Practical System Dynamics and Control

Daniel "lilatomic" Goldman

February 2, 2019

0 Background Refresher

Chapter 1

Inverting Matrices

1.1 2x2

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

1 Basics of Modelling

Chapter 2

Open and Closed Loop Control

2.1 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.

2.2 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 3

Block Diagrams

3.1 Definition

There are several basic elements in a block diagram:

- Summing Junction
- Branch or Takeoff Junction
- Function Block

3.1.1 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:

$$O_0 = G_{00}I_0 + G_{01}I_1$$
$$O_1 = G_{10}I_0 + G_{11}I_1$$

3.2 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.

3.2.1 Series Multiplication

Blocks in series can be multiplied together.

3.2.2 Parallel Addition

Blocks in parallel can simply be added

3.2.3 Move Summing Junction Before Function

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

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

3.2.4 Move Summing Junction After Function

The Function must be distributed to both signals.

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

3.2.5 Move Takeoff Junction Before Function

The Function must be distributed to both signals.

3.2.6 Move Takeoff Junction After Function

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

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

3.2.7 Eliminate Feedback Loop

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

$$\frac{G}{1 + GH} = \frac{1}{1/G + H} \tag{3.1}$$

These assumes a negative feedback loop.

3.3 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 4

Signal Flow Graphs

4.1 Definitions

Mixed node: A node with incoming and outgoing branches

4.2 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.

4.3 Operations

4.3.1 Series Multiplication

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

4.3.2 Parallel Addition

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

4.3.3 Unzip Y Node

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

4.3.4 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.

4.3.5 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.

4.3.6 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.

4.3.7 Collapse 1-Loop to Self-Loop

An 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.

4.3.8 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:

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

4.3.9 Eliminate Loop

As a prepared procedue, 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:

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

4.4 Converting Block Diagrams

4.5 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

4.6 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

2a Transfer Functions

Chapter 5

Laplace Transforms

5.1 Definition

The mathematical definition of the $\not L\!\!\rightarrow$ is

$$F(s) = \not L\!\!\rightarrow[f(t)] = \int_0^{\infty} [f(t)e^{-st}]dt \tag{5.1}$$

5.2 Useful Description

The $\not L\!\!\rightarrow$ 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.

$\not L\!\!\rightarrow$ are very effective at resolving the Frequency Dynamics of a system. The $\leftarrow\!\!\!\not L$ can also be used to turn a solution in the Frequency Domain into one in the Time Domain.

5.3 Neato Math Properties

5.3.1 Linearity

$$\not L\!\!\rightarrow[a \cdot f_1(t) + b \cdot f_2(t)] = \not L\!\!\rightarrow[a \cdot f_1(t)] + \not L\!\!\rightarrow[b \cdot f_2(t)]$$

5.3.2 Differentiation

$$\not L\!\!\rightarrow[\frac{d}{dt}[f(t)]] = s \cdot F(s) - f(0)$$

$$\not L\!\!\rightarrow[\frac{d^2}{dt^2}[f(t)]] = s^2 \cdot F(s) - s \cdot f(0) - \dot{f}(0)$$

$$\not L\!\!\rightarrow[\frac{d^n}{dt^n}[f(t)]] = s^n F(s) - \sum_1^n s^{n-i} f'^{i-1}$$

5.3.3 Integration

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

5.4 Initial and Final Value Theorems

5.4.1 Initial Value Theorem

$$f(0^+) = \lim_{s\to\infty}[s \cdot F(s)] \tag{5.2}$$

5.4.2 Final Value Theorem

$$f(\infty) = \lim_{s\to 0}[s \cdot F(s)] \tag{5.3}$$

Chapter 6

Partial Fraction Decomposition

6.1 Definition

6.2 Special Cases

6.2.1 Repeated Factor

6.2.2 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.

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

Chapter 7

Inverse Laplace Transform

7.1 Definition

The mathematical definition of the $\overleftarrow{\mathcal{L}}$ is

$$f(t) = \overleftarrow{\mathcal{L}}\,[F(s)] = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty}[F(s)e^{st}]dt \tag{7.1}$$

Hopefully you will never have to use it.

7.2 Simple Inverse Transforms

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

7.3 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
$\frac{1}{s}$	$u(t)$
$\frac{1}{s+a}$	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
$\frac{1}{(s+a)^2}$	$t * e^{-at}$
1	δ
$\frac{\omega}{(s+a)^2+\omega^2}$	$e^{-at} * \sin(\omega t)$
$\frac{(s+a)}{(s+a)^2+\omega^2}$	$e^{-at} * \cos(\omega t)$
$\frac{1}{(1+a_1 s)(1+a_2 s)}$	$\frac{e^{-s/a_1}-e^{-s/a_2}}{a_1-a_2}$
$\frac{s+b}{(s+a)^2+\omega^2}$	$\frac{1}{\omega}\sqrt{(b-a)^2+\omega^2 e^{-at}} * \sin(\omega t+\arctan(\frac{\omega}{b-a}))$
$\frac{\omega_n^2}{s^2+2\zeta\omega_n s+\omega_n^2}$	$\frac{\omega_n^2}{\omega_d} e^{-\zeta\omega_n t}\sin(\omega_d t) if \zeta < 1$
$\frac{1}{s((s+a)^2+w^2)}$	$\frac{1}{a^2+\omega^2}+\frac{1}{\omega\sqrt{a^2+\omega^2}} e^{-at} sin(\omega t-\arctan(\frac{\omega}{-a}))$
$\frac{\omega_n^2}{s((s^2+2\zeta\omega_n s+\omega_n^2)}$	$1-\frac{\omega_n}{\omega_d} e^{-\zeta\omega_n t}\sin(\omega_d t+\arccos(\zeta))$

Chapter 8

Transfer Functions

8.1 Definition

The transfer function is defined as:

$$G(s) = \frac{\mathcal{L}[Output]}{\mathcal{L}[Input]} \tag{8.1}$$

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:

$$G(s) = \frac{Y(s)}{X(s)} = \frac{\{a_n * s^n\}}{\{b_m * s^m\}} \tag{8.2}$$

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.

Chapter 9

Mason's Gain Rule

9.1 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 take 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:

$$T_{ij} = \frac{\sum_k P_k \Delta_k}{\Delta_\bullet} \tag{9.1}$$

where

$$\begin{aligned} P_k &= transission\ of\ the\ k^{th}\ path \\ \Delta_k &= the\ cofactor\ of\ the\ k^{th}\ path \\ \Delta_\bullet &= the\ graph\ determinant \end{aligned}$$

Which is more helpful as:

$$\begin{aligned} \Delta_k &= the\ cofactor\ of\ the\ k^{th}\ path \\ \Delta_\bullet &= 1 - \sum_i L_i + \sum_{i,j} L'_i L'_j - \sum_{i,j,k} L''_i L''_j L''_k + ... \end{aligned}$$

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.

9.2 Examples

Chapter 10

State Space to Transfer Function

10.1 Definition

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

$$G(s) = \frac{Y}{U} = (C \cdot INV[s\vec{1} - A] \cdot B + D) \tag{10.1}$$

10.2 Derivation

With

$$\dot{X} = AX + BU \tag{10.2}$$
$$Y = CX + DU \tag{10.3}$$

The $L\!\!\to$ gives

$$L\!\!\to X(s) = sX(s) = AX + BU(s) \tag{10.4}$$
$$L\!\!\to Y(s) = \ Y(s) = CX(s) + DU(s) \tag{10.5}$$

Rearanging gives

$$(s\vec{1} - A)X(s) = BU(s) \tag{10.6}$$
$$X(s) = (s\vec{1} - A)BU(s) \tag{10.7}$$
$$\tag{10.8}$$

And substituting $X(s)$ from above

$$Y(s) = (C \cdot INV[s\vec{1} - A] \cdot B + D)U(s) \tag{10.9}$$

2b State Space Representation

Chapter 11

State Space Representation

11.1 Definition

11.2 Characteristic Equation

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

$$DET[s\vec{1} - A] \tag{11.1}$$

11.3 State Transition Matrix

The STM is given as

$$\Phi(s) = INV(s\vec{1} - A) \tag{11.2}$$

Chapter 12

Block Diagram to State Space

12.1 Steps

12.1.1 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.

12.1.2 Label State Variable Derivatives

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

12.1.3 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.

12.1.4 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 13

Solution of State Space Problems

13.1 Zero Input Solution

13.1.1 Definition

For a system defined as

$$\dot{q} = Aq + Bu \tag{13.1}$$
$$y = Cq + Du \tag{13.2}$$

the Zero Input Problem is given as

$$\dot{q} = Aq \tag{13.3}$$
$$y = Cq \tag{13.4}$$

on account of the input being zero and all that.

13.1.2 Solution

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

$$Q = \Phi \cdot q(0^-) \tag{13.5}$$

The inverse laplace transform gives

$$q = \phi \cdot q(0^-) \tag{13.6}$$

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

$$y(t) = C \cdot q(t) = C\phi \cdot q(0^-) \tag{13.7}$$

13.2 Zero State Solution

13.3 General Solution

13.4 References

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

Chapter 14

Principal Coordinates

14.1 Utility

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

14.2 Steps

14.2.1 Find Eigenvalues and Eigenvectors

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

14.2.2 Transform

$$\tilde{A} = E^{-1}AE = \Lambda \tag{14.1}$$

$$\tilde{B} = E^{-1}B \tag{14.2}$$

$$\tilde{C} = CE \tag{14.3}$$

$$\tilde{D} = D \tag{14.4}$$

Chapter 15

Transfer Function to State Space

15.1 Strictly Proper

15.1.1 The long way

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

This gives $\dddot{y} + a_1\ddot{y} + a_2\dot{y} + a_3 y = b_0 u$, from which we can define

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

Then we get

$$\begin{aligned} \dot{q_1} &= \dot{y} &&= q_2 \\ \dot{q_2} &= \ddot{y} &&= q_3 \\ \dot{q_3} &= \dddot{y} &&= b_0 u - a_1\ddot{y} - a_2\dot{y} - a_3 y = b_0 u - a_1 q_3 - a_2 q_2 - a_3 q_1 \end{aligned}$$

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:

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

15.1.2 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 Cannonical Form. Transposing the A matrix and interchanging the B and C matrices gives the Observable Cannonical Form.

15.2 Proper

15.3 Improper