from $13.6dB$ to $-6.2dB$ at the expense of worsening the phase lag after the cut off frequency.

The step response of the filtered system has now less oscillations as can be seen in Figure 6.6.

6.3.2 Design of Phase-Lead Controller

The transfer function of the phase lead system is :

$$
\begin{aligned}
C(s) &= \frac{1 + \alpha T s}{1 + Ts} \left\{ \ \alpha > 1 \quad \alpha T \succ T \right. \\
C(s) &= \alpha\frac{s + \frac{1}{\alpha T}}{s + \frac{1}{T}} \left\{ \ \alpha > 1 \quad \frac{1}{\alpha T} \prec \frac{1}{T} \right.
\end{aligned}
$$

Since $\frac{1}{\alpha T} < \frac{1}{T}$, zero of the system is closer to the imaginary axis than the pole. Lets assume that the desired extra phase lead at frequency ω_m is identified to be ϕ_m from the open loop frequency response of the system. The frequency ω_m is placed between the geometric mean of $\frac{1}{\alpha T}$ and $\frac{1}{T}$ as:

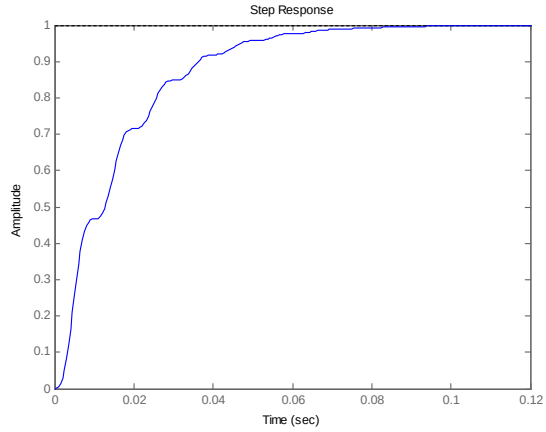


Figure 6.6: Step response of low pass filtered system (GC).

$$\omega_m^2 = \frac{1}{\alpha T} \cdot \frac{1}{T} \quad \rightarrow \quad \omega_m = \sqrt{\frac{1}{\alpha T} \cdot \frac{1}{T}} = \frac{1}{T\sqrt{\alpha}}$$

The phase contributed by the lead compensator is :

$$\begin{aligned}
\phi(\omega) &= \tan^{-1}(\alpha T\omega) - \tan^{-1}(T\omega) \\
\tan\phi(\omega) &= \frac{\alpha T\omega - T\omega}{1 + \alpha T\omega \cdot T\omega} \\
\tan\phi_m(\omega_m = \frac{1}{T\sqrt{\alpha}}) &= \frac{\alpha T \frac{1}{T\sqrt{\alpha}} - T\frac{1}{T\sqrt{\alpha}}}{1 + \alpha T \frac{1}{T\sqrt{\alpha}} \cdot T\frac{1}{T\sqrt{\alpha}}} = \frac{\alpha - 1}{2\sqrt{\alpha}} \\
\sin\phi_m &= \frac{\alpha - 1}{\alpha + 1} \rightarrow \alpha = \frac{1 + \sin\phi_m}{1 - \sin\phi_m}
\end{aligned}$$

Note: $\tan^2\phi_m = \frac{\sin^2\phi_m}{\cos^2\phi_m} = \frac{\sin^2\phi_m}{1-\sin^2\phi_m} = \left(\frac{\alpha-1}{2\sqrt{\alpha}}\right)^2$, $\left[4\alpha + (\alpha-1)^2\right]\sin^2\phi_m = (\alpha-1)^2$, $\sin^2\phi_m = \frac{(\alpha-1)^2}{4\alpha+\alpha^2-2\alpha+1} = \frac{(\alpha-1)^2}{(\alpha+1)^2} \rightarrow \sin\phi_m = \frac{\alpha-1}{\alpha+1}$

The frequency (ω_m), where the phase lead (ϕ_m) is desired, is already selected from the frequency response of the uncompensated open loop system. The corresponding frequency span factor is found :

$$\omega_m = \frac{1}{T\sqrt{\alpha}} \rightarrow T = \frac{1}{\omega_m\sqrt{\alpha}}$$

The phase lead compensator adds a zero on the right side of the pole with the following effects.

- Making the zero closer to the imaginary axis or origin, the system's rise time becomes shorter but the overshoot may increase since the open loop zero appears as the zero of the closed loop transfer function of the system as well..
- Since the net gain increase at steady state ($s \rightarrow 0$) is unity, the steady state error is not affected by the lead compensator.
- The phase of the forward path transfer function around the gain cross over frequency increases, which improves the phase margin of the closed loop system.

- The slope of the system around gain cross over frequency reduces, leading to improved gain and phase margins.
- The bandwidth of the close loop system increases, which leads to faster response. If the original system is unstable or with low stability margin, the parameter α may need to be too high which creates unfavorable amplification of noise at the cross over frequency zone. Also, a single stage phase lead compensator may not be enough if the system has low stability margin. It must be noted that the maximum phase lead can be only 90 deg, and additional lead compensator may need to be cascaded to obtain higher phase margin.

Example: Consider a position control system with the following type 1 plant dynamics which has a time constant of $\tau = 0.01s$ or an open loop pole at 100 [rad/s]:

$$G(s) = \frac{3000}{s(s+100)} = \frac{30}{s(0.01s+1)}$$

We wish to design a lead compensator which pushes the zero towards origin to reduce the rise time. Since this is a second order open loop stable system, the phase lag will never be less than -180 deg hence the gain margin is infinite. The gain cross over frequency is $\omega_g = 28.8$ [rad/s] with a phase margin of 74 degrees. If we wish to add additional 30 deg to the phase margin, the parameter α is evaluated as:

$$\alpha = \frac{1+\sin\phi(\omega_m)}{1-\sin\phi(\omega_m)} = \frac{1+\sin\pi/6}{1-\sin\pi/6} = 3.0$$

We wish to place the geometric mean of the system at the gain cross over frequency. The uncompensated system has a gain cross over frequency of 28.8 [rad/s], where the phase is $\phi = -106$ deg with 74 deg phase margin. If we wish to add +30 deg phase at the gain cross over frequency $\omega_g = \omega_m = 28.8[rad/s]$.

$$\omega_m = \frac{1}{\sqrt{\alpha}T} \rightarrow T = \frac{1}{\omega_m\sqrt{\alpha}} = \frac{1}{28.8\sqrt{3.0}} = 0.02$$

The compensator becomes

$$C(s) = \frac{1+\alpha Ts}{1+Ts} = C(s) = \frac{1+3.0\times 0.02s}{1+0.02s} = \frac{1+0.06s}{1+0.02s} = 3\frac{s+16.67}{s+50}$$

The compensated system has now extra +30 deg phase at $\omega_m = 28.8[rad/s]$, and new phase margin is 82 deg at new gain cross over frequency of $\omega_g = 61.6[rad/s]$.The gain can be tuned to achieve desired rise time and overshoot using root locus technique. Bode plots of uncompensated (G), lead compensator (C), and the compensated (C*G) are shown in Figure 6.7. The design requires several iterations until the performance is found to be satisfactory. An interactive routine can be written in MATLAB to reduce the design efforts. A sample MATLAB code is given below for this example.

```
% Yusuf Altintas, University of British Columbia
% November 2010
% MECH 467 Notes
%
% LEAD - COMPENSATOR DESIGN%
clear all;
close all;
% open loop tf of the plant in s domain
G=tf([30],[0.01 1 0]);
w=[1:1:1000]; % Create an array of frequencies in rad/s
figure(1); clf;
[mag,phase]=bode(G,w); % Obtain Bode Plot of G
[gms,pms,ws1,ws2]=margin(G);
margin(G);
title(sprintf('Bode Plot of the open position loop \n Gms = %4.4g at %4.3g rad/sec, PM = %4.4g at %4.3g rad/sec ', gms, ws1, pms, ws2));
```

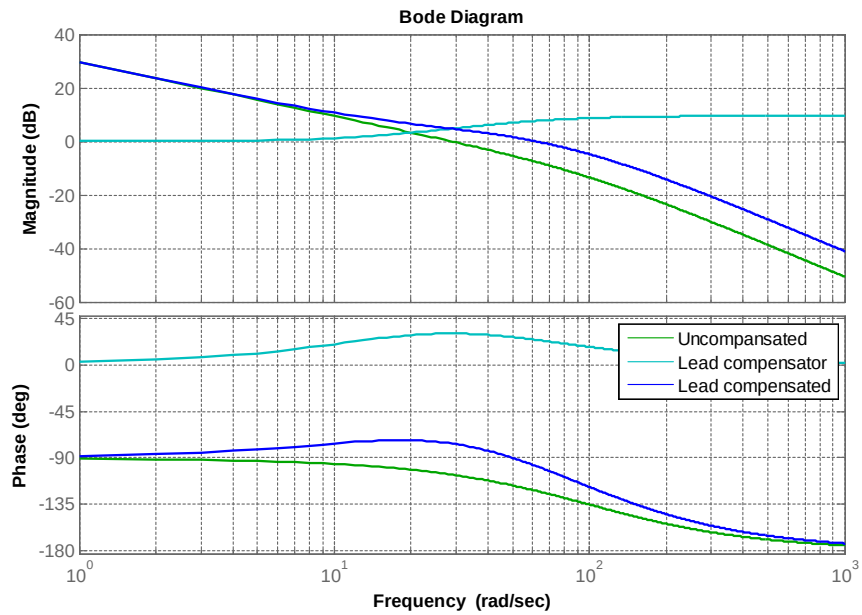


Figure 6.7: Uncompansated system: $G(s) = \frac{30}{s(0.01s+1)}$. Lead compensator : $C(s) = \frac{1+0.06014s}{1+0.02005s}$. Compensated system: $C(s)G(s) = \frac{1.804s+30}{s(0.002s^2+0.03s+1)}$

```
grid
% Lead Compensator
phim=pi/6;
wm=28.8; % Gain cross over frequency
a=(1+sin(phim))/(1-sin(phim)) ;
T=1/(wm*sqrt(a));
C=tf([a*T 1],[T 1]);
CG=C*G;
figure(2); clf;
bode(G,'g',C,'c',CG,'b',w);
legend('uncompansated','lead compensator','compensated');
grid
figure(3); clf;
[gmsc,pmsc,wsc1,wsc2]=margin(CG);
margin(CG);
grid
```

6.3.3 Design of Phase-Lag Controller

The phase lag controller or low pass filter system is defined by the following transfer function:

$$C(s) = \frac{s + \frac{1}{\alpha T}}{s + \frac{1}{T}} = \frac{1}{\alpha}\frac{1 + \alpha T s}{1 + T s}, \ \alpha < 1$$

which has the following properties:

- The zero frequency gain of the controller is greater than unity ($\frac{1}{\alpha} > 1 \longleftarrow \alpha < 1$)
- The damping will increase. The pole and zero must be placed close to each other. Especially, the zero and pole of the lag compensator for Type 0 and 1 systems must be placed near the origin.

- The gain cross over frequency is decreased, hence the bandwidth is reduced.
- The magnitude of the forward path transfer function is attenuated near the gain cross over frequency, hence the stability is improved. The attenuation is about $20 \log \alpha$
- The system becomes slower to respond, i.e. the rise and settling times become longer due to reduced bandwidth.
- The system becomes more sensitive to the changes in the transfer function of the plant.

The Bode plot based lag compensator design procedure is outlined as follows:

- Bode plot of the forward transfer function is drawn, and the forward gain is set based on the steady state performance requirements. The phase and gain margins of the uncompensated system are determined.
- The new cross over frequency where the phase needs to be improved must be decided on the Bode plot.
- The amount of gain reduction at the new cross over frequency can be obtained from the Bode plot, which leads to estimation of α

$$|G_p(j\omega_g)| = -20 \log_{10} \alpha \quad [dB] \quad \longleftarrow \alpha < 1$$
$$\alpha = 10^{-|G_p(j\omega_g)|/20}$$

- As a rule of thumb, put the pole $\frac{1}{\alpha T} = \frac{\omega_g}{10}$, which leads to $\frac{1}{T} = \alpha \frac{\omega_g}{10}$ [rad/s].
- Check the Bode plot of the compensated system, and adjust α, T until the system is found to be satisfactory.
- It is possible to tune the forward loop gain of the compensated system by considering the overshoot or rise time using Root locus technique.

Example: Consider a position control system with the following type 1 plant dynamics which has a time constant of $\tau = 0.01s$ or an open loop pole at 100 [rad/s]:

$$G(s) = \frac{3000}{s(s+100)} = \frac{30}{s(0.01s+1)}$$
$$G(j\omega) = \frac{3000}{\omega\sqrt{\omega^2+100^2}}, \phi = -\frac{\pi}{2} - \tan^{-1}\frac{\omega}{100}$$

The gain margin is infinite and the gain cross over frequency is $\omega_{ug} = 28.8[rad/s]$ with a corresponding phase margin of $180 - 106 = 74\,\text{deg}$. Lets select a new corner frequency as 70 [rad/s] where we demand higher phase margin. The gain at 70 [rad/s] is $-8dB$, which gives $\alpha = 10^{-8/20} = 0.3981$, $T = \frac{10}{\alpha\omega_g} = 10/(\alpha 70) = 0.3588$. The lag compensator becomes:

$$\begin{aligned} C(s) &= \frac{s+\frac{1}{\alpha T}}{s+\frac{1}{T}} = \frac{1}{\alpha}\frac{1+\alpha T s}{1+Ts}, \ \alpha < 1 \\ C(s) &= \frac{1}{0.3981}\frac{1+0.1429s}{1+0.3588s} = \frac{1}{0.3981}\frac{0.1429}{0.3588}\frac{s+\frac{1}{0.1429}}{s+\frac{1}{0.3588}} = \frac{s+6.9979}{s+2.7871} \end{aligned}$$

Example 82 *Loop Shaping controller design.*

Transfer function of a machine with a seond order lag dynamics is given as:

$$G(s) = \frac{K_0}{(\tau_1 s+1)(\tau_2 s+1)} = \frac{10}{(0.1s+1)(0.01s+1)}$$

Design a loop shaping controller to meet the following specifications (Figure 6.8):

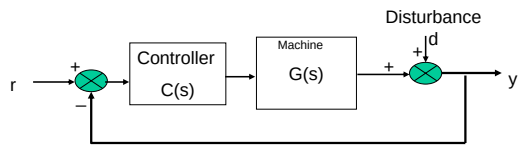


Figure 6.8: Loop shaping controller design for a second order lag system.

- Steady state error to step reference input and constant disturbance are zero and less than 0.01 for a unit ramp input
- The phase margin at open loop gain cross over frequency is 60 degrees

The transfer function of the closed loop system is derived as:

$$\begin{aligned} y &= (r-y)CG + d \\ (1+CG)y &= CGr + d \\ y(s) &= \frac{CG}{1+CG}r(s) + \frac{1}{1+CG}d(s) \end{aligned}$$

The tracking error $e(s)$of the closed loop system is:

$$\begin{aligned} e(s) &= r(s) - y(s) = r(s) - \frac{CG}{1+CG}r(s) - \frac{1}{1+CG}d(s) \\ e(s) &= \left[1 - \frac{CG}{1+CG}\right] r(s) - \frac{1}{1+CG}d(s) \\ &= \left[\frac{1+CG-CG}{1+CG}\right] r(s) - \frac{1}{1+CG}d(s) = \left[\frac{1}{1+CG}\right] r(s) - \frac{1}{1+CG}d(s) \\ e(s) &= \left[\frac{1}{1+CG}\right] [r(s) - y(s)] \end{aligned}$$

Let the reference step input is $r(t) = R_0 \rightarrow r(s) = R_0/s$ and the disturbance has a constant amplitude $D_0 \rightarrow d(s) = D_0/s$.The tracking error becomes:

$$e(s) = \left[\frac{1}{1+CG}\right] \frac{R_0 - D_0}{s}$$

The steady state error when the reference and disturbance inputs are constant (i.e. step) is:

$$\begin{aligned} e_{ss} &= \lim_{s=0} s \cdot e(s) = \lim_{s=0} s \cdot \left[\frac{1}{1+CG}\right] \frac{R_0 - D_0}{s} = \lim_{s=0} \left[\frac{1}{1+CG}\right] (R_0 - D_0) \\ e_{ss} &= \lim_{s=0} \left[\frac{1}{1 + C\frac{K_0}{(\tau_1 s+1)(\tau_2 s+1)}}\right] (R_0 - D_0) = \lim_{s=0} \left[\frac{(\tau_1 s+1)(\tau_2 s+1)}{(\tau_1 s+1)(\tau_2 s+1) + K_0 C}\right] (R_0 - D_0) \end{aligned}$$

If only a proportional controller with a gain C_0 is used, the steady state error will not be zero as shown below:

$$e_{ss} = \lim_{s=0} \left[\frac{(\tau_1 s+1)(\tau_2 s+1)}{(\tau_1 s+1)(\tau_2 s+1) + K_0 C_0}\right] (R_0 - D_0) = \left[\frac{1}{1+K_0 C_0}\right] (R_0 - D_0)$$

We need to have s in the numerator to push $e_{ss} \to 0$. In other words, the controller must have an integral action to accumulate the error and push it towards zero, i.e. $C_1(s) = K_I/s$

$$\begin{aligned} e_{ss} &= \lim_{s=0} s \left[\frac{1}{1 + \frac{K_I}{s} \frac{K_0}{(\tau_1 s+1)(\tau_2 s+1))}} \right] \frac{R_0 - D_0}{s} \\ &= \lim_{s=0} \left[\frac{(\tau_1 s+1)(\tau_2 s+1)}{(\tau_1 s+1)(\tau_2 s+1) + K_0 K_I} \right] (R_0 - D_0) = \frac{0}{0 + K_0 K_I} (R_0 - D_0) = 0 \end{aligned}$$

The required steady state error for a unit ramp input $r(t) = t \to r(s) = \frac{1}{s^2}$ must be $e_{ss,r} \preceq 0.01$.

$$\begin{aligned} e_{ss} &= \lim_{s=0} s \left[\frac{1}{1 + \frac{K_I}{s} \frac{K_0}{(\tau_1 s+1)(\tau_2 s+1))}} \right] \frac{1}{s^2} = \lim_{s=0} s \left[\frac{s(\tau_1 s+1)(\tau_2 s+1)}{s(\tau_1 s+1)(\tau_2 s+1) + K_0 K_I} \right] \frac{1}{s^2} \\ &= \lim_{s=0} \left[\frac{(\tau_1 s+1)(\tau_2 s+1)}{s(\tau_1 s+1)(\tau_2 s+1) + K_0 K_I} \right] = \frac{1}{K_0 K_I} \preceq 0.01 \\ K_I &\succeq \frac{1}{10 \cdot 0.01} \to K_I \succeq 10 \to \text{let} \quad K_I = 10 \end{aligned}$$

The loop transfer function of the combined integral controller and plant becomes:

$$L_1(s) = \frac{K_I}{s} G(s) = \frac{10}{s} \frac{10}{(0.1s+1)(0.01s+1)}$$

The FRF of the loop transfer function $L_1(s)$ is:

$$\begin{aligned} L_1(j\omega) &= \frac{K_I}{j\omega} G(j\omega) = \frac{K_I}{s} \frac{K_0}{(\tau_1 s+1)(\tau_2 s+1)} = \frac{10}{j\omega} \frac{10}{(0.1j\omega+1)(0.01j\omega+1)} \\ |L_1(j\omega)| &= \frac{10}{\omega} \frac{10}{\sqrt{1+(0.1\omega)^2}\sqrt{1+(0.01\omega)^2}} \text{ linear scale} \\ |L_1(j\omega)| &= 20 - 20\log\omega + 20 - 20\log\sqrt{1+(0.1\omega)^2} - 20\log\sqrt{1+(0.01\omega)^2} \\ \phi &= -\frac{\pi}{2} - \tan^{-1} 0.1j\omega - \tan^{-1} 0.01j\omega \end{aligned}$$

Bode plots show that the gain cross over frequency $\omega_g = 30.1$ rad/s with a corresponding phase margin of 1.6 deg .The phase cross over frequency is $\omega_p = 31.6$ with a gain margin of only $0.8dB$.

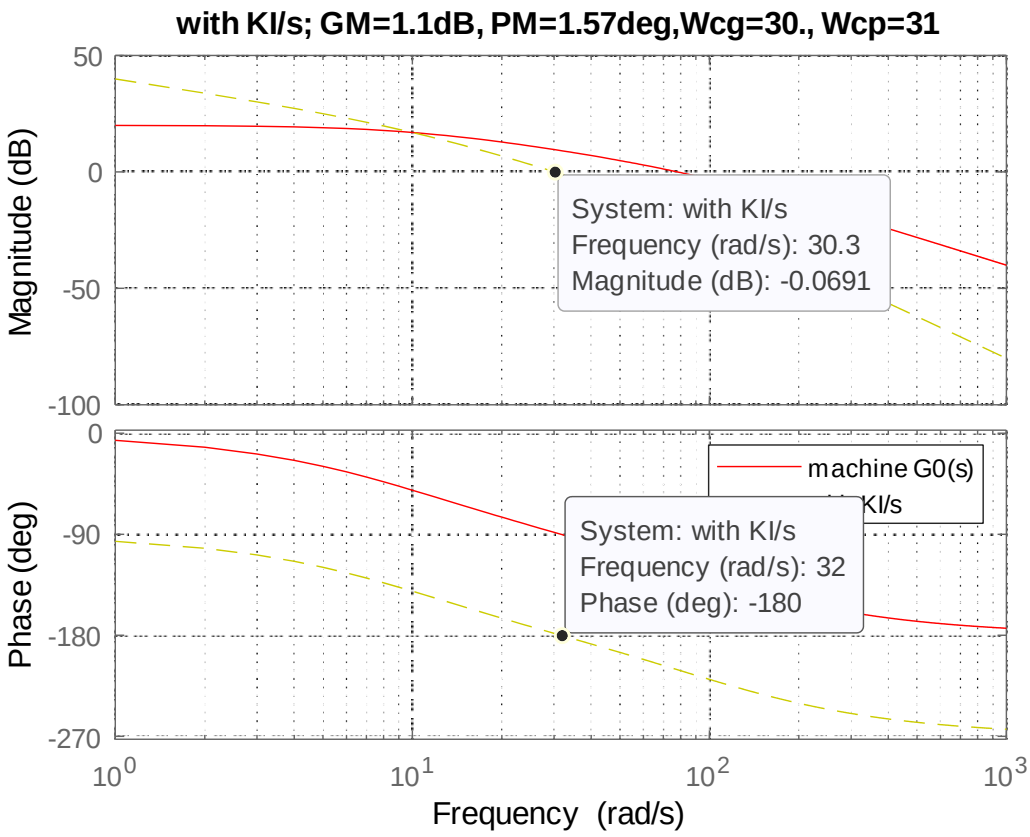
The requirement is to increase the phase margin to 60 deg at around 30Hz. Since we already have 1.6 deg phase margin, we need to add additional $\phi_m = 60 - 1.6 = +58.4$ deg phase lead at $\omega_m = 30rad/s$.

The lead compensator is:

$$C(s) = C_0 \frac{1 + \alpha T s}{1 + T s}$$

$$\begin{aligned} \alpha &= \frac{1+\sin\phi_m}{1-\sin\phi_m} = \frac{1+\sin 58.4}{1-\sin 58.4} = 12.4886 \\ T &= \frac{1}{\omega_m\sqrt{\alpha}} = \frac{1}{30\sqrt{12.4886}} = 0.0094 \\ C'(s) &= C_0 \frac{1 + 12.4886 \cdot 0.0094s}{1 + 0.0094s} = C_0 \frac{1 + 0.1178s}{1 + 0.0094s} \end{aligned}$$

The magnitude change brought by $\frac{1+\alpha Ts}{1+Ts}$ at $\omega_m = 30\ rad/s$ must be compensated by C_0 in order to keep the oorignal system gain .


with KI/s; GM=1.1dB, PM=1.57deg,Wcg=30., Wcp=31
Magnitude (dB)
50
0
-50
-100
System: with KI/s
Frequency (rad/s): 30.3
Magnitude (dB): -0.0691
Phase (deg)
0
-90
-180
-270
machine G0(s)
KI/s
System: with KI/s
Frequency (rad/s): 32
Phase (deg): -180
10^0
10^1
10^2
10^3
Frequency (rad/s)

$$
\begin{aligned}
\left|\frac{1+j\alpha T\omega}{1+jT\omega}\right| &= \frac{\sqrt{1+(\alpha T\omega_m)^2}}{\sqrt{1+(T\omega_m)^2}} = \frac{\sqrt{1+(0.1178\times 30)^2}}{\sqrt{1+(0.0094\times 30)^2}} = 3.534 \\
C_0 &= \frac{1}{3.534} = 0.283
\end{aligned}
$$

Hence the loop shaping controller must have the following transfer function:

$$
C(s) = C_0\frac{1+\alpha Ts}{1+Ts}\frac{K_I}{s} = 0.283\frac{1+0.1178s}{1+0.0094s}\frac{10}{s}
$$

Now the controller is designed, we can look at FRF of the loop and its gain-phase margins.

$$
\begin{aligned}
L(s) &= C_0\frac{1+\alpha Ts}{1+Ts}\frac{K_I}{s}G(s) = 0.283\frac{1+0.1178s}{1+0.0094s}\frac{10}{s}\frac{10}{(0.1s+1)(0.01s+1)} \\
L(j\omega) &= 0.283\frac{1+j0.1178\omega}{1+j0.0094\omega}\frac{10}{j\omega}\frac{10}{(0.1j\omega+1)(0.01j\omega+1)} \\
&= 20\log 0.283 + 20\log\sqrt{1+(0.1178\omega)^2} - 20\log\sqrt{1+(0.0094\omega)^2} + 20\log 10 - 20\log\omega \\
&\quad +20\log 10 - 20\log\sqrt{1+(0.1\omega)^2} - 20\log\sqrt{1+(0.01\omega)^2} \\
\phi_L &= +\tan^{-1}0.1178\omega - \tan^{-1}0.0094\omega - \frac{\pi}{2} - \tan^{-1}(0.1\omega) - \tan^{-1}(0.01\omega)
\end{aligned}
$$

Now, the gain cross over frequency is $\omega_{Lg} = 30rad/s$ with a $\phi_{pm} = 60\,\text{deg}$ phase margin as designed. The phase cross over frequency is $\omega_{Lp} = 104\ rad/s$ with a gain margin of $16dB$.

We can now close the loop and check the FRF of the closed loop system and its step response.

$$
\begin{aligned}
y(s) &= \frac{CG}{1+CG}r(s) + \frac{1}{1+CG}d(s) \\
G_{cl}(s) &= \frac{C(s)G(s)}{1+C(s)G(s)} = \frac{C_0\frac{1+\alpha Ts}{1+Ts}\frac{K_I}{s}\frac{K_0}{(\tau_1 s+1)(\tau_2 s+1)}}{1+C_0\frac{1+\alpha Ts}{1+Ts}\frac{K_I}{s}\frac{K_0}{(\tau_1 s+1)(\tau_2 s+1)}} \\
&= C_0K_IK_0\frac{1+\alpha Ts}{s\,(1+Ts)\,(\tau_1 s+1)(\tau_2 s+1) + C_0K_IK_0\,(1+\alpha Ts)}
\end{aligned}
$$

If the step response does not have a desired performance, i.e. overshoot, C_0 can be replaced by K_pC_0 in $C(s)$.Bode plot of loop and step response of the closed loop system can be iteratively solved entire a satisfactory K_p is designed through root locus with a desired damping damping ratio(i.e. overshoot).

The FRF of the closed loop system describes its bandwidth and the full performance of the entire control system.

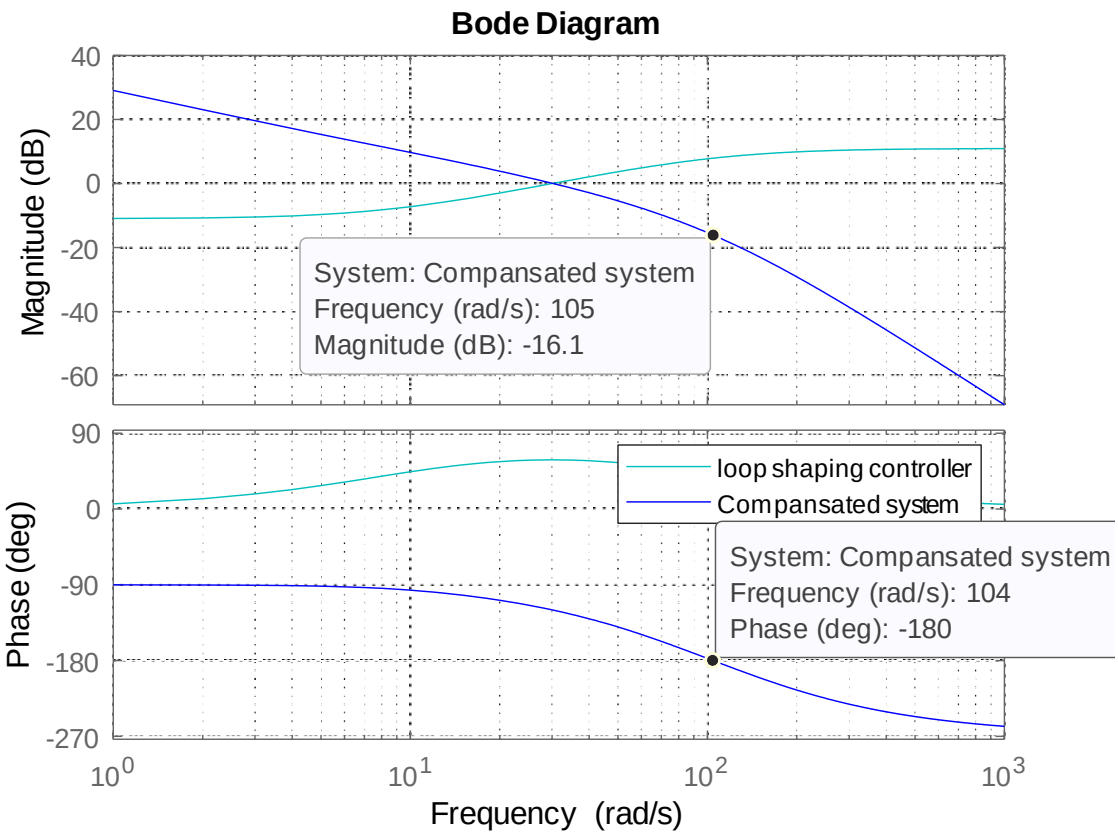
$$
G_{cl}(s=j\omega) = C_0K_IK_0\frac{1+\alpha Ts}{s\,(1+Ts)\,(\tau_1 s+1)(\tau_2 s+1) + C_0K_IK_0\,(1+\alpha Ts)}\Bigg|_{s=j\omega}
$$

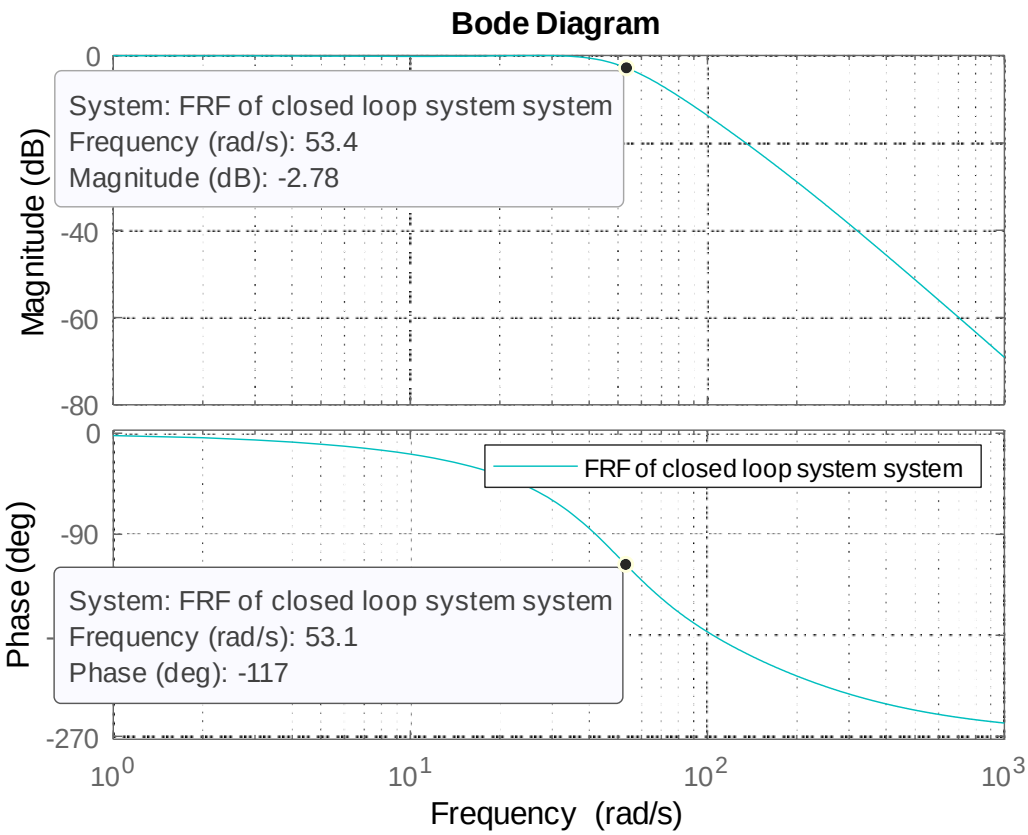
The bandwidth is the frequency when the magnitude starts deviating from unity (or $0dB$ in log scale), which is about $53rad/s$ with the designed controller. Higher the bandwidth is, faster the response will be.

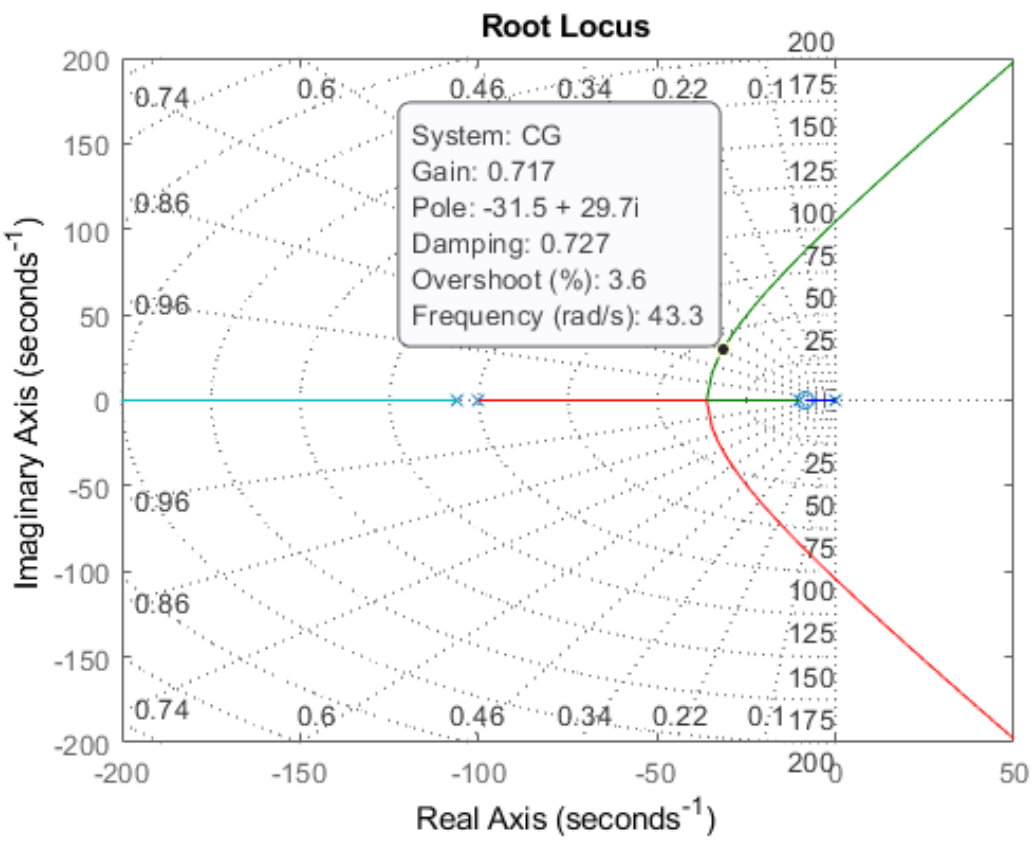
The root locus design can be iteratively repeated until a satisfactory performance is achieved.

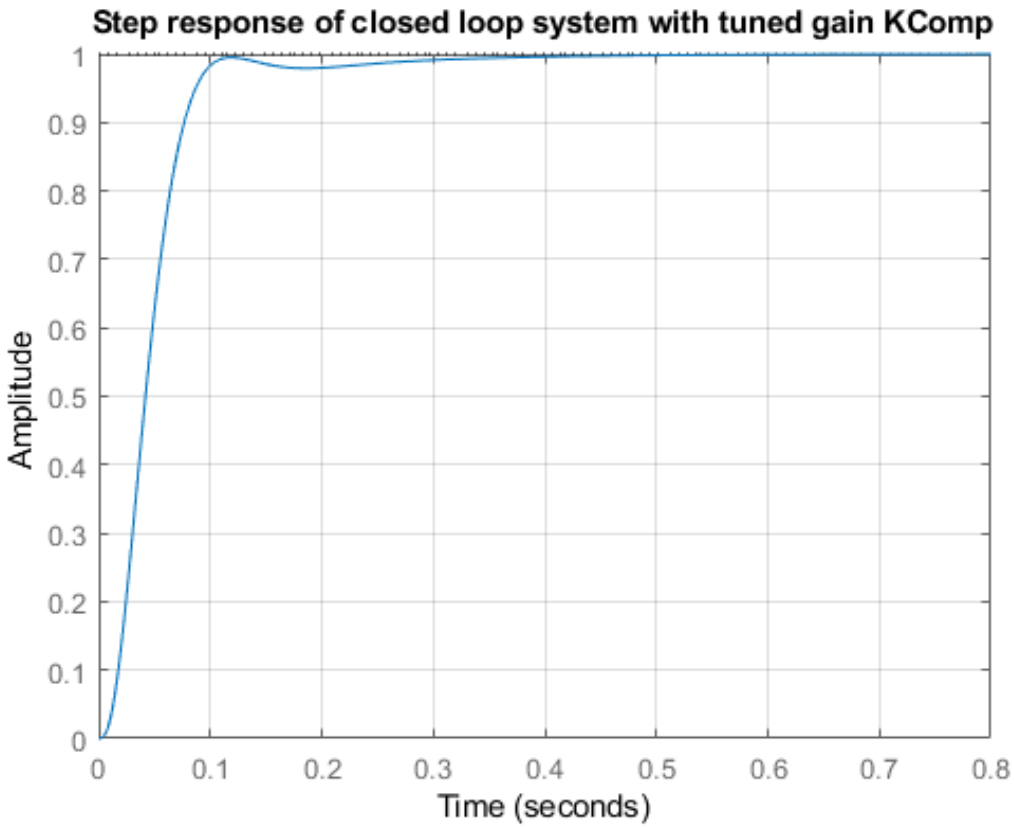
K_p =0.7259 for $\zeta = 0.707$

MATLAB Porgram for the loop shaping example.


Bode Diagram
Magnitude (dB)
40
20
0
-20
-40
-60
System: Compansated system
Frequency (rad/s): 105
Magnitude (dB): -16.1
Phase (deg)
90
0
-90
-180
-270
loop shaping controller
Compansated system
System: Compansated system
Frequency (rad/s): 104
Phase (deg): -180
10^0
10^1
10^2
10^3
Frequency (rad/s)


Bode Diagram
System: FRF of closed loop system system
Frequency (rad/s): 53.4
Magnitude (dB): -2.78
Magnitude (dB)
0
-40
-60
-80
FRF of closed loop system system
System: FRF of closed loop system system
Frequency (rad/s): 53.1
Phase (deg): -117
Phase (deg)
0
-90
-270
10^0
10^1
10^2
10^3
Frequency (rad/s)


Root Locus
System: CG
Gain: 0.717
Pole: -31.5 + 29.7i
Damping: 0.727
Overshoot (%): 3.6
Frequency (rad/s): 43.3
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
200
150
100
50
0
-50
-100
-150
-200
-200
-150
-100
-50
0
50
0.74
0.6
0.46
0.34
0.22
0.1
0.86
0.96
175
125
75
25


Step response of closed loop system with tuned gain KComp
Amplitude
Time (seconds)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

```
% MECH 467/589: Loop Shaping controller (C(s)=(KI/s)(1+alfa*T*s)/(1+Ts))
% design for a second order plant P(s)=10/[(0.1s+1)(0.01s)]
% Author: Yusuf Altintas, Professor
% Date: October 28, 2019
% University of British Columbia
% Manufacturing Automation Laboratory www.mal.mech.ubc.ca
%Steps
% 1- Plot FRF of the plant P(s)
% 2- Add an integral controller KI/s to minimize the steady state error for
% ramp input
% 3- Design a lead compnsator to improve the Gain-Phase margin as desired
% 4- Plot FRF of the loop transfer function with a complete controller
% 5- Plot Closed loop frf and check the bandwidth; Plot Step response
% 6- Apply Root Locus on the loop FRF including controller. If yo wish to
% increase damping (i.e. reduce overshoot); select a damping to find
% scale gain factor KComp for the Controller
% 7- Multiply C(S) with KComp and check the Bode of loop FRF to see new
% gain-phase margins
% 8- Check closed loop FRF - bandwidth and step response
% 9- Iterate until you are satisfied with the design
%
close all;
clear all;
T1 = 1/10;
T2 = 1/100;
K0 =10;
w=1:1:1000;
den= conv([T1 1], [T2 1]); %product of two tf s
G0 =tf(K0,den) ;% G(s)=K0/[(T1*s+1)(T2*s+1)];
% hold on;
KI=10;
KIs=tf(KI,[1 0]);
L1=KIs*G0; ;% G(s)=(KI/s)K0/[(T1*s+1)(T2*s+1)];
L1=zpk(L1);
figure(1);
bode(G0,'r',L1,'y–',w);
grid on;
[G0m,P0m,W0cp,W0cg]=margin(G0);
[Gm1,Pm1,Wcp1,Wcg1]=margin(L1);
marginsu=[20*log10(Gm1),Pm1,Wcg1,Wcp1];
legend('machine G0(s)','with KI/s');
title('with KI/s; GM=1.1dB, PM=1.57deg,Wcg=30., Wcp=31');
% Design a phase lead compensator at 30 rad/s add + 18.4deg phase
wm=30; %rad/s where the phase lead is needed
phim=58.4*pi/180;
alfa=(1+sin(phim))/(1-sin(phim));
T=1/(sqrt(alfa)*wm);
MagC=sqrt(1+(alfa*T*wm)^2)/sqrt(1+(T*wm)^2);
C=(1/MagC)*tf([alfa*T 1],[T 1]);
figure(2);
bode(C,w);
grid on;
title('Lead compensator C(s)=(1+alfaT)/(1+T); T=0.0.02404; alfa=1.9223');
CG=C*L1;
```

```
figure(3);
bode(C,'c',CG,'b',w);
legend('loop shaping controller','Compansated system');
grid on;
[Gmc,Pmc,Wcpc,Wcgc]=margin(CG);
margins=[20*log10(Gmc),Pmc,Wcgc,Wcpc];
sprintf('Gain Margin GM =%g dB',20*log10(Gmc))
sprintf('Phase Margin PM =%g deg',Pmc)
sprintf('Gain crossover wg =%g (rad/s)',Wcgc)
sprintf('Phase cross over =%g (rad/s)',Wcpc)
% Analyze the performance of closed loop system
CGcl=feedback(CG,1); % Transfer function of closed loop system
figure(4);
bode(CGcl,'c',w);
grid on;
legend('FRF of closed loop system system');
%
% Generate step response of the system
figure (5);
step(CGcl);
grid on;
title('Step response of closed loop system');
% Tune the gain using root locus for a desired overshoot
%
% Loop until you are happy with Loop FRF and step response of closed
% loop system with damping gain compansation (KComp)
for i=1:5
figure(6);
zoom on;
rlocus(CG);                          % display root locus
sgrid;                               % show grid (as wn vs. zeta)
axis([-200 50 -200 200])
%
% Dashed circles represent constant natural frequency loci, or magnitude of poles
% Lines extending from the coordinate center (zero point) represents constant
% damping loci
% Cross (x) points represent poles of the open loop system
%
disp('Select points on the locus with the mouse');
[KComp,poles]=rlocfind(CG);                          % user selects feedback gain
sprintf('Kcomp =%g',KComp);
disp(['Poles of the designed closed system with Kcomp: ' num2str(KComp) ]);
poles                % Print the designed poles
figure(7);
bode(CG,'c',KComp*CG,'b',w);
legend('lead-lag compansated','+Damping compansated');
grid on;
% Obtain the closed loop transfer function
GcKcomp=feedback(CG*KComp,1);                        % closed-loop t.f.
figure(8);
step(GcKcomp);
grid on;
title('Step response of closed loop system with tuned gain KComp');
pause
```

end % end the root locus design iteration

6.4 Summary of Stability of Closed Loop Control Systems

A block diagram of a closed loop system is given in 6.9. The physical machine, the plant, is always modeled ($G_p(s)$) in Laplace domain. If the sensor has analog circuitry with dynamics, it has to be modeled in Laplace domain ($H(s)$) as well like the plant. The controller can be designed in either Laplace domain ($D(s)$) in continuous time domain or in z domain ($D(s)$) in discrete time domain.

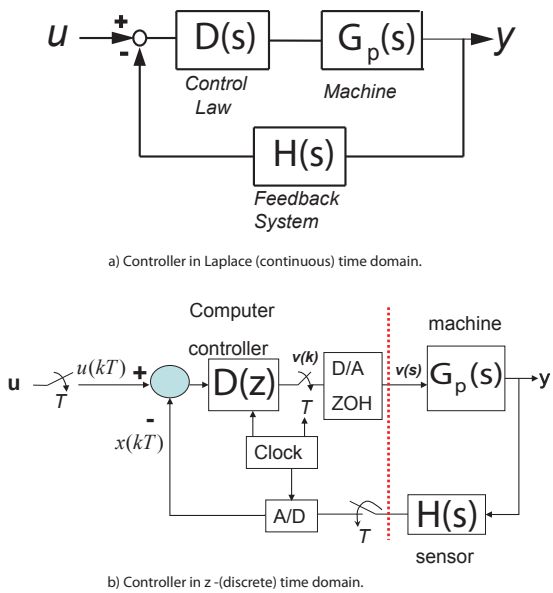


Figure 6.9: Close loop control systems with continuous (Laplace) and discrete time (z) domain controllers.

6.4.1 Controller Design in Laplace (continuous time) Domain :

The block diagram of a continuous controller design is given in Fig. 6.9 .a where $D(s)$ is designed in Laplace domain. The closed loop transfer function of the entire system is:

$$G_{cl}(s) = \frac{y(s)}{u(s)} = \frac{D(s)G_p(s)}{1 + D(s)G_p(s)H(s)} \tag{6.11}$$

The **loop transfer function** of the system is: $D(s)G_p(s)H(s)$
The stability of the system is determined by the poles of the characteristic equation :

$$\begin{gathered} 1 + D(s)G_p(s)H(s) = (s + p_1)(s + p_2) \cdot \cdot (s + p_n) \\ p = \sigma + j\omega_n \end{gathered} \tag{6.12}$$

Stability analysis on s plane: If any of the poles of the characteristic equation has a positive real part (i.e. $\sigma > 0$) the system will be unstable regardless of other poles and any input ($u(s)$) supplied to the closed loop system.

Stability analysis using open loop transfer function in frequency domain: The loop transfer function of the system, which is also often called open loop transfer function of the entire closed loop system,

is $D(s)G_p(s)H(s)$. It is possible to check the stability of the system from the frequency response function (**FRF**) of the loop transfer function.

$$\begin{aligned}\text{FRF of loop transfer function } \; G_{ol}(j\omega) &= D(s)G_p(s)H(s)|_{j\omega} \\ &= \mathrm{Re}[G_{ol}(j\omega)] + j\,\mathrm{Im}[G_{ol}(j\omega)]\end{aligned}$$

Nyquist Diagram: FRF of the open loop system can be plotted on complex frequency domain plane (polar or Nyquist) by entering the real part ($\mathrm{Re}[G_{ol}(j\omega)]$) in abscissa, and imaginary part ($\mathrm{Im}[G_{ol}(j\omega)]$) in normal axes. If the plot encircles -1 (negative real axis), the system is unstable and it indicates that the characteristic equation (6.12) has at least one or more unstable pole (i.e. $\sigma > 0$). If the open loop FRF has a real axis crossing which is greater than -1, (i.e.-0.6) the system is stable with a gain margin of $GM = \frac{-1}{-0.6} = 1.67$. That means, the gain of the controller ($D(s)$) can be increased by $K = 1.67$ times before the system can reach to -1 where it becomes critically stable by having poles (Eq. (6.12)) with zero real part ($\sigma = 0$).

Bode Diagram: FRF of the open loop system can be evaluated by plotting the magnitude ($|G_{ol}(_{j\omega})| = \sqrt{(\mathrm{Re}[G_{ol}(j\omega)])^2 + (\mathrm{Im}[G_{ol}(j\omega)])^2}$) and phase $\measuredangle\, G_{ol}(_{j\omega}) = \tan^{-1}\frac{\mathrm{Im}[G_{ol}(j\omega)]}{\mathrm{Re}[G_{ol}(j\omega)]}$ as a function of frequency (ω). If the gain margin and phase margins are positive, the system is stable; otherwise the closed loop system will be unstable and the characteristic equation (Eq. (6.12)) has positive real pole (i.e. $\sigma > 0$).

Nyquist and Bode diagrams allow the stability analysis the system has in frequency domain without having to find the roots of the characteristic equation. They also indicate how far (the margin) we have before the system becomes critically stable. One can design $D(s)$ and check the margin, and increase or decrease its gain to make it stable with acceptable gain - phase margin which indicates the distance from the critical stability limit.

Note: Controller ($D(s)$) is designed in Laplace domain when it has to be implemented using analog circuits, i.e. velocity controller and amplifier design using analog tachogenarator. If $D(s)$ is designed in Laplace domain for convenience, but implemented in the computer (z), $D(s)$ must be converted into z domain by using one of the approximations ($s = f(z)$ using $Z[D(s)]$ or one of Tustin, Euler or Forward approximations as shown in Fig. 3.7). The Zero Order Hold must not be considered in this approximation except the gain (K_d) of D/A controller which will affect the total gain of the system.

6.4.2 Controller Design in z (discrete time) Domain

If the machine is controlled by the computer (Fig.6.9.b) , it is more proper to design the controller ($D(z)$) in discrete time (z) domain as follows. First the discrete time equivalent of the physical machine must be evaluated. If feedback is unity $H = 1$ or just a gain H, the physical part is only $G_p(s)$ whose zero order hold equivalent transfer function is :

$$G_p(z) = ZOH[G_p(s)] = (1 - z^{-1})Z\left[\frac{G_p(s)}{s}\right] \tag{6.13}$$

If feedback has dynamics in Laplace domain (i.e. $H(s)$), it has to be considered as explained in Eq. 3.2.

The closed loop transfer function of the system with a constant feedback gain (H) becomes :

$$G_{cl}(z) = \frac{y(k)}{u(k)} = \frac{D(z)G_p(z)}{1 + HD(z)G_p(z)} \tag{6.14}$$

The **loop transfer function** of the system is: $HD(z)G_p(z)$
The stability of the system is determined by the poles of the characteristic equation :

$$1 + HD(z)G_p(z) = (z + p_1)(z + p_2) \cdot \cdot (z + p_n) \tag{6.15}$$
$$p = \alpha + j\beta$$

Stability analysis on z plane: If any of the poles of the characteristic equation has a magnitude greater than unity (i.e. $|p| > 1$) the system will be unstable regardless of other poles and any input ($u(k)$) supplied to the closed loop system. If $|p| < 1$ the closed loop system is stable, and if $|p| = 1$ the system will be critically stable.

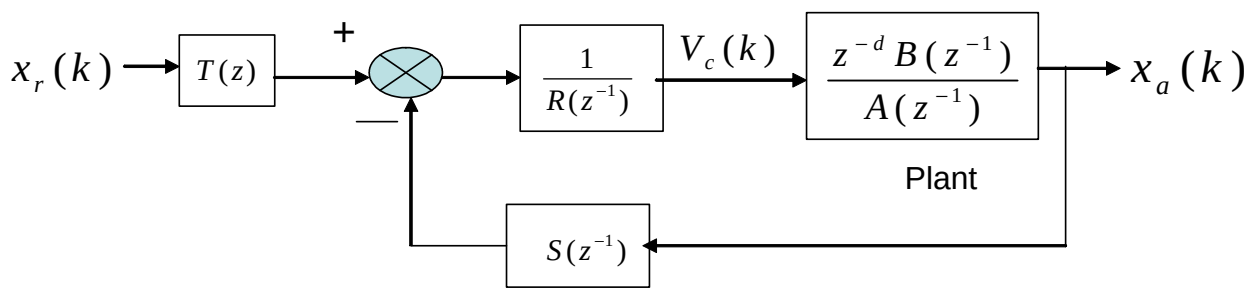
Stability analysis using open loop transfer function in frequency domain: It is possible to apply the same Nyquist and Bode Plot analysis on the loop transfer function of the system represented in z domain by replacing $z = e^{sT}\big|_{s=j\omega} = e^{j\omega T} = \cos(\omega T) + j\sin(\omega T)$.

$$\begin{aligned} \text{FRF of loop transfer function} \quad G_{ol}(j\omega) &= HD(z)G_p(z)|_{z=e^{j\omega T}} \\ &= \mathrm{Re}[G_{ol}(j\omega)] + j\,\mathrm{Im}[G_{ol}(j\omega)] \end{aligned}$$

Note that the sampling period (T) affects the FRF in discrete time domain. The same procedure as in continuous time domain is applied on $G_{ol}(j\omega)$. If the system has a negative gain - phase margin, the closed loop poles of the characteristic equation on z plane are outside the unit circle. MATLAB command for the frequency response function analysis in z domain is freqz.

6.5 Pole Placement Control System Design

Pole placement control strategy is designed to assign desired poles to the closed loop transfer function of the system. A sample block diagram of a position control system is given in Figure 6.5.



Block Diagram of a Pole Placement Control system for a plant $G_p(z^{-1}) = \frac{z^{-d}B(z^{-1})}{A(z^{-1})}$

The plant's transfer function ($G_p(z^{-1}) = \frac{z^{-d}B(z^{-1})}{A(z^{-1})}$) has a delay of d sampling intervals with the open loop polynomials with degrees of n and m as follows:

$$\begin{aligned} B(z^{-1}) &= z^{-d}\left(b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdot\cdot + b_n z^{-n}\right) \quad \deg(B) = n \\ A(z^{-1}) &= 1 + a_1 z^{-1} + a_2 z^{-2} + \cdot\cdot + a_m z^{-m} \qquad \deg(A) = m \end{aligned}$$

The polynomials $S(z^{-1})$, $T(z^{-1})$ and $R(z^{-1})$ represent feedback, feed-forward and error regulators respectively, which represent the control law.

$$\begin{aligned} R(z^{-1}) &= 1 + r_1 z^{-1} + r_2 z^{-2} + \cdot\cdot + r_p z^{-p} \quad \deg(R) = p \\ S(z^{-1}) &= s_0 + s_1 z^{-1} + s_2 z^{-2} + \cdot\cdot + s_f z^{-f} \quad \deg(S) = f \\ T(z^{-1}) &= t_0 \end{aligned}$$

Once $R(z^{-1}), S(z^{-1})$ and $T(z^{-1}) = t_0$ are designed, the control law must be executed to send $V_c(k)$ commands to the machine's amplifier - drive as:

$$R(z^{-1})\ V_c(k) = t_0 x_r(k) - S(z^{-1})\ x_a(k) \tag{6.16}$$

$$V_c(k) = -\left[r_1 V_c(k-1) + r_2 V_c(k-2) + .. + r_p V_c(k-p)\right] \tag{6.17}$$

$$+\ t_0 x_r(k) - \left[s_0 x_a(k) + s_1 x_a(k-1) + .. + s_f x_a(k-f)\right] \tag{6.18}$$

The control polynomials $R(z^{-1}), S(z^{-1})$ and $T(z^{-1}) = t_0$ are designed as follows.

The closed loop transfer function of the control system is given by,

$$G_{cl}(z) = \frac{x_a(k)}{x_r(k)} = \frac{T(z^{-1})z^{-d}B(z^{-1})}{A(z^{-1})R(z^{-1}) + z^{-d}B(z^{-1})S(z^{-1})} \tag{6.19}$$

If the plant has a stable zero inside the unit circle, B can be partitioned into two parts: $B = B^+B^-$ where B^+ has stable and B^- has unstable zeros. Stable zeros (B^+) can be canceled by selecting control polynomial $R = R^{'}B^+$, which leads to the following transfer function:

$$G_{cl}(z) = \frac{x_a(k)}{x_r(k)} = \frac{T(z^{-1})z^{-d}B^+(z^{-1})B^-(z^{-1})}{A(z^{-1})B^+(z^{-1})R'(z^{-1}) + z^{-d}B^+(z^{-1})B^-(z^{-1})S(z^{-1})} \tag{6.20}$$

$$= \frac{T(z^{-1})z^{-d}B^-(z^{-1})}{A(z^{-1})R'(z^{-1}) + z^{-d}B^-(z^{-1})S(z^{-1})} \tag{6.21}$$

where

$$R(z^{-1}) = R^{'}(z^{-1})B^+(z^{-1})$$

However, if the zeros are unstable or very close to the unit circle, they must be retained and model zero must contain the unstable plant zero $B_m(z^{-1}) = B^-(z^{-1})b_m$. Otherwise, if the unstable zero is assigned as a closed loop pole, the controller will be unstable. The control algorithm regulates the velocity command $V_c(k)$ as,

$$R\ V_c(k) = T x_r(k) - S\ x_a(k) \tag{6.22}$$

The aim of the pole placement design is that the closed loop transfer function between the reference position command x_r and the actual position x_a obey the following desired model dynamics

$$G_m(z) = \frac{x_a(k)}{x_r(k)} = \frac{z^{-d}B_m(z^{-1})}{A_m(z^{-1})} \tag{6.23}$$

$A_m(z^{-1})$ is the desired characteristic equation of the closed loop system which is designed to satisfy transient response crequirements of the controller. For example, an overshoot of $M_p = 1\%$ and peak time of t_p may be the performance requirements of a step input response. The corresponding damping ratio ζ_m and natural frequency ω_m can be found as follows:

$$\begin{aligned} M_p &= e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \rightarrow \zeta = \frac{-\ln M_p}{\sqrt{\pi^2 + (\ln M_p)^2}} \\ \omega_d t_p &= \pi \ , \ \ \omega_d = \omega_m\sqrt{1-\zeta^2} \rightarrow \omega_m = \frac{\pi}{t_p\sqrt{1-\zeta^2}} \end{aligned}$$

A second order dynamic model is selected to represent the response with a desired damping ratio of ζ_m and natural frequency of ω_m,

$$A_m(s) = s^2 + 2\zeta_m\omega_m s + \omega_m^2 \ = (s - s_{m1})(s - s_{m2}) \tag{6.24}$$

$$s_{m1} = -\zeta_m\omega_m + j\omega_m\sqrt{1-\zeta_m^2}, \quad s_{m2} = -\zeta_m\omega_m - j\omega_m\sqrt{1-\zeta_m^2}$$

The desired poles can be mapped to the z domain by,

$$z_{m1} = e^{s_{m1}T} = e^{-\zeta_m \omega_m T} \cdot e^{j\omega_m \sqrt{1-\zeta_m^2}T},$$
$$z_{m2} = e^{s_{m2}T} = e^{-\zeta_m \omega_m T} \cdot e^{-j\omega_m \sqrt{1-\zeta_m^2}T}$$

$$\begin{aligned} A_m(z) &= (z - z_{m1})(z - z_{m2}) = z^2 - z(z_{m1} + z_{m2}) + z_{m1}z_{m2} \\ &= z^2 - ze^{-\zeta_m \omega_m T}(e^{j\omega_m \sqrt{1-\zeta_m^2}T} + e^{-j\omega_m \sqrt{1-\zeta_m^2}T}) + e^{-2\zeta_m \omega_m T} \\ &= z^2 - 2e^{-\zeta_m \omega_m T} cos(\omega_m \sqrt{1-\zeta_m^2}T)z + e^{-2\zeta_m \omega_m T} \\ &\text{or} \\ A_m(z^{-1}) &= 1 - 2e^{-\zeta_m \omega_m T} cos(\omega_m \sqrt{1-\zeta_m^2}T)z^{-1} + e^{-2\zeta_m \omega_m T} z^{-2} = 1 + m_1 z^{-1} + m_2 z^{-2} \end{aligned}$$

For a selected damping ratio of $\zeta_m \geq 0.707$, and the rise time can be estimated from the acceleration as follows. The maximum acceleration can be determined from the peak motor torque without saturating its limit. If the peak torque is T_d for accelerating the inertia J_e, the maximum acceleration is :

$$\ddot{\theta}_{\max} = \frac{T_d}{J_e}$$

If the maximum velocity of the drive is $\dot{\theta}_{\max}$, the rise time is $t_r = \frac{\dot{\theta}_{\max}}{\ddot{\theta}_{\max}}$. The corresponding natural frequency ω_m[rad/sec].

$$\omega_m \simeq \frac{1 - 0.4167\zeta + 2.917\zeta^2}{t_r} \rightarrow \zeta < 1$$

Alternatively, the peak time can be used to identify the natural frequency ($\omega_m = \pi/t_p\sqrt{1-\zeta^2}$) With this criteria, the discrete closed loop model transfer function with a unit gain is

$$\frac{x_a(k)}{x_r(k)} = \frac{z^{-d}B_m(z^{-1})}{A_m(z^{-1})} = \frac{z^{-d}B_m(z^{-1})}{1 + m_1 z^{-1} + m_2 z^{-2}} \tag{6.25}$$

where $m_1 = -2\ e^{-\zeta_m\ \omega_m T}\ cos(\omega_m \sqrt{1-\zeta_m^2}T)$ and $m_2 = +e^{-2\zeta_m \omega_m T}$. In general, the order of the desired characteristic equation (A_m) must be equal to the order of plant's poles ($A(z^{-1})$, i.e. $\deg(A_m) = \deg(A)$. In order for the system to behave like the desired model, the following equivalence must hold.

$$\frac{x_a(k)}{x_r(k)} = \frac{z^{-d}B\ T}{AR + z^{-d}BS} \equiv \frac{z^{-d}B_m(z^{-1})}{A_m(z^{-1})} \tag{6.26}$$

The characteristic equation ($AR + z^{-d}B$) of the closed loop system must be equal to the desired characteristic equation $A_m(z^{-1})$:

$$\begin{aligned} AR + z^{-d}BS &\equiv A_m(z^{-1}) \\ &\left(1 + a_1 z^{-1} + \cdot\cdot + a_m z^{-m}\right)\left(1 + r_1 z^{-1} + \cdot\cdot + r_p z^{-p}\right) \\ &+ z^{-d}\left(b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdot\cdot + b_n z^{-n}\right)\left(s_0 + s_1 z^{-1} + \cdot\cdot + s_f z^{-f}\right) \\ &= (1 + m_1 z^{-1} + m_2 z^{-2}) \end{aligned}$$

which is called Diophentine Equation. The highest powers of backward operator (z^{-1}) are $\deg(A) + \deg R = m + p$ and $d + \deg B + \deg S = d + n + f$ .The number of unknowns in control polynomials $R(z^{-1})$ and $S(z^{-1})$ is $p + (f + 1)$. The degree of control polynomials are found as:

- Let the degrees of polynomials in AR and $z^{-d}BS$ are designed to be equal $\deg(A) + \deg R = d + \deg B + \deg S$ or $m + p = d + n + f$
- Degree of polynomial $\deg(R) = p$: The number of unknowns $(\deg(R) + \deg(S) + 1 = p + (f + 1))$ are equal to the number of simultaneous difference equations contributed by AR and $z^{-d}BS$ polynomials. $p + (f + 1) = d + n + f$ and $p + (f + 1) = m + p$
- $p+(f+1) = d+n+f \rightarrow \deg(R) = p = d+n-1 = d+\deg(B)-1 \rightarrow R(z^{-1}) = 1+r_1z^{-1}+r_2z^{-2}+\cdots+r_pz^{-p}$
- $p+(f+1) = m+p \rightarrow \deg(S) = f = m-1 = \deg(A)-1$ $\quad S(z^{-1}) = s_0+s_1z^{-1}+s_2z^{-2}+\cdots+s_fz^{-f}$

Let $T(z^{-1}) = t_0$ as a gain to make steady state ($s = 0$ in Laplace domain and $z = 1$ in z domain) output of the closed loop system to be equal to input. The numerators must have a balanced power as well,

$$z^{-d}B(z^{-1})\,t_0 = z^{-d}B_m(z^{-1}) \rightarrow B(z^{-1})t_0 = B_m(z^{-1}) \;\; ; \;\; AR + z^{-d}BS = A_m(z^{-1})$$

$$\left[\frac{z^{-d}B(z^{-1})t_0}{AR + z^{-d}BS}\right]_{z=1} = \left[\frac{z^{-d}B_m(z^{-1})}{A_m(z^{-1})}\right]_{z=1} = 1 \text{ and } B_m(z^{-1}) = B(z^{-1})t_0 \rightarrow t_0 = \left[\frac{AR + z^{-d}BS}{z^{-d}B(z^{-1})}\right]_{z=1} = \left[\frac{A_m(z^{-1})}{z^{-d}B(z^{-1})}\right]_{z=1}$$

The desired and closed loop control systems must have equal characteristic equations:

$$AR + z^{-d}BS \equiv A_m(z^{-1})$$

$$\left(1 + a_1z^{-1} + \cdots + a_mz^{-m}\right)\left(1 + r_1z^{-1} + \cdots + r_pz^{-p}\right) + z^{-d}\left(b_0 + b_1z^{-1} + b_2z^{-2} + \cdots + b_nz^{-n}\right)\left(s_0 + s_1z^{-1} + \cdots + s_fz^{-f}\right)$$
$$= 1 + m_1z^{-1} + m_2z^{-2}$$

After simplification,

$$\left(1 + a_1z^{-1} + \cdots + a_mz^{-m}\right)\left(r_1z^{-1} + \cdots + r_pz^{-p}\right) + z^{-d}\left(b_0 + b_1z^{-1} + b_2z^{-2} + \cdots + b_nz^{-n}\right)\left(s_0 + s_1z^{-1} + \cdots + s_fz^{-f}\right) = (m_1 - a$$
$$\left(1 + \sum_{i=1}^{m} a_iz^{-i}\right)\sum_{j=1}^{p} r_jz^{-j} + z^{-d}\sum_{i=0}^{n} b_iz^{-i}\sum_{j=0}^{f} s_jz^{-j} = (m_1 - a$$

After collecting the unknown parameters on the left, and the terms which do not contain any unknown parameter on the right, $p + (f + 1)$ number of simultaneous equations can be expressed in a matrix form as :

$$[\Phi]\{\theta\} = \{C\} \rightarrow \{\theta\} = [\Phi]^{-1}\{C\}$$

where the parameter vector is $\{\theta\} = \left\{ \begin{array}{cccccccc} r_1 & r_2 & \cdot & \cdot & r_p & s_0 & s_1 & \cdot & \cdot & s_f \end{array} \right\}^T$.Note that the matrix needs to be written by opening the characteristic equation for each case.

Example 83 *Let the open loop transfer function of an electrohydraulic actuator is defined as*

$$G_p(z^{-1}) = \frac{z^{-2}(b_0 + b_1z^{-1})}{1 + a_1z^{-1}}\frac{1}{1 - z^{-1}} = \frac{z^{-2}(b_0 + b_1z^{-1})}{1 + (a_1 - 1)z^{-1} - a_1z^{-2}} = \frac{z^{-2}(0.3676 + 0.3387z^{-1})}{1 - 1.9919z^{-1} + 0.9919z^{-2}}$$
$$G_p(z) = \frac{(0.3676z + 0.3387)}{z(z^2 - 1.9919z + 0.9919)} = \frac{0.3676(z + 0.92138)}{z(z^2 - 1.9919z + 0.9919)} = \frac{0.3676z^{-2}(1 + 0.92138z^{-1})}{1 - 1.9919z^{-1} + 0.9919z^{-2}}$$

Let's not cancel the stable zero of the plant ($(z + 0.92138) = 0 \rightarrow z_1 = -0.92138$). The desired closed loop controller is expected to have a second order characteristic equation with a natural frequency of $\omega_n = 50[rad/s]$ and damping ratio of $\zeta = 0.8$. The control interval is $T_s = 2[ms]$.

$$
\begin{aligned}
z_{m1} &= e^{s_{m1}T} = e^{-0.8\cdot 50\cdot 0.002}\cdot e^{j50\sqrt{1-0.8^2}0.002} = 0.923\,12e^{0.06j}\\
z_{m2} &= e^{s_{m2}T} = e^{-0.8\cdot 50\cdot 0.002}\cdot e^{-j50\sqrt{1-0.8^2}0.002} = 0.923\,12e^{-0.06j}\\
m_1 &= z_{m1}+z_{m2} = -2e^{-\zeta_m\omega_m T}cos(\omega_m\sqrt{1-\zeta_m^2}T) = -2e^{-0.8\cdot 50\cdot 0.002}\cdot\cos(50\sqrt{1-0.8^2}0.002) = -1.8429\\
m_2 &= e^{-2\zeta_m\omega_m T} = e^{-2\cdot 0.8\cdot 50\cdot 0.002} = 0.85214\\
A_m(z^{-1}) &= 1-1.8429z^{-1}+0.85214z^{-2}
\end{aligned}
$$

$$
\begin{aligned}
\deg(R) &= p = d+\deg(B)-1 = 2+1-1 = 2 \rightarrow R(z^{-1}) = 1+r_1z^{-1}+r_2z^{-2}\\
\deg(S) &= \deg(A)-1 = 2-1 = 1 \rightarrow S(z^{-1}) = s_0+s_1z^{-1}\\
t_0 &= \left[\frac{A_m(z^{-1})}{z^{-d}B(z^{-1})}\right]_{z=1} = \frac{1+m_1+m_2}{b_0+b_1}
\end{aligned}
$$

By writing the characteristic equation

$$
\begin{aligned}
&AR+z^{-d}BS \equiv A_m(z^{-1})\\
&\quad\left(1+a_1z^{-1}+a_2z^{-2}\right)\left(1+r_1z^{-1}+r_2z^{-2}\right)+z^{-2}\left(b_0+b_1z^{-1}\right)\left(s_0+s_1z^{-1}\right) = 1+m_1z^{-1}+m_2z^{-2}\\
&\quad 1+\left(a_1+r_1\right)z^{-1}+\left(a_2+r_2+a_1r_1+b_0s_0\right)z^{-2}+\left(a_1r_2+a_2r_1+b_0s_1+b_1s_0\right)z^{-3}+a_2r_2+b_1s_1\\
&\quad = 1+m_1z^{-1}+m_2z^{-2}
\end{aligned}
$$

$$
\begin{aligned}
&r_1z^{-1}+\left(r_2+a_1r_1+b_0s_0\right)z^{-2}+\left(a_1r_2+a_2r_1+b_0s_1+b_1s_0\right)z^{-3}+a_2r_2+b_1s_1\\
&= (m_1-a_1)z^{-1}+(m_2-a_2)z^{-2}
\end{aligned}
$$

By equating the powers of each z^- operator,

$$
\begin{aligned}
r_1 &= m_1-a_1\\
r_2+a_1r_1+b_0s_0 &= m_2-a_2\\
a_1r_2+a_2r_1+b_0s_1+b_1s_0 &= 0\\
a_2r_2+b_1s_1 &= 0
\end{aligned}
$$

Or in a matrix form:

$$
\begin{bmatrix} 1 & 0 & 0 & 0\\ a_1 & 1 & b_0 & 0\\ a_2 & a_1 & b_1 & b_0\\ 0 & a_2 & 0 & b_1 \end{bmatrix}
\begin{Bmatrix} r_1\\ r_2\\ s_0\\ s_1 \end{Bmatrix}
=
\begin{Bmatrix} m_1-a_1\\ m_2-a_2\\ 0\\ 0 \end{Bmatrix}
$$

The unknown parameter vector can be expressed as:

$$\begin{Bmatrix} r_1 \\ r_2 \\ s_0 \\ s_1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ a_1 & 1 & b_0 & 0 \\ a_2 & a_1 & b_1 & b_0 \\ 0 & a_2 & 0 & b_1 \end{bmatrix}^{-1} \begin{Bmatrix} m_1 - a_1 \\ m_2 - a_2 \\ 0 \\ 0 \end{Bmatrix}$$

$$\begin{Bmatrix} r_1 \\ r_2 \\ s_0 \\ s_1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1.9919 & 1 & 0.3676 & 0 \\ 0.9919 & -1.9919 & 0.3387 & 0.3676 \\ 0 & 0.9919 & 0 & 0.3387 \end{bmatrix}^{-1} \begin{Bmatrix} -1.8429 + 1.9919 \\ 0.85214 - 0.9919 \\ 0 \\ 0 \end{Bmatrix}$$

$$= \begin{bmatrix} 1.0 & 0 & 0 & 0 \\ 0.708\,6 & 0.230\,93 & -0.250\,64 & 0.272\,02 \\ 3.\,491 & 2.\,092\,1 & 0.681\,82 & -0.74 \\ -2.\,075\,2 & -0.676\,3 & 0.734\,01 & 2.\,155\,8 \end{bmatrix} \times \begin{bmatrix} 0.149 \\ -0.139\,76 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{Bmatrix} r_1 \\ r_2 \\ s_0 \\ s_1 \end{Bmatrix} = \begin{bmatrix} 0.1490 \\ 0.0733 \\ 0.2277 \\ -0.2147 \end{bmatrix},$$

$$t_0 = \left[\frac{1 + m_1 + m_2}{b_1 + b_2}\right] = \left[\frac{1 - 1.8429 + 0.85214}{0.3676 + 0.3387}\right] = 0.013082$$

The controller can be simulated by the following equivalent transfer functions used in a Simulink model:

$$\frac{B(z)}{A(z)} = \frac{z^{-2}(0.3676 + 0.3387z^{-1})}{1 - 1.9919z^{-1} - 0.9919z^{-2}} = \frac{(0.3676z + 0.3387)}{z^3 - 1.9919z^2 - 0.9919z}$$

$$\frac{T(z)}{R(z)} = \frac{t_0}{1 + r_1z^{-1} + r_2z^{-2}} = \frac{t_0z^2}{z^2 + r_1z + r_2z} = \frac{0.01308\,2z^2}{z^2 + 0.1490z + 0.0733}$$

$$\frac{S(z)}{R(z)} = \frac{s_0 + s_1z^{-1}}{1 + r_1z^{-1} + r_2z^{-2}} = \frac{s_0z^2 + s_1z}{z^2 + r_1z + r_2z} = \frac{0.2277z^2 - 0.2147z}{z^2 + 0.1490z + 0.0733}$$

Example 84 *Control law $u(k)$ can be expressed as,*

$$R(z^{-1})\ u(k) = T(z^{-1})y_r(k) - S(z^{-1})\ y(k)$$
$$u(k) = t_0y_r(k) - r_1u(k-1) - r_2(u(k-2) - s_0y(k) - s_1y(k-1)$$

Sample MATLAB code to solve and simulate the controller is given below

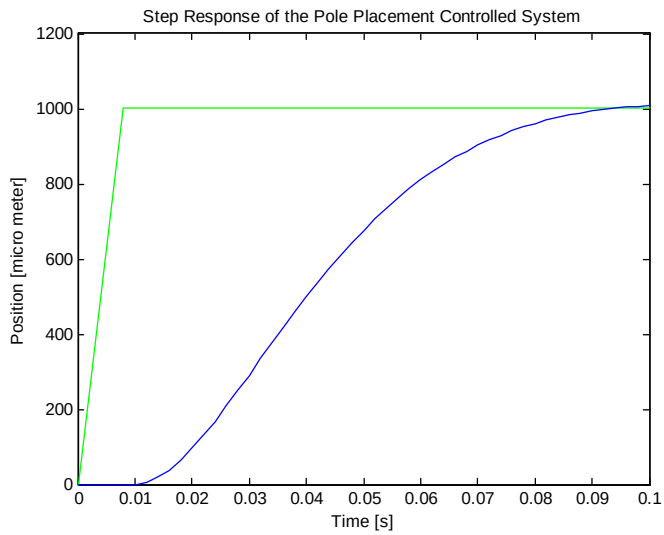
```
%MECH 467 Hydraulic Actuator Control Project
% Author : Prof. Yusuf Altintas
% November 25, 2007 UBC
% Input yr, output: y, Controller output command u(k)
close all;
clear;
%
% Identified dynamics of the hydraulic actuator
Gov=tf([0.3676  0.3387],[1 -0.9919 0 0],Ts); % Input is voltage, output is the velocity
Gi=tf([1],[1 -1],Ts); %Integrator to obtain position
Gp=Gov*Gi; % Open loop tf of position system
%Plant tf parameters
a1=Gp.den{[1]}(2);
a2=Gp.den{[1]}(3);
b1=Gp.num{[1]}(4);
b2=Gp.num{[1]}(5);
```

```
% Desired Model Transfer Function
zeta_m=0.8; wn_m=50 ; % Desired damping ratio and natural frequency
wd=wn_m*sqrt(1-zeta_m^2);
s1m=-zeta_m*wn_m+j*wn_m*sqrt(1-zeta_m^2); % Desired poles in s domain
s2m=-zeta_m*wn_m-j*wn_m*sqrt(1-zeta_m^2);
Ts=0.002 ; %[s] Sampling interval
z1_m=exp(s1m*Ts); z2_m=exp(s2m*Ts); % Desired poles are mapped on the z domain
Am=[1 -(z1_m+z2_m) z1_m*z2_m]; % Desired characteristic eqn. of the closed loop controller
m1=Am(2);m2=Am(3);
% Control polynomials
bm0=(1+m1+m2)/(b1+b2);
t0=bm0;
% Express the indetification matrix
FI=([1 0 0 0;a1 1 b1 0;a2 a1 b2 b1;0 a2 0 b2]); %Plant param matrix
C=([m1-a1 m2-a2 0 0]); % model matrix
Teta=FI^(-1)*C'; % Obtain the parameters from
% Idnetified parameters are
r1=Teta(1);r2=Teta(2);
s0=Teta(3); s1=Teta(4);
%Initialize the states for recursive calculation
u(1)=0;y(1)=0;yr(1)=0;yr(2)=0;yr(3)=0;y(2)=0;y(3)=0;u(2)=0;u(3)=0;
for k=4:50,
t(k)=k*Ts;
yr(k)=1000;
y(k)=-a1*y(k-1)-a2*y(k-2)+b1*u(k-2)+b2*u(k-3);
u(k)=-r1*u(k-1)-r2*u(k-2)+t0*yr(k)-s0*y(k)-s1*y(k-1);
end
% Plot the results
figure;
plot(t,yr,'g',t,y,'b');
title('Step Response of the Pole Placement Controlled System');
xlabel('Time [s]'); ylabel('Position [micro meter]');
figure;
plot(t,u);
title('Input command for the step input input response');
xlabel('Time [s]'); ylabel('Command input [V]');
% You can simulate the system also using the attached Simulink as well
% See attached simulink file
% You either use the ZOH and continuous system together
% or if yiu already have the discrete model (as you identified in the hydraulic
%lab using least squares. That already included DAQ and ZOH during measurements
% Click on the scope and right click the mouse and select Auto scale to
% see the signals.
pole_placement_hydraulic.mdl ;% The same simulation can be done by SIMULINK
```

Example 85 *The same machine with transfer function :*

$$G_p(z) = \frac{z^{-2}(b_1 + b_2 z^{-1})}{1 + (a_1 - 1)z^{-1} - a_1 z^{-2}} = \frac{0.3676 z^{-2}(1 + 0.92138 z^{-1})}{1 - 1.9919 z^{-1} + 0.9919 z^{-2}} = \frac{0.3676(z + 0.92138)}{z(z^2 - 1.9919 z + 0.9919)}$$

is considered but with a pole zero cancellation. Since the zero is $z_1 = -b_2/b_1 = -0.92138$ is stable (within unit circle), we can cancel it by selecting control polynomial R as a product of the stable zero. .



$$G_p(z^{-1}) = \frac{z^{-1}(b_1 z^{-1} + b_2 z^{-2})}{1 + (a_1 - 1)z^{-1} - a_1 z^{-2}} == \frac{0.3676 z^{-2}(1 + 0.921\,38 z^{-1})}{1 - 1.9919 z^{-1} + 0.9919 z^{-2}} = \frac{B^-(z^{-1})B^+(z^{-1})}{A(z^{-1})}$$
$$d = 2; B^-(z^{-1}) = 0.3676, \quad B^+(z^{-1}) = 1 + 0.921\,38 z^{-1}$$
$$A(z) = 1 - 1.9919 z^{-1} + 0.9919 z^{-2}$$

Example 86 *Let's select* $R(z^{-1}) = R'(z^{-1})B^+(z^{-1}) = R'(z^{-1})(1 + 0.921\,38 z^{-1})$. *The closed loop transfer function becomes:*

$$\frac{x_a(k)}{x_r(k)} = \frac{t_0 z^{-d} B^+(z) B^-(z^{-1})}{A(z^{-1})B^+(z^{-1})R'(z^{-1}) + z^{-d}B^+(z)B^-(z^{-1})S(z^{-1})}$$
$$= \frac{t_0 z^{-2} b_1}{A(z^{-1})R'(z^{-1}) + z^{-2} b_1 S(z^{-1})}$$

The number of unknowns in control polynomials $R'(z^{-1})$ *and* $S(z^{-1})$ *is* $p' + (f+1)$. *The order balances become*

$$\deg A + \deg R' = d + \deg S + \deg B^- \equiv p' + f + 1 \rightarrow m + p' = d + f + 0 = p' + f + 1$$

$$d + f + 0 = p' + f + 1 \rightarrow p' = \deg R' = d - 1 = 2 - 1 = 1 \rightarrow R'(z^{-1}) = 1 + r_1' z^{-1}$$
$$m + p' = p' + f + 1, \ \deg(S) = f = m - 1 = \deg(A) - 1 = 2 - 1 = 1 \rightarrow S(z^{-1}) = s_0 + s_1 z^{-1}$$
$$T(z^{-1}) = t_0 = b_m = \left[\frac{A_m(z^{-1})}{z^{-d}B^-(z^{-1})}\right]_{z=1} = \frac{1 + m_1 + m_2}{b_1}$$

$$\left(1 + a_1 z^{-1} + a_2 z^{-2}\right)\left(1 + r_1' z^{-1}\right) + z^{-1}\left(b_1 z^{-1}\right)\left(s_0 + s_1 z^{-1}\right)$$
$$= \left(1 + a_1 z^{-1} + a_2 z^{-2}\right) + \left(1 + a_1 z^{-1} + a_2 z^{-2}\right)\left(r_1' z^{-1}\right) + z^{-1}\left(b_1 z^{-1}\right)\left(s_0 + s_1 z^{-1}\right)$$
$$= 1 + m_1 z^{-1} + m_2 z^{-2}$$

$$\left(1 + a_1 z^{-1} + a_2 z^{-2}\right)\left(r_1' z^{-1}\right) + z^{-1}(b_1 z^{-1})\left(s_0 + s_1 z^{-1}\right)$$
$$= (m_1 - a_1)z^{-1} + (m_2 - a_2)z^{-2}$$

Example 87

$$\begin{bmatrix} 1 & 0 & 0 \\ a_1 & b_1 & 0 \\ a_2 & 0 & b_1 \end{bmatrix} \begin{Bmatrix} r_1' \\ s_0 \\ s_1 \end{Bmatrix} = \begin{Bmatrix} m_1 - a_1 \\ m_2 - a_2 \\ 0 \end{Bmatrix}$$

$$\begin{Bmatrix} r_1' \\ s_0 \\ s_1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ a_1 & b_1 & 0 \\ a_2 & 0 & b_1 \end{bmatrix}^{-1} \begin{Bmatrix} m_1 - a_1 \\ m_2 - a_2 \\ 0 \end{Bmatrix}$$

$$\begin{Bmatrix} r_1' \\ s_0 \\ s_1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1.9919 & 0.3676 & 0 \\ 0.9919 & 0 & 0.3676 \end{bmatrix}^{-1} \begin{Bmatrix} -1.8429 + 1.9919 \\ 0.85214 - 0.9919 \\ 0 \end{Bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 5.4187 & 2.7203 & 0 \\ -2.6983 & 0 & 2.7203 \end{bmatrix} \times \begin{bmatrix} 0.149 \\ -0.13976 \\ 0 \end{bmatrix}$$

$$\begin{Bmatrix} r_1' \\ s_0 \\ s_1 \end{Bmatrix} = \begin{bmatrix} 0.149 \\ 0.42720 \\ -0.40205 \end{bmatrix}$$

$$R(z) = (1 + 0.149\, z^{-1})(1 + 0.92138 z^{-1}) = 1 + 1.0704 z^{-1} + 0.13729 z^{-2}$$

$$S(z) = 0.42720 - 0.40205 z^{-1}$$

$$t_0 = \frac{1 + m_1 + m_2}{b_1} = \frac{1 - 1.8429 + 0.85214}{0.3676} = 0.025136$$

The control system can be simulated in Simulink by constructing the following transfer functions:

$$\frac{B(z)}{A(z)} = \frac{z^{-1}(0.3676 z^{-1} + 0.3387 z^{-2})}{1 - 1.9919 z^{-1} - 0.9919 z^{-2}} = \frac{(0.3676 z + 0.3387)}{z^3 - 1.9919 z^2 - 0.9919 z}$$

$$\frac{T(z)}{R(z)} = \frac{t_0}{1 + r_1 z^{-1} + r_2 z^{-2}} = \frac{t_0 z^2}{z^2 + r_1 z + r_2 z} = \frac{0.025136 z^2}{z^2 + 0.1490 z + 0.13729}$$

$$\frac{S(z)}{R(z)} = \frac{s_0 + s_1 z^{-1}}{1 + r_1 z^{-1} + r_2 z^{-2}} = \frac{s_0 z^2 + s_1 z}{z^2 + r_1 z + r_2 z} = \frac{0.42720 z^2 - 0.40205 z}{z^2 + 0.1490 z + 0.13729}$$

Control law can be expressed as,

$$R(z^{-1})\, u(k) = T(z^{-1}) y_r(k) - S(z^{-1})\, y(k)$$

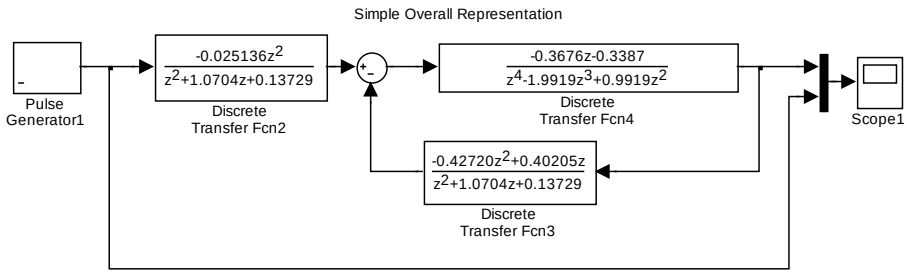
$$u(k) = t_0 y_r(k) - r_1 u(k-1) - r_2 (u(k-2) - s_0 y(k) - s_1 y(k-1)$$

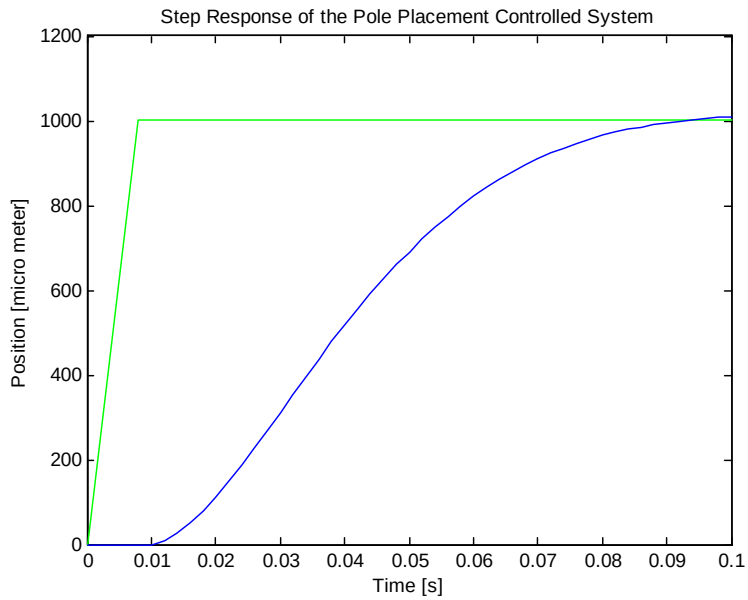
6.6 Sliding Mode Controller

Sliding mode controller is a typical example of robust, nonlinear control system. An application and design of a sliding mode controller to feed drive mechanism is given here as an example. The control law was developed at UBC Manufacturing Automation Laboratory, and applied to piezo actuators with hysteresis and feed drive control system.

Open loop block diagram of a feed drive system is sown in Figure 6.10.

The ball screw drive system application considered here is controlled in a current mode, and its open loop transfer function between the table position ($x[mm]$) and amplifier command generated in the CNC ($u[V]$) is :





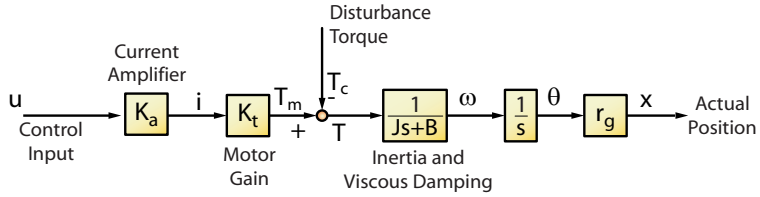


Figure 6.10: Open loop block diagram of a feed drive system powered under current mode of the amplifier.

$$
\begin{aligned}
x(s) &= [K_a K_t u(s) - T_c] \frac{r_g}{(Js+B)s} \\
&= \frac{K_a K_t r_g}{s(Js+B)}[u(s) - \frac{1}{K_a K_t} T_c(s)] \\
&= \frac{K}{s(s+p)}[u(s) - \frac{1}{K_a K_t} T_c(s)]
\end{aligned} \tag{6.27}
$$

where $K = K_a K_t r_g / J$ is the gain, $p = B/J$, is the velocity loop pole, and $T_c(s)$.is the torque disturbance caused by friction and cutting process In the linear motor drives, the inertia (J) is replaced by the table-workpiece mass, and the disturbance torque (T_c) is replaced by the cutting force and friction in the linear motor driven systems. The differential equation of the drive system can be reorganized as,

$$
\begin{aligned}
\frac{1}{K} s^2 x(s) + \frac{p}{K} s x(s) &= u(s) - \frac{1}{K_a K_t} T_c(s) \\
\underbrace{\frac{J}{K_a K_t r_g}}_{J_e} \ddot{x}(t) + \underbrace{\frac{B}{K_a K_t r_g}}_{B_e} \dot{x}(t) &= u(t) - \underbrace{\frac{1}{K_a K_t} T_c(t)}_{d(t)}
\end{aligned} \tag{6.28}
$$

or normalizing the disturbance torque with respect to input,

$$
J_e \ddot{x}(t) + B_e \dot{x}(t) = u(t) - d(t) \tag{6.29}
$$

where the reflected disturbance at the input is $d(t) = T_c(t)/(K_a K_t)$. The acceleration of the drive can be separated from Eq. 6.29 as,

$$
\ddot{x}(t) = \frac{1}{J_e}[-B_e \dot{x}(t) + u(t) - d(t)] \tag{6.30}
$$

The objective of the feed drive control system is that the controller must be capable of following the commanded trajectory and velocity with minimum error at high speeds, regardless of the slight variations in the inertia (J_e), viscous friction (B_e) and external disturbances (d) such as Coulomb friction and cutting forces reflected on the drive input. The conventional drive control systems, such as pole placement and feed-forward control techniques work quite well when the system is modeled accurately, and external disturbances are rejected by having a large bandwidth in linear drives. In addition, large transmission ratio in the ball screw drives reduce the reflected torque on the rotating drive motor shaft significantly. However, when the friction is not modeled and compensated accurately, and when the external forces such as in linear drives are not determined before, conventional control techniques do not exhibit high tracking accuracy in high speed machining, leading to inaccuracies in contour machining operations. The sliding mode controller belongs to a class of nonlinear control strategies, which are robust to such uncertainties and time variation in the drive dynamics.

There are two fundamental steps in designing a sliding mode controller: the selections of a sliding surface and a Lyapunov function. For accurate tracking of position and velocity, which are the key requirements from high speed machine tools, the sliding surface (S) is selected as,

$$
S = \lambda(x_r - x) + (\dot{x_r} - \dot{x}) \tag{6.31}
$$

where $\lambda[1/s]$ is the desired but achievable bandwidth of the drive, x_r, x are the reference command and actual positions, and $\dot{x_r}, \dot{x}$ are the reference command and actual velocities of the drive, respectively. The control input (u) must be manipulated in such a way that after a finite time, both the position and velocity of the drive approach to reference command values ($x \to x_r, \dot{x} \to \dot{x_r}$) hence forcing the value of position and velocity errors to be zero, i.e. $S \to 0$.

Inertia (J_e) and viscous damping (B_e) on the machine drive are assumed to change slowly and insignificantly. The variation in the external disturbance caused by the cutting process and friction is considered to

be strong, but with upper (d^+) and lower (d^-) limits measured on the machine. The external disturbance can be tracked by the following simple observer,

$$\dot{\widehat{d}} = \rho\kappa S, \quad \widehat{d}(k) = \widehat{d}(k-1) + \rho\kappa S \cdot T \tag{6.32}$$

where T(sec) is the control period, k is the control interval counter in discrete time domain, ρ is parameter adaptation gain ($\rho \approx 0.005$), and κ is used to limit the integral control of the disturbance as,

$$\kappa = \left\{ \begin{array}{ll} 0 & \text{if } \widehat{d}(k) \leq d^- \text{ and } S \leq 0 \\ 0 & \text{if } \widehat{d}(k) \geq d^+ \text{ and } S \geq 0 \\ 1 & \text{otherwise} \end{array} \right\} \tag{6.33}$$

Hence the estimated disturbances are always kept within the predetermined bounds, i.e $\widehat{d}(k) \in [d^-, d^+]$.

The second step in sliding mode controller design is to select a Lypanov Function which is used to obtain a stable control law for a nonlinear system, i.e. feed drive controlled by the nonlinear sliding mode control strategy. The following Lyapunov function

$$V(t) = \frac{1}{2}[J_e S^2 + \frac{(d-\widehat{d})^2}{\rho}] \tag{6.34}$$

which resembles the summation of kinetic energy and the square of disturbance prediction error. Like the sliding surface, the selection of specific Lyapunov function is based on the experience and intuition. For asymptotic stability of nonlinear systems, the derivative of Lyapunov function must be negative or zero, meaning that the rate of change in the energy and prediction error must decrease.

$$\dot{V}(t) = J_e S\dot{S} - \dot{\widehat{d}}\frac{(d-\widehat{d})}{\rho} \leq 0 \tag{6.35}$$

By substituting $\dot{S} = \lambda(\dot{x_r} - \dot{x}) + (\ddot{x_r} - \ddot{x})$ from Eq. 6.31, $\dot{\widehat{d}} = \rho\kappa S$ from Eq. 6.32 and $\ddot{x}$ from Eq. 6.30,

$$\dot{V}(t) = J_e S\left[\lambda(\dot{x_r} - \dot{x}) + \ddot{x_r}\right] + SB_e\dot{x} - Su + Sd - S\kappa(d-\widehat{d}) \leq 0 \tag{6.36}$$

Expressing $Sd - S\kappa(d-\widehat{d}) = S\widehat{d} + S(d-\widehat{d})(1-\kappa)$,and noting that $S(d-\widehat{d})(1-\kappa) < 0$ always due to imposed limit conditions ($\widehat{d}(k) \in [d^-, d^+], \kappa = 0$ or 1, as well as the value of the sliding surface S) in Eq. 6.33, never allowed to be outside the following criteria will ensure the asymptotic stability condition at any condition ($\dot{V}(t) \leq 0$),

$$J_e S\left[\lambda(\dot{x_r} - \dot{x}) + \ddot{x_r}\right] + SB_e\dot{x} - Su + S\widehat{d} = -K_s S^2 \tag{6.37}$$

where $K_s > 0$ is the control gain to be selected. The control law (u) is obtained from Eq. 6.37 as,

$$\begin{aligned} u(k) &= J_e\left[\lambda(\dot{x_r}(k) - \dot{x}(k)) + \ddot{x_r}(k)\right] + B_e\dot{x}(k) + \widehat{d}(k) + K_s S(k) \\ S(k) &= \lambda[x_r(k) - x(k)] + [\dot{x_r}(k) - \dot{x}(k)] \end{aligned} \tag{6.38}$$

where (k) is the control interval counter. The reference position, $x_r(k)$, velocity $\dot{x_r}(k)$, and acceleration $\ddot{x_r}(k)$ are obtained from the command generation algorithm running in the CNC system. The actual position $x(k)$ is measured from the encoder, and the actual velocity $\dot{x}(k)$ is estimated by taking the derivative of the measured position from the linear drive or measuring directly from a tachogenerator. However, evaluation of velocity and acceleration from discrete position commands and encoder readings may be noisy. The following simple low pass filter can be used to smoothen them,

‘

$$\dot{x_r}(k) = \alpha \dot{x_r}(k-1) + \frac{1-\alpha}{T}[x_r(k) - x_r(k-1)]$$
$$\ddot{x_r}(k) = \alpha \ddot{x_r}(k-1) + \frac{1-\alpha}{T}[\dot{x}(k) - \dot{x_r}(k-1)]$$
$$\dot{x}(k) = \alpha \dot{x}(k-1) + \frac{1-\alpha}{T}\dot{x}_m(k)$$

where $\dot{x}_m(k)$ is the measured velocity from the tachogenerator. The filter gain is typically set to $\alpha \simeq 0.6$.

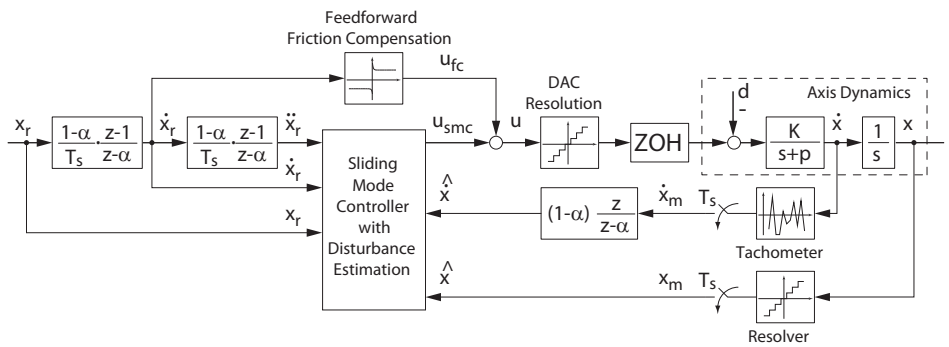


Figure 6.11: Sliding mode controller with feed forward friction compensation.

The disturbance compensation through on-line estimation is useful to minimize the effects of slowly varying cutting forces and friction. However, when the drive changes the direction of the velocity especially at the corners and quadrants of the circular paths, the friction force reverses its direction and leaves glitches on the surface. If major Coulomb friction force is known, it may be more advantageous to pre-compansate it at the feed-forward command generation as follows.

$$u_{fc}(k) = \left\{ \begin{array}{lcl} u_{fc}^{+} = T_f^{+}/(K_a K_t) & \rightarrow & \dot{x_r}(k) > 0 \\ 0 & \rightarrow & \dot{x_r}(k) = 0 \\ u_{fc}^{-} = T_f^{-}/(K_a K_t) & & \dot{x_r}(k) < 0 \end{array} \right\}$$

where T_f^{+}, T_f^{-} are the magnitudes of Coulomb friction measured in positive and negative directions of the motion. Hence the overall control signal is obtained as :

$$u(k) = u_{smc}(k) + u_{fc}(k)$$

The sliding mode controller can attenuate the influence of external disturbances, while providing a good tracking performance with high bandwidth.

Chapter 7

Trajectory Generation for Computer Controlled Machines

7.1 Introduction

The CNC motion control computer receives NC blocks that contain the mode of interpolation (i.e., linear – G01, circular – G02 or G03, or spline G05), the coordinates of end point ($P_e(x_e, y_e)$), arc radius ($R = \sqrt{I_c^2 + J_c^2}$), and vector feed (f). The coordinates and derivatives of series of points along a trajectory are provided for interpolation.

The acceleration (A), deceleration (D), and jerk (J) values are either set to default values within the CNC or given by the NC programmer within the NC part program. The units are converted into *counts* in the CNC system. The tool path segment (i.e., line, arc, or spline in an NC block) is broken into N number of small segments in axis directions at interpolation time intervals of T_i.

The minimum interpolation time has to be equal to or an integer multiple of the axis position control loop closure time (T). The feed f is provided by the NC part program, and the minimum interpolation period $T_{\min}$ is set within the CNC control software. The interpolation step size is calculated as

$$\Delta u = f T_{\min} \tag{7.1}$$

for set values of f and $T_{\min}$. Either step size Δu or interpolation period T_i is kept constant during velocity command generation ($f = \Delta u / T$). Since the computers became quite powerful, the current CNC systems keep the interpolatoin period T_i constant and equal to the servo control period.

7.1.1 Acceleration - Limited Velocity Profile Generation with Constant Interpolation Period

The interpolation period T_i is kept constant and typically it is set to be equal to the control interval if the motion control computer has sufficient computation power in real time, or integer multiples of it.

The acceleration, velocity and displacement profiles can be expressed as a function of time (t) as follows:

$$a(t) = \begin{Bmatrix} A & 0 \le t < t_1 & A > 0 \\ 0 & t_1 \le t < t_2 & A = 0 \\ D & t_2 \le t < t_3 & D < 0 \end{Bmatrix} \tag{7.2}$$

where A, D are the acceleration and deceleration, respectively. In most applications $D = -A$. The velocity profile is expressed as:

$$f(t) = \begin{Bmatrix} f_s + \int A dt = f_s + At & 0 \le t < t_1 & f_1 = f_s + AT_1 = f_c \\ f_c & t_1 \le t < t_2 & f_2 = f_c \\ f_c + \int D dt = f_c + Dt & t_2 \le t < t_3 & f_3 = f_c + DT_3 = f_e \end{Bmatrix} \tag{7.3}$$

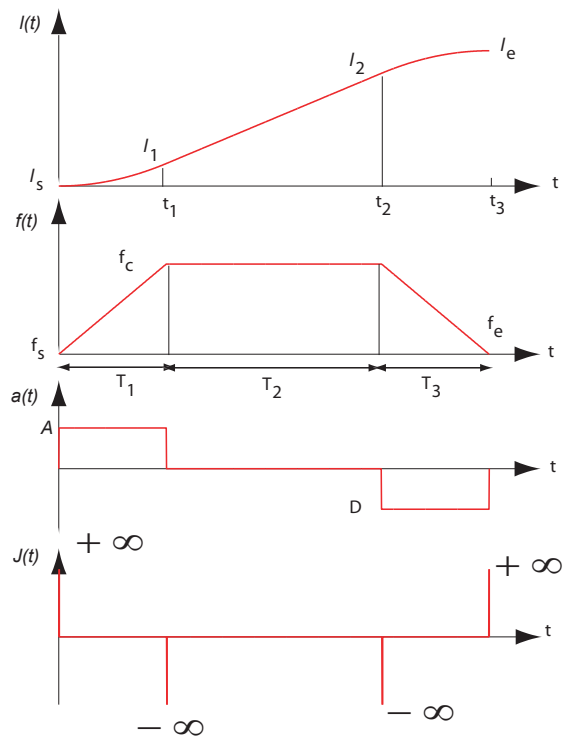


Figure 7.1: Trajectory generation with constant acceleration and fixed interpolation period.

where f_s, f_c, f_e are the start, steady and end feed values of the motion segments, respectively. The time segments are $T_1 = t_1$, $T_2 = t_2 - t_1$, $T_3 = t_3 - t_2 = T - (T_1 + T_2)$ where T is the total travel time along the path segment. The integration of feed leads to the displacement traveled at each motion segment as :

$$l(t) = \left\{ \begin{array}{lll} l_s + \int (f_s + At)dt = l_s + f_s t + \frac{1}{2}At^2 & 0 \leq t < t_1 & l_1 = l_s + f_s T_1 + \frac{1}{2}AT_1^2 \\ l_1 + \int f_c dt = l_1 + f_c t & t_1 \leq t < t_2 & l_2 = l_1 + f_c T_2 \\ l_2 + \int (f_c + Dt)dt = l_2 + f_c t + \frac{1}{2}Dt^2 & t_2 \leq t < t_3 & l_3 = l_2 + f_c T_3 + \frac{1}{2}DT_3^2 = l_s + L \end{array} \right\} \tag{7.4}$$

where L is the total path segment length. By substituting $l_1$and l_2 into the l_3, the total path length L can be expressed as a function of motion time zones, feed, acceleration and deceleration.

$$\begin{aligned} L &= l_3 - l_s = l_2 + f_c T_3 + \frac{1}{2}DT_3^2 - l_s \\ &= l_1 + f_c T_2 + f_c T_3 + \frac{1}{2}DT_3^2 - l_s \\ &= l_s + f_s T_1 + \frac{1}{2}AT_1^2 + f_c T_2 + f_c T_3 + \frac{1}{2}DT_3^2 - l_s \\ L &= f_s T_1 + \frac{1}{2}AT_1^2 + f_c T_2 + f_c T_3 + \frac{1}{2}DT_3^2 \end{aligned} \tag{7.5}$$

The time segments T_1, T_2 and T_3must be identified in order to generate trajectory commands at each interpolation period. The acceleration (T_1) and deceleration (T_3) periods can be evaluated as:

$$\begin{array}{lll} f_s + AT_1 = f_c & \rightarrow & T_1 = \frac{f_c - f_s}{A} \\ f_c + DT_3 = f_e & \rightarrow & T_3 = \frac{f_e - f_c}{D} \end{array} \tag{7.6}$$

By substituting T_1, T_3 into the total displacement length (L), the unknown T_2 can be found as :

$$
\begin{aligned}
L &= f_s T_1 + \frac{1}{2} A T_1^2 + f_c T_2 + f_c T_3 + \frac{1}{2} D T_3^2 \\
L &= f_s \left(\frac{f_c - f_s}{A} \right) + \frac{1}{2} A \left(\frac{f_c - f_s}{A} \right)^2 + f_c T_2 + f_c \left(\frac{f_e - f_c}{D} \right) + \frac{1}{2} D \left(\frac{f_e - f_c}{D} \right)^2 \\
T_2 &= \frac{L}{f_c} - \left[\left(\frac{1}{2A} - \frac{1}{2D} \right) f_c + \left(\frac{-f_s^2}{2A} + \frac{f_e^2}{2D} \right) \frac{1}{f_c} \right]
\end{aligned} \tag{7.7}
$$

Solution 88 *Details of derivation for T_2*

$$
\begin{aligned}
T_2 &= \left\{ L - \left[f_s \left(\frac{f_c - f_s}{A} \right) + \frac{1}{2} A \left(\frac{f_c - f_s}{A} \right)^2 + f_c \left(\frac{f_e - f_c}{D} \right) + \frac{1}{2} D \left(\frac{f_e - f_c}{D} \right)^2 \right] \right\} \frac{1}{f_c} \\
&= \left\{ L - \left[\frac{f_s f_c - f_s^2}{A} + \frac{f_c^2 - 2 f_s f_c + f_s^2}{2A} + \frac{f_c f_e - f_c^2}{D} + \frac{f_e^2 - 2 f_e f_c + f_c^2}{2D} \right] \right\} \frac{1}{f_c} \\
&= \left\{ L - \left[\frac{2 f_s f_c - 2 f_s^2 + f_c^2 - 2 f_s f_c + f_s^2}{2A} + \frac{2 f_e f_c - 2 f_c^2 + f_e^2 - 2 f_e f_c + f_c^2}{2D} \right] \right\} \frac{1}{f_c} \\
&= \left\{ L - \left[\frac{-f_s^2 + f_c^2}{2A} + \frac{-f_c^2 + f_e^2}{2D} \right] \right\} \frac{1}{f_c} \\
&= \left\{ L - \left[\left(\frac{1}{2A} - \frac{1}{2D} \right) f_c^2 + \left(\frac{-f_s^2}{2A} + \frac{f_e^2}{2D} \right) \right] \right\} \frac{1}{f_c}
\end{aligned}
$$

When the acceleration and deceleration values are set equal ($D = -A$), the steady state velocity time T_2 becomes:

$$
T_2 = \frac{L}{f_c} + \left[-\frac{f_c}{A} + \left(\frac{f_e^2 + f_s^2}{2A} \right) \frac{1}{f_c} \right] \tag{7.8}
$$

The total motion duration becomes $T_t = T_1 + T_2 + T_3$. The number of commands generated at the acceleration, constant velocity and deceleration zones are found as:

Acceleration	$N_1 = \frac{T_1}{T_i}$
Constant speed	$N_2 = \frac{T_2}{T_i}$
Deceleration	$N_3 = \frac{T_3}{T_i}$

The real time computation can be optimized by converting the Eq. 7.4 into difference equations by substituting $t = kT_i$ where T_i is the constant interpolation period:

$$
l(k) = \left\{ \begin{array}{lll}
l_s + f_s(kT_i) + \frac{1}{2} A (kT_i)^2 & k = 1, 2, .., (N_1 = T_1/T_i) & \begin{array}{c} l(k-1) + f_s T_i + \Delta_1 (k - \frac{1}{2}) \\ \Delta_1 = A T_i^2,\ l_1 = l(N_1) \end{array} \\
l_1 + f_c(kT_i) & k = 1, 2, .., (N_2 = T_2/T_i) & \begin{array}{c} l(k-1) + \Delta_2, \\ \Delta_2 = f_c T_i,\ l_2 = l(N_2) \end{array} \\
l_2 + f_c(kT_i) + \frac{1}{2} D (kT_i)^2 & k = 1, 2, ..(N_3 = T_3/T_i) & \begin{array}{c} l(k-1) + f_c T_i + \Delta_3 (k - \frac{1}{2}), \\ \Delta_3 = D T_i^2,\ l_3 = l(N_3) \end{array}
\end{array} \right\} \tag{7.9}
$$

The constant displacements $(\Delta_1, \Delta_2, \Delta_3)$ can be calculated during initialization and simply added recursively during real time trajectory generation. Note that $l(k)$ is discrete displacement command in space, and must be decoupled into individual axis using the kinematics of the machine.

Solution 89 *Details of derivation for the acceleration phase of trajectory generation.*

$$
\begin{aligned}
l_s + f_s(kT_i) + \frac{1}{2}A(kT_i)^2 &= l_s + f_s(k-1)T_i + f_sT_i + \frac{1}{2}A\left[(k-1)\,T_i + T_i\right]^2 \\
&= l_s + f_s(k-1)T_i + f_sT_i + \frac{1}{2}A[(k-1)\,T_i]^2 + A\,(k-1)\,T_i^2 + \frac{1}{2}AT_i^2 \\
&= \underbrace{l_s + f_s(k-1)T_i + \frac{1}{2}A[(k-1)\,T_i]^2}_{l(k-1)} + f_sT_i + AT_i^2(k-\frac{1}{2}) \\
&= l(k-1) + f_sT_i + AT_i^2(k-\frac{1}{2})
\end{aligned}
$$

When the path segments are too small, the machine may not be able to reach to the command feed (f_c), and the steady state velocity zone may be zero or negative, i.e. $T_2 \leq 0$. If the limited acceleration leads such an incompatibility with the trajectory generation, the command feed must be reduced to a compatible value which can be achieved without saturating the drives.

If $T_2 \leq 0$ in Eqn.7.7, that it is forced to become zero to identify the possible feed from a given set of acceleration, deceleration and the start and end feed values,

$$
\begin{aligned}
\frac{L}{f_{cm}} - \left[\left(\frac{1}{2A} - \frac{1}{2D}\right) f_{cm} + \left(\frac{-f_s^2}{2A} + \frac{f_e^2}{2D}\right)\frac{1}{f_{cm}}\right] &= T_2 = 0 \qquad (7.10) \\
\frac{2ADL}{2AD} - \left[\left(\frac{D-A}{2AD}\right) f_{cm}^2 + \left(\frac{f_e^2A - f_s^2D}{2AD}\right)\right] &= 0 \\
2ADL - (D-A)f_{cm}^2 - (f_e^2A - f_s^2D) &= 0 \\
2ADL - (f_e^2A - f_s^2D) &= f_{cm}^2(D-A) \\
\sqrt{\frac{2ADL - (f_e^2A - f_s^2D)}{D-A}} &= f_{cm}
\end{aligned}
$$

Note that if $D = -A$, the the allowable feed becomes

$$
f_{cm} = \sqrt{AL + \frac{(f_e^2 + f_s^2)}{2}} \qquad (7.11)
$$

The new acceleration and deceleration periods are computed as ($T_2 = 0$) :

$$
\begin{array}{lcl}
f_s + AT_1 = f_{cm} & \rightarrow & T_1 = \frac{f_{cm} - f_s}{A} \\
f_{cm} + DT_3 = f_e & \rightarrow & T_3 = \frac{f_e - f_{cm}}{D}
\end{array} \qquad (7.12)
$$

In addition to considering the zero steady state travel zone ($T_2 = 0$), it is important to consider numerical round off errors in trajectory generation. If the interpolation period T_i is fixed in the CNC system, the total number of interpolation steps must be an integer number :

$$
N = ceil(\frac{T_t}{T_i})
$$

However, if the $\frac{T_t}{T_i}$is not perfectly integer, which is always possible, than the fraction must be considered in order to correct the duration of motion according to trajectory generation. Since the interpolation period (T_i) is constant, the algorithm leads to actual total travel time $T_t' = NT_i$ which can be different than T_t if $\frac{T_t}{T_i}$ is not integer. The acceleration, steady state and deceleration time lengths (T_1, T_2, T_3) are adjusted to guarantee the total time is equal to T_t,

$$T_j^{'} = \left(\frac{T_t'}{T_t}\right) T_j \qquad \text{where } j = 1, 2, 3.$$

Since T_1', T_2' and T_3' are now different than previously estimated travel periods (T_1, T_2 and T_3), it is necessary to re-modify the nominal feed, acceleration and deceleration again. From Eq. 7.7, the new feed f_{cn} is calculated as :

$$f_{cns} = \frac{2L - f_s T_1^{'} - f_e T_3'}{T_1' + 2T_2' + T_3^{'}}$$
$$A_n = \frac{f_{cn} - f_s}{T_1^{'}}, \quad D_n = \frac{f_e - f_{cns}}{T_3^{'}}$$

The new kinematic profiles must be regenerated again by following the new feed (f_{cn}), acceleration (A_n) and deceleration (B_n) values by substituting them in Eqs. 7.2,7.3,7.4. The trajectory generation can be rearranged as;

$$l(t) = \begin{Bmatrix} l_s + f_s\tau + \frac{1}{2}A_n\tau^2 & 0 \leq \tau < T_1' & l_{1n} = l_s + f_s T_1' + \frac{1}{2}A_n T_1'^2 \\ l_{1n} + f_{cn}\tau & 0 \leq \tau < T_2' & l_{2n} = l_{1n} + f_{cn}T_2^{'} \\ l_{2n} + f_{cn}\tau + \frac{1}{2}D_n\tau^2 & 0 \leq \tau < T_3^{'} & = l_s + L \end{Bmatrix} \tag{7.13}$$

The real time computation is executed as shown in Eq.7.9.

7.2 Jerk-Limited Velocity Profile Generation

The trapezoidal velocity profile introduced in the previous section is simple to implement and suitable for most machines. However, since it employs a constant acceleration, the jerk or the derivative of the acceleration is zero, which leads to various oscillations and noise on the feed and acceleration when interpolating along complex tool paths. The acceleration times inertia or mass acts as a dynamic torque or force on the feed drive structure. If the feed drive acceleration command produced by the trajectory generator is not smooth, the resulting acceleration torque for ball screw and force for linear motor drives contain high frequency components that excite the structural dynamics of the feed drives and cause undesired vibrations. To obtain smooth velocity and acceleration profiles, jerk-limited trajectory generation algorithms are used, and these are presented in this section.

7.2.1 Kinematic Profiles.

The kinematic time profiles of jerk (J), acceleration (a), feed rate (f), and trajectory command position (l) are shown in Figure 7.2. Before the NC block motion is started, the initial and final values of position ($l_\mathrm{s}, l_\mathrm{e}$) and feed rate ($f_\mathrm{s}, f_\mathrm{e}$), maximum acceleration (A), deceleration (D), and jerk (J) limits are defined. The maximum acceleration/deceleration limits are identified from the maximum torque and force limits of the drive motors. The acceleration time is set depending on the peak torque/force delivery periods of the amplifier. The jerk limit is set by the maximum acceleration divided by the acceleration time. From Figure 7.2, the jerk ($J(\tau)$), acceleration ($a(\tau)$),feed ($f(\tau)$), and displacement ($l(\tau)$) along the tool path can be expressed. The jerk is constant when the acceleration increases linearly as:

$$J(\tau) = \begin{Bmatrix} J_1 & 0 \leq t < t_1 \\ 0 & t_1 \leq t < t_2 \\ -J & t_2 \leq t < t_3 \\ 0 & t_3 \leq t < t_4 \\ -J_5 & t_4 \leq t < t_5 \\ -0 & t_5 \leq t < t_6 \\ J_7 & t_6 \leq t < t_7 \end{Bmatrix} \tag{7.14}$$

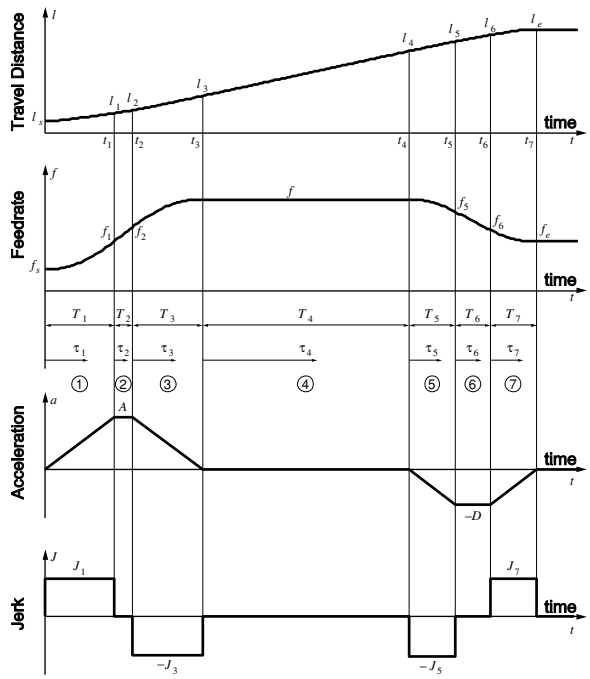


Figure 7.2: Kinematic profiles for jerk-limited feedrate generation.

The acceleration is found by integrating the jerk while considering its initial conditions:

$$
\begin{aligned}
a(t) &= a(t_i) + \textstyle\int_{t_i}^{t} J(\tau)d\tau, \\
f(t) &= f(t_i) + \textstyle\int_{t_i}^{t} a(\tau)d\tau, \\
l(t) &= l(t_i) + \textstyle\int_{t_i}^{t} f(\tau)d\tau.
\end{aligned}
\tag{7.15}
$$

The jerk is constant during acceleration (T_1, T_3) and deceleration (T_5, T_7) periods and is zero during constant acceleration (T_2, T_4, T_6). By integrating the jerk at each period shown in Figure 7.2, the acceleration profiles are obtained:

$$
a(\tau) = \left\{ \begin{array}{ll}
J_1\tau_1, & 0 \le t < t_1 \\
A, & t_1 \le t < t_2 \\
A - J_3\tau_3, & t_2 \le t < t_3 \\
0, & t_3 \le t < t_4 \\
-J_5\tau_5, & t_4 \le t < t_5 \\
-D, & t_5 \le t < t_6 \\
-D + J_7\tau_7, & t_6 \le t < t_7
\end{array} \right\},
\tag{7.16}
$$

where τ_k $(k = 1, 2, \ldots, 7)$ denotes relative time (i.e., $\tau_k = t - t_{k-1}$) at each each motion zone. By integrating the acceleration (Eq. 7.16) at each period, the feed velocity profile for each phase in Figure 7.2 is found to be

$$f(\tau)=\begin{cases} f_{\mathrm{s}}+\frac{1}{2}J_1\tau_1^2, & f_{\mathrm{s}}:\text{initial feed}, & 0\le t<t_1 \\ f_1+A\tau_2, & f_1=f_s+\frac{1}{2}J_1T_1^2, & t_1\le t<t_2 \\ f_2+A\tau_3-\frac{1}{2}J_3\tau_3^2, & f_2=f_1+AT_2, & t_2\le t<t_3 \\ f_3, & f=f_3=f_2+AT_3-\frac{1}{2}J_3T_3^2, & t_3\le t<t_4 \\ f_4-\frac{1}{2}J_5\tau_5^2, & f=f_4=f_3, & t_4\le t<t_5 \\ f_5-D\tau_6, & f_5=f_4-\frac{1}{2}J_5T_5^2, & t_5\le t<t_6 \\ f_6-D\tau_7+\frac{1}{2}J_7\tau_7^2, & f_6=f_5-DT_6, & t_6\le t<t_7 \end{cases} \tag{7.17}$$

where T_k is the period of phase k. Integrating (7.17) once more yields the displacement profile as

$$l(\tau)=\begin{cases} l_{\mathrm{s}}+f_{\mathrm{s}}\tau_1+\frac{1}{6}J_1\tau_1^3, & l_s:\text{initial position}, & 0\le t<t_1 \\ l_1+f_1\tau_2+\frac{1}{2}A\tau_2^2, & l_1=l_s+f_sT_1+\frac{1}{6}J_1T_1^3, & t_1\le t<t_2 \\ l_2+f_2\tau_3+\frac{1}{2}A\tau_3^2-\frac{1}{6}J_3\tau_3^3, & l_2=l_1+f_1T_2+\frac{1}{2}AT_2^2, & t_2\le t<t_3 \\ l_3+f_3\tau_4, & l_3=l_2+f_2T_3+\frac{1}{2}AT_3^2-\frac{1}{6}J_3T_3^3, & t_3\le t<t_4 \\ l_4+f_4\tau_5-\frac{1}{6}J_5\tau_5^3, & l_4=l_3+f_3T_4, & t_4\le t<t_5 \\ l_5+f_5\tau_6-\frac{1}{2}D\tau_6^2, & l_5=l_4+f_4T_5-\frac{1}{6}J_5T_5^3, & t_5\le t<t_6 \\ l_6+f_6\tau_7-\frac{1}{2}D\tau_7^2+\frac{1}{6}J_7\tau_7^3, & l_6=l_5+f_5T_6-\frac{1}{2}DT_6^2, & t_6\le t<t_7 \end{cases}, \tag{7.18}$$

where l_k is the total displacement reached at the end of phase k. The total path segment length during the motion is :

The incremental distance (L_k) travelled at each phase (k) is

$$L_k=l_k-l_{k-1}, \tag{7.19}$$

where the initial displacement is $l_0=l_{\mathrm{s}}$. The sum of the distances traveled at each step should be equal to the total distance to be traveled (Eqs. 7.18 and 7.19):

$$L=l_{\mathrm{e}}-l_{\mathrm{s}}=\sum_{k=1}^{7}L_k, \tag{7.20}$$

where L is the total distance traveled within the NC tool path segment. From the trapezoidal acceleration/ deceleration profiles,

$$A=J_1T_1=J_3T_3,\quad D=J_5T_5=J_7T_7 \tag{7.21}$$

should hold, although this may require readjustment after the first initialization step. Considering that the desired feed (f) is reached at the end of phase three, we have

$$f_3=f\rightarrow T_2=\frac{1}{A}\left[f-f_{\mathrm{s}}-\frac{1}{2}J_1T_1^2-AT_3+\frac{1}{2}J_3T_3^2\right], \tag{7.22}$$

and similarly, considering that the final feed (f_{e}) is reached at the end of phase seven, we have

$$\begin{aligned} f_7&=f_{\mathrm{e}}=f_6-DT_7+\tfrac{1}{2}J_7T_7^2, \\ T_6&=\tfrac{1}{D}\left[f-f_{\mathrm{e}}-\tfrac{1}{2}J_5T_5^2-DT_7+\tfrac{1}{2}J_7T_7^2\right]. \end{aligned} \tag{7.23}$$

7.2.2 Initialization.

Before generating incremental position commands in the interpolation stage, the number of interpolation steps and the time interval between each of them must be predetermined to implement the jerk-limited kinematic profile given in Figure 7.2. To design a general algorithm, the following conditions are evaluated at the initialization step.

The total number of interpolation steps (N) is checked. If $2 < N \leq 4$, then $N = 4$ is selected to guarantee at least the presence of acceleration and deceleration phases $(1, 3, 5, 7)$ in Figure 7.2. If $N \leq 2$, then $N = 2$ is selected to allow an acceleration and deceleration. Note that these conditions would only occur when the motion is very small, such as in high speed spline interpolation applications or in precision positioning. If the acceleration stage exists, the desired feed (f) must be reached within the first three phases, which implies that $T_2 \geq 0$. If the jerk values are equal ($J_1 = J_3$) (Eq. 7.21), the acceleration condition requires that

$$T_1 = T_3 = A/J_1, \quad T_2 = \frac{f - f_\mathrm{s}}{A} - \frac{A}{J_1} \geq 0. \tag{7.24}$$

If Eq. (7.24) does not hold, than the magnitude of the acceleration must be reduced to its maximum possible limit as

$$A = \mathrm{sgn}(A) \cdot \sqrt{J_1(f - f_\mathrm{s})} \tag{7.25}$$

and T_2 is set to zero. Similarly, if the deceleration stage exists,

$$T_5 = T_7 = D/J_5, \quad T_6 = \frac{f - f_\mathrm{e}}{D} - \frac{D}{J_5} \geq 0. \tag{7.26}$$

If the deceleration stage does not exist, the deceleration limit is reduced as

$$D = \mathrm{sgn}(D) \cdot \sqrt{J_5(f - f_\mathrm{e})} \tag{7.27}$$

and T_6 is set to zero. If the displacement length is large enough to encompass the constant feed stage, $T_4 \geq 0$ and must be identified. The total travel length is calculated from Eqs. (7.18) and (7.20) by substituting T_2 and T_6 from Eqs. (7.23) and (7.26) as

$$L = \left(\frac{1}{2A} + \frac{1}{2D}\right) f^2 + \left(\frac{A}{2J_1} + \frac{D}{2J_5} + T_4\right) f + \left(\frac{Af_\mathrm{s}}{2J_1} + \frac{Df_\mathrm{e}}{2J_5} - \frac{f_\mathrm{s}^2}{2A} - \frac{f_\mathrm{e}^2}{2D}\right). \tag{7.28}$$

If the tool path distance is too short to reach the target feed (f), its corresponding terms must not be considered in Eq. (7.28). In order to have a constant feed (f) phase, $T_4 \geq 0$ should hold:

$$\begin{aligned} T_4 = \frac{1}{f}\Bigg[L - \Bigg\{&\left(\frac{1}{2A} + \frac{1}{2D}\right) f^2 + \left(\frac{A}{2J_1} + \frac{D}{2J_5}\right) f \\ &+ \left(\frac{Af_\mathrm{s}}{2J_1} + \frac{Df_\mathrm{e}}{2J_5} - \frac{f_\mathrm{s}^2}{2A} - \frac{f_\mathrm{e}^2}{2D}\right)\Bigg\}\Bigg] \geq 0. \end{aligned} \tag{7.29}$$

If Eq. (7.29) is not satisfied, then $T_4 = 0$ and the magnitude of the feed rate is reduced to its maximum possible value by solving Eq. (7.29). If the quadratic equation yields complex roots, the initial and final feeds are set to zero ($f_\mathrm{s} = f_\mathrm{e} = 0$), and the path re-initialization is repeated by adjusting the initial parameters (A, D, J).

7.2.3 Number of Interpolation Steps in Each Phase.

The incremental displacement Δu is decided by the linear, circular, or spline interpolation algorithms. Since the displacement in each phase is defined in Eq. (7.19), the number of interpolation steps in phases $(1, 3, 5, 7)$ can be expressed as

$$\begin{aligned} N_1 &= \mathrm{round}\,(L_1/\Delta u), \quad N_3 = \mathrm{round}\,(L_3/\Delta u), \\ N_5 &= \mathrm{round}\,(L_5/\Delta u), \quad N_7 = \mathrm{round}\,(L_7/\Delta u). \end{aligned} \tag{7.30}$$

If any of (N_1, N_3, N_5, N_7) are zero for nonzero L_1, L_3, L_5, L_7 due to rounded integers, they are set to one. The total number of steps for the acceleration (N_{ac}) and deceleration (N_{dec}) stages are calculated as

$$N_{\text{ac}} = \text{round}\,\frac{(L_1 + L_2 + L_3)}{\Delta u}, \quad N_{\text{dec}} = \text{round}\,\frac{(L_5 + L_6 + L_7)}{\Delta u}, \tag{7.31}$$

from which the number of steps for constant acceleration (phase 2) and constant deceleration (phase 6) are found:

$$N_2 = N_{\text{ac}} - (N_1 + N_3), \quad N_6 = N_{\text{dec}} - (N_5 + N_7). \tag{7.32}$$

The number of interpolation steps for the constant feed (phase 4) becomes

$$N_4 = N - (N_{\text{ac}} + N_{\text{dec}}). \tag{7.33}$$

The number of interpolation steps $(N_1, \ldots, N_7)$ for each phase of the jerk-limited velocity profile generation (Fig. 7.2) is now defined. Although the total number of interpolation steps is still N, the displacement at each phase may be changed because of constraints, that is,

$$L'_k = N_k \cdot \Delta u, \quad k = 1, 2, \ldots, 7, \tag{7.34}$$

where L'_k is the new displacement value, which may be different than the value given in Eq. 7.19). This requires readjustment of all acceleration, deceleration, jerk, and time periods again.

For the acceleration stage (phase 1,2,3) when $T_2 > 0$, Eq. (7.22) for T_2 and the new displacements L'_1, L'_2, L'_3 are substituted in Eqs. (7.18) and (7.19), resulting in the following equations:

$$\left.\begin{aligned} & f_{\text{s}}T_1 + \tfrac{1}{6}AT_1^2 - L'_1 = 0 \\ & -\tfrac{1}{8}AT_1^2 + \tfrac{1}{8}AT_3^2 - \tfrac{1}{2}f_{\text{s}}T_1 - \tfrac{1}{2}fT_3 + \tfrac{f^2 - f_{\text{s}}^2}{2A} - L'_2 = 0 \\ & fT_3 - \tfrac{1}{6}AT_3^2 - L'_3 = 0 \end{aligned}\right\} \rightarrow \text{for } T_2 > 0. \tag{7.35}$$

If $T_2 = 0$, Eq. (7.19) can not be used. Instead, the feed rate condition for the end of acceleration stage (f_3 in Eq. 7.17) is used, which leads to

$$\left.\begin{aligned} & f_{\text{s}}T_1 + \tfrac{1}{6}AT_1^2 - L'_1 = 0 \\ & \tfrac{1}{2}AT_1 + \tfrac{1}{2}AT_3 + f_{\text{s}} - f = 0 \\ & \tfrac{1}{3}AT_3^2 + \tfrac{1}{2}AT_1T_3 + f_{\text{s}}T_3 - L'_3 = 0 \end{aligned}\right\} \rightarrow \text{for } T_2 = 0. \tag{7.36}$$

A similar approach is applied to the deceleration stage (phases 5,6,7). By substituting T_6 and the new displacements L'_5, L'_6, L'_7 in Eqs. (7.18) and (7.19), we get

$$\left.\begin{aligned} & fT_5 - \tfrac{1}{6}DT_5^2 - L'_5 = 0 \\ & \tfrac{1}{8}DT_5^2 - \tfrac{1}{8}DT_7^2 - \tfrac{1}{2}fT_5 - \tfrac{1}{2}f_{\text{e}}T_7 + \tfrac{f^2 - f_{\text{e}}^2}{2D} - L'_6 = 0 \\ & f_{\text{e}}T_7 + \tfrac{1}{6}DT_7^2 - L'_7 = 0 \end{aligned}\right\} \rightarrow \text{for } T_6 > 0. \tag{7.37}$$

For the case $T_6 = 0$ the feed rate condition at the end of deceleration is used:

$$\left.\begin{aligned} & fT_5 - \tfrac{1}{6}DT_5^2 - L'_5 = 0 \\ & \tfrac{1}{2}DT_5 + \tfrac{1}{2}DT_7 + f_{\text{e}} - f = 0 \\ & -\tfrac{1}{3}DT_7^2 - \tfrac{1}{2}DT_5T_7 + fT_7 - L'_7 = 0 \end{aligned}\right\} \rightarrow \text{for } T_6 = 0. \tag{7.38}$$

For the acceleration stage, either Eq. (7.35) for $T_2 > 0$ or Eq. (7.36) for $T_2 > 0$ is used to find the new values of T_1, T_3, A, and T_2. For the deceleration stage, either Eq. (7.37) for $T_6 > 0$ or (7.38) for $T_6 = 0$

is used to find the updated values of T_5, T_7, D, and T_6. These equations are nonlinear and can be solved iteratively using the Newton–Raphson numerical algorithm. T_4 is found in Eq. (7.29). Once the values of $T_1, T_2, \ldots, T_7$ and A, D are found, the jerk values are updated using Eqs. (7.21) and (7.26), the final feed rate values reached at the end of each phase are recalculated from Eq. (7.17).

In short, the number of interpolation steps $(N_1, \ldots, N_7)$, time intervals $(T_1, \ldots, T_7)$, feed values $(f_1, \ldots, f_7)$, displacements $(L'_1, \ldots, L'_7)$, and possible accelerations and decelerations (A, D, J) are found for each phase of the motion block before the real time interpolation starts.

7.2.4 Recursively Executed Real Time Part.

Here the interpolation period at each step must be identified. After the initialization part is completed as explained above, the continuously executed real time part of the jerk-limited feed rate generation algorithm is invoked at each interpolation step. For any of the seven phases in Figure 7.2, the following general displacement formula can be used:

$$l(\tau_k) = \frac{1}{6} J_{0,k} \tau_k^3 + \frac{1}{2} a_{0,k} \tau_k^2 + f_{0,k} \tau_k + l_{0,k} \tag{7.39}$$

where $J_{0,k}$, $a_{0,k}$, $f_{0,k}$, and $l_{0,k}$ are jerk, acceleration, feed rate, and displacement values respectively, and they are calculated at the initialization stage. τ_k is the relative time parameter at the beginning of each phase k. Since the incremental displacement step Δu is predetermined at the interpolation, the displacement at interpolation step number m can be found as

$$l(\tau_{k,m}) = m \cdot \Delta u = \frac{1}{6} J_{0,k} \tau_{k,m}^3 + \frac{1}{2} a_{0,k} \tau_{k,m}^2 + f_{0,k} \tau_{k,m} + l_{0,k}, \tag{7.40}$$

which leads to the solution of accumulated time $(\tau_{k,m})$ within the phase k. Although the above equation can be solved analytically, it is more efficient to solve it in real time by using the Newton–Raphson iterative algorithm. The interpolation period for within phase k at interpolation step m can be evaluated by

$$T_{i(k,m)} = \tau_{k,m} - \tau_{k,m-1}. \tag{7.41}$$

Alternatively, by replacing the initial conditions at each step, the interpolation period can be solved from Eq. (7.40) as well but at the expense of numerical round-off errors.

7.2.5 Reconstruction of Reference Trajectory at Servo Loop Control Frequency.

Reference trajectories generated with varying interpolation periods may cause sudden changes in the discrete position commands at each axis and hence cause undesirable jerks on the drives. To smoothen the generated trajectory, the discrete position commands of each drive at varying interpolation periods must be re-sampled at the servo loop frequency. Note that the re-sampling is done after the trajectory is generated by the interpolation stage presented in the following section, although it is presented here because of its relevance to jerk-limited trajectory generation.

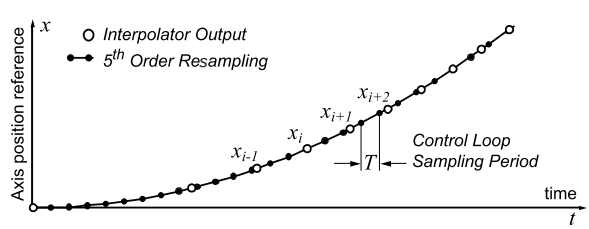


Figure 7.3: Resampling of interpolator output Trajectory at Control Loop Frequency.

A sample trajectory is shown in Figure 7.3, where the interpolator and re-sampled trajectory at servo control interval T are shown. If we consider one of the drives, for example the x axis, two consecutive displacements of the drive are denoted by x_i, x_{i+1} where i is the interpolation counter. A fifth-order polynomial

is fit between the two points as

$$\tilde{x}(\tau) = A_{\mathrm{r}}\tau^5 + B_{\mathrm{r}}\tau^4 + C_{\mathrm{r}}\tau^3 + D_{\mathrm{r}}\tau^2 + E_{\mathrm{r}}\tau + F_{\mathrm{r}}, \tag{7.42}$$

where the time varies as $0 \leq \tau \leq (t_{i+1} - t_i)$. The coefficients $(A_{\mathrm{r}}, B_{\mathrm{r}}, C_{\mathrm{r}}, D_{\mathrm{r}}, E_{\mathrm{r}}, F_{\mathrm{r}})$ are identified from the boundary conditions as

$$\begin{aligned} &\tilde{x}(0) = x_i, \quad \tilde{x}(t_{i+1} - t_i) = x_{i+1}, \\ &\frac{d\tilde{x}(0)}{d\tau} = \dot{x}_i, \quad \frac{d\tilde{x}(t_{i+1} - t_i)}{d\tau} = \dot{x}_{i+1}, \\ &\frac{d^2\tilde{x}(0)}{d\tau^2} = \ddot{x}_i, \quad \frac{d^2\tilde{x}(t_{i+1} - t_i)}{d\tau^2} = \ddot{x}_{i+1}, \end{aligned} \tag{7.43}$$

where $(\dot{x}_i, \ddot{x}_i; \dot{x}_{i+1}, \ddot{x}_{i+1})$ are the feed rate and acceleration estimates at the beginning and end of the interpolated position command segment, respectively. They can be estimated using a third-order cubic polynomial similar to the method described in the quintic or fifth-order spline interpolation as explained in Section 7.3.3. For each new reference point coming out of the interpolator algorithm, the polynomial coefficients are recalculated, and the reference axis position commands are generated at servo control loop interval T from Eq. (7.42) (i.e., $\tau = 0, T, 2T, \ldots (t_{i+1} - t_i)$).

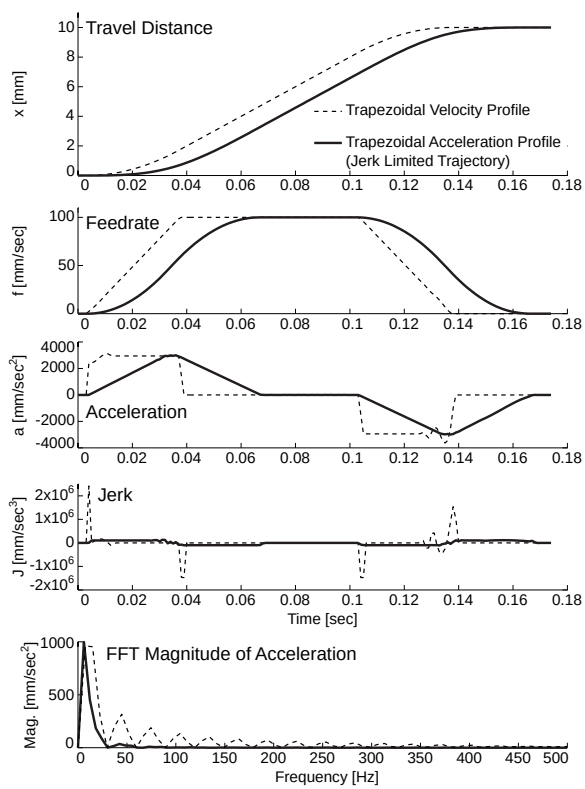


Figure 7.4: Comparison trajectories generated with jerk-limited and trapezoidal velocity profiles.

Example. The trajectory generation with simple trapezoidal velocity and jerk-limited trapezoidal acceleration profiles are compared in Figure 7.4 during a single axis (x) motion. Transient oscillations are present in the acceleration and jerk when the trapezoidal velocity profile is used. However, it can be seen that both acceleration and jerk become very smooth when the trapezoidal acceleration is used instead. The presence of many harmonics in the acceleration indicates that the trapezoidal velocity profile may excite structural modes of the feed drive; hence it is not recommended for high speed machine tool drives. Further smoothing

is achieved on the acceleration and jerk when the interpolated axis displacements are re-sampled using a fifth-order polynomial (see Fig. 7.5).

7.3 Real Time Interpolation Methods

CNC systems must be capable of following the complex trajectories that may be present on parts to be machined. Most of the common geometries can be constructed from linear and circular segments. However, high speed machining of dies, molds, and aerospace parts with sculptured surfaces requires real time spline interpolation as well.

In this section, common design methods for real time two-axis linear, circular, and quintic spline interpolation methods are presented. The algorithms are general and can easily be extended to motions containing more than two active axes.

7.3.1 Linear Interpolation Algorithm

Method I- Constant Displacement- varying Interpolation Period :

This interpolation method is based on the digital integration of the velocity components in two axes.

Let us assume that a cutting tool center is to follow the linear path shown in Figure 7.6. The starting point of the tool is $P_{\mathrm{s}}(x_{\mathrm{s}}, y_{\mathrm{s}})$ and the end point is $P_{\mathrm{e}}(x_{\mathrm{e}}, y_{\mathrm{e}})$. At time t the position of the axes will be at

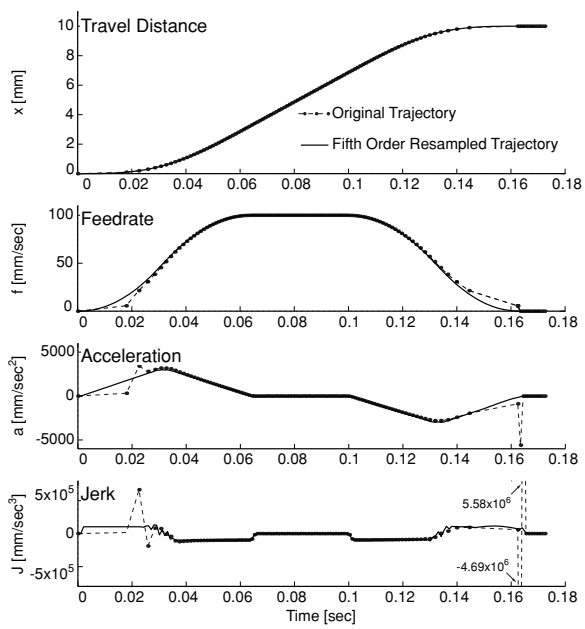


Figure 7.5: Original trajectory generated by the interpolator and smoothed trajectory with 5th order resampling at the axis level.

$$\begin{aligned} x(t) &= x(k \cdot \Delta t) = x_{\mathrm{s}} + \int_0^{\mathrm{t}} f_x(t)dt, \\ y(t) &= y(k \cdot \Delta t) = y_{\mathrm{s}} + \int_0^{\mathrm{t}} f_y(t)dt, \end{aligned} \tag{7.44}$$

where the axis velocities f_x and f_y and interpolation time interval T_i vary with time during acceleration and deceleration periods but remain constant during the constant feed zone as explained in the previous section. Since the interpolation algorithm is executed N times at time intervals T_i, the above equation (7.44) can be

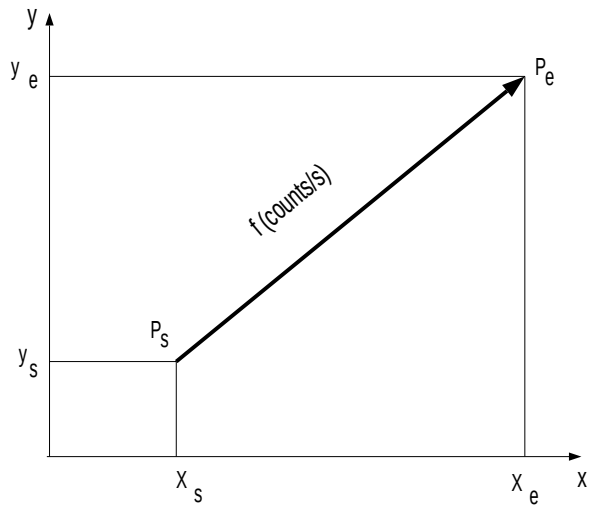


Figure 7.6: Linear interpolation.

expressed in discrete form as

$$x(k) = x_{\mathrm{s}} + \sum_{j=1}^{k} f_x(j)T_i(j) = x_{\mathrm{s}} + \sum_{j=1}^{k-1} f_x(j)T_i(j) + f_x(k)T_i(k),$$

$$y(k) = y_{\mathrm{s}} + \sum_{j=1}^{k} f_y(j)T_i(j) = y_{\mathrm{s}} + \sum_{j=1}^{k-1} f_y(j)T_i(j) + f_y(k)T_i(k),$$

or

$$\begin{aligned} x(k) &= x(k-1) + f_x(k)T_i(k), \\ y(k) &= y(k-1) + f_y(k)T_i(k). \end{aligned} \tag{7.45}$$

Note that the axis velocities at time interval k are

$$f_x(k) = \frac{\Delta x}{T_i(k)}, \quad f_y(k) = \frac{\Delta y}{T_i(k)}. \tag{7.46}$$

Manipulation of interpolation time interval T_i results in manipulation of the axis feeds according to the vector feed and the acceleration/deceleration velocity profile. However, the incremental displacements in both axes are constant and are given by

$$\Delta x = \frac{x_{\mathrm{e}} - x_{\mathrm{s}}}{N}, \quad \Delta y = \frac{y_{\mathrm{e}} - y_{\mathrm{s}}}{N}. \tag{7.47}$$

Substituting Eqs. (7.46) and (7.47) into (7.45) gives the recursive digital linear interpolation equations:

$$\begin{aligned} x(k \cdot \Delta t) &= x_{\mathrm{s}} + k \cdot \Delta x = x(k-1) + \Delta x, \\ y(k \cdot \Delta t) &= y_{\mathrm{s}} + k \cdot \Delta y = y(k-1) + \Delta y. \end{aligned} \tag{7.48}$$

The integration increments Δx and Δy are constant and calculated once at the beginning of the interpolation routine. A real time computer implementation of the linear interpolation is given below.

T_i	The interpolation period.
f	The requested feed rate.
$x_{\mathrm{s}}, y_{\mathrm{s}}$	Start position.
$x_{\mathrm{e}}, y_{\mathrm{e}}$	End position.
δx	Total distance in x axis.
δy	Total distance in y axis.
$\mathrm{sign}(x)$	Direction of x movement.

$\text{sign}(y)$	Direction of y movement.
N	The number of interpolation iterations.
dx, dy	Step size for each axis.
x_{rem}	Remainder for x axis step size.
y_{rem}	Remainder for y axis step size.

(All values are integer except T_i and f)

The initialization calculations in the CNC are

$$
\begin{aligned}
&\delta x = \text{abs}(x_{\text{e}} - x_{\text{s}}) \\
&\delta y = \text{abs}(y_{\text{e}} - y_{\text{s}}) \\
&\text{sign}(x) = \text{sign}(x_{\text{e}} - x_{\text{s}}) \\
&\text{sign}(y) = \text{sign}(y_{\text{e}} - y_{\text{s}}) \\
&\Delta u = fT_i \\
&N = \sqrt{\delta x^2 + \delta y^2}/\Delta u \\
&dx = \text{fix}(\delta x/N) \\
&dy = \text{fix}(\delta y/N) \\
&x_{\text{rem}} = \delta x - (dx \cdot N) \\
&y_{\text{rem}} = \delta y - (dy \cdot N) \\
&\text{line}(x_{\text{s}}, dx, x_{\text{rem}}, \text{sign}(x), N) \qquad \text{; sent to } x \text{ axis position controller} \\
&\text{line}(y_{\text{s}}, dy, y_{\text{rem}}, \text{sign}(y), N) \qquad \text{; sent to } y \text{ axis position controller}
\end{aligned}
$$

After initialization of as many axes as are needed, each can be calculated at every period T_i with the following:

```
function line (x_s, dx, x_rem, sign(x), N)
x(1) = x_s
x_error = 0
for i = 2, N + 1
   x(i) = x(i − 1) + sign(x) · dx
   x_error = x_error + x_rem
   if (x_error >= N)
      x(i) = x(i) + sign(x)
      x_error = x_error − N
   end
end
```

The generated position commands at each interpolation time interval T_i are then sent to the digital servo control algorithm of each axis in the CNC.

Example. The following NC block is entered to the CNC in *counts*:

N010 G01 G90 X24 Y32 F1000

This NC program provides $P_{\text{e}}(24, 32)$ and $f = 1000$ [counts/s]. The initial coordinates were given as $P_{\text{s}}(5, 6)$, the acceleration and deceleration values of the CNC are set as $A = D = 50{,}000$ [counts/s^2], and the trapezoidal velocity profile is used. The minimum interpolation time is $T_{\min} = 0.002\,\text{s}$ in the CNC. The interpolation algorithm is set as follows:

$$
\begin{aligned}
&\delta x = x_{\text{e}} - x_{\text{s}} = 19, \quad \delta y = y_{\text{e}} - y_{\text{s}} = 26, \quad L = \sqrt{\delta x^2 + \delta y^2} = 32, \\
&\Delta u = fT_{\min} = 2, \quad N = L/\Delta u = 16, \quad N_1 = N_3 = f^2/(2A\Delta u) = 5, \\
&\Delta x = \delta x/N = 1.1875, \quad \Delta y = \delta y/N = 1.625.
\end{aligned}
$$

The numerical interpolation results are shown in Table 7.1 and Figure 7.7.

Table 7.1: Linear Interpolation Steps between Points $P_s(5, 6)$ and $P_e(24, 32)$

Clock Pulses k	**Feed** $f(k)$ **[counts/s]**	**Interpolation Interval** T_i **(ms)**	**Time** t **[ms]**	$x(k)$ **[counts]**	$y(k)$ **[counts]**
0	0.00	0.00000	0.0	5.00	6.00
1	447.21	0.00894	8.9	6.19	7.63
2	632.46	0.00370	12.6	7.38	9.25
3	774.60	0.00284	15.5	8.56	10.88
4	894.43	0.00240	17.9	9.75	12.50
5	1000.00	0.00211	20.0	10.94	14.13
6	1000.00	0.00200	22.0	12.13	15.75
7	1000.00	0.00200	24.0	13.31	17.38
8	1000.00	0.00200	26.0	14.50	19.00
9	1000.00	0.00200	28.0	15.69	20.63
10	1000.00	0.00200	30.0	16.88	22.25
11	1000.00	0.00200	32.0	18.06	23.88
12	894.43	0.00211	34.1	19.25	25.50
13	774.60	0.00240	36.5	20.44	27.13
14	632.46	0.00284	39.4	21.63	28.75
15	447.21	0.00370	43.1	22.81	30.38
16	0.00	0.00894	52.0	24.00	32.00

Method II- Varying Displacement - Constant Interpolation Period

The acceleration limited trajectory generation was derived previously as:

$$l(k) = \begin{cases} l_s + f_s\left(kT_i\right) + \frac{1}{2}A\left(kT_i\right)^2 & k = 1, 2, .., N_1\left(= T_1/T_i\right) & \begin{array}{c} l(k-1) + f_sT_i + \Delta_1\left(k - \frac{1}{2}\right), \\ \Delta_1 = AT_i^2, l_1 = l\left(N_1\right) \end{array} \\ l_1 + f_ckT_i & k = 1, 2, .., N_2\left(= T_2/T_i\right) & \begin{array}{c} l(k-1) + \Delta_2, \\ \Delta_2 = f_cT_i, l_2 = l\left(N_2\right) \end{array} \\ l_2 + f_c\left(kT_i\right) + \frac{1}{2}D\left(kT_i\right)^2 & k = 1, 2, .., N_3\left(= T_3/T_i\right) & \begin{array}{c} l(k-1) + f_cT_i + \Delta_3\left(k - \frac{1}{2}\right), \\ \Delta_3 = DT_i^2, l_3 = l\left(N_3\right) \end{array} \end{cases} \tag{7.49}$$

Note: Details of derivation

$$\begin{aligned} l(k) &= l_s + f_s(kT_i) + \frac{1}{2}A(kT_i)^2 \\ &= l_s + f_s(k-1)T_i + f_sT_i + \frac{1}{2}A\left[(k-1)+1\right]^2 T_i^2 \\ &= l_s + f_s(k-1)T_i + f_sT_i + \frac{1}{2}A\left[(k-1)^2 + 2(k-1) + 1\right] T_i^2 \\ &= l_s + f_s(k-1)T_i + f_sT_i + \frac{1}{2}A\left[(k-1)T_i\right]^2 + \frac{1}{2}A(2k-1)T_i^2 \\ &= l(k-1) + f_sT_i + A\left[k - \frac{1}{2}\right] T_i^2 \end{aligned}$$

where the displacement $l(k)$ is calculated at each time interval k at constant interpolation period T_i. If the tool travels from point $P_s(x_s, y_s)$ to point $P_e(x_e, y_e)$ along the linear path, the direction is described by the following unit vector:

$$\frac{\overrightarrow{P_1P_2}}{\left|\overrightarrow{P_1P_2}\right|} = \frac{(x_e - x_s)}{L}\overrightarrow{i} + \frac{(y_e - y_s)}{L}\overrightarrow{j} \tag{7.50}$$

$$\Delta x = \frac{x_e - x_s}{L} \quad ; \Delta y = \frac{y_e - y_s}{L} \tag{7.51}$$

The tool position at interpolation period k can be evaluated as:

$$\overrightarrow{l(k)} = \left(\Delta x\overrightarrow{i} + \Delta y\overrightarrow{j}\right) l(k) = \overrightarrow{l(k)} = \Delta x \cdot l(k)\overrightarrow{i} + \Delta y \cdot l(k)\overrightarrow{j}$$
$$\overrightarrow{l(k)} = x(k)\overrightarrow{i} + y(k)\overrightarrow{i}$$

Alternatively, the same expression can be evaluated from the line geometry:

$$x(k) = l(k)\cos\theta = l(k)\frac{(x_e - x_s)}{L} = l(k)\Delta x$$
$$y(k) = l(k)\sin\theta = l(k)\frac{(y_e - y_s)}{L} = l(k)\Delta y$$

Substituting $l(k)$ into the linear interpolation leads to following recursive difference equations which need to be executed at constant time interval T_i.

$$l(k) = \left\{ \begin{array}{ll} x(k) = \Delta x\left[l(k-1) + f_sT_i + \Delta_1(k-\frac{1}{2})\right] = x(k-1) + \Delta x\left[f_sT_i + \Delta_1(k-\frac{1}{2})\right] & \Delta_1 = AT_i^2 \\ y(k) = \Delta y\left[l(k-1) + f_sT_i + \Delta_1(k-\frac{1}{2})\right] = y(k-1) + \Delta y\left[f_sT_i + \Delta_1(k-\frac{1}{2})\right] & k = 1,2,..,(N_1 = T_1/T_i) \\ x(k) = \Delta x\left[l(k-1) + \Delta_2\right] = x(k-1) + \Delta x\Delta_2 & \Delta_2 = f_cT_i \\ y(k) = \Delta y\left[l(k-1) + \Delta_2\right] = y(k-1) + \Delta y\Delta_2 & k = 1,2,..,(N_2 = T_2/T_i) \\ x(k) = \Delta x\left[l(k-1) + f_cT_i + \Delta_3(k-\frac{1}{2})\right] = x(k-1) + \Delta x\left[f_cT_i + \Delta_3(k-\frac{1}{2})\right] & \Delta_3 = DT_i^2 \\ y(k) = \Delta y\left[l(k-1) + f_cT_i + \Delta_3(k-\frac{1}{2})\right] = y(k-1) + \Delta y\left[f_cT_i + \Delta_3(k-\frac{1}{2})\right] & k = 1,2,..(N_3 = T_3/T_i) \end{array} \right\} \tag{7.52}$$

The algorithm is first initialized by identifying $(\Delta x, \Delta y)$, $(\Delta_1 = AT_i^2, N_1 = T_1/T_i), (\Delta_2 = f_cT_i, N_2 = T_2/T_i), (\Delta_3 = DT_i^2, N_3 = T_3/T_i)$, and followed by the recursive calculations listed above at each time interval T_i. Since the displacement $l(k)$ is found by integrating the feed, which is in turn calculated by integrating the acceleration, each displacement command at time interval k inherently contains projected velocities in x and y directions.

7.3.2 Circular Interpolation Algorithm

Consider a circular segment whose center is at the origin of the CNC's coordinate system (see Fig. 7.8). The circular tool path has a length of

$$L = R(\theta_{\mathrm{e}} - \theta_{\mathrm{s}}) = R\theta_{\mathrm{t}}. \tag{7.53}$$

The arc must be broken into N small segments for digital interpolation. The length of each segment is Δu, with a corresponding angular segment of $\Delta\theta$ as shown in Figure 7.8. The chord error (chord_error) must be kept less than the resolution of position sensing systems (i.e., 1 count):

$$\text{chord_error} = R\left(1 - \cos\frac{\Delta\theta}{2}\right) \leq 1, \tag{7.54}$$

which is satisfied when the angular segment $\Delta\theta$ is selected as:

$$\Delta\theta \leq 2\cos^{-1}\left(\frac{R-1}{R}\right). \tag{7.55}$$

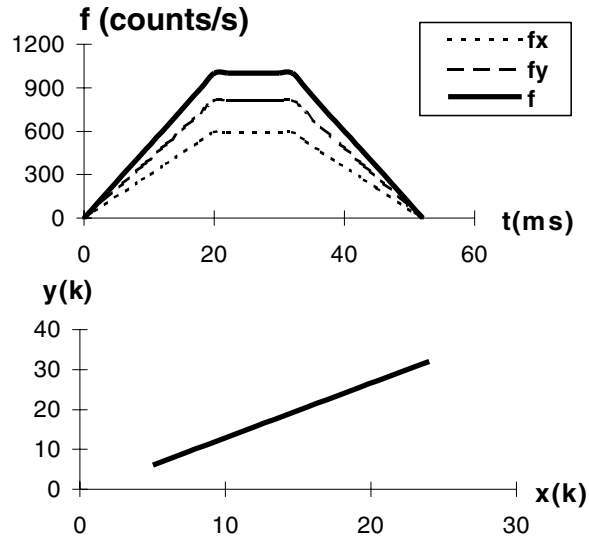


Figure 7.7: Linear interpolation example.

The selection of

$$\Delta\theta = \cos^{-1}\left(\frac{R-1}{R}\right), \quad \cos\Delta\theta = \frac{R-1}{R} \tag{7.56}$$

guarantees half a count chord error and is more advantageous for computation. The corresponding chord segment is found as

$$\Delta u = R\Delta\theta. \tag{7.57}$$

The circular tool path is broken into N_1, N_2, N_3 segments for trapezoidal velocity profiling as explained earlier. The tool has a feed speed of f [counts/s] on the arc, and the arc segment Δu is traveled in one interpolation period T_i. The algorithm is implemented in real time either by keeping the interpolation period T_i or angular displacement $\Delta\theta$ (or chord displacement Δu constant.

Method I- Constant Displacement- varying Interpolation Period : The interpolation period T_i varies during acceleration and deceleration zones (N_1, N_3), but it remains constant during the steady state feed zone (N_2) as shown in Figure **??**. The angular feed velocity $\omega(t)$ and instantaneous angular position $(\theta(t))$ of the cutter at time t are

$$\omega = \frac{f}{R}, \quad \theta(t) = \omega t = \frac{f}{R}t \tag{7.58}$$

respectively. Coordinates of a point on the arc can be expressed as

$$\begin{aligned} x(t) &= R\ \cos\theta(t) = R\cos\left(\frac{f}{R}t\right), \\ y(t) &= R\ \sin\theta(t) = R\sin\left(\frac{f}{R}t\right). \end{aligned} \tag{7.59}$$

The velocities of the x and y axis feed drives are

$$\begin{aligned} f_x &= \frac{dx}{dt} = -\frac{f}{R}R\sin\left(\frac{f}{R}t\right) = -\frac{f}{R}\cdot y(t), \\ f_y &= \frac{dy}{dt} = \frac{f}{R}R\cos\left(\frac{f}{R}t\right) = \frac{f}{R}\cdot x(t). \end{aligned} \tag{7.60}$$

Because the velocities are coupled with the positions, digital integration of the above equations always produces some errors. A decoupled recursive circular interpolation explained in the following can be easily implemented on CNC systems that are capable of handling floating point arithmetic operations.

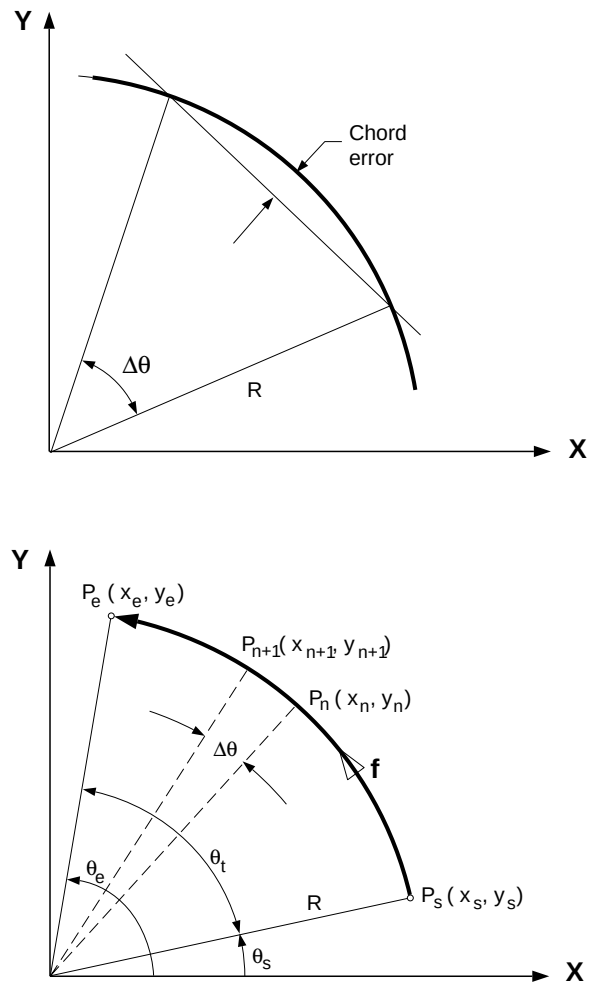


Figure 7.8: Circular interpolation.

Consider two points, P_{n} and $P_{\mathrm{n+1}}$ on the arc, which are $\Delta\theta$ away from each other (Figure 7.8):

$$\begin{aligned} x_{\mathrm{n}} &= R\ \cos(\theta_{\mathrm{s}} + n\Delta\theta), \quad y_{\mathrm{n}} = R\ \sin(\theta_{\mathrm{s}} + n\Delta\theta), \\ x_{\mathrm{n+1}} &= R\ \cos[\theta_{\mathrm{s}} + (n+1)\Delta\theta], \quad y_{\mathrm{n+1}} = R\ \sin[\theta_{\mathrm{s}} + (n+1)\Delta\theta]. \end{aligned} \tag{7.61}$$

The coordinates of $P_{\mathrm{n+1}}$ can be expanded as

$$\begin{aligned} x_{\mathrm{n+1}} &= R\ \cos(\theta_{\mathrm{s}} + n\Delta\theta + \Delta\theta), \\ y_{\mathrm{n+1}} &= R\ \sin(\theta_{\mathrm{s}} + n\Delta\theta + \Delta\theta), \end{aligned}$$

or

$$\begin{aligned} x_{\mathrm{n+1}} &= R\cos(\theta_{\mathrm{s}} + n\Delta\theta)\cos\Delta\theta - R\sin(\theta_{\mathrm{s}} + n\Delta\theta)\sin\Delta\theta, \\ y_{\mathrm{n+1}} &= R\sin(\theta_{\mathrm{s}} + n\Delta\theta)\cos\Delta\theta + R\cos(\theta_{\mathrm{s}} + n\Delta\theta)\sin\Delta\theta. \end{aligned}$$

The products of trigonometric functions can be expanded as

$$\sin(\theta_{\mathrm{s}} + n\Delta\theta)\sin\Delta\theta = \frac{1}{2}\{\cos[\theta_{\mathrm{s}} + (n-1)\Delta\theta] - \cos[\theta_{\mathrm{s}} + (n+1)\Delta\theta]\},$$

$$\cos(\theta_{\mathrm{s}} + n\Delta\theta)\sin\Delta\theta = \frac{1}{2}\{-\sin[\theta_{\mathrm{s}} + (n-1)\Delta\theta] + \sin[\theta_{\mathrm{s}} + (n+1)\Delta\theta]\}.$$

Substituting the trigonometric expansion into the coordinates of $P_{\mathrm{n+1}}$, we get

$$x_{\mathrm{n+1}} = R\ \cos(\theta_{\mathrm{s}} + n\Delta\theta)\cos\Delta\theta - \frac{R}{2}\cos[\theta_{\mathrm{s}} + (n-1)\Delta\theta] + \frac{R}{2}\cos[\theta_{\mathrm{s}} + (n+1)\Delta\theta],$$

$$y_{\mathrm{n+1}} = R\ \sin(\theta_{\mathrm{s}} + n\Delta\theta)\cos\Delta\theta - \frac{R}{2}\sin[\theta_{\mathrm{s}} + (n-1)\Delta\theta] + \frac{R}{2}\sin[\theta_{\mathrm{s}} + (n+1)\Delta\theta].$$

When the coordinates of x_{n} and y_{n} are substituted in Eq. (7.61), the following difference equations are obtained:

$$x_{\mathrm{n+1}} = x_{\mathrm{n}}\cos\Delta\theta - \frac{1}{2}x_{\mathrm{n-1}} + \frac{1}{2}x_{\mathrm{n+1}},$$

$$y_{\mathrm{n+1}} = y_{\mathrm{n}}\cos\Delta\theta - \frac{1}{2}y_{\mathrm{n-1}} + \frac{1}{2}y_{\mathrm{n+1}}.$$

The discrete position equations can be reduced to a set of decoupled recursive axis equations:

$$\begin{aligned} x_{\mathrm{n+1}} &= 2x_{\mathrm{n}}\cos\Delta\theta - x_{\mathrm{n-1}}, \\ y_{\mathrm{n+1}} &= 2y_{\mathrm{n}}\cos\Delta\theta - y_{\mathrm{n-1}}. \end{aligned} \tag{7.62}$$

Note that the integration interval $\Delta\theta$ is selected according to the maximum chord error criterion, and $\cos\Delta\theta$ is given by Eq. (7.56). The interpolation time intervals are calculated according to the velocity profile as explained in Section 7.1. The numerical value of $\Delta\theta$ is precomputed and stored in the memory before the real time, recursive interpolation algorithm starts.

The algorithm has been tested on the research milling machine controlled by an in-house developed CNC system. A two-axis 117 count (0.148 mm) linear move, followed by an 80 count (0.102 mm) move in the x direction and a full circular path with 80 count (0.102 mm) radius were executed on the machine tool at an intended feed of 4,000 counts/s. The command and measured tool paths, actual feed along the tool path, and the corresponding axis velocities are shown in Figure 7.9. With $A = 50{,}000$ counts/s^2 acceleration, the target feed is never reached because of the short travel distances. However, the path is still followed with a satisfactory accuracy.

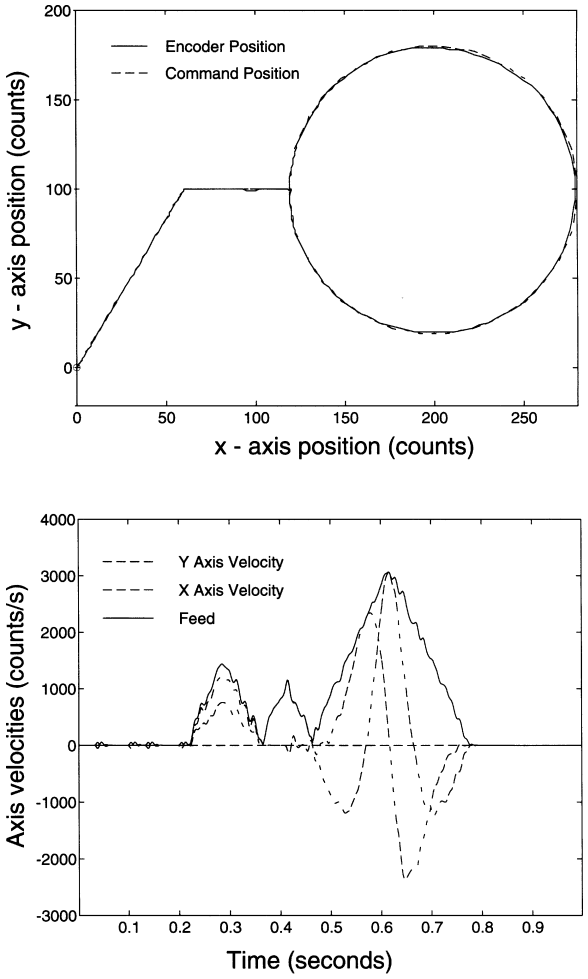


Figure 7.9: A sample tool path with combined linear and circular interpolation segments. The actual position and velocities were evaluated from encoder measurements. 1 count = 0.00127 mm.

Table 7.2: Circular Interpolation Steps between Points $P_s(180, 0)$ and $P_e(90, -90)$

Clock Pulses k	**Feed** $f(k)$ **[counts/s]**	**Interpolation Interval** T_i **(ms)**	**Time** t **[ms]**	$x(k)$ **[counts]**	$y(k)$ **[counts]**
0	0.00	0.0000	0	180.00	0.00
1	463.53	0.1158	116	176.02	26.46
2	655.53	0.0480	164	164.44	50.58
3	802.86	0.0368	201	146.28	70.23
4	927.06	0.0310	232	123.14	83.68
5	1000.0	0.0278	260	97.08	89.72
6	1000.0	0.0268	286	70.38	87.84
7	1000.0	0.0268	313	45.43	78.19
8	1000.0	0.0268	340	24.41	61.63
9	1000.0	0.0268	367	9.19	39.62
10	1000.0	0.0268	394	1.11	14.11
11	1000.0	0.0268	421	0.89	−12.65
12	886.08	0.0284	449	8.55	−38.29
13	755.17	0.0327	482	23.41	−60.54
14	596.17	0.0397	522	44.15	−77.45
15	374.90	0.0553	577	68.95	−87.50
16	0.00	0.1432	720	90.00	−90.00

Example. The following NC block is entered to the CNC in counts:

N010 G90 G03 X90 Y-90 I- 90 J0 F1000

This NC program provides $P_e(90, -90)$, $f = 1{,}000$ [counts/s], and $R = \sqrt{I^2 + J^2} = 90$ [counts]. The initial coordinates were given as $P_s(180, 0)$, and the acceleration (A) and deceleration (D) values of the CNC are set to $A = D = 4{,}000$ [counts/s^2] with a trapezoidal velocity profile. The minimum interpolation time is $T_{\min} = 0.002\,\text{s}$ in the CNC. The interpolation algorithm is set as follows:

$$\theta_s = 0, \quad \theta_e = \theta_t = 3\pi/2 = 4.71239, \quad \Delta\theta = 0.298419,$$
$$\Delta u = R\Delta\theta = 26.857723, \quad N_1 = N_3 = f^2/(2A\Delta u) = 5, \quad N = \theta_t/\Delta\theta = 16.$$

The numerical interpolation results are shown in Table 7.2 and Figure 7.10.

Method II- Varying Displacement - Constant Interpolation Period :

The instantaneous angular position of the tool along the circular tool path is :

$$\theta(k) = \frac{l(k)}{R} \tag{7.63}$$

The circular tool path is broken into N_1, N_2, N_3 segments for trapezoidal velocity profiling as explained earlier. The tool has a feed speed of f [counts/s] on the arc, and the arc segment $\Delta\theta(k)$ is traveled in one interpolation period T_i. The angular feed velocity $\omega(t)$ and instantaneous angular position ($\theta(t)$) of the cutter at time t are:

$$\omega(t) = \frac{f(t)}{R}, \quad \theta(t) = \omega(t) \cdot t = \frac{f(t)}{R} t \tag{7.64}$$

respectively. The angular velocity and position can be expressed in discrete time interval $t = kT_i \in k = 0, 1, 2, ...$

$$\omega(kT_i) = \frac{f(kT_i)}{R}, \quad \theta(k) = \omega(k) \cdot kT_i = \frac{f(k)}{R}(kT_i) \tag{7.65}$$

The variable arc segment displacement at time interval k can be expressed as:

$$\Delta u(k) = R\Delta\theta(k) = f(k) \cdot T_i$$

Considering that the maximum angular displacement must meet the following constrain to maintain the chord error less than one encoder unit:

$$\Delta\theta_{\max} \leq 2\cos^{-1}\frac{R-1}{R} \rightarrow \Delta\theta(k) \leq \Delta\theta_{\max}$$

Since the interpolation period T_i is constant, the maximum feed must be kept as follows:

$$f_{\max} = \frac{R\Delta\theta_{\max}}{T_i}$$

which overrides the operator's desired feed.

The coordinates of a point on the arc can be expressed as

$$\begin{aligned} x(t) &= R\ \cos\theta(t) = R\cos\left(\frac{f}{R}t\right), \\ y(t) &= R\ \sin\theta(t) = R\sin\left(\frac{f}{R}t\right). \end{aligned} \tag{7.66}$$

The velocities of the x and y axis feed drives are

$$\begin{aligned} f_x &= \frac{dx}{dt} = -\frac{f}{R}R\sin\left(\frac{f}{R}t\right) = -\frac{f}{R}\cdot y(t), \\ f_y &= \frac{dy}{dt} = \frac{f}{R}R\cos\left(\frac{f}{R}t\right) = \frac{f}{R}\cdot x(t). \end{aligned} \tag{7.67}$$

Because the velocities are coupled with the positions, digital integration of the above equations always produces some errors. A decoupled recursive circular interpolation explained in the following can be easily implemented on CNC systems that are capable of handling floating point arithmetic operations.

Consider two points, P_{k} and $P_{\mathrm{k+1}}$ on the arc, which are $\Delta\theta$ away from each other (Figure 7.8):

$$\begin{aligned} x_{\mathrm{k}} &= R\ \cos\theta(k), \quad y_{\mathrm{k}} = R\ \sin\theta(k), \\ x_{\mathrm{k+1}} &= R\ \cos[\theta(k) + \Delta\theta(k)], \quad y_{\mathrm{k+1}} = R\ \sin[\theta(k) + \Delta\theta(k)]. \end{aligned} \tag{7.68}$$

The coordinates of $P_{\mathrm{k+1}}$ can be expanded as

$$\begin{aligned} x_{\mathrm{k+1}} &= R\cos\theta(k)\cos\Delta\theta(k) - R\sin\theta(k)\sin\Delta\theta(k), \\ y_{\mathrm{k+1}} &= R\sin\theta(k)\cos\Delta\theta(k) + R\cos\theta(k)\sin\Delta\theta(k). \end{aligned}$$

Considering the following trigonometric laws:

$$\begin{array}{ccc} \cos(a-b) = \cos a\cos b + \sin a\sin b & & \sin(a+b) = \sin a\cos b + \cos a\sin b \\ \cos(a+b) = \cos a\cos b - \sin a\sin b & \text{and} & \sin(a-b) = \sin a\cos b - \cos a\sin b \\ \sin a\sin b = \frac{\cos(a-b)-\cos(a+b)}{2} & & \cos a\sin b = \frac{-\sin(a-b)+\sin(a+b)}{2} \end{array}$$

The products of trigonometric functions can be expanded as

$$\sin\theta(k)\sin\Delta\theta(k) = \frac{1}{2}\{\cos[\theta(k) - \Delta\theta(k)] - \cos[\theta(k) + \Delta\theta(k)]\},$$

$$\cos\theta(k)\sin\Delta\theta(k) = \frac{1}{2}\{-\sin[\theta(k) - \Delta\theta(k)] + \sin[\theta(k) + \Delta\theta(k)]\}.$$

Substituting the trigonometric expansion into the coordinates of $P_{\mathrm{k}+1}$, we get

$$x_{\mathrm{k}+1} = R\cos\theta(k)\cos\Delta\theta(k) - \frac{R}{2}\cos[\theta(k) - \Delta\theta(k)] + \frac{R}{2}\cos[\theta(k) + \Delta\theta(k)],$$

$$y_{\mathrm{k}+1} = R\sin\theta(k)\cos\Delta\theta(k) - \frac{R}{2}\sin[\theta(k) - \Delta\theta(k)] + \frac{R}{2}\sin[\theta(k) + \Delta\theta(k)]].$$

When the coordinates of x_{k} and y_{k} are substituted in Eq. (7.61), the following difference equations are obtained:

$$x_{\mathrm{k}+1} = x_{\mathrm{k}}\cos\Delta\theta(k) - \frac{1}{2}x_{\mathrm{k}-1} + \frac{1}{2}x_{\mathrm{k}+1},$$

$$y_{\mathrm{k}+1} = y_{\mathrm{k}}\cos\Delta\theta(k) - \frac{1}{2}y_{\mathrm{k}-1} + \frac{1}{2}y_{\mathrm{k}+1}.$$

The discrete position equations can be reduced to a set of decoupled recursive axis equations:

$$\begin{array}{l} x_{\mathrm{k}+1} = 2x_{\mathrm{k}}\cos\Delta\theta(k) - x_{\mathrm{k}-1}, \rightarrow \Delta x(k) = x(k+1) - x(k) \\ y_{\mathrm{k}+1} = 2y_{\mathrm{k}}\cos\Delta\theta(k) - y_{\mathrm{k}-1}. \rightarrow \Delta x(k) = y(k+1) - y(k) \end{array} \tag{7.69}$$

Remembering the constant acceleration based trajectory generation algorithm as:

$$l(k) = \left\{ \begin{array}{lll} l_s + f_s(kT_i) + \frac{1}{2}A(kT_i)^2 & k = 1,2,..,(N_1 = T_1/T_i) & \begin{array}{l} l(k-1) + f_sT_i + \Delta_1(k - \frac{1}{2}) \\ \Delta_1 = AT_i^2,\ l_1 = l(N_1) \end{array} \\ l_1 + f_c(kT_i) & k = 1,2,..,(N_2 = T_2/T_i) & \begin{array}{l} l(k-1) + \Delta_2, \\ \Delta_2 = f_cT_i,\ l_2 = l(N_2) \end{array} \\ l_2 + f_c(kT_i) + \frac{1}{2}D(kT_i)^2 & k = 1,2,..(N_3 = T_3/T_i) & \begin{array}{l} l(k-1) + f_cT_i + \Delta_3(k - \frac{1}{2}), \\ \Delta_3 = DT_i^2,\ l_3 = l(N_3) \end{array} \end{array} \right\}$$

The real time computation of $\cos\Delta\theta(k)$ becomes costly since $\Delta\theta(k)$ changes during acceleration and deceleration periods when time interval is kept constant:

$$\Delta\theta(k) = \frac{l(k) - l(k-1)}{R} \rightarrow \cos\Delta\theta(k) = \cos\frac{l(k) - l(k-1)}{R} \tag{7.70}$$

Since k is increasing, $\Delta\theta(k)$ becomes time varying at each interpolation interval.

$$\Delta\theta(k) = \frac{l(k) - l(k-1)}{R} = \left\{ \begin{array}{lll} \frac{1}{R}\left[f_sT_i + AT_i^2(k - \frac{1}{2})\right] & k = 1,2,..,(N_1 = T_1/T_i) & l_1 = l(N_1) \\ \left[\frac{f_cT_i}{R}\right] & k = 1,2,..,(N_2 = T_2/T_i) & l_2 = l(N_2) \\ \frac{1}{R}\left[f_sT_i + DT_i^2(k - \frac{1}{2})\right] & k = 1,2,..(N_3 = T_3/T_i) & l_3 = l(N_3) \end{array} \right\}$$

It must be noted that $\Delta\theta(k)$ is quite a small number, hence it must be handled with precision algorithm in real time in order to avoid accumulation of round-off errors.

7.3.3 Quintic Spline Interpolation within CNC Systems

In addition to linear and circular interpolation, modern CNCs offer parabolic, helical, and spline interpolations. Instead of breaking sculptured paths into small linear and circular segments, it is preferred to move the cutting tool along the complex path using the path geometry itself. This reduces the length of the NC program and improves the smooth generation of velocity and acceleration in addition to the contouring along the path. Real time quintic spline interpolation is most useful in high speed machining of dies and molds, and its implementation in the CNC is presented here.

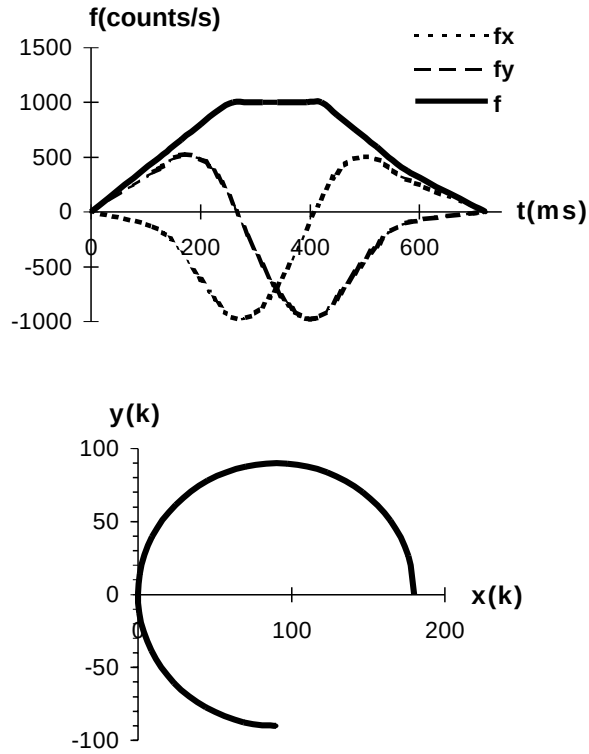


Figure 7.10: Circular interpolation example.

A series of n knots $(P_1, P_2, \ldots, P_{\mathrm{n}})$ along the tool path are to be connected by quintic spline segments with smooth transitions (see Fig. 7.11). A spline segment S_i connecting two knots (P_i, P_{i+1}) is expressed by a fifth-order polynomial as follows:

$$\{S_{qi}\} = \{A_{qi}\}u^5 + \{B_{qi}\}u^4 + \{C_{qi}\}u^3 + \{D_{qi}\}u^2 + \{E_{qi}\}u + \{F_{qi}\}, \tag{7.71}$$

where $q : \{x, y, z\}, u \in [0, l]$ and $\{A_{qi}\}, \{B_{qi}\}, \{D_{qi}\}, \{E_{qi}\}, \{F_{qi}\}$ are vectors containing the coefficients for each coordinate (i.e., $q : x, y, z$ for a three-axis machine), and u is the spline parameter that changes between zero and spline segment length l_i. The number of elements in each vector is therefore equal to the number of axes in motion. The real time interpolation algorithm requires the values of all spline coefficients (A, B, C, D, E, F), the spline length (l_i), and the feed, acceleration, and jerk limits of the machine tool drives as input parameters. The quintic spline interpolation has both off-line and on-line mathematical steps. First, a quintic spline is fit to a series of knots along a sculptured tool path, and the coefficients are identified. The fitting can either be done on the CAD/CAM system during tool path generation, and the coefficients of the spline can be transferred to CNC via NC codes, or they can be identified within the preprocessing stage of CNC by using the raw knot coordinates, supplied within the NC program. The length of the spline segment is estimated by approximating its chord length. The coefficients, knot coordinates, and approximated chord length are used to generate individual axis coordinates in real time by considering the feed, acceleration, and jerk limits of the machine tool drives. Each step in the implementation is shown in Figure 7.12 and explained in the following sections.

Evaluation of Quintic Spline Parameters

Since there are six coefficients, we need six boundary conditions to evaluate them. These are the coordinates (P_i, P_{i+1}), first derivatives $(\frac{dP_i}{du}, \frac{dP_{i+1}}{du})$, and the second derivatives $(\frac{d^2P_i}{du^2}, \frac{d^2P_{i+1}}{du^2})$ at the two knots (P_i, P_{i+1}).

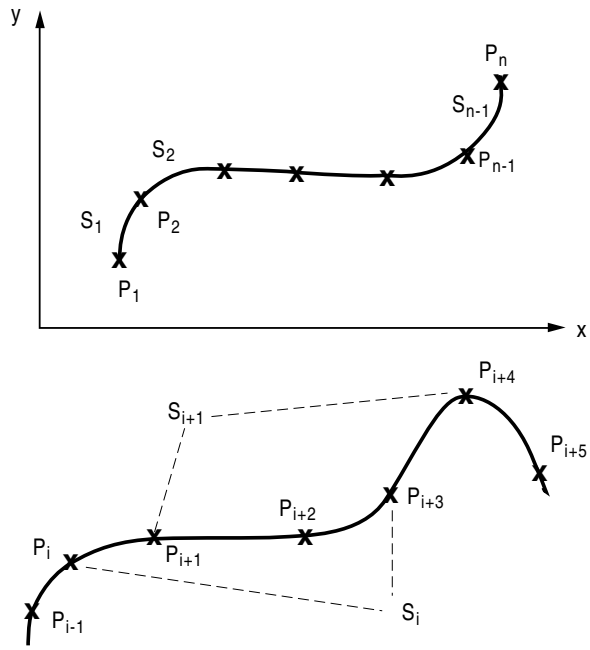


Figure 7.11: Spline passing through knots and the use of cubic polynomials in estimating the first and second derivatives.

The first and second derivatives of the quintic spline (7.71) with respect to u are

$$\begin{aligned} \left\{\frac{dS_i}{du}\right\} &= 5\left\{A_i\right\}u^4 + 4\left\{B_i\right\}u^3 \\ &\quad +3\left\{C_i\right\}u^2 + 2\left\{D_i\right\}u + \left\{E_i\right\}, \\ \left\{\frac{d^2S_i}{du^2}\right\} &= 20\left\{A_i\right\}u^3 + 12\left\{B_i\right\}u^2 \\ &\quad +6\left\{C_i\right\}u + 2\left\{D_i\right\}. \end{aligned} \tag{7.72}$$

The coefficients are evaluated from the following matrix:

$$\begin{bmatrix} u_0^5 & u_0^4 & u_0^3 & u_0^2 & u_0 & 1 \\ u_1^5 & u_1^4 & u_1^3 & u_1^2 & u_1 & 1 \\ 5u_0^4 & 4u_0^3 & 3u_0^2 & 2u_0 & 1 & 0 \\ 5u_1^4 & 4u_1^3 & 3u_1^2 & 2u_1 & 1 & 0 \\ 20u_0^3 & 12u_0^2 & 6u_0 & 2 & 0 & 0 \\ 20u_1^3 & 12u_1^2 & 6u_1 & 2 & 0 & 0 \end{bmatrix} \begin{Bmatrix} A_i \\ B_i \\ C_i \\ D_i \\ E_i \\ F_i \end{Bmatrix} = \begin{Bmatrix} P_i \\ P_{i+1} \\ \frac{dP_i}{du} \\ \frac{dP_{i+1}}{du} \\ \frac{d^2P_i}{du^2} \\ \frac{d^2P_{i+1}}{du^2} \end{Bmatrix}, \tag{7.73}$$

which leads to the solution of six quintic spline coefficients with $u_0 = 0, u_1 = l_i$:

$$\begin{aligned} \{A_i\} &= \frac{1}{u_1^5}\left[6\left(P_{i+1} - P_i\right) - 3\left(\frac{dP_{i+1}}{du} + \frac{dP_i}{du}\right)u_1 + \frac{1}{2}\left(\frac{d^2P_{i+1}}{du^2} - \frac{d^2P_i}{du^2}\right)u_1^2\right], \\ \{B_i\} &= \frac{1}{u_1^4}\left[-15\left(P_{i+1} - P_i\right) + \left(7\frac{dP_{i+1}}{du} + 8\frac{dP_i}{du}\right)u_1 - \left(\frac{d^2P_{i+1}}{du^2} - \frac{3}{2}\frac{d^2P_i}{du^2}\right)u_1^2\right], \\ \{C_i\} &= \frac{1}{2u_1^3}\left[20\left(P_{i+1} - P_i\right) - 4\left(2\frac{dP_{i+1}}{du} + 3\frac{dP_i}{du}\right)u_1 + \left(\frac{d^2P_{i+1}}{du^2} - 3\frac{d^2P_i}{du^2}\right)u_1^2\right], \\ \{D_i\} &= \frac{1}{2}\frac{d^2P_i}{du^2}, \\ \{E_i\} &= \frac{dP_i}{du}, \\ \{F_i\} &= P_i. \end{aligned}$$

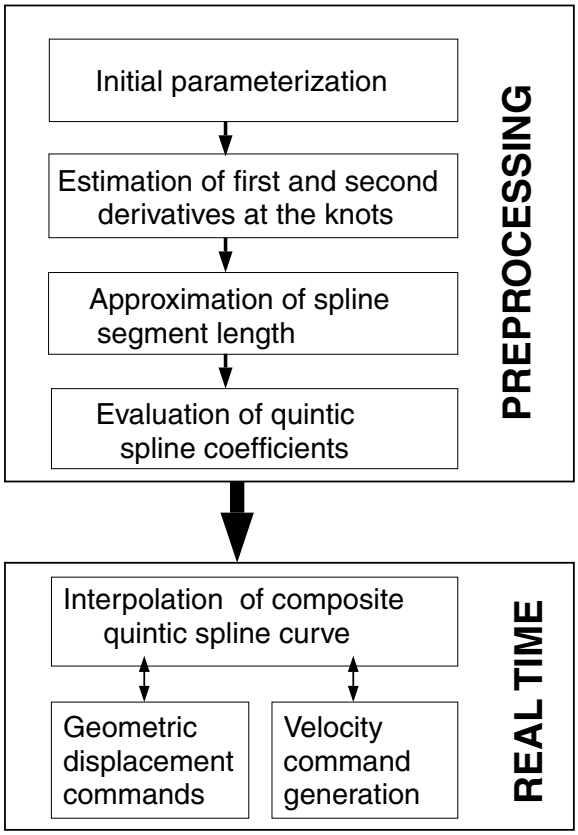


Figure 5.20: Preprocessing and real time processing steps of quintic spline interpolation.

Figure 7.12: Preprocessing and real time processing steps of quintic spline interpolation.

In order to evaluate the coefficients (7.74), the first and second derivatives at the two ends (P_i, P_{i+1}) of the spline segment must be known.

Identification of Derivatives Using Parabolic Approximation

The derivatives at the knots are estimated by approximating a parabola between three consecutive knots. Consider the following parabola:

$$\{S_i(u)\} = \{a_i\}\, u^2 + \{b_i\}\, u + \{c_i\}\,, \quad u \in [0, L_i, L_{i+1}]. \tag{7.74}$$

passing through three consecutive knots P_i, P_{i+1}, P_{i+2}, that is,

$$\begin{bmatrix} u_0^2 & u_0 & 1 \\ u_1^2 & u_1 & 1 \\ u_2^2 & u_2 & 1 \end{bmatrix} \begin{Bmatrix} \{a_i\} \\ \{b_i\} \\ \{c_i\} \end{Bmatrix} = \begin{Bmatrix} \{P_i\} \\ \{P_{i+1}\} \\ \{P_{i+2}\} \end{Bmatrix}, \tag{7.75}$$

where $u_0 = 0, u_1 = L_i,$ and $u_2 = L_i + L_{i+1}$. The distance between each knot is approximated as a line:

$$L_i = |P_{i+1} - P_i| = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \tag{7.76}$$

for a three-axis path. The parameters of the parabola can be identified from the matrix expression as Eq. (7.75) as

$$\begin{aligned} \{a_i\} &= \frac{(P_{i+2} - P_i)(u_1 - u_0) - (P_{i+1} - P_i)(u_2 - u_0)}{\left(u_2^2 - u_0^2\right)(u_1 - u_0) - \left(u_1^2 - u_0^2\right)(u_2 - u_0)}, \\ \{b_i\} &= \frac{(P_{i+1} - P_i) - \left(u_1^2 - u_0^2\right)\{a_i\}}{(u_1 - u_0)}, \\ \{c_i\} &= P_i - u_0^2\{a_i\} - u_0\{b_i\}. \end{aligned} \tag{7.77}$$

The first and second derivatives of the parabola at knot P_i are

$$\begin{aligned} \left\{\frac{dS_i(u)}{du}\right\} &= 2\{a_i\}u + \{b_i\}, \\ \left\{\frac{d^2S_i(u)}{du^2}\right\} &= 2\{a_i\}. \end{aligned} \tag{7.78}$$

Note that the second derivative has a unique solution, and the first derivative has three solutions using $u_0 = 0, u_1 = L_i,$ and $u_2 = L_i + L_{i+1}$ at subsequent knots. An average value of the first derivative can be used for better accuracy at the mid knots. The solution is unique at the first and last knot.

Identification of Derivatives Using Cubic Spline Approximation

Experience has indicated that the parabolic approximation of the spline with a linear chord length assumption does not lead to smooth feed rate changes along the quintic spline path. Although computational more demanding, a cubic spline approximation for the derivatives at the knots and a near–arc length estimation for the chord length between the knots are found to generate smoother feed and acceleration during quintic spline interpolation.

Consider a cubic spline passing through the knots $P_i, P_{i+1}, P_{i+2}, P_{i+3}$ shown in Figure 7.11:

$$\{S_i(u)\} = \{a_i\}\, u^3 + \{b_i\}\, u^2 + \{c_i\}\, u + \{d_i\}, \quad u \in [0, L_i + L_{i+1} + L_{i+2}], \tag{7.79}$$

where L_i is the length of the chord segment between knots (P_i, P_{i+1}) and is to be estimated using near–arc length parameterization. The number of elements in each vector is equal to the number of active axes on

the machine. The boundary conditions at each knot are

$$\{S_i(u)\} = \left\{ \begin{matrix} P_i, & u_0 = 0, \\ P_{i+1}, & u_1 = L_i, \\ P_{i+2}, & u_2 = L_i + L_{i+1}, \\ P_{i+3}, & u_3 = L_i + L_{i+1} + L_{i+2}. \end{matrix} \right\} \tag{7.80}$$

By substituting the boundary conditions at four knots (7.80) into the cubic spline equation (7.79) the spline parameters can be evaluated:

$$\begin{bmatrix} u_0^3 & u_0^2 & u_0 & 1 \\ u_1^3 & u_1^2 & u_1 & 1 \\ u_2^3 & u_2^3 & u_2^3 & 1 \\ u_3^3 & u_3^3 & u_3^3 & 1 \end{bmatrix} \begin{Bmatrix} \{a_i\} \\ \{b_i\} \\ \{c_i\} \\ \{d_i\} \end{Bmatrix} = \begin{Bmatrix} P_i \\ P_{i+1} \\ P_{i+2} \\ P_{i+3} \end{Bmatrix}, \tag{7.81}$$

which leads to the following solution:

$$\{a_i\} = \left[\frac{(P_{i+1} - P_i)u_2u_3(u_2 - u_3) + (P_{i+2} - P_i)u_3u_1(u_3 - u_1) + (P_{i+3} - P_i)u_1u_2(u_1 - u_2)}{\Delta_{\mathrm{c}}} \right],$$

$$\{b_i\} = \left[\frac{(P_{i+1} - P_i)u_2u_3(u_3^2 - u_2^2) + (P_{i+2} - P_i)u_3u_1(u_1^2 - u_3^2) + (P_{i+3} - P_i)u_1u_2(u_2^2 - u_1^2)}{\Delta_{\mathrm{c}}} \right],$$

$$\{c_i\} = \left[\frac{(P_{i+1} - P_i)u_2^2u_3^2(u_2 - u_3) + (P_{i+2} - P_i)u_3^2u_1^2(u_3 - u_1) + (P_{i+3} - P_i)u_1^2u_2^2(u_1 - u_2)}{\Delta_{\mathrm{c}}} \right],$$

$$\{d_i\} = \{P_i\},$$

where

$$\Delta_{\mathrm{c}} = u_1u_2u_3\left[u_1^2(u_2 - u_3) + u_2^2(u_3 - u_1) + u_3^2(u_1 - u_2)\right].$$

The derivatives of the spline at each knot are

$$\left. \begin{aligned} \frac{dS_i}{du} &= 3\{a_i\}u^2 + 2\{b_i\}u + \{c_i\} \\ \frac{d^2S_i}{du^2} &= 6\{a_i\}u + 2\{b_i\} \end{aligned} \right\} \quad u \in [0,\ u_3]. \tag{7.82}$$

If there are n knots on the spline, the derivatives at the first two knots (P_1, P_2) and the last two knots are evaluated using the coordinates of the first four (P_1, P_2, P_3, P_4) and last four $(P_{\mathrm{n}-3}, P_{\mathrm{n}-2}, P_{\mathrm{n}-1}, P_{\mathrm{n}})$. The derivatives at the remaining knots are evaluated for two mid knots of each spline segment that spans over four points. For example, spline segment S_i is used to evaluate the derivatives at knots P_{i+1}, P_{i+2} in Figure 7.11. This will lead to two values for each derivative, which are obtained from two consecutive spline segments. An average of the two derivative values is used in the quintic spline equation (7.74).

Near–Arc Length Parameterization

The length of a spline segment between the two consecutive knots (P_i, P_{i+1}) is better approximated by assuming an arc length instead of a straight line. When the spline (dS_i) is parameterized by its arc length, it must have a unit tangent along the spline segment. A closed arc length can be found when the deviation of (dS/du) is minimized. The least square solution is formulated as

$$\frac{d}{du}\int_0^l \left(\frac{dS}{du} - 1\right)^2 du = 0. \tag{7.83}$$

An analytical solution of the length (l) from this equation is difficult. A reasonable solution can be obtained by considering the midpoint of the chord;

$$\left| \frac{dS}{du}\left(u = \frac{l}{2}\right) \right| = 1. \tag{7.84}$$

By substituting $u = l/2$ and quintic spline coefficients (7.74) into the first derivative of quintic spline (Eq. 7.72), we get

$$\left|\frac{15}{8l_i}(P_{i+1} - P_i) - \frac{7}{16}\left(\frac{dP_{i+1}}{du} + \frac{dP_i}{du}\right) + \frac{1}{32}\left(\frac{d^2P_{i+1}}{du^2} - \frac{d^2P_i}{du^2}\right) l_i\right| = 1. \tag{7.85}$$

Squaring both sides of the equation leads to

$$\begin{aligned}&\left(\tfrac{15}{8l_i}\Delta P\right)^2 + \left(\tfrac{-7}{16}\left(\tfrac{dP_{i+1}}{du} + \tfrac{dP_i}{du}\right)\right)^2 + \left(\tfrac{l_i}{32}\Delta^2 P\right)^2 \\ &\quad - \tfrac{105}{64l_i}\Delta P\left(\tfrac{dP_{i+1}}{du} + \tfrac{dP_i}{du}\right)^2 - \tfrac{7}{256}\Delta^2 P\left(\tfrac{dP_{i+1}}{du} + \tfrac{dP_i}{du}\right) l_i \\ &\quad + \tfrac{15}{128}\Delta P \cdot \Delta^2 P = 1,\end{aligned} \tag{7.86}$$

where $\Delta P = P_{i+1} - P_i$ and $\Delta^2 P = \frac{d^2P_{i+1}}{d^2u} - \frac{d^2P_i}{d^2u}$. Equation (7.86) is rearranged as a fourth-order polynomial,

$$al_i^4 + bl_i^3 + cl_i^2 + dl_i + e = 0, \tag{7.87}$$

where the coefficients are

$$\begin{aligned}
a &= \left(\frac{d^2P_{i+1}}{d^2u} - \frac{d^2P_i}{d^2u}\right)^2, \\
b &= -28\left(\frac{d^2P_{i+1}}{d^2u} - \frac{d^2P_i}{d^2u}\right)\left(\frac{dP_{i+1}}{du} + \frac{dP_i}{du}\right), \\
c &= -1024 + 196\left(\frac{dP_{i+1}}{du} + \frac{dP_i}{du}\right)^2 + 120\left(P_{i+1} - P_i\right)\left(\frac{d^2P_{i+1}}{d^2u} - \frac{d^2P_i}{d^2u}\right), \\
d &= -1680\left(P_{i+1} - P_i\right)\left(\frac{dP_{i+1}}{du} + \frac{dP_i}{du}\right), \\
e &= 3600\left(P_{i+1} - P_i\right)^2.
\end{aligned}$$

Equation (7.87) can be solved by an iteration scheme such as the Newton–Raphson method. Estimation of spline segment length (l_i) by using the near–arc length approximation (Eq. 7.87) minimizes the deviation of the unit tangent vector much better than the linear chord length parameterization; hence it leads to a smoother feed rate and acceleration during quintic spline interpolation.

Spline interpolation is summarized as follows (Fig. 7.12):

Estimate the derivatives at the knots from Eq. (7.82).

Estimate the chord length (l_i) from Eq. (7.87).

Evaluate the quintic spline coefficients from Eq. (7.74) by substituting the chord length (l_i) and derivatives.

Generate incremental motion commands from the quintic spline expression (7.71) by varying

$u = 0, \delta u, 2\delta u, \ldots, k\delta u, \ldots, N\delta u$, where $\delta u = l_i/N$.

Adjust the number of interpolation steps and the time intervals depending on the feed, acceleration, and position feedback measurement resolution as explained in the linear and circular interpolation algorithms.

Chapter 8

Problems

1. Find and sketch the response $y(t)$of a system $G(s)$ for a given input $r(t)$.

Transfer function $G(s)$	Initial condition $y(0)$	Input $r(t)$	Output response $y(t)$ =?
$\frac{3}{s+3}$	10	6	$6+4e^{-3t}$
$\frac{1}{s+10}$	0	$3\cos(10t)$	$0.212\cos(10t-\pi/4)-0.15e^{-10t}$
$\frac{-4s+20}{s+300}$	0	10	$0.67-40.67e^{-300t}$
$\frac{3}{s^2+0.5s+4}$	$y(0)=y^{'}(0)=0$	2	$1.5+1.51e^{-0.25t}\cos(1.98t+173^0)$

2. Find the steady state response (y_{ss}) and steady state error e_{ss} of system $G(s)$ for a given input $r(t)$.

$$y_{ss}=\lim_{s=0} s\left[G(s)\cdot R(s)\right]$$
$$e_{ss}=\lim_{s=0} s\left[1-G(s)\right]R(s)$$

Transfer function $G(s)$	Input $r(t)$	$R(s)$	y_{ss}	e_{ss}
$\frac{3}{s+3}$	6	$\frac{6}{s}$	6	0
$\frac{-4s+20}{s+300}$	10	$\frac{10}{s}$	0.67	9.333
$\frac{3}{s^2+0.5s+4}$	2	$\frac{2}{s}$	1.5	0.5
$\frac{3}{s^2+0.5s+4}$	$2t$	$\frac{2}{s^2}$	∞	∞
$\frac{10}{s^2+3s+10}$	$2t$	$\frac{2}{s^2}$	∞	0.6

MATLAB code to obtain step and ramp response of systems [Source: Stefani et al. "Design of Feedback Control Systems", Oxford University Press, 2002. pp: 196]

$G(s)=\frac{9s+20}{s^3+6s^2+9s+20}$

```
ta=tf([9 20],[1 6 9 20]); % System TF
step(ta)
r1=zpk([], 0, 1)
syst=ta*r1;
step(sys);
```

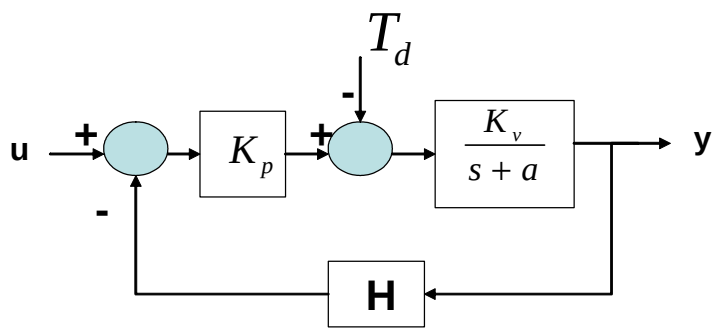
3. Discrete Time Modeling of Systems

Obtain the Zero Order Hold Equivalent of following plant transfer functions with a sampling period of $T=0.01\,\text{sec}$.You can check your results by using MATLAB Command "Gpz = C2D(Gps,Ts,METHOD) where METHOD must be replaced by 'zoh'. Gps is the plant in s domain, and the ZOH equivalent result is Gpz with sampling time Ts. Assume that the D/A converter gain is unity.

- $G_p(s)=0.81\frac{s+0.2}{s+2}$ (Matlab solution: Gps=0.81*tf([1 0.2],[1 2]) ; Gpz=c2d(Gps,0.01,'zoh')

- $G_p(s) = \frac{100}{s^2+14s+100}$
- If the ZOH equivalent of a system is $G_p(z) = \frac{z(z+b)}{z^2+az_1+a_2}$, and controller has a transfer function of $D(z) = K_p + K_I \frac{z}{z-1}$,what is the closed loop transfer function of the system with unity feedback?
- What is the steady state gain and delay of the following system ? Express $y(k)$ in terms of past/present inputs and outputs ? $G(z) = \frac{y(k)}{u(k)} = \frac{b_0 z^2+b_1 z+b_2}{z^3(z^2+a_1 z+a_2)}$

4 Basics of block diagram algebra, transient and frequency response analysis (Figure 8).



Closed loop system with disturbance.

- Find the loop (i.e. open loop) transfer function of the system ?
- Express the closed loop transfer functions $G_{yu}(s) = y(s)/u(s)$ and disturbance transfer function $G_{yd}(s) = y(s)/T_d(s)$?
- Express the response of y for a unit step input of u ?
- Evaluate the frequency response function between the output (y) and input (u)?
- Evaluate the steady state error of the closed loop system for a ramp input ($u(t) = Ut$) with constant disturbance T_d.
- Obtain the Zero Order Hold Equivalent of $G_{yu}(s) = y(s)/u(s)$ when the sampling period is T ?

Tables:

$\mathbf{x(t)}$	1	e^{-at}	t
$\mathbf{X(s)}$	$\frac{1}{s}$	$\frac{1}{s+a}$	$\frac{1}{s^2}$
$X(z)$	$\frac{1}{1-z^{-1}}$	$\frac{1}{1-e^{-aT}z^{-1}}$	$\frac{Tz^{-1}}{(1-z^{-1})^2}$

5. An open loop block diagram of a feed drive system is given in Figure 8.1. The given feed drive system is to be controlled with a Proportional+Derivative (PD) type controller that has a transfer function of $\frac{u(s)}{e(s)} = K_p + K_d s$ where the position tracking error in [count] is $e(s) = x_r(s) - x(s)$, and $u(s)$ is the control signal [V].($K_g = 8.956\ Nm/V, K_m = 50\ rad/s/Nm$,$\tau_m = 1.565s, K_e = 3183$ count/rad.)

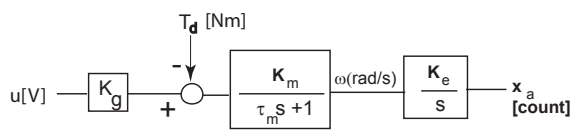


Figure 8.1: Open Loop transfer function of axis drive system

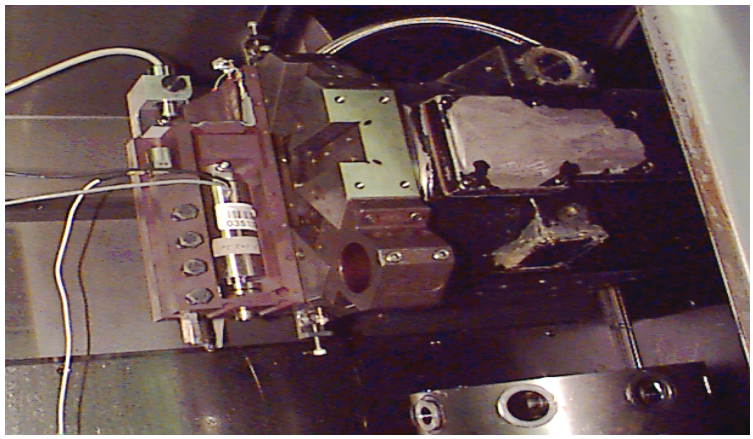
- Draw the closed loop system's block diagram, considering the commanded position $x_r(s)$[count] and the disturbance torque $T_d(s)$[Nm] as the inputs, and the measured table position $x(s)$[count] as the output.
- Obtain the closed loop expression for the measured table position as a function of the commanded position and the disturbance torque in the form $x(s) = G_x(s)x_r(s) + G_d(s)T_d(s)$
- Design the PD controller gains (i.e. $K_p$and K_d) so that the closed loop response to a step position command has a rise time of $t_r = 0.025[s]$ and a maximum overshoot of $M_p = 1$ [%]. Design the controller in Laplace domain.
- Assuming that position control loop is to be closed digitally at a sampling period of $T = 0.001[s]$, obtain the control law (i.e. difference equation) that needs to be implemented in the control computer by transforming the continuous-time PD into the z-domain using Euler (backward difference) approximation ($s \cong \frac{z-1}{Tz}$).
- Sketch the frequency response graph (only magnitude at 3 points, $\omega = 0, \omega_n, 10\omega_n$) of the closed loop disturbance transfer function (i.e. $G_d(s) = x(s)/T_d(s)$).
- Compute the steady state tracking error for a machining operation where a feed motion of $f = 10^5[count/s]$ is commanded against a constant cutting torque of $T_d = 10Nm$ in the opposing direction.

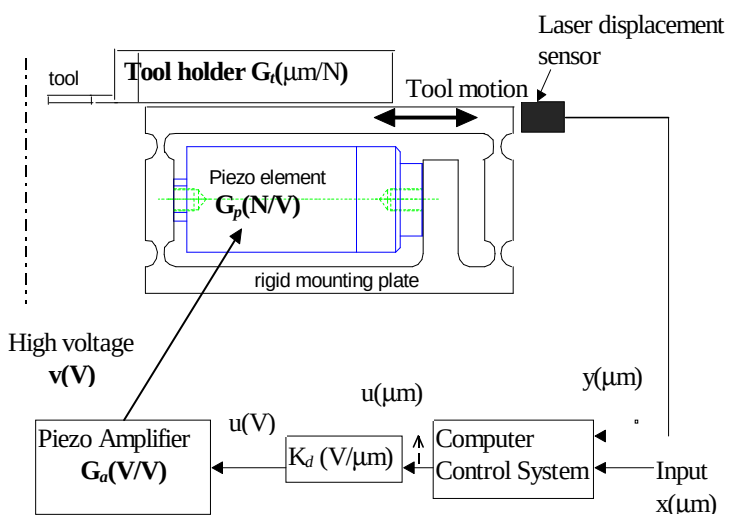
6. A machine table must be driven by a ball screw drive powered by an electric servo motor. The table and maximum workpiece mass is 2000kg. The viscous friction reflected to feed ball screw shaft is $B = 0.02Nm/(rad/s)$. The total static friction originating from the weight, bearings and cutting forces are reflected to the ball screw is estimated to be $10Nm$. The machine has to travel with a maximum feed speed of $0.25m/s$ with a maximum acceleration of $2m/s^2$. The diameter of the ball screw is $50mm$ with a pitch length of $10mm$ and mass of $m_l = 20kg$. A servo motor with a matching amplifier was already selected with the following parameters:

Torque Constant of the motor : $K_t = 0.5Nm/A$. with a maximum current limit of $i_{\max} = 30A$. and continuous current supply of $10A$. The inertia of the servo motor shaft is $0.0026kgm^2$.

- Assume that the motor is directly connected to the ball screw. Calculate reflected inertial to the motor, maximum continuous and peak torque required from the motor-amplifier set?
- Does the motor have sufficient peak and continuous torque rating ? If not, how much gear reduction do you need ? The pinion and gear have negligible inertias.

7. You are asked to design a pole placement control of a precision displacement device which is actuated by a piezo stack shown in Figure 8. The piezo stack receives voltage and delivers force by expanding linearly, which in turn deflects the tool holder assembly through flexures. The combined transfer function of Piezo amplifier and piezo element is approximated just as a constant gain $K_a[N/V] = G_a(s)G_p(s)$.The tool holder assembly, which moves the tool in horizontal (x) direction is modeled as a single mass (m)-damping (c) and spring element (k).system. The displacement is measured with a laser sensor in μm directly, and the system is controlled at T_s[sec] discrete time intervals. Assume that the D/A converter has a gain of K_d, and has one sampling period delay in transmitting the command from the computer to the piezo stack. (Do not use any numerical value until the last sub-section of the question.)





A fast tool servo deevloped by Altintas et al at UBC, 1996.

- – ∗ Obtain the open loop transfer function of the piezo actuator system $G_p =?$ $[\mu m/V]$? Since the sampling period (T_s[sec]) is very small, neglect the zero order hold, and use Euler approximation $s \equiv \frac{z-1}{zT_s}$ to obtain the discrete time (z domain) equivalent of the actuator..
  - ∗ Design a pole placement control system which has a dominant damping ratio of ζ_m and natural frequency of ω_{nm}? Summarize the control law polynomials and the final control command $u(k)$?
  - ∗ Evaluate the steady state position error for a unit step input for x_r?
  - ∗ Use the following parameters, and express the control command $u(k)[V] =?$ at discrete intervals T_s.
    The D/A converter has 12 bits with $\pm 10[V]$ range. The system parameters are given as:

$K_a = 100[N/V]$, $m = 3[kg]$, $c = 0.001E + 06[N/(m/s)]$, $k = 40E + 6[N/m]$, $\zeta_m = 0.8$, $\omega_{nm} = 300[rad/s]$,

8. A model of a lead screw feed drive system with direct table position feedback encoder is shown in Figure 8.2. The motor is powered by the current amplifier and drives the lead screw system which delivers motion to the machine table, which is supported by lubricated guideways. The position of the table is measured using a linear encoder, and this measurement is used by the servo controller for generating the necessary control signals, so that the table closely follows the commanded motion profile.(Linear Encoder: a linear optic scale attached to the table, and it replaces rotary encoder).

No rotary velocity or position feedback is used on the motor. It is assumed that current regulation loop between the amplifier and the motor armature is much faster than the rest of the system dynamics, hence the amplifier is regarded as a static gain that produces motor current proportional to the applied control signal. The electrical winding of the motor can be neglected. The effect of disturbances such as cutting force are to be modeled as equivalent disturbance torque (T_d) acting on the motor shaft.

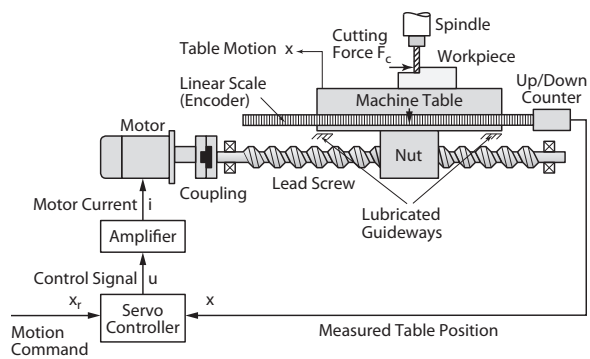


Figure 8.2: Machine tool feed drive control system.

a) Draw the block diagram of the open loop system (i.e. servo controller not connected to the physical system). Express the transfer function of each part in the system in Laplace domain. Use symbolic representation of the system on the block diagram. The input is a signal voltage [V] to the amplifier, and the output is in [counts]. The motor disturbance torque is in [Nm]. Substitute the values of each parameter in the open loop transfer function block diagram using the following parameters of the machine:

Current Amplifier Gain	K_a	4 [A/V]
Motor Torque Gain	K_t	2.239 [Nm/A]
Lead Screw Pitch Length	h_p	20 [mm/rev]
Rotational Inertia of Rotor, Coupling, and Lead Screw	J	0.027 [kgm^2]
Mass of Table and Nut	m_t	225 [kg]
Mass of Workpiece	m_w	200 [kg]
Reflected viscous friction at the motor shaft	B_e	0.020 [Nm/(rad/s)]
Linear Encoder Resolution of Quadrature Decoding		1 [$count/\mu m$]
Digital to Analog Converter Gain (**neglect**)		1

b) Express the open loop transfer function of the system both symbolically and numerically.

9. Open loop block diagram of a feed drive system is given in Figure 8.3.a where $u[V]$ is the voltage command to the current amplifier with a gain of $K_a = 6.5[A/V]$. The current $i[A]$ supplied to the servo motor which has torque constant of $K_t = 0.47[NmA]$. The torque produced by the motor is $T_m[Nm]$, and disturbance torque is given as $T_d[Nm]$.

i) Identify the equivalent inertia ($J[kgm^2]$) and viscous damping ($B[Nm/(rad/s)]$]) from the frequency response function FRF measurement (Ω/u) obtained between the angular velocity $\Omega[rad/s]$ and input voltage ($u[V]$) as shown in Figure 1.a. Neglect the effect of disturbance torque (T_d) in FRF measurement.

ii) An analog PI controller ($K_p + \frac{K_i}{s}$) is used in controlling the velocity of the feed drive as shown in Figure 1.b. A tachogenerator with a gain of T_g is used as a velocity feedback sensor, and $S_g[V/V]$ is the operational amplifier gain. Express the closed loop velocity response $\Omega(s)$ symbolically in terms of velocity command ($V_c(s)$) and disturbance torque ($T_d(s)$), i.e. $\Omega(s) = G_V(s)V_c(s)+G_d(s)T_d(s)$ →where direct transfer function is $G_v(s) =?$ and disturbance transfer function is $G_d(s)$.

iii) Tune the PI controller ($K_p =?, K_i =?$) to achieve a second order characteristic equation with a natural frequency of $\omega_n = 500$[rad/s] and damping ratio of $\zeta = 0.8$.The operational amplifier gain (S_g) needs to be tuned to have static gain of $20[(rad/s)/V]$. Use $T_g = 0.07[V/(rad/s)]$.

iv) If the closed loop servo has a constant input of $V_c[V]$ and disturbed by a constant friction torque $T_d[Nm]$ at the same time, what is the steady state velocity response of the closed loop system? What is the minimum required command voltage ($V_c[V]$) to overcome the constant friction torque ($T_d[Nm]$) before the motor can move the machine?

10. A table is positioned by a linear motor drive as shown in Figure 8.4. The friction in the drives is negligible due to the use of ball bearings, and the disturbance is negligible in the application.

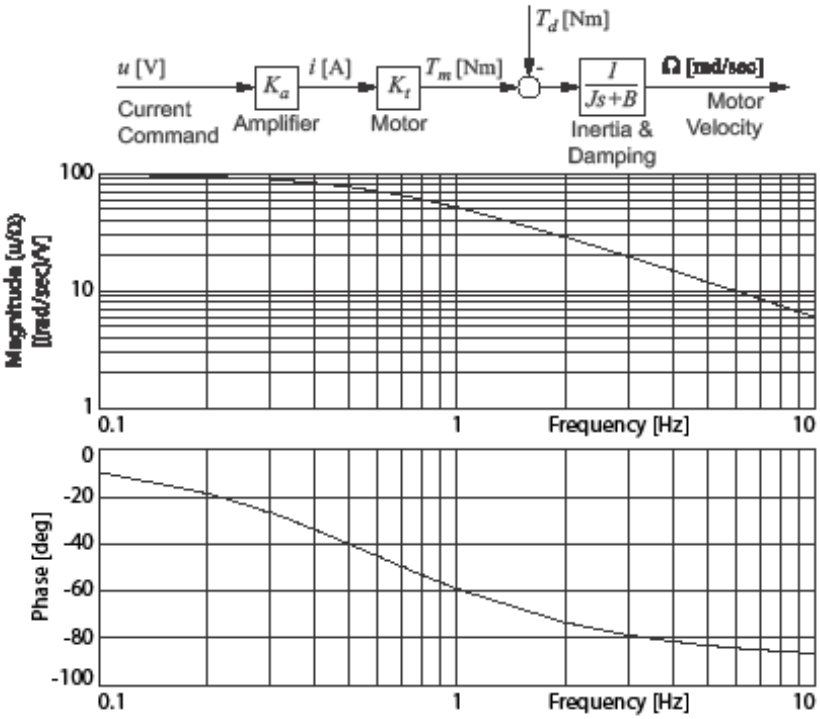
$$G_p(s) = \frac{K_a K_t K_e}{ms^2} = \frac{60}{10s^2} = \frac{6}{s^2}$$

i) Obtain the zero order hold equivalent of the drive $G_p(s)$ using a sampling period of $T_s = 0.001[s]$

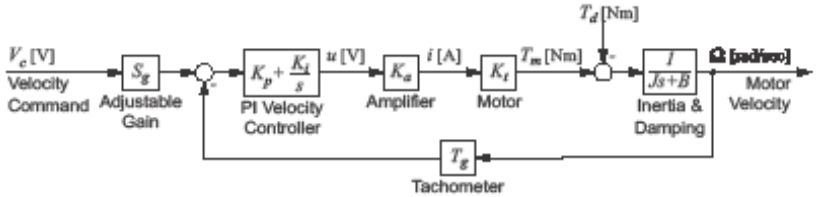
ii) Design a pole placement controller with a damping ratio of $\zeta = 0.8$ and natural frequency of $\omega_m = 100[rad/s]$

11. A CNC interpolator has been designed to generate trapezoidal feedrate profiles by modulating the interpolation period between uniformly spaced position commands. The interpolator has been designed to realize smooth feedrate transitions with acceleration/deceleration magnitudes of $A = D = 10000[mm/s^2]$. The position commands computed are sent to the servo controller running at a sampling period of $T = 0.001s$. Using the provided information, interpret the NC code fragment given below and compute the x-axis trajectory generated at each interpolation step. Initial feed of the machine is zero. Provide your answer in the table at the bottom of this sheet. (1count=1μm)

```
N010 G71 G91 G94          ; metric [mm] mode, relative coordinates, feedrate in [mm/min]
N020 G01 X1.4F6000        ;
```



a) Open Loop Dynamics of Drive and its Frequency Response



b) PI Controlled Velocity Servo

Figure 8.3: PI controller design for a feed drive velocity controller.

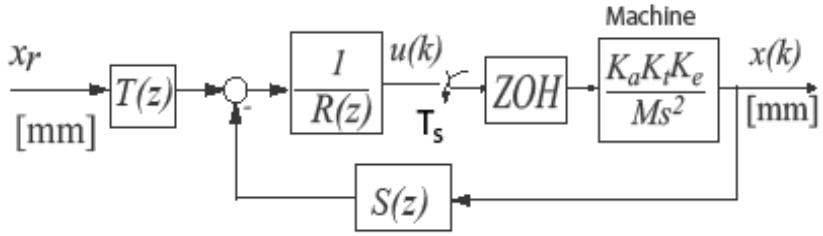


Figure 8.4: Pole placement controller for a linear drive with rigid mass.

k	$\mathbf{x}_r$**[count]**	**f(k)[count/s]**	$\mathbf{T}_i$**[s]**
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			

12. Design a circular interpolator for the following NC program. The units are in microns (μm), and the feed is in microns/minute. The acceleration and deceleration are 3000 $\mu m/s^2$,the minimum interpolation period is $T_{int} = 0.010[s]$.

N010 G02 X15 Y26 I30 J0 F12000 (Assume that the cutter's starting point was at (30,0) before block number N010.)

- Set-up the interpolation algorithm by identifying the parameters.
- Fill the table provided below
- Plot the tool path

Clock pulse (k)	Clock pulse t(sec)	Ti(s) (s)	f($\mu m/s$)	x(k) [μm]	y(k) [μm]
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					

13. A CNC interpolator has been designed to generate trapezoidal feedrate profiles at a constant sampling period of $T_s = 0.01$[sec]. The acceleration and deceleration magnitudes have been set to $A = D = 250mm/\sec^2$ and the desired feedrate is $f = 10[mm/sec]$ with starting and end feedrates of $f_{st} = 0, f_{end} = 0$, respectively

i) Using this information, interpolate a circular trajectory with a radius of $R = 0.5[mm]$ where the starting point is $x_s = y_s = 0$. Calculate the acceleration (T_1), constant feed (T_2) and deceleration (T_3) time lengths of the trajectory; process them with respect to the interpolation period; compute the number of interpolation steps for each section, and interpolate x_r and y_r axis reference commands in each section. Please fill in the table with your results. You need to calculate only one sample per acceleration, constant velocity and deceleration phases. The center of the circle is given by ($x_c = 0.5mm, y_c = 0$) and the feed is in clockwise direction.

ii) Depending on the feedrate (f) and the radius of the circle (R), this circular tool-path is also generating sinusoidal reference commands at different frequencies to x and y axis. Calculate the approximate excitation frequency (Hz) of this circular tool-path (Hint: Do not consider any acceleration/deceleration section and assume that the feedrate is achieved immediately as in a step velocity change).

Bibliography

[1] Ogata, K., Modern Control Engineering, Prentice Hall, 1983

[2] Altintas, Y., Manufacturing Automation, Cambridge University Press, 2000.

[3] Dorf, R.C., Modern Control Systems, Addison Wesley, 1980.

[4] Nise, S. N., Control Systems Engineering, John Wiley & Sons, 2004

[5] Kuo, B., Golnaraghy, F., Automatic Control Systems, Prentice-Hall, 2002.

[6] Richard Bronson, Schaum's Outline of Theory and Problems of Matrix Operations", McGraw Hill, 1989.