

Practical System Dynamics and Control

Daniel "lilatonic" Goldman

February 4, 2019

0 Background Refresher

Chapter 1

Inverting Matrices

1.1 2x2

$$INV \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \frac{1}{ad-bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

Chapter 2

Complex Numbers

2.1 Representations

2.1.1 Coordinate

$$z = x + j y$$

2.1.2 Phasor

$$z = A e^{j \phi}$$

2.2 Conversion

2.2.1 Coordinate to Phasor

$$\begin{cases} x = A \cos \phi \\ y = A \sin \phi \end{cases}$$

2.2.2 Phasor to Coordinate

$$\begin{cases} A = \sqrt{x^2 + y^2} \\ \phi = \arctan(y/x) \end{cases}$$

1 Basics of Modelling

Chapter 3

Open and Closed Loop Control

3.1 Open Loop Control

Open Loop Control systems do not feed the information of the process results back to the controller. That is to say the actual result does not influence the action taken by the controller.

It might seem like this would be a really bad control system. After all, it can't take into account disturbances to the state of the system and has no way of addressing drift of the system. However, if the system is well understood and the controller is calibrated to the specifics of the system, results can be pretty decent.

For example, stepper motors rotate in discrete steps, so a controller can simply count the steps it bade the motor to take. This gives it confidence in the motor's position and velocity despite not having feedback on it. Another example of a partial open loop is a water heater which only heats a reservoir and does not monitor the temperature of the outflow where it is used. If the heat losses from reservoir are well understood, the controller can have a good guess at what the reservoir temperature needs to be for the actual temperature of the outflow to be as desired.

3.2 Closed Loop Control

In contrast, Closed Loop Control systems have a mechanism for feedback. This gives a mechanism to communicate unanticipated error back to the controller. Unanticipated error includes disturbances to the system, system drift, miscalibration, and linearisation errors, among others.

For example, encoders on the joints of a robotic arm would report the actual positions of the links. This would allow the controller to account for the arm

bumping into something, the motors having backlash, the effects of friction, and so on.

Chapter 4

Block Diagrams

4.1 Definition

There are several basic elements in a block diagram:

- Summing Junction
- Branch or Takeoff Junction
- Function Block

4.1.1 MIMO

Some blocks are Multi-Input-Multi-Output (MIMO). These blocks are actually composed of a matrix of functions, with a function for each input-output pair.

For example, a block with I_0 and I_1 inputs and O_0 and O_1 outputs has:

$$\begin{aligned}O_0 &= G_{00}I_0 + G_{01}I_1 \\O_1 &= G_{10}I_0 + G_{11}I_1\end{aligned}$$

4.2 Operations

Many common graphical modifications have equivalent mathematical operations, allowing us to simplify a diagram. Presented here are several of the more common, listed by their graphical operation.

4.2.1 Series Multiplication

Blocks in series can be multiplied together.

4.2.2 Parallel Addition

Blocks in parallel can simply be added

4.2.3 Move Summing Junction Before Function

The reciprocal of the function must be applied to the signal to keep it equivalent.

$$aG + b = G(a + bG^{-1})$$

4.2.4 Move Summing Junction After Function

The Function must be distributed to both signals.

$$(a + b) * G = aG + bG$$

4.2.5 Move Takeoff Junction Before Function

The Function must be distributed to both signals.

4.2.6 Move Takeoff Junction After Function

The reciprocal of the Function must be applied to keep the signal equivalent.

$$a = (G * a) * G^{-1}$$

4.2.7 Eliminate Feedback Loop

A Feedback Loop can be reduced to the combined transfer function:

$$\frac{G}{1 + GH} = \frac{1}{1/G + H} \quad (4.1)$$

These assumes a negative feedback loop.

4.3 Converting Models

The easiest way to convert a physical model to a block diagram is:

1. Build the mathematical model of the system. Consider using FBD or equivalent
2. Convert each operation into its corresponding block
3. Combine blocks to form chains. For example, the second integral of acceleration can use the output from the first integral of acceleration. This step can be done intuitively at the same time as the previous one.

4. Take the Laplace transform
5. Combine and simplify to one Transfer Function

Chapter 5

Signal Flow Graphs

5.1 Definitions

Mixed node: A node with incoming and outgoing branches

5.2 Definition

The most succinct way of describing an SFG is as a Block Diagram but where we view the edges as vertices and the vertices as edges. That is: In a Block Diagram, the nodes of the graph are the functional blocks, and the edges between them are signals. The Block Diagram places emphasis on the functional blocks, with the signals passed between them relegated to second thought. The SFG instead views the signals as the important thing, and the functional blocks are the way to get from one signal to another.

5.3 Operations

5.3.1 Series Multiplication

A Cascade can be eliminated. The new edge is the product of the eliminated cascade.

5.3.2 Parallel Addition

Two edges between the same nodes can be combined. The resulting edge is the sum of the two.

5.3.3 Unzip Y Node

If a node functions as a summing junction, the summing can be moved to after the next edge by distribution of the function of the edge.

5.3.4 Zip Y Node

If a node functions as a summing junction and the previous edges have the same value, the edges can be joined and the summing junction moved previous.

5.3.5 Collapse Y Node

If a node functions as a summing junction and has only one output, the function of the output can be distributed to the input functions, and then edge associated with it thereby eliminated.

5.3.6 Split X Node

If a node has multiple inputs and multiple outputs, the node can be replicated with each replicate having one of the outputs.

5.3.7 Collapse 1-Loop to Self-Loop

A pair of vertexes with only a forward edge and a feedback edge between them can be collapsed to a single vertex. The procedure is to view the first vertex as a Y Node with the feedback edge as one of its inputs. Collapse Y Node is then used. This will create a Self-Loop.

5.3.8 Eliminate Self-Loop

A node with an input a and a feedback output to itself c can be reduced to a single-input node. The input edge then has the value:

$$\frac{a}{1 - c}$$

5.3.9 Eliminate Loop

As a prepared procedure, a 1-Loop can be quickly eliminated in one action: for input a , forward b , and feedback c , the loop node and edges can be eliminated. In their place, the input edge now has the value:

$$\frac{ab}{1 - bc} = \frac{a}{1/b - c}$$

5.4 Converting Block Diagrams

5.5 Drawing from Equations

Usually, you won't have to do this. If you do, here are a few tricks:

- Just do it. The equations will describe exactly what needs to get done.

- If things get busy, consider beginning by making a separate diagram for each equation. If you put the nodes in the same places in each graph, combining the graphs will be straightforward

5.6 References

- Modern Control Engineering, Ogata. 4th edition. It seems that this information is not present in the 3rd or 5th editions of the textbook. Go figure. You can check out pages 106-112 for the information.
- The Wikibook on the subject has a nice worked-through example. Give it a look: https://en.wikibooks.org/wiki/Control_Systems/Signal_Flow_Diagrams#Signal-flow_graphs

Chapter 6

Canonical Form of 2nd Order Equations

6.1 Definition

The Canonical Form is the form where the parameters have some relation to the behaviour of the system which society has deemed useful.

For a system defined as

$$\frac{K}{J * s^2 + B * s + K} = \frac{\frac{K}{J}}{s^2 + \frac{B}{J}s + \frac{K}{J}} \quad (6.1)$$

the canonical form is given as

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (6.2)$$

which leads naturally to the following mapping

$$\omega_n = \sqrt{\frac{K}{J}} \quad (6.3)$$

$$\zeta = \frac{B}{2\sqrt{JK}} \quad (6.4)$$

6.2 Element Definitions

DEF ω_n : The natural frequency

DEF ω_d : $\omega_d = \omega_n \sqrt{1 - \zeta^2}$

DEF ζ : The damping coefficient. Indicates the strength of the damping

of the system.

$$\begin{cases} \zeta \in (0, 1) & \textit{subcriticallydamped} \\ \zeta = 1 & \textit{criticallydamped} \\ \zeta > 1 & \textit{supercriticallydamped} \end{cases}$$

$$\boxed{\text{DEF}} \quad \sigma : \zeta \omega_n$$

6.3 Application

2a Transfer Functions

Chapter 7

Laplace Transforms

7.1 Definition

The mathematical definition of the $L\rightarrow$ is

$$F(s) = L\rightarrow[f(t)] = \int_0^{\infty} [f(t)e^{-st}]dt \quad (7.1)$$

7.2 Useful Description

The $L\rightarrow$ moves functions from the Time Domain to the Frequency Domain. This has the added effect of transforming convolution integrals into simple multiplication. This makes solving the underlying DE much easier.

$L\rightarrow$ are very effective at resolving the Frequency Dynamics of a system. The $\Leftarrow L$ can also be used to turn a solution in the Frequency Domain into one in the Time Domain.

7.3 Neato Math Properties

7.3.1 Linearity

$$L\rightarrow[a \cdot f_1(t) + b \cdot f_2(t)] = L\rightarrow[a \cdot f_1(t)] + L\rightarrow[b \cdot f_2(t)]$$

7.3.2 Differentiation

$$\begin{aligned}L\left[\frac{d}{dt}[f(t)]\right] &= s \cdot F(s) - f(0) \\L\left[\frac{d^2}{dt^2}[f(t)]\right] &= s^2 \cdot F(s) - s \cdot f(0) - \dot{f}(0) \\L\left[\frac{d^n}{dt^n}[f(t)]\right] &= s^n F(s) - \sum_1^n s^{n-i} f^{(i-1)}(0)\end{aligned}$$

7.3.3 Integration

$$L\left[\int_0^t [f(\lambda)]d\lambda\right] = \frac{1}{s}F(s)$$

7.4 Initial and Final Value Theorems

7.4.1 Initial Value Theorem

$$f(0^+) = \lim_{s \rightarrow \infty} [s \cdot F(s)] \quad (7.2)$$

7.4.2 Final Value Theorem

$$f(\infty) = \lim_{s \rightarrow 0} [s \cdot F(s)] \quad (7.3)$$

Chapter 8

Partial Fraction Decomposition

8.1 Definition

8.2 Special Cases

8.2.1 Complex Conjugate

With a second-order or higher denominator, it is possible that the denominator has complex factors. One solution is to simply accept that the roots will be complex and use complex conjugates:

$$\frac{U}{s^2 + 2s + 5} = \frac{A}{s + 1 - 2j} + \frac{B}{s + 1 + 2j}$$

Alternatively, you can leave it uncomplexified, and simply as a second order term.

$$\frac{U}{s^2 + 1} = \frac{C_1 s + C_0}{s^2 + 1}$$

Doing this makes it easy to rearrange it into a form amenable for sin or cos.

$$\frac{C_1 s + C_0}{s^2 + 2s + 5} = \frac{C_1 s}{(s + 1)^2 + 2^2} + \frac{C_0}{(s + 1)^2 + 2^2}$$

Or you can just use the table of common \mathcal{L} to not have to do the rearranging.

8.2.2 Repeated Factor

If a repeated factor appears in the denominator, the partial fraction decomposition needs to contain increasing powers of the factor (up to the power of the root).

$$\frac{U}{(s + 1)^3} = \frac{A}{(s + 1)} + \frac{B}{(s + 1)^2} + \frac{C}{(s + 1)^3}$$

Note that instead of that, you can have a numerator of higher order.

$$\frac{U}{s(s^2 + 1)} = \frac{As + B}{s^2 + 1} + \frac{C}{s}$$

8.2.3 Improper Fraction

A fraction is improper when the degree of the numerator is not less than the degree of the denominator. The degree of mismatch is the difference between the former and the latter.

In the case of an improper fraction, a polynomial of the degree of mismatch must be added to the PDF side.

$$\begin{aligned}\frac{(s+1)^2}{(s-2)^2} &= \frac{A}{(s-2)^2} + \frac{B}{(s-2)} + \textcolor{blue}{C} \\ &= \frac{9}{(s-2)^2} + \frac{6}{(s-2)} + \textcolor{blue}{1}\end{aligned}$$

Chapter 9

Inverse Laplace Transform

9.1 Definition

The mathematical definition of the \mathcal{L}^{-1} is

$$f(t) = \mathcal{L}^{-1}[F(s)] = \frac{1}{2\pi j} \int_{c-j\infty}^{c+j\infty} [F(s)e^{st}] dt \quad (9.1)$$

Hopefully you will never have to use it.

9.2 Simple Inverse Transforms

Following is a table of simple inverse transforms. These are useful for general exercises.

9.3 Common Inverse Transforms

Following is a table of inverse transforms which are tailored to situations which you will commonly see. This will minimise the amount of work which actually needs to get done.

F	f
$\frac{1}{s}$	$\tilde{r}u(t)$
$\frac{1}{s+a}$	e^{-at}
$\frac{\omega}{s^2+\omega^2}$	$\sin(\omega t)$
$\frac{s}{s^2+\omega^2}$	$\cos(\omega t)$
$\frac{n!}{s^{n+1}}$	t^n
$\frac{1}{(s+a)^2}$	$t * e^{-at}$
$\frac{1}{s}$	δ
$\frac{\omega}{(s+a)^2+\omega^2}$	$e^{-at} * \sin(\omega t)$
$\frac{(s+a)}{(s+a)^2+\omega^2}$	$e^{-at} * \cos(\omega t)$
$\frac{1}{(1+a_1 s)(1+a_2 s)}$	$\frac{e^{-s/a_1} - e^{-s/a_2}}{a_1 - a_2}$
$\frac{s+b}{(s+a)^2+\omega^2}$	$\frac{1}{\omega} \sqrt{(b-a)^2 + \omega^2} e^{-at} * \sin(\omega t + \arctan(\frac{\omega}{b-a}))$
$\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$	$\frac{\omega_n^2}{\omega_d} e^{-\zeta\omega_n t} \sin(\omega_d t) \text{ if } \zeta < 1$
$\frac{1}{s((s+a)^2+\omega^2)}$	$\frac{1}{a^2+\omega^2} + \frac{1}{\omega\sqrt{a^2+\omega^2}} e^{-at} \sin(\omega t - \arctan(\frac{\omega}{-a}))$
$\frac{\omega_n^2}{s((s^2+2\zeta\omega_n s+\omega_n^2))}$	$1 - \frac{\omega_n}{\omega_d} e^{-\zeta\omega_n t} \sin(\omega_d t + \arccos(\zeta))$

Chapter 10

Transfer Functions

10.1 Definition

The transfer function is defined as:

$$G(s) = \frac{L\{Output\}}{L\{Input\}} \quad (10.1)$$

For a system defined as $\{a_n * D_n[y]\} = \{b_m * D_m[x]\}$ where y is the output and x is the input, the transfer function is given as:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{\{a_n * s^n\}}{\{b_m * s^m\}} \quad (10.2)$$

Note that the transfer function is not unique; there are many systems which reduce to the same transfer function. Note also that the transfer function is not tied to the input function. The transfer function does not provide information about the physical system underlying the model, and many different systems with different physical quantities being studied can have the same transfer function. The transfer function is limited to linear, time-invariant differential equations.

10.2 TF of a Closed-Loop System

For a general closed-loop system, there are several transfer functions of interest.

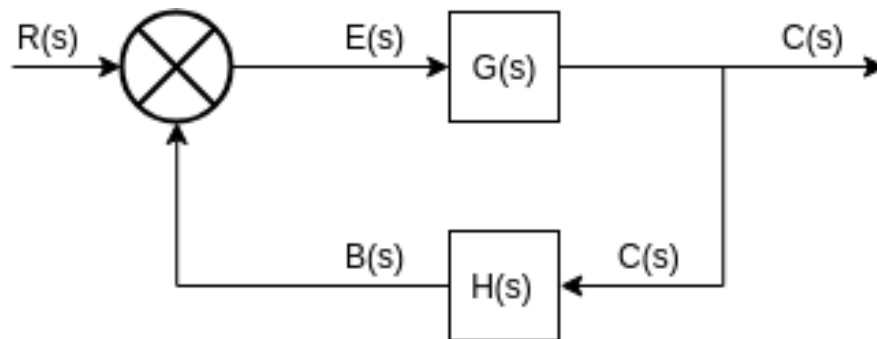


Figure 10.1: General Closed-Loop system block diagram

- | | | |
|-----|----------------|---|
| DEF | Closed-Loop : | $\frac{C(s)}{R(s)} = \frac{G(s)}{1+G(s)H(s)}$ |
| DEF | Feed-Forward : | $\frac{C(s)}{E(s)} = G(s)$ |
| DEF | Error Ratio : | $\frac{E(s)}{R(s)} = \frac{1}{1-G(s)H(s)}$ |

Chapter 11

Mason's Gain Rule

11.1 Definition

Given a SFG, Mason's gain rule provides a catchall formula for evaluating the gain of a system. It uses a generalised form of the SFG simplifications to allow you to replace what would have taken a little ingenuity with a lot of accounting.

The amount of accounting rises little with complexity, but the amount of ingenuity rises dramatically, so occasionally it's useful.

The total gain between node i and j is given as:

$$T_{ij} = \frac{\sum_k P_k \Delta_k}{\Delta_{\bullet}} \quad (11.1)$$

where

$P_k = \text{transmission of the } k^{\text{th}} \text{ path}$

$\Delta_k = \text{the cofactor of the } k^{\text{th}} \text{ path}$

$\Delta_{\bullet} = \text{the graph determinant}$

Which is more helpful as:

$\Delta_k = \text{the cofactor of the } k^{\text{th}} \text{ path}$

$$\Delta_{\bullet} = 1 - \sum_i L_i + \sum_{i,j} L'_i L'_j - \sum_{i,j,k} L''_i L''_j L''_k + \dots$$

with $L'_i L'_j$ being the product of any two loops that do not have a node or branch in common, and so on for the ones with more loops.

11.2 Examples

Chapter 12

State Space to Transfer Function

12.1 Definition

The TF can be derived simply with the application of the formula

$$G(s) = \frac{Y}{U} = (C \cdot \text{INV}[s\vec{I} - A] \cdot B + D) \quad (12.1)$$

12.2 Derivation

With

$$\dot{X} = AX + BU \quad (12.2)$$

$$Y = CX + DU \quad (12.3)$$

The $L\rightarrow$ gives

$$L\rightarrow X(s) = sX(s) = AX + BU(s) \quad (12.4)$$

$$L\rightarrow Y(s) = Y(s) = CX(s) + DU(s) \quad (12.5)$$

Rearranging gives

$$(s\vec{I} - A)X(s) = BU(s) \quad (12.6)$$

$$X(s) = (s\vec{I} - A)BU(s) \quad (12.7)$$

$$(12.8)$$

And substituting $X(s)$ from above

$$Y(s) = (C \cdot \text{INV}[s\vec{I} - A] \cdot B + D)U(s) \quad (12.9)$$

2b State Space Representation

Chapter 13

State Space Representation

13.1 Definition

DEF **State** : The collection of variables which completely defines the present and future of the system

DEF **State Space Equation** : Equations which relate the state variables and input variables to the output variables. These will take the form of a Vector-Matrix Differential Equation.

The equations take the form

$$\dot{\vec{x}} = A\vec{x} + Bu \quad (13.1)$$

$$\vec{y} = C\vec{x} + Du \quad (13.2)$$

The matrices above have names:

DEF **A** : state matrix

DEF **B** : input matrix

DEF **C** : output matrix

DEF **D** : direct transmission matrix

These can be represented as a block diagram:

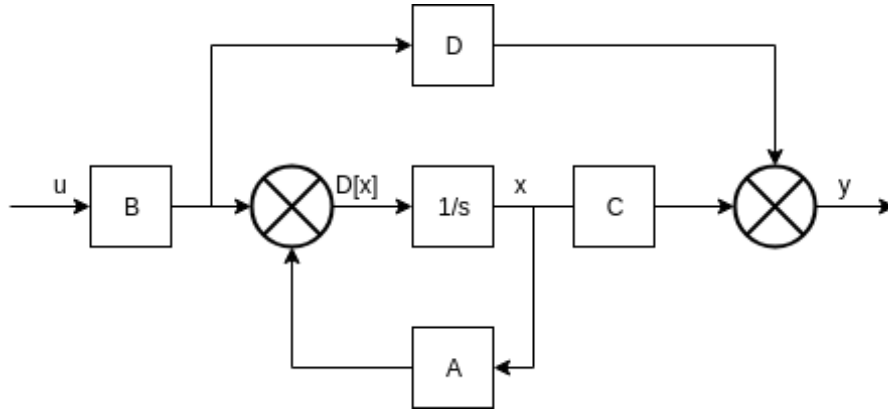


Figure 13.1: General Closed-Loop system block diagram

13.2 Characteristic Equation

The Characteristic Equation is the denominator of the TF and is given by

$$DET[s\vec{1} - A] \quad (13.3)$$

13.3 State Transition Matrix

The STM is given as

$$\Phi(s) = INV(s\vec{1} - A) \quad (13.4)$$

Chapter 14

Block Diagram to State Space

14.1 Steps

14.1.1 Label State Variables

Begin with the labelling of state variables. The easiest way to do this is to start at the last integrator block. Place an x_1 after it. Working backwards, place another x_i after each integrator block.

14.1.2 Label State Variable Derivatives

Simply place the derivative of each state variable before the integrator blocks.

14.1.3 Equate Derivatives with Values

Use the block diagrams to assign values to each of the state variable derivatives in terms of the state variables and the input variables. This is as simple as tracing the lines which feed into the signal for labelled with the state variable derivative until the lines become a state variable or input variable.

14.1.4 Construct Matrices

The A-matrix and B-matrix are straightforward to construct from the equations. The B-matrix pulls all the terms involving the input variables, and the A-matrix is simply the matrixification of the state-space terms.

The C-matrix and D-matrix terms are also easy to construct. Simply express Y as a function of the state and input variables. The D-matrix pulls all the terms involving the input variables, and the C-matrix is simply the matrixification of the state-space terms.

Chapter 15

Solution of State Space Problems

15.1 Zero Input Solution

15.1.1 Definition

For a system defined as

$$\dot{q} = Aq + Bu \quad (15.1)$$

$$y = Cq + Du \quad (15.2)$$

the Zero Input Problem is given as

$$\dot{q} = Aq \quad (15.3)$$

$$y = Cq \quad (15.4)$$

on account of the input being zero and all that.

15.1.2 Solution

We begin by taking the $L\lambda$. We then solve for $Q(s)$. The shortmode gives

$$Q = \Phi \cdot q(0^-) \quad (15.5)$$

The inverse laplace transform gives

$$q = \phi \cdot q(0^-) \quad (15.6)$$

Which can be substituted back into the equation for y to give

$$y(t) = C \cdot q(t) = C\phi \cdot q(0^-) \quad (15.7)$$

15.2 Zero State Solution

15.3 General Solution

15.4 References

<http://lpsa.swarthmore.edu/Transient/TransMethSS.html>

Chapter 16

Principal Coordinates

16.1 Utility

This decouples the equations, which makes manipulating them mathematically much easier.

16.2 Steps

16.2.1 Find Eigenvalues and Eigenvectors

Letting the eigenvalues be λ_n , the matrix of corresponding eigenvectors be $E = [e_1, \dots, e_n]$, and the matrix of $DIAG[\lambda_i] = \Lambda$

16.2.2 Transform

$$\tilde{A} = E^{-1}AE = \Lambda \quad (16.1)$$

$$\tilde{B} = E^{-1}B \quad (16.2)$$

$$\tilde{C} = CE \quad (16.3)$$

$$\tilde{D} = D \quad (16.4)$$

Chapter 17

Transfer Function to State Space

17.1 Strictly Proper

17.1.1 The long way

The first thing to do is to convert from the TF to a DE.

This gives $\ddot{y} + a_1\dot{y} + a_2y = b_0u$, from which we can define

$$\begin{aligned}q_1 &= y \\q_2 &= \dot{y} \\q_3 &= \ddot{y}\end{aligned}$$

Then we get

$$\begin{aligned}\dot{q}_1 &= \dot{y} = q_2 \\ \dot{q}_2 &= \ddot{y} = q_3 \\ \dot{q}_3 &= \dddot{y} = b_0u - a_1\ddot{y} - a_2\dot{y} - a_3y = b_0u - a_1q_3 - a_2q_2 - a_3q_1\end{aligned}$$

We can then represent this in a matrix

$$\dot{q} = Aq + Bu = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} q + \begin{bmatrix} 0 \\ 0 \\ b_0 \end{bmatrix} u$$

Also, the corresponding equations for y are simply:

$$y = C\vec{q} + D\vec{u} = [1, 0, 0]\vec{q} + 0u$$

17.1.2 The short way

For a TF defined as

$$G(s) = \frac{\sum_{i=1}^k n_i s^{k-i}}{\sum_{i=0}^k d_i s^{k-i}}$$

Specifically, as $G(s) = \frac{n_1 s^3 + n_2 s^2 + n_3 s + n_4}{s^4 + d_1 s^3 + d_2 s^2 + d_3 s + d_4}$ simply build the system with

$$\dot{q} = Aq + Bu = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -d_4 & -d_3 & -d_2 & -d_1 \end{bmatrix} q + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u$$

and

$$y = C\vec{q} + D\vec{u} = [n_4, n_3, n_2, n_1]\vec{q}$$

Note that this gives the Controllable Canonical Form. Transposing the A matrix and interchanging the B and C matrices gives the Observable Canonical Form.

17.2 Proper

17.3 Improper

3 Time Domain Performance Characteristics

Chapter 18

Free and Forced Response

18.1 Definition

DEF **Free Response** : The free response of a system is the response to the initial conditions, with no forcing functions.

DEF **Forced Response** : The forced response of a system is the response to the forcing function with 0 initial conditions.

18.2 From Differential Equation

Given a DE with initial conditions

$$\ddot{y} + 2\dot{y} + 2y = u \quad (18.1)$$

$$(s^2 + 2s + 2)Y = U + sy(0) + \dot{y}(0) + 2y(0) \quad (18.2)$$

$$Y = \frac{U}{(s^2 + 2s + 2)} + \frac{\dot{y}(0)}{(s^2 + 2s + 2)} + \frac{(s + 2)y(0)}{(s^2 + 2s + 2)} \quad (18.3)$$

$$Y = \text{ForcedResponse} + \text{FreeResponse} \quad (18.4)$$

Chapter 19

Transient Response Characteristics

19.1 Definitions

All of these refer in general to the Unit Step Response.

DEF **rise time** $\approx \frac{1.8}{\omega_n}$: time from a percentage from the initial values to the 1st time encountering a percentage from the final value. Most commonly, it is from 0.1 of initial to 0.9 of final.

DEF **peak time** $\approx \frac{\pi}{\omega_d}$: time to hit the highest value. More generally, the time until the first peak.

DEF **settling time** $\approx \frac{4}{\zeta\omega_n}$ (*for 0.02*) : time before the system has converged to within a specified percentage of the final value.

DEF **overshoot** $\approx \exp[\frac{-\pi\zeta}{\sqrt{1-\zeta^2}}]$: ratio of the height of the 1st peak above the final value to the final value.

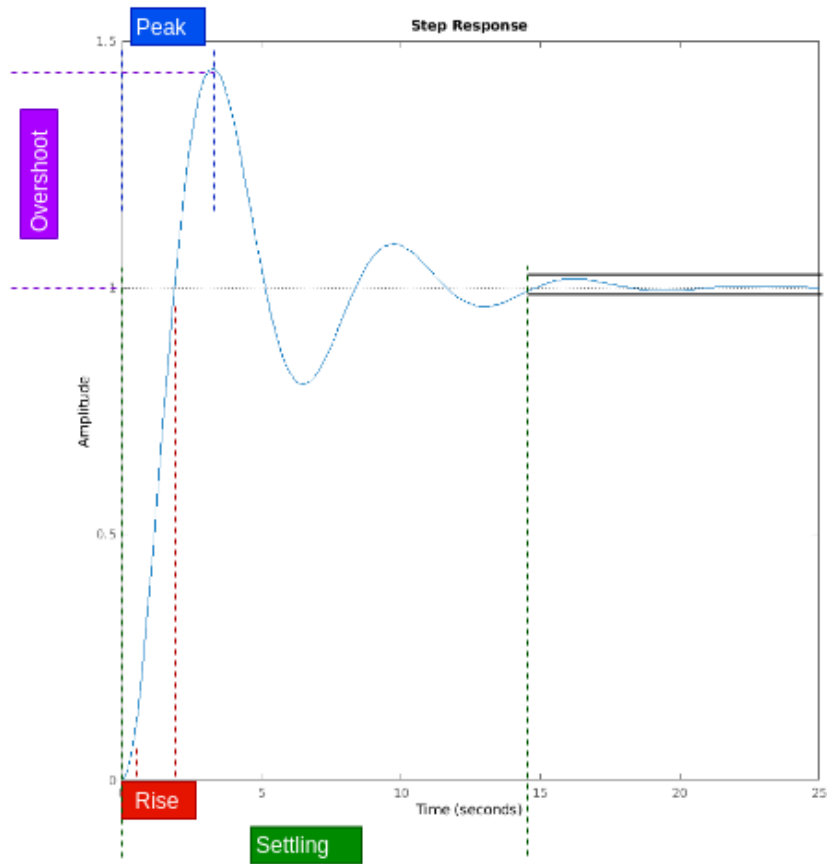


Figure 19.1: Time Response Characteristics on a sample function

19.2 Designing for Characteristics

19.2.1 Overshoot

$$\zeta = \sqrt{\frac{(\ln M_p)^2}{\pi^2 + (\ln M_p)^2}}$$

Chapter 20

The Complex Plane

20.1 Defintion

The complex plain is the plane formed by the Real and Complex number lines.

It is useful when the roots and poles of a Transfer Function are plotted on it. It becomes a graphical tool which allows us to estimate the performance of a system.

20.1.1 Reading Parameters

The (x, y) coordinates of a root correspond to the values (σ, ω_d) for that root.

The distance of the root to the origin corresponds to ω_n for that root.

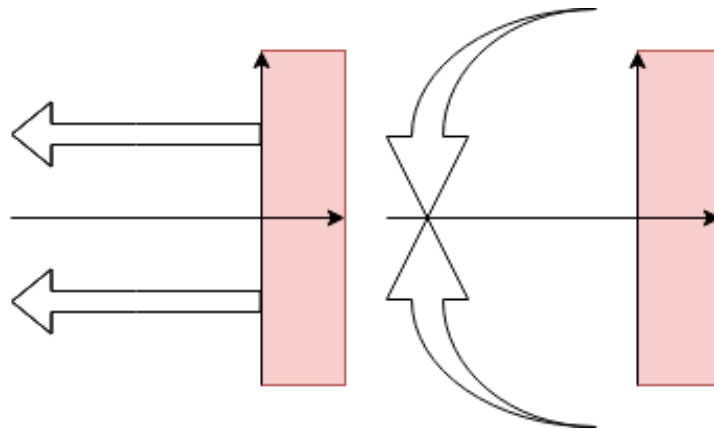
With the angle formed between the negative real number line and the root defined as α , $\cos \alpha = \zeta$.

20.2 Expressing Requirements on the Plane

Simple requirements can be expressed as follows:

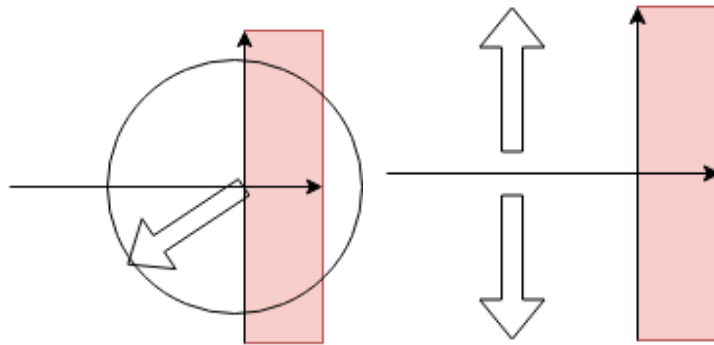
Simply identify the value which makes the the condition true, and then block off the area below it on the graph.

Composite requirements can be made by superimposing simple requirements.



(a) Quicker Convergence

(b) Less Overshoot



(c) Shorter Rise (radial)

(d) Shorter Peak