

# **Notes on Dynamics and Control Systems**

**Rizwan Ahmed**

# Contents

<b>Table of Contents</b>	<b>I</b>
<b>I Newtonian Mechanics</b>	<b>1</b>
<b>1 Newtons Laws and Kinematics</b>	<b>3</b>
1.1 Method for solving Dynamics Problem . . . . .	3
1.2 Derivative Rule from vector geometry . . . . .	3
1.2.1 Framing derivative rule for unit vectors . . . . .	5
1.3 Velocity and Acceleration in moving reference frames . . . . .	7
1.3.1 Velocity of a vector in a moving frame . . . . .	7
1.3.2 Acceleration of a vector in a moving frame . . . . .	9
1.4 Velocity and acceleration w.r.t moving frame in polar coordinates . . . . .	10
1.4.1 Velocity in polar coordinates . . . . .	11
1.4.2 Acceleration in polar coordinates . . . . .	13
1.5 Rotation Problem 1 . . . . .	13
1.5.1 Explaining Coriolis force using Law of conservation of angular momentum . . . . .	15
1.6 Rotation Problem 2 . . . . .	15
1.7 Rotation Problem 3 . . . . .	16
1.8 Derivative on angular velocity $\omega$ . . . . .	17
<b>2 Impulse, Torque and Angular Momentum for a system of particles</b>	<b>21</b>
2.1 Linear Impulse and Momentum . . . . .	21
2.1.1 Problem on Impulse - Momentum . . . . .	21
2.2 Momentum for system of particles . . . . .	22
2.2.1 Problem on momentum on system of particles . . . . .	23
2.3 Angular Momentum of a particle . . . . .	23
2.3.1 Angular momentum and torque problem for a particle . . . . .	26
2.3.1.1 Angular momentum . . . . .	26
2.3.1.2 Torque . . . . .	26
2.4 Angular momentum and Torque for rigid bodies . . . . .	27
2.4.1 Torque and angular momentum Problem 2 . . . . .	27
2.5 Exercise Problem . . . . .	30
2.5.1 Chose coordinates . . . . .	30

2.5.2	DOF . . . . .	31
2.5.3	Deriving EOM . . . . .	31
<b>3</b>	<b>Fictitious Forces using Free Body Diagrams</b>	<b>33</b>
<b>4</b>	<b>Deriving Equations of Motion</b>	<b>37</b>
4.1	Method for deriving EOMs . . . . .	37
4.2	Exercise 1 . . . . .	37
4.2.0.1	Assigning coordinates . . . . .	37
4.2.1	DOF . . . . .	37
4.2.2	Determine angular momentum of the system . . . . .	38
4.2.3	External Forces and Moments . . . . .	38
4.2.4	Complete EOM . . . . .	40
4.3	Exercise 2 - self solved . . . . .	40
4.3.1	Choosing coordinates . . . . .	40
4.3.2	DOF . . . . .	40
4.3.3	Determining angular momentum . . . . .	41
4.3.4	Determining external moments . . . . .	41
4.3.5	Finishing EOM . . . . .	41
4.4	Exercise 3 . . . . .	41
4.4.1	Choosing coordinates . . . . .	42
4.4.2	DOF . . . . .	42
4.4.3	Determining external forces on the system . . . . .	42
4.4.4	Determining accelerations along x . . . . .	43
4.4.5	Finishing EOM . . . . .	43
4.5	Exercise 4 . . . . .	44
4.5.1	Coordinates . . . . .	44
4.5.2	DOF . . . . .	44
4.5.3	External forces . . . . .	45
4.5.4	External moments . . . . .	45
4.5.5	Finishing the EOMs . . . . .	46
<b>5</b>	<b>Mass Moment of Inertia</b>	<b>47</b>
5.1	Principal axes . . . . .	49
5.2	Angular momentum of rigid bodies . . . . .	50
5.3	Products of inertia . . . . .	51
5.4	Exercise 1: Cart and pendulum problem . . . . .	52
5.4.1	Coordinates . . . . .	52
5.4.2	DOF . . . . .	53
5.4.3	External forces . . . . .	54
5.5	Finishing EOM . . . . .	55
5.6	Exercise 2: Translating System . . . . .	58
5.6.1	Coordinate selection . . . . .	58

5.6.2	DOF . . . . .	58
5.6.3	External forces . . . . .	59

## **II Classical Control Theory** **61**

<b>6</b>	<b>Preliminaries</b>	<b>63</b>
6.1	Euler's Identity . . . . .	63
6.2	Linear differential equations . . . . .	65
6.2.1	Null Solution . . . . .	66
6.2.1.1	Deriving characteristic equation for LDE . . . . .	66
6.2.2	Particular Solution . . . . .	66
6.2.3	Laplace transform . . . . .	68
6.3	Example of solving a LDE . . . . .	69
6.4	Basic definitions of Linear system . . . . .	70
6.5	Solution to differential of $e^{\lambda t}$ . . . . .	70
6.6	Solution to differential of an exponential matrix $e^{\mathbf{A}t}$ . . . . .	71
6.6.1	Defining $e^{\mathbf{A}t}$ . . . . .	71
6.6.2	Time derivative of exponential matrix $e^{\mathbf{A}t}$ . . . . .	72
6.6.3	State Transition Matrix . . . . .	73
6.6.4	Solutions to free and forced response of a system using results from State Transition Matrix . . . . .	73
6.6.4.1	Null Solution . . . . .	73
6.6.4.2	Particular Solution . . . . .	74
6.7	Newton's Law's for rotating bodies . . . . .	74
6.8	Harmonic Oscillator . . . . .	75
<b>7</b>	<b>System Modeling</b>	<b>77</b>
7.1	Three phase method of modeling . . . . .	78
7.1.0.1	Phase I: Subsysteming and choosing inputs and outputs . . . . .	79
7.1.0.2	Phase II: Mathematical modeling . . . . .	79
7.1.0.3	Phase III: Writing State-space form . . . . .	80
7.2	Model verification and validation . . . . .	80
7.3	Mathematical Modeling . . . . .	81
7.3.1	State Space Representation . . . . .	81
7.4	Transfer function representation . . . . .	81
7.5	General form of Second order System . . . . .	82
7.6	Non-Canonical Systems . . . . .	83
7.6.1	Higher order systems . . . . .	83
7.6.1.1	Model reduction . . . . .	83

7.6.2	Effect of zeros . . . . .	88
7.6.2.1	Effect 1: Contribution of each pole to the transfer function . . . . .	88
7.6.2.2	Effect 2: Increases the overshoot . . . . .	88
7.6.2.3	Effect 3: Non-minimum Phase Behavior . . . . .	91
7.6.2.4	Effect 4: Canceling poles that are near them . . . . .	91
7.6.3	Nonlinear Systems . . . . .	92
<b>8</b>	<b>System Analysis</b>	<b>93</b>
8.1	Time-domain analysis . . . . .	93
8.2	Frequency domain analysis . . . . .	93
8.3	Analysis of first order systems . . . . .	94
8.3.1	Impulse response . . . . .	94
8.3.2	General first order responses . . . . .	95
8.3.2.1	$u(t) = 0$ , non-zero initial condition . . . . .	96
8.3.2.2	Step response . . . . .	97
8.3.2.3	step response with non-zero initial condition . . . . .	98
8.4	Stability analysis of PT-2 systems . . . . .	100
8.4.1	Stability in linear systems . . . . .	100
8.4.2	Stability Analysis in Non-linear systems . . . . .	101
8.4.3	Solving / Analysing PT-2 using Laplace transform . . . . .	104
8.4.3.1	Determining the system behavior . . . . .	104
8.4.4	Poles determine the system behavior . . . . .	112
8.4.5	Solving / Analysis PT-2 using characteristic polynomial . . . . .	112
8.4.6	Numerical solution for PT-2 systems using Matlab . . . . .	113
<b>9</b>	<b>Introduction to Control</b>	<b>115</b>
9.1	Control Architectures . . . . .	115
9.1.1	Open-loop control / Feedforward control . . . . .	115
9.1.2	Closed Loop Architecture / Feedback control . . . . .	116
9.2	Block diagram reduction . . . . .	119
9.3	Linearity and Principle of Superposition . . . . .	121
9.4	Cruise control example and drawbacks of proportional regulator . . . . .	121
9.4.1	Open-loop system model . . . . .	121
9.4.2	Closed Loop Model . . . . .	122
9.4.3	Effects of Noise with Proportional Regulator . . . . .	122
<b>10</b>	<b>Eigen Value Analysis</b>	<b>125</b>
10.1	How Eigen Value Analysis Works . . . . .	127
<b>11</b>	<b>Classical Control System Design</b>	<b>129</b>
11.1	Control Specifications . . . . .	129
11.2	Control Expectations (Performance expectations) . . . . .	129

11.3	Pole Placement . . . . .	130
11.4	Calculating steady-state error . . . . .	133
11.5	PID Control . . . . .	134
11.5.1	Understanding PID using a canonical system . . . . .	134
11.5.1.1	P Control . . . . .	136
11.5.1.2	D Control . . . . .	137
11.5.1.3	I Control . . . . .	140
11.5.2	PD Control . . . . .	142
11.5.2.1	PI Control . . . . .	142
11.5.2.2	Summarizing PID control actions . . . . .	143
11.6	System Type - Steady-state error . . . . .	143
<b>12</b>	<b>Root Locus</b>	<b>147</b>
12.1	Theory on Root Locus . . . . .	147
12.1.1	OL, forward and CL TF . . . . .	148
12.1.2	Defining the constraint equation for the root locus . . . . .	149
12.1.2.1	Constraint 1 . . . . .	149
12.1.2.2	Constraint 2 . . . . .	150
12.1.3	Rules for drawing rot locus . . . . .	151
12.1.4	Root locus using Maltab . . . . .	151
12.2	Examples of control design with Root Locus . . . . .	152
12.2.1	Example: Using a simple P control . . . . .	152
12.2.2	Example: Altering root locus using I or D poles or zeros . . . . .	156
12.2.2.1	Case 1: Zero to the right of two poles . . . . .	157
12.2.2.2	Case 2: Zero in between two poles . . . . .	158
12.2.2.3	Case 3: Zero to the left of two poles . . . . .	159
12.2.3	General trends of adding a zero or a pole . . . . .	162
<b>13</b>	<b>Freqeucy Response Analysis</b>	<b>163</b>
13.1	Plotting bode plots . . . . .	164
13.1.1	Approach 1: Point-by-Point approach . . . . .	165
13.1.2	Straight line approximations . . . . .	165
13.1.2.1	Bode plot of Gain $K$ . . . . .	166
13.1.2.2	Bode plot of Differentiator $s$ . . . . .	167
13.1.2.3	Bode Plot of Integrator $1/s$ . . . . .	168
13.1.2.4	Bode Plot for simple Zero . . . . .	170
13.1.2.5	Bode Plot for simple Pole . . . . .	170
13.2	Stability analysis from Frequency response . . . . .	172
13.2.1	Relative Stability: How far is the system from going unstable . . . . .	173
13.2.1.1	Defining Relative stability . . . . .	174
13.3	Caclulation of gain and phase shifts . . . . .	176
13.3.1	System Indentification using Bode Plots . . . . .	177

13.4 Nyquist Diagram . . . . .	179
13.4.1 An example of plotting Nyquist plot for a simple system . . . . .	180
<b>14 Frequency Response for Controller Design</b>	<b>183</b>
14.1 Relationship between domains . . . . .	183
14.2 Design with Frequency Response . . . . .	185
14.2.1 Design with a lead compensator . . . . .	187
14.2.2 Design with a PID controller . . . . .	191
14.2.2.1 Design with a PD regulator . . . . .	191
14.3 Comparison of Frequency based control design with Time based control design . . . . .	192
14.3.1 Advantages of Frequency based control design . . . . .	192
<b>15 Introduction to Robust Control</b>	<b>195</b>
15.1 Robustness in a MIMO system . . . . .	195
15.1.1 Attempting a control design . . . . .	196
15.1.2 One solution from frequency response . . . . .	197
15.2 Pure Time Delay (Transport Lag) . . . . .	198
<b>16 Fundamentals of Electric circuits</b>	<b>201</b>
16.1 Basics of Electricity . . . . .	201
16.2 Modeling Electrical System . . . . .	202
16.3 Principles of electrical systems . . . . .	204
16.3.1 Kirchhoff's Laws . . . . .	204
16.3.2 Modeling electrical circuits . . . . .	204
16.3.3 Modeling DC converter . . . . .	205
16.4 Modeling circuits in Laplace Domain . . . . .	210
16.4.1 TF example of voltage divider . . . . .	211
<b>17 Advanced control architectures</b>	<b>213</b>
17.1 Two degrees of freedom control . . . . .	213
<b>III Model Based Design of Control Systems</b>	<b>215</b>
<b>18 Introduction</b>	<b>217</b>
18.1 Adaptive cruise control example . . . . .	217
18.1.1 Open-loop control design . . . . .	217
18.2 Model Based Control Design general methodology . . . . .	219
18.3 Control design methods . . . . .	220
18.4 State-space control design methodology . . . . .	220

<b>19 State space representation</b>	<b>223</b>
19.1 Laplace transform from state-space model . . . . .	223
19.2 Solution of State-space model . . . . .	224
19.3 Transition Matrix or Matrix exponential . . . . .	225
19.4 Poles and zeros and Eigenvalues of State-space . . . . .	225
19.5 State-space transformations . . . . .	226
19.6 State space for a 2D particle . . . . .	226
19.7 Dimension check for matrices [EgeND, p.3.1.6] . . . . .	227
<b>20 Modeling Automotive Systems</b>	<b>229</b>
20.1 Longitudinal vehicle dynamics model . . . . .	229
20.2 Modeling Drivetrain . . . . .	232
20.2.1 Phase I . . . . .	233
20.2.2 Phase II . . . . .	233
20.2.2.1 Equation for engine . . . . .	235
20.2.2.2 Equations for transmission . . . . .	235
20.2.2.3 Equations for drivetrain . . . . .	235
20.2.2.4 Equations for wheel and chassis . . . . .	235
20.2.3 Phase III . . . . .	236
20.3 Modeling suspension system . . . . .	237
20.3.0.1 Phase I . . . . .	237
20.3.0.2 Phase II . . . . .	238
20.3.0.3 Phase III . . . . .	239
20.4 Single track vehicle model . . . . .	240
20.4.1 Modeling assumptions . . . . .	240
20.4.2 Phase I . . . . .	240
20.4.3 Phase II . . . . .	241
20.4.4 Phase III . . . . .	243
20.4.5 Constructing State-Space equation . . . . .	244
20.5 Differential drive robot . . . . .	245
20.5.1 Part I: Deriving Equations of Motion for differential drive robot	245
<b>21 Linearization</b>	<b>249</b>
21.1 Linearizing EoM of a pendulum . . . . .	251
<b>22 State space behavior</b>	<b>253</b>
22.1 State-Transition matrix $e^{\mathbf{A}t}$ . . . . .	253
<b>23 State Feedback Design</b>	<b>255</b>
23.1 Principles . . . . .	255
23.1.1 Poles of the closed loop system . . . . .	257
23.1.2 Reference tracking . . . . .	257
23.2 Integral control . . . . .	258

23.3 Example of State-feedback design . . . . .	259
23.4 Example of reaching non-controllability . . . . .	262
<b>24 Tuning of state feedback systems</b>	<b>263</b>
24.1 Ackermann's Formula for pole placements . . . . .	263
24.1.1 Ackermann's approach . . . . .	264
24.2 Example solving for K using Ackermann's formula . . . . .	266
24.3 Example with Single track model . . . . .	267
24.4 Linear Quadratic Regulator . . . . .	267
24.4.1 Solution to the optimization problem . . . . .	268
24.5 Examples on LQR . . . . .	269
24.5.0.1 Mass damper system . . . . .	269
24.5.1 Choosing penalty matrices . . . . .	271
<b>25 Controllability and Observability</b>	<b>273</b>
25.1 Stability of State-space systems . . . . .	273
25.1.1 Equilibrium . . . . .	273
25.1.2 Stable Equilibrium . . . . .	273
25.1.3 Asymptotic stability . . . . .	274
25.1.4 BIBO . . . . .	274
25.2 Defining controllability . . . . .	274
25.3 Controllability vs Trajectories . . . . .	275
<b>26 State Estimation</b>	<b>279</b>
26.1 Principle of State estimation . . . . .	279
26.2 Goal of State estimation . . . . .	280
26.3 Observers design . . . . .	281
26.3.1 Theory . . . . .	281
26.3.2 State estimation gain . . . . .	283
26.3.3 Predictor-Corrector . . . . .	283
26.3.4 Example of observer design . . . . .	284
26.3.5 Example observer design fo Single Track Model . . . . .	285
26.3.6 Control using the estimated states . . . . .	286
26.3.7 Full System with Observer and Feedback controller . . . . .	287
26.3.8 Implementation - Digital Computer Language . . . . .	288
26.4 Observability . . . . .	288
26.4.1 Observability Theorem 1 . . . . .	289
26.4.2 Observability Theorem 2 . . . . .	289
26.5 Separation Principle . . . . .	289
26.5.1 Method of separation principle . . . . .	290
26.5.2 Choosing eigenvalues for controller and observer . . . . .	291
26.6 Kalman Filter . . . . .	291
26.6.0.1 Kalman - Bucy Filter . . . . .	292

26.6.1	Vehicle Steering Example . . . . .	292
26.6.2	Matlab's Kalman Filter . . . . .	293
26.6.2.1	Kalman filter for a single state system . . . . .	294
26.6.2.2	Kalman filter for Sensor Fusion . . . . .	298
26.6.3	Kalman Filter Matrix Equations . . . . .	299
<b>27</b>	<b>Example Systems with control applications</b>	<b>301</b>
27.1	Rotor of a Humanoid Robotic arm . . . . .	301
27.2	Full system control design for a single track model . . . . .	301
27.2.1	Linearizing single track model . . . . .	303
27.2.2	Controllability . . . . .	304
27.2.3	Feedback Control . . . . .	305
27.2.4	Estimator design using Kalman Filter . . . . .	306
<b>28</b>	<b>Robust Control</b>	<b>307</b>
28.1	Reason for a Robust control . . . . .	307
<b>Bibliography</b>		<b>A</b>

X

# List of Figures

1.1	Geometry of two points using vectors . . . . .	4
1.2	Vector of a point represented from body frame to global frame . . . . .	5
1.3	Rotation of unit vectors . . . . .	6
1.4	Vectors in moving frames . . . . .	8
1.5	2 DOF motion problem stated in polar coordinates . . . . .	10
1.6	Top view of 2 DOF motion problem stated in polar coordinates . . . . .	11
1.7	Taking time-derivative of unit vector in polar coordinates . . . . .	12
1.8	Rotation Problem 1 . . . . .	14
1.9	Rotation problem 2 . . . . .	16
1.10	Rotation Problem 3 . . . . .	17
1.11	Angular velocity with zero tilt angle . . . . .	18
2.1	Impulse - Momentum Problem . . . . .	22
2.2	Momentum problem on system of particles . . . . .	23
2.3	Angular momentum of B w.r.t O . . . . .	24
2.4	Angular momentum of B w.r.t A . . . . .	24
2.5	Angular momentum problem . . . . .	26
2.6	Problem 2 on Torque and Angular Momentum . . . . .	28
2.7	Problem 2 on Torque and Angular Momentum, case 2 . . . . .	29
2.8	Exercise Problem on Inclined plane and polar coordinates . . . . .	31
3.1	Rotation problem to describe fictitious forces . . . . .	34
3.2	Naming fictitious forces using FBD's . . . . .	35
4.1	Pendulum attached to a torsional spring . . . . .	38
4.2	FBD for pendulum BOB . . . . .	39
4.3	FBD for pendulum Link . . . . .	39
4.4	FBD for pendulum Link . . . . .	40
4.5	Two sliding masses . . . . .	42
4.6	FBD mass 1 . . . . .	43
4.7	FBD mass 2 . . . . .	43
4.8	Two mass suspension system . . . . .	44
4.9	FBD of two mass . . . . .	45
4.10	FBD of wheel . . . . .	45
5.1	Finding mass moment of inertia (MMI) . . . . .	47

5.2	Similar problem on finding MMI . . . . .	48
5.3	Axis of symmetry for rotating objects . . . . .	50
5.4	Mass placed asymmetrically about the rotating axis . . . . .	51
5.5	Products of inertia results in $H_A$ on non-rotating axis . . . . .	52
5.6	Cart and pendulum . . . . .	53
5.7	FBD of pendulum . . . . .	54
5.8	FBD of cart . . . . .	55
5.9	Translating masses . . . . .	58
5.10	FBD of mass 1 . . . . .	59
6.1	Flow of solution to LDE to and from Laplace Transforms . . . . .	67
7.1	Model verification and validation definition . . . . .	80
7.2	Step response at $\zeta = 0.2$ . . . . .	84
7.3	Step response at $\zeta = 0.4$ . . . . .	85
7.4	Step response at $\zeta = 0.8$ . . . . .	85
7.5	Step response for pole at $s = -2$ . . . . .	86
7.6	Step response for pole at $s = -10$ . . . . .	86
7.7	Step response for whole system with both the poles at $\{-2, -10\}$ . .	87
7.8	Zeros increase the overshoot of the system . . . . .	89
7.9	Zeros increase the overshoot of the system . . . . .	90
7.10	Zeros which are close to poles cancel them . . . . .	91
8.1	Impulse response of a first order system . . . . .	95
8.2	Step response of a first order system . . . . .	97
8.3	Neighborhood of a non-linear system stability . . . . .	102
8.4	Lagrange Stability . . . . .	103
8.5	Lyaponov Stability . . . . .	103
8.6	Asymptotic stability . . . . .	104
8.7	Poles in the PT2 system . . . . .	105
8.8	Solution to PT-2 system with step input using Laplace transform . . .	106
8.9	Important behvarial propertial seen in PT-2 systems . . . . .	107
8.10	Delay and Rise times seen in PT-2 systems . . . . .	108
8.11	Settling time seen in PT-2 systems . . . . .	109
8.12	Peak time seen in PT-2 systems . . . . .	110
8.13	Maximun overshoot seen in PT-2 systems . . . . .	111
8.14	Poles determine the system behavior . . . . .	112
9.1	Open Loop Control Architecture . . . . .	115
9.2	Closed Loop Control Architecture . . . . .	117
9.3	Use of Feedback . . . . .	117
9.4	Use of Feedback: Making the system robust . . . . .	118
9.5	Block diagram reduction: Algebraic . . . . .	119
9.6	Block diagram reduction: Memorizing rules . . . . .	120

9.7	Block diagram reduction: Series and Parallel . . . . .	120
9.8	Block diagram of Cruise controller with P-type regulator . . . . .	123
9.9	Tracking achieved by Hight Propotional controller gain . . . . .	123
9.10	Tracking failed by high Proportional controller gain with noise . . . . .	124
10.1	Plot of Eigenvectors and the matrix A evolution along Eigenvectors . .	126
11.1	PI Controller for Automatic Cruise Control . . . . .	131
11.2	Pole location of the roots given in equations (11.4) and (11.5) . . . . .	132
11.3	Example calculating SS error . . . . .	133
11.4	PT-2 canonical system with P-Control . . . . .	135
11.5	Steady-state error with P-Control . . . . .	135
11.6	Reduced Steady-state error with P-Control . . . . .	137
11.7	Overshoot elimination using derivative control . . . . .	139
11.8	Performance of Derivative Control with noise . . . . .	140
11.9	Noise amplification of derivative control into the system . . . . .	141
12.1	Root locus definition . . . . .	147
12.2	Root locus definition . . . . .	148
12.3	Root locus constraint ratio plot on the complex plane . . . . .	150
12.4	Control design using root locus (non-canonical systems) . . . . .	153
12.5	Control design using root locus (non-canonical systems) . . . . .	153
12.6	System response for $K = 4.07$ . . . . .	154
12.7	System response for $K = 5.46$ . . . . .	155
12.8	Example using root locus altering for this system . . . . .	156
12.9	Root locus with area of control interest enveloped between semi-circle and angled lines . . . . .	157
12.10	Altering root locus by placing a zero to the left of two poles . . . . .	158
12.11	Altering root locus by placing a zero in-between two poles . . . . .	159
12.12	Altering root locus by placing a zero in-between two poles . . . . .	160
12.13	Altering root locus by placing a zero in-between two poles . . . . .	161
12.14	Altering root locus by placing a zero in-between two poles . . . . .	162
13.1	Graphical Represenataion of the Analytical frequency response . . . . .	163
13.2	Bode plot of gain $K$ . . . . .	167
13.3	Bode plot of Differentiator $s$ . . . . .	168
13.4	Bode plot of Integrator $1/s$ . . . . .	169
13.5	Bode plot of simple zero $(Ts + 1)$ . . . . .	171
13.6	Bode plot of simple pole $1/(Ts + 1)$ . . . . .	172
13.7	A Closed-loop system with input as frequency . . . . .	173
13.8	Gain and phase margins on Bode plot . . . . .	175
13.9	Effect of $K$ on relative stability . . . . .	176
13.10	Determining Magnitude and Phase using Bode Plots . . . . .	177
13.11	system Identification using Bode Plot . . . . .	178

13.12 Plotting example of complex numbers in Nyquist plot . . . . .	181
14.1 Step Response . . . . .	184
14.2 Frequency Response . . . . .	184
14.3 Overview of time and frequency relationships . . . . .	186
14.4 Bode plot of the lead compensator . . . . .	188
14.5 Bode plot of the lead compensator . . . . .	189
14.6 Bode plot of the lead compensator and plant . . . . .	190
14.7 Bode plot of plant with LC . . . . .	191
14.8 Bode plot of plant with LC . . . . .	192
15.1 MIMO system with noise and disturbance . . . . .	196
15.2 Control strategy for noise and disturbance attenuation . . . . .	197
15.3 Time Delay theory 1 . . . . .	198
15.4 Time Delay theory 2 . . . . .	199
16.1 Charged particles creates electric fields . . . . .	201
16.2 Electric fields exert force on charged particles . . . . .	201
16.3 Resistors in series . . . . .	204
16.4 Resistors in parallel . . . . .	205
16.5 Schematic of a boost converter . . . . .	206
16.6 Boost converter CKT in ON mode . . . . .	206
16.7 Boost converter CKT in OFF mode . . . . .	207
16.8 Performance of boost converter . . . . .	209
16.9 Charging frequency of the boost converter . . . . .	209
16.10 Boost converter charging cycles . . . . .	210
16.11 Example for circuit modeling using TF . . . . .	211
16.12 System modeled in the form representing a voltage divider . . . . .	212
18.1 ACC in OL control . . . . .	218
18.2 ACC in OL control . . . . .	218
18.3 Visualizing state space modeling . . . . .	221
20.1 Subsystems of entire vehicle model . . . . .	229
20.2 Subsystems of entire vehicle model . . . . .	230
20.3 Further Subsystems of vehicle model . . . . .	230
20.4 Free body diagram chassis model . . . . .	231
20.5 Detailed subsystems of drivetrain . . . . .	233
20.6 FBD of subsystems of drivetrain . . . . .	234
20.7 Suspension model . . . . .	238
20.8 Suspension model . . . . .	238
20.9 Single track model . . . . .	241
20.10 Single track model . . . . .	241
20.11 Differential drive robot . . . . .	246

20.12 Kinematic configuration with $v_r \neq 0$ and $v_l = 0$ . . . . .	246
20.13 Kinematic configuration with $v_r = 0$ and $v_l \neq 0$ . . . . .	247
21.1 Adding and subtracting OP's in feedback loop . . . . .	251
23.1 Block diagram of state-space model . . . . .	256
23.2 Block diagram of state feedback model . . . . .	256
23.3 Block diagram of state feedback model without D . . . . .	256
23.4 Simulation results for different design parameters . . . . .	260
23.5 Simulation results for sinusoidal trajectory . . . . .	261
23.6 Simulation results for sinusoidal trajectory with better tracking . . . . .	261
25.1 A model for trajectory vs Controllability . . . . .	276
25.2 Non possible trajectory . . . . .	276
25.3 Possible trajectory . . . . .	277
26.1 State-feedback control design principle . . . . .	279
26.2 State estimation principle with observer design . . . . .	280
26.3 Generic Observer Design . . . . .	280
26.4 Open-loop observer design . . . . .	281
26.5 Closed-loop observer design . . . . .	282
26.6 Control algorithm using estimated states . . . . .	287
26.7 Poles of controller and observer . . . . .	291
26.8 Gaussian distribution of sensor disturbances $v$ . . . . .	294
26.9 Understanding Kalman filter optimization using graph . . . . .	295
26.10 Definition of $P_k^-$ . . . . .	297
26.11 Definition of $P_k^-$ . . . . .	297
26.12 Definition of $P_k^-$ . . . . .	298
26.13 Definition of $P_k^-$ . . . . .	299
27.1 Global coordinate system for single track model . . . . .	302
28.1 Input/Output model of a plant . . . . .	307
28.2 Control system developed using the measurements from the input/output model . . . . .	307
28.3 Control system developed using the measurements from the input/output model . . . . .	308
28.4 open-loop margins of an example system . . . . .	310
28.5 adding gain to the example system . . . . .	311
28.6 adding phase to the example system . . . . .	311
28.7 adding phase to the example system . . . . .	312
28.8 adding phase to the example system . . . . .	313



# List of Tables

8.1	Table summarizing PT-1 response for non-zero initial conditions . . . . .	96
8.2	Table summarizing step response of PT-1 system . . . . .	98
8.3	Table summarizing response of PT-1 due to step input and IC . . . . .	99
11.1	Table summarizing standard system types with standard input types .	145
13.1	Table showing $G(j\omega)$ calculation for Nyquist plot . . . . .	180
27.1	Single track model parameters . . . . .	303



# Nomenclature

## Abbreviations

<b>LC</b>	Lead Compensator
<b>TF</b>	Transfer Function
<b>FrRe</b>	Frequency Response
<b>MgRe</b>	Magnitude Response
<b>EoM</b>	Equations of motion
<b>ODE</b>	Ordinary Differential Equations
<b>IC</b>	Initial conditions
<b>LDE</b>	Linear Differential Equation
<b>StSp</b>	State-space

## Small letters

**ss** steady-state

## Big letters

$P_m$  Phase margin  
 $G_m$  Gain margin

## Greek letters

*Nomenclature*

# **Part I**

# **Newtonian Mechanics**



# 1 Newtons Laws and Kinematics

**Note:** Vector calculus cannot be applied to angles!! Vector calculus can be applied to positions, linear velocities and accelerations, angular velocity and angular acceleration but not on angles!!

Because angles do not add up like any vector. To add up angles we need to use DCM's (Direction cosine matrix).

## 1.1 Method for solving Dynamics Problem

- state the problem
- Draw figures including FBD's
- Describe motion
  - determine degrees of freedom
  - from DOF, define the coordinates of the motion problem
  - also from DOF, equations of motion can be described for each of the DOF
- assign the coordinates (generalized coordinates)
  - match generalized coordinates
  - match generalized velocities
- explain correct physical laws
- do the math

## 1.2 Derivative Rule from vector geometry

Consider figure 1.1, in which there are tow points  $A$  and  $B$  which can be located from a reference point  $O$  in an arbitrary reference frame  $F$ . The three position vectors can be drawn from  $O$  which are labeled using vectors  $r_{i/o}$  as shown in figure 1.1. Where  $i$  is any reference point  $\{A, B\}$  and  $o$  denotes the reference point chosen to measure the vector  $r$ . The derivative rule gives a mathematical tool for differentiating vectors relative to the frames of reference that they originally do not belong to. The

## Nomenclature

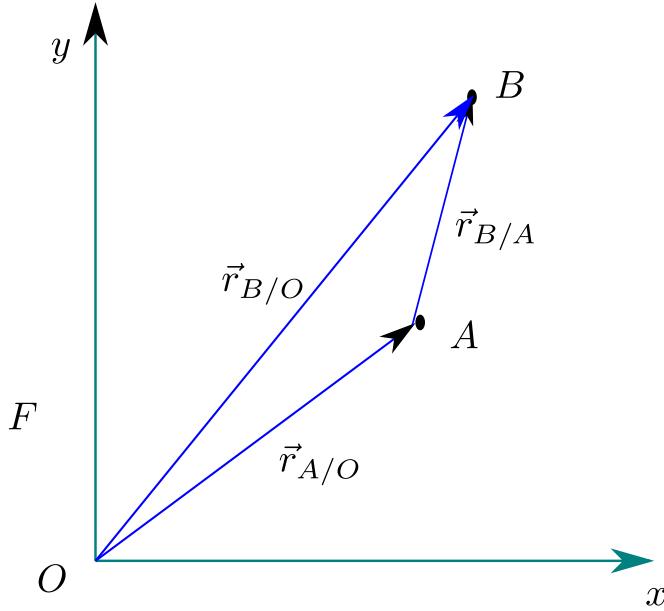


Figure 1.1: Geometry of two points using vectors

method involves a geometrical approach to differentiating vectors first as described in the following:

Starting from the positions, using the vector formula for addition, we get:

$$\mathbf{r}_{B/O} = \mathbf{r}_{A/O} + \mathbf{r}_{B/A} \quad (1.1)$$

To find velocities of these, while applying differential operator also note the frames that they are being differentiated:

$$\frac{^F d\mathbf{r}_{B/O}}{dt} = \frac{^F d\mathbf{r}_{A/O}}{dt} + \frac{^F d\mathbf{r}_{B/A}}{dt} \quad (1.2)$$

The superscript  $F$  applied on the differential operator suggests that differentiation is applied w.r.t frame F. Lets consider first term of equation (1.2), this vector as it is referenced directly in frame F and has its tail attached to O, it is directly present in the Frame F and can be therefore, differentiated directly:

$$\frac{^F d\mathbf{r}_{A/O}}{dt} = \dot{\mathbf{r}}_{A/O} = \mathbf{v}_{A/O} \quad (1.3)$$

For the second term, the tail of the vector is attached to point A which is translating, therefore, vector  $\mathbf{r}_{B/A}$  is not in the Frame F but that of translating Frame A.

$$\frac{^F d\mathbf{r}_{B/A}}{dt} = \frac{^A d\mathbf{r}_{B/A}}{dt} + \boldsymbol{\omega} \times \mathbf{r}_{B/A} \quad (1.4)$$

This above derivative rule comes into practice due to differentiation of unit vectors that are changing w.r.t to the reference Frame F. The proof of this derivation is as described in the following section.

### 1.2.1 Framing derivative rule for unit vectors

Consider figure 1.2. Consider a point P, a position vector can be expressed from O of Frame F. There are two frames in this figure 1.2 with their respective centers and unit vectors as described below

$$F : O, \hat{I}, \hat{J}, \hat{K}$$

$$B : P, \hat{r}, \hat{\theta}, \hat{k}$$

Therefore, using Frame B, the position vector of point P can be expressed either in

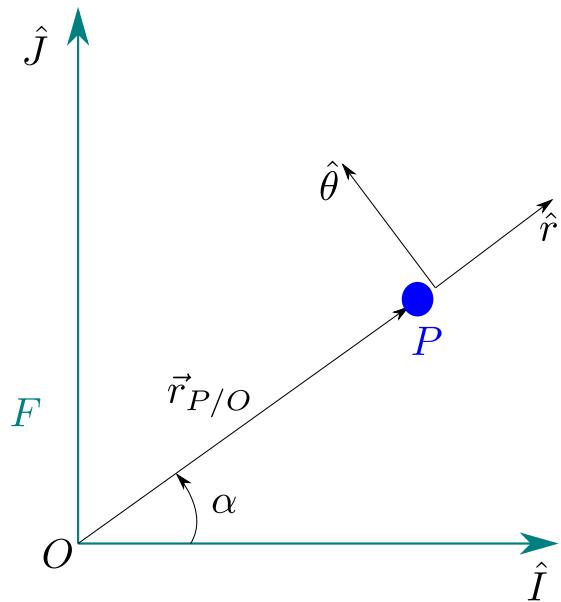


Figure 1.2: Vector of a point represented from body frame to global frame

body frame as

$$\mathbf{r}_{P/0} = l\hat{r} \quad (1.5)$$

where  $l$  is the length of vector from point O to point P. In order to represent vector  $\mathbf{r}_{P/0}$  on global frame F, consider the rotation of unit vectors as shown in figure 1.3:

## Nomenclature

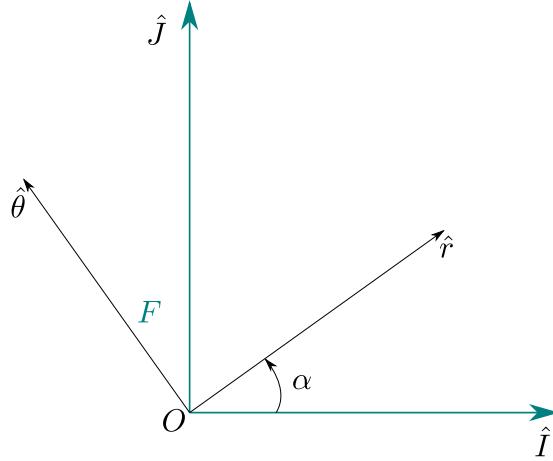


Figure 1.3: Rotation of unit vectors

due to the rotation the unit vectors  $\hat{r}$  and  $\hat{\theta}$  can be expressed as

$$\hat{r} = \hat{I} \cos \alpha + \hat{J} \sin \alpha \quad (1.6)$$

$$\hat{\theta} = -\hat{I} \sin \alpha + \hat{J} \cos \alpha \quad (1.7)$$

now the velocities of these unit vectors can be determined using differentiation as they have been expressed in frame F as:

$${}^F \frac{d\hat{r}}{dt} = \dot{\hat{r}} = (-\sin \alpha \hat{I} + \cos \alpha \hat{J}) \dot{\alpha} = \dot{\alpha} \hat{\theta} \quad (1.8)$$

$${}^F \frac{d\hat{\theta}}{dt} = \dot{\hat{\theta}} = (-\cos \alpha \hat{I} - \sin \alpha \hat{J}) \dot{\alpha} = -\dot{\alpha} \hat{r} \quad (1.9)$$

therefore, using equation 1.5 vector  $\mathbf{r}_{P/0}$  the velocity of this vector can be expressed as:

$${}^F \frac{d\mathbf{r}_{P/0}}{dt} = \dot{\mathbf{v}}_{P/0} = \frac{dl}{dt} \hat{r} + l \frac{d\hat{r}}{dt} \quad (1.10)$$

using equation (1.8), we get

$$\dot{\mathbf{v}}_{P/0} = l \hat{r} + l \dot{\alpha} \hat{\theta} \quad (1.11)$$

this can further be expressed using cross products between the unit vectors as:

$$\dot{\mathbf{v}}_{P/0} = l \hat{r} + l \dot{\alpha} (\hat{K} \times \hat{r}) \quad (1.12)$$

here the terms  $\dot{\alpha} \hat{K}$  is nothing but the angular velocity of Frame B w.r.t Frame F and the term  $l \hat{r}$  is nothing but the position vector  $\mathbf{r}_{P/0}$ , therefore

$$\dot{\alpha} \hat{K} = \omega$$

$$l \hat{r} = \mathbf{r}_{P/0}$$

the velocity equation can now be re-written as

$$\dot{\mathbf{v}}_{P/0} = \dot{l}\hat{\mathbf{r}} + \boldsymbol{\omega} \times \mathbf{r}_{P/0} \quad (1.13)$$

the above equation is used for differentiating vectors in different frames of references. It is also called ***Transport Theorem***.

## 1.3 Velocity and Acceleration in moving reference frames

### 1.3.1 Velocity of a vector in a moving frame

The transport theorem can be used to determine velocity and acceleration of vector quantities in any moving frames they are attached to. Consider a point B which is attached to a moving reference frame A as shown in figure 1.4.

## Nomenclature

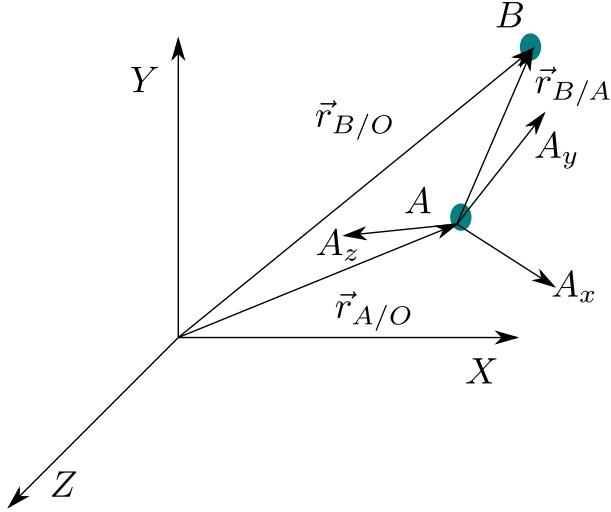


Figure 1.4: Vectors in moving frames

Assigning frames:

$$\begin{aligned} F &: \{O, \hat{I}, \hat{J}, \hat{K}\} \\ A &: \{A, \hat{i}, \hat{j}, \hat{k}\} \\ B &: \{B, \hat{B}_i, \hat{B}_j, \hat{B}_k\} \end{aligned}$$

The object is to find velocity and acceleration of vectors of point B that is attached to a moving frame A. Using vector geometry the position of B from O can be determined as:

$$\mathbf{r}_{B/O} = \mathbf{r}_{A/O} + \mathbf{r}_{B/A} \quad (1.14)$$

for velocities, differentiating the above equation

$$\begin{aligned} {}^F \frac{d}{dt} \mathbf{r}_{B/O} &= {}^F \frac{d}{dt} \mathbf{r}_{A/O} + {}^F \frac{d}{dt} \mathbf{r}_{B/A} \\ \mathbf{v}_{B/O} &= \mathbf{v}_{A/O} + {}^F \frac{d}{dt} \mathbf{r}_{B/A} \end{aligned}$$

the term  $\mathbf{v}_{A/O}$  is fixed to the global frame, therefore it is referenced to the unit vectors that are attached to global frame. The second term however, is due to a vector that is attached to a moving frame. This vector needs to be differentiated using transport theorem in order to determine its components properly.

using transport theorem (let point B be described in frame A with position vector  $\mathbf{r}_{B/A} = (A_x \hat{i} + A_y \hat{j} + A_z \hat{k})$ ):

$${}^F \frac{d}{dt} \mathbf{r}_{B/A} = \frac{d}{dt} \mathbf{r}_{B/A} + \omega_{/O} \times \mathbf{r}_{B/A}$$

therefore, velocity of B which is fixed to a moving reference frame is expressed as:

$$v_{B/O} = \mathbf{v}_{A/O} + \frac{d}{dt} \mathbf{r}_{B/A} + \omega_{/O} \times \mathbf{r}_{B/A} \quad (1.15)$$

**Note:** The extra term (third term in equation (1.15)) is due to differentiating unit vectors of moving frame w.r.t the global frame. The first two terms are literally scalar quantities of the velocities that is, their differentiation does not have any direction associated with them. Just like when energy is differentiated, it is not considered in which frame it is differentiated. Simply a time derivative of these scalar quantities are taken directly. Also, the first two terms of equation (1.15) are linear velocity of the moving frame itself and the relative velocity of the frame into consideration that is fixed to a moving frame.

**Note:** For all 2D and 3D motion problems the angular velocity term is always chosen to be expressed in the succeeding (frame which is moving compared to the global chosen frame of reference) than the preceding (global frame of reference) frame, mainly because it is always easier to represent  $\omega$  in the succeeding frame than the preceding frame for calculation purposes.

In summary, when deriving relative velocity of a vector that is attached to a moving frame has these three following components:

- Velocity of the moving frame itself
- Velocity of frame in which the vector is being measured (relative to the moving frame)
- velocity due to the rotation of the moving frame w.r.t to the global frame

### 1.3.2 Acceleration of a vector in a moving frame

$$a_{B/O} = \frac{^F d}{dt} v_{B/O} = a_{A/O} + {}^A a_{B/A} + \{2\omega_{/O} \times {}^A v_{B/A}\} + \{\dot{\omega}_{/O} \times r_{B/A}\} + \{\omega_{/O} \times (\omega_{/O} \times r_{B/A})\} \quad (1.16)$$

in equation (1.16), the following terms are as described

- $a_{A/O}$  is due to the acceleration of moving frame itself
- ${}^A a_{B/A} + \{2\omega_{/O} \times {}^A v_{B/A}\}$  is due to the derivative of second term in equation (1.15), where the first term is the relative acceleration of frame B w.r.t frame A, the second term is called Coriolis acceleration and this happens due to a velocity vector that is being rotated by the moving frame. (example of water flowing through channels when these channels are under rotation, the water particles experience Coriolis acceleration)
- $\dot{\omega}_{/O} \times r_{B/A}$  is due to the angular acceleration of moving frame (frame A). Also called Eulerian acceleration term.
- $\omega_{/O} \times (\omega_{/O} \times r_{B/A})$  is due to the centripetal acceleration (of the form  $r\omega^2$ ).

## Nomenclature

### 1.4 Velocity and acceleration w.r.t moving frame in polar coordinates

Consider figure 1.5, this is a 2 DOF problem with rotation about  $\hat{k}$  and translation of B on the arm. Consider the following frames in the figure

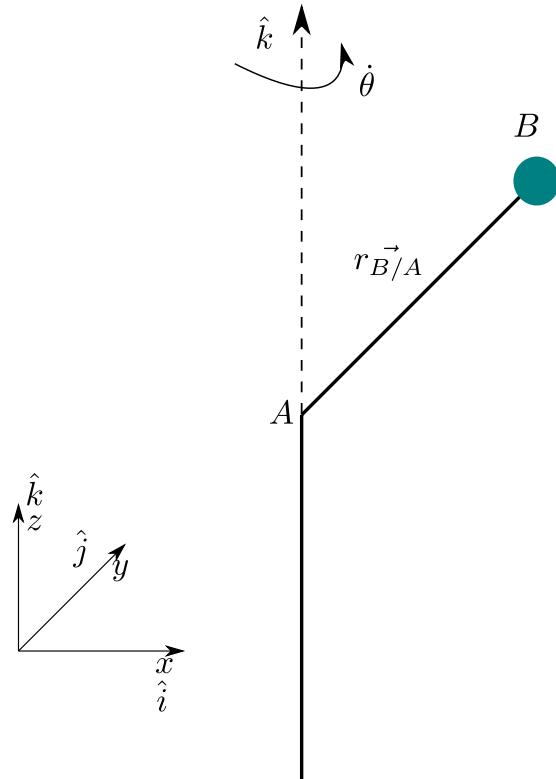


Figure 1.5: 2 DOF motion problem stated in polar coordinates

$$F : \{O, \hat{i}, \hat{j}, \hat{k}\}$$

$$A : \{A, \hat{r}, \hat{\theta}, \hat{k}\}$$

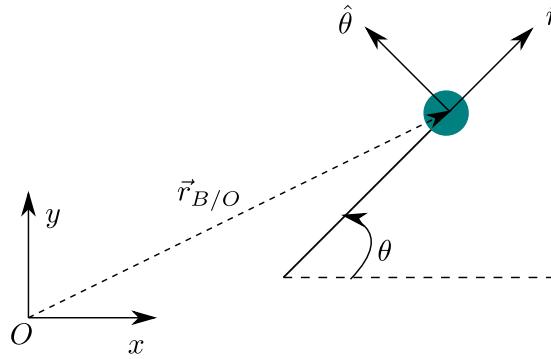


Figure 1.6: Top view of 2 DOF motion problem stated in polar coordinates

### 1.4.1 Velocity in polar coordinates

Using the top-view from figure 1.6, the following relations can be derived. The velocity of point B can be expressed as:

$$v_{B/O} = v_{A/O} + \overset{F}{\frac{d}{dt}} r_{B/A} \quad (1.17)$$

The acceleration of point B can be expressed as:

$$a_{B/O} = a_{A/O} + \overset{F}{\frac{d^2}{dt^2}} r_{B/A} \quad (1.18)$$

In general, in rotations problems in 2D, the rotation is about a fixed axis, therefore,  $v_{A/O} = a_{A/O} = 0$ . This simplification reduces the simple terms from the above equations which can be put back later if needed.

Since this is a 1D rotation problem, the angular velocity can be expressed as:

$$\omega_{/O} = \dot{\theta} \hat{k} \quad (1.19)$$

From equation (1.17), using the position vector  $r_{B/O} = rr\hat{r} + z\hat{k}$ :

$$v_{B/O} = 0 + \overset{F}{\frac{d}{dt}} r_{B/A} = \frac{d}{dt}(rr\hat{r} + z\hat{k}) = \dot{r}\hat{r} + r\dot{\hat{r}} + \dot{z}\hat{k} + z\dot{\hat{k}} \quad (1.20)$$

where  $\dot{\hat{k}} = 0$ , as  $\hat{k}$  will never change its direction, therefore, the time derivative of this unit vector is zero. Therefore, the above equation can be re-written as:

$$v_{B/O} = \dot{r}\hat{r} + r\dot{\hat{r}} + \dot{z}\hat{k} \quad (1.21)$$

there is only one time-derivative of a unit vector that is,  $\dot{\hat{r}}$  in this problem. In order to solve for this time-derivative the following figure 1.7 in considered.

## Nomenclature

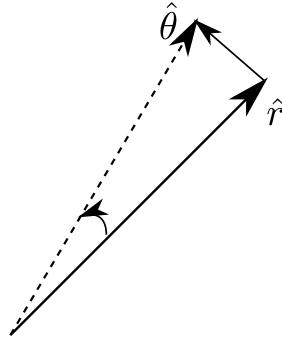


Figure 1.7: Taking time-derivative of unit vector in polar coordinates

using this figure 1.7, it can be seen that for an infinitesimally small time  $\Delta t$ , the unit vector  $\hat{r}$  moves about an infinitesimally small length  $\dot{\theta}\Delta t$  in the direction of  $\hat{\theta}$ , therefore:

$$\Delta\hat{r} = \dot{\theta}\hat{\theta}\Delta t \quad (1.22)$$

where  $\Delta\hat{r}$  in figure 1.7 is the small vector that indicates the change in the direction of this unit vector  $\hat{r}$ . By re-formulating the above expression, the following can be written:

$$\frac{\Delta\hat{r}}{\Delta t} = \dot{\theta}\hat{\theta} \quad (1.23)$$

from the above equation, if the limit of  $\Delta t$  goes to zero, the expression reaches the definition of a derivative:

$$\lim_{\Delta t \rightarrow 0} \frac{\Delta\hat{r}}{\Delta t} = \frac{d}{dt}\hat{r} = \dot{\theta}\hat{\theta} \quad (1.24)$$

therefore, deriving the time derivative of the required unit vector. The velocity of B in polar coordinates can therefore be expressed as (including velocity of the moving frame itself):

$$v_{B/O} = v_{A/O} + \dot{r}\hat{r} + r\dot{\theta}\hat{\theta} + \dot{z}\hat{k} \quad (1.25)$$

The above expression for velocity in polar coordinates can simply be determined using Transport Theorem, without getting into the details of polar coordinates themselves as described below. Using equation (1.17)

$${}^F \frac{d}{dt} r_{B/O} = {}^A \frac{d}{dt} r_{B/A} + \omega_{/O} \times r_{B/A} \quad (1.26)$$

substituting for  $r_{B/A}$  in the above equation:

$$\begin{aligned} {}^F \frac{d}{dt} r_{B/O} &= {}^A \frac{d}{dt} (r\hat{r} + z\hat{k}) + \omega_{/O} \times r\hat{r} + z\hat{k} \\ {}^F \frac{d}{dt} r_{B/O} &= \dot{r}\hat{r} + \dot{z}\hat{k} + \dot{\theta}\hat{k} \times (r\hat{r} + z\hat{k}) \\ {}^F \frac{d}{dt} r_{B/O} &= \dot{r}\hat{r} + \dot{z}\hat{k} + r\dot{\theta}\hat{\theta} \end{aligned}$$

### 1.4.2 Acceleration in polar coordinates

From equation (1.18), acceleration is expressed as:

$$a_{B/O} = a_{A/O} + \frac{F}{dt^2} r_{B/A} = 0 + \frac{F}{dt} v_{B/A} = \frac{d}{dt} (\dot{r}\hat{r} + \dot{z}\hat{k} + r\dot{\theta}\hat{\theta}) \quad (1.27)$$

simplifying the above expression:

$$\begin{aligned} a_{B/O} &= \ddot{r}\hat{r} + \dot{r}\dot{\hat{r}} + \ddot{z}\hat{k} + \dot{z}\dot{\hat{k}} + \dot{r}\dot{\theta}\hat{\theta} + \dot{r}\ddot{\theta}\hat{\theta} + r\dot{\theta}\dot{\hat{\theta}} \\ a_{B/O} &= \ddot{r}\hat{r} + \dot{r}\dot{\theta}\hat{\theta} + \ddot{z}\hat{k} + 0 + \dot{r}\dot{\theta}\hat{\theta} + r\ddot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \end{aligned}$$

therefore, the acceleration in polar coordinates including  $a_{A/O}$  can be expressed as:

$$a_{B/O} = a_{A/O} + (\ddot{r}\hat{r} + \ddot{z}\hat{k}) + 2\dot{r}\dot{\theta}\hat{\theta} + r\ddot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \quad (1.28)$$

where

- $a_{A/O}$  acceleration of the moving frame itself
- $(\ddot{r}\hat{r} + \ddot{z}\hat{k})$  relative acceleration of a moving point B which is connected to moving frame A
- $2\dot{r}\dot{\theta}\hat{\theta}$  is the Coriolis acceleration
- $r\ddot{\theta}\hat{\theta}$  is Euler's acceleration due to  $\ddot{\theta}$
- $r\dot{\theta}^2\hat{r}$  is the centripetal (or centrifugal acceleration)

**Note:** The same exact method is employed while deriving expressions for 3D dynamics problems both in Cartesian and polar coordinates.

## 1.5 Rotation Problem 1

Consider a 1D rotation problem as shown in figure 1.8. Consider the following frames

## Nomenclature

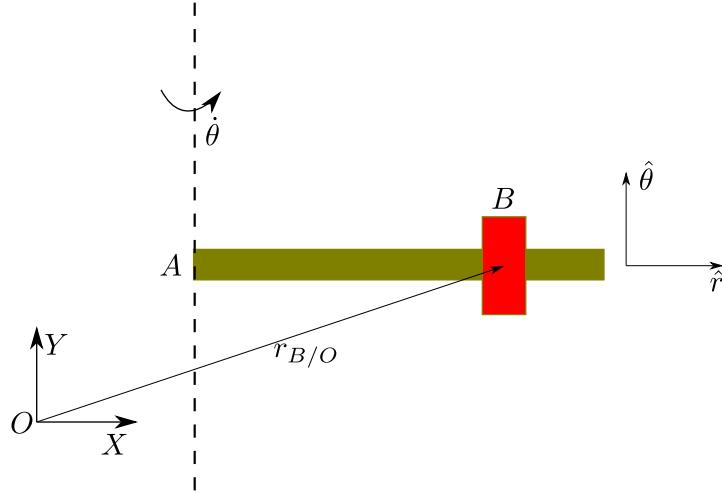


Figure 1.8: Rotation Problem 1

$$F : \{O, X(\hat{i}), Y(\hat{j}), Z(\hat{k})\}$$

$$A : \{A, \hat{r}, \hat{\theta}, \hat{k}\}$$

Frame O is the inertial reference frame, A is the moving frame w.r.t Frame O. The green bar is a pipe which is rotating at a speed  $\omega_{/O} = \dot{\theta}\hat{k}$ . B is a nut which is constrained to moving along the green bar at a constant velocity  $\dot{r}\hat{r}$ . The object is to find the velocity and acceleration of B  $r_{B/O}$ .

In order to solve this problem, consider the velocity formula in polar coordinates (consider polar coordinates while solving problems related to 1D rotation).

$$v_{B/O} = v_{A/O} + \dot{r}\hat{r} + r\dot{\theta}\hat{\theta} + \dot{z}\hat{k} \quad (1.29)$$

in the above equation  $v_{A/O} = 0$  as well as  $\dot{z}\hat{k} = 0$  as this problem is constrained only to 1D rotation and 1D translation  $\dot{r}\hat{r}$ . Therefore, reducing the velocity to the following terms:

$$v_{B/O} = \dot{r}\hat{r} + r\dot{\theta}\hat{\theta} \quad (1.30)$$

For acceleration, consider the acceleration equation in polar coordinates:

$$a_{B/O} = a_{A/O} + (\ddot{r}\hat{r} + \ddot{z}\hat{k}) + 2\dot{r}\dot{\theta}\hat{\theta} + r\ddot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \quad (1.31)$$

where  $a_{A/O} = 0$ ,  $(\ddot{r}\hat{r} + \ddot{z}\hat{k}) = 0$  and  $r\ddot{\theta}\hat{\theta} = 0$ , therefore reducing the above equation to:

$$a_{B/O} = 2\dot{r}\dot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \quad (1.32)$$

Now, its all about finding the forces using Newton's second law  $\sum F_{ext} = ma_{B/O}$

$$\sum F_{ext}\hat{r} = ma_{B/O} \cdot \hat{r} = -mr\dot{\theta}^2\hat{r} \quad (1.33)$$

$$\sum F_{ext} \hat{\theta} = ma_{B/O} \cdot \hat{\theta} = 2m\dot{r}\dot{\theta}\hat{\theta} \quad (1.34)$$

here,  $-mr\dot{\theta}^2\hat{r}$  is the centripetal force, this is a necessary force that is required to maintain a circular motion. The other force  $2m\dot{r}\dot{\theta}\hat{\theta}$  is a Coriolis force, the existence of this force can be explained using *law of conservation of momentum*.

### 1.5.1 Explaining Coriolis force using Law of conservation of angular momentum

Using the definition of angular momentum, we get:

$$h_{B/O} = r_{B/O} \times p_{B/O} = r_{B/O} \times mv_{B/O} \quad (1.35)$$

$$h_{B/O} = r\hat{r} \times (m\dot{r}\hat{r} + r\dot{\theta}\hat{\theta}) = mr^2\dot{\theta}\hat{k} \quad (1.36)$$

torque is defined as the time-derivative of angular momentum:

$$\frac{d}{dt} h_{B/O} = 2mrr\dot{\theta}\hat{k} = r_{B/O} \times F_{cor} \quad (1.37)$$

it can be seen that the  $F_{cor}$  term in the above equation is nothing but the term  $2m\dot{r}\dot{\theta}\hat{\theta}$  from equation of  $\sum F_{ext} \hat{\theta}$ . Therefore, this Coriolis force is the force that comes into action when a particle / body in translation is also under rotation due to the frame that it is fixed to. Similar to centripetal force, Coriolis force is also a necessary force that is required to conserve the angular momentum in case of problems concerning 1D rotation with 1D translation to it.

## 1.6 Rotation Problem 2

Consider a problem as shown in figure 1.9. Unlike problem 1, the nut is here replaced with a ball that is not constrained so additionally we have  $\ddot{r}$ , ie., the ball can accelerate due to the rotation of the pipe as it is no more constrained by the centripetal acceleration. the arrow pointing out of the ball is the velocity  $\dot{r}$  along  $\hat{r}$  direction.

## Nomenclature

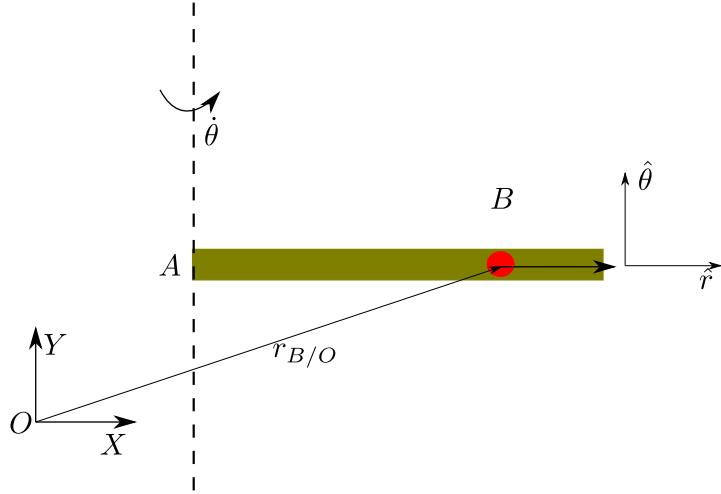


Figure 1.9: Rotation problem 2

applying velocity formula for the polar coordinates:

$$v_{B/O} = \dot{r}\hat{r} + r\dot{\theta}\hat{\theta} \quad (1.38)$$

acceleration:

$$a_{B/O} = \ddot{r}\hat{r} + 2\dot{r}\dot{\theta}\hat{\theta} + r\ddot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \quad (1.39)$$

The forces acting on the ball along the two directions are:

$$\sum F_{\hat{r}} = m(\ddot{r} - r\dot{\theta}^2)\hat{r} \quad (1.40)$$

$$\sum F_{\hat{\theta}} = m(2\dot{r}\dot{\theta} + r\ddot{\theta})\hat{\theta} \quad (1.41)$$

consider  $\sum F_{\hat{r}}$ , it can be seen that since  $m \neq 0$ , the expression follows as  $\ddot{r} = r\dot{\theta}^2$ . So basically, the acceleration along the  $\hat{r}$  direction is equal to the centripetal acceleration. In other words, in absence of centripetal acceleration, the ball will now accelerate by a quantity of  $r\dot{\theta}^2$ . Which means that  $r\dot{\theta}^2$  is the required centripetal acceleration if the ball is supposed to be constrained to move along a circular path. Therefore, centripetal acceleration is a necessary condition for an object that is moving along a circular path.

## 1.7 Rotation Problem 3

Consider figure 1.10.

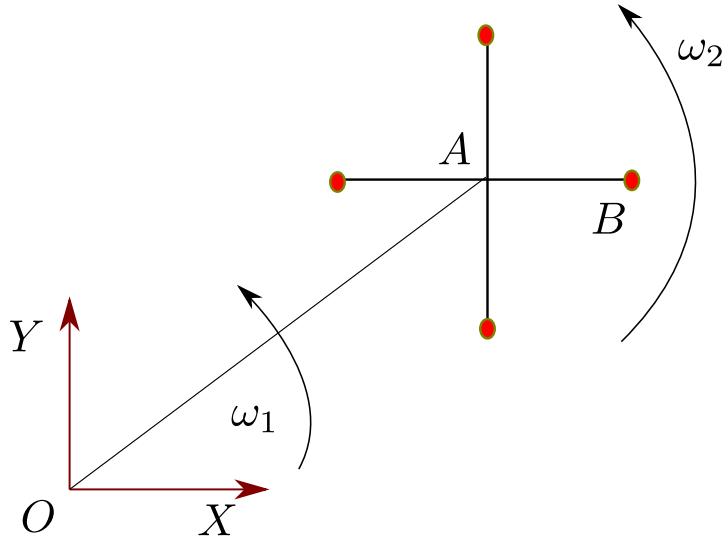


Figure 1.10: Rotation Problem 3

Frames:

$$\begin{aligned} F &: \{O, \hat{i}_1, \hat{j}_1, \hat{k}_1\} \\ arm &: \{c, \hat{i}_2, \hat{j}_2, \hat{k}_2\} \\ A &: \{A, \hat{i}_3, \hat{j}_3, \hat{k}_3\} \end{aligned}$$

Angular velocity of B w.r.t A:

$$\omega_{B/O} = (\omega_2 + \omega_1)\hat{k} \quad (1.42)$$

velocity:

$$v_{B/O} = v_{A/O} + v_{B/A} + \omega_{B/O} \times r_{B/A} \quad (1.43)$$

further determining  $v_{A/O}$ : (Not shown in the diagram A is the center of the rotation spokes, but the arm to which A is fixed has its own frame, lets call this frame center as c and let it coincide with point O). Let R be the arm length:

$$v_{A/O} = v_{c/O} + v_{A/c} + \omega_1 \hat{k}_1 \times r_{A/c} = 0 + 0 + \omega_1 \hat{k}_1 \times R \hat{i}_1 = R \omega_1 \hat{j}_1 \quad (1.44)$$

simplifying the above equation

$$v_{B/O} = 0 + R \omega_1 \hat{j}_1 + (\omega_2 + \omega_1) \hat{k} \times l \hat{i}_2 = R \omega_1 \hat{j}_1 + l(\omega_2 + \omega_1) \hat{j}_2 \quad (1.45)$$

## 1.8 Derivative on angular velocity $\omega$

Consider figure 1.11. The axis-y of both the wheel and the cross are aligned with zero angle between them.

## Nomenclature

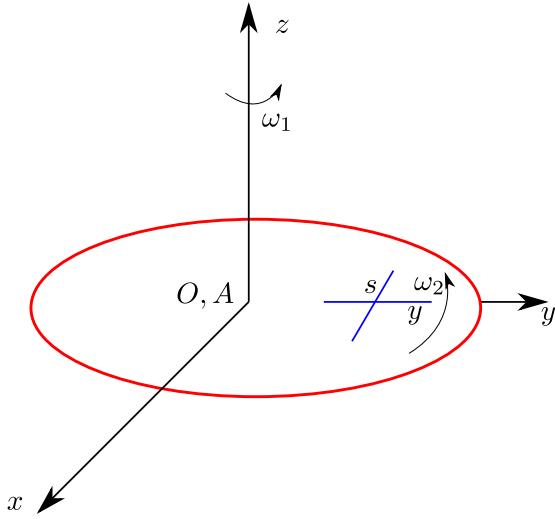


Figure 1.11: Angular velocity with zero tilt angle

In such a case, in order to find the angular velocity of  $s$ , both  $\omega_1$  and  $\omega_2$  can be added directly, as  $\omega$  is a vector and a linear operation is possible.

**Note:** This is not possible with angles as they are not vectors and need to be added up using DCM's or related methods using DCM's such as Euler angles.

Let the frames of the wheel and the cross be respectively as follows:

$$F : \{O, \hat{I}, \hat{J}, \hat{K}\}$$

$$C : \{A, \hat{i}, \hat{j}, \hat{k}\}$$

The angular velocity  $\omega_s$  is expressed using addition theorem (not covered in this book):

$$\omega_s = \omega_2 \hat{j} + \omega_1 \hat{I} \quad (1.46)$$

**Note:** Care should be taken to do time-derivative on any vectors that belong to  $s$  as  $s$  is attached to a moving frame  $F$ .

differentiating above equation w.r.t frame F:

$${}^F \frac{d}{dt} \omega_s = \dot{\omega}_s = \frac{d}{dt} (\omega_2 \hat{j}) + \frac{d}{dt} (\omega_1 \hat{K}) \quad (1.47)$$

where

- $\frac{d}{dt} (\omega_2 \hat{j})$  can be obtained using transport theorem as this is a vector attached to a moving frame A
- $\frac{d}{dt} (\omega_1 \hat{K}) = \dot{\omega}_1 \hat{K}$ , as derivative of  $\hat{K}$  w.r.t frame F will be zero

further using transport theorem:

$$\frac{d}{dt} \left( \omega_2 \hat{j} \right) = \dot{\omega}_2 \hat{j} + \omega_1 \hat{K} \times \omega_2 \hat{j} = \dot{\omega}_2 \hat{j} - \omega_1 \omega_2 \hat{i} \quad (1.48)$$

**Note:** Transport theorem (TT) works this way, the first term in TT is the term which has time derivative of a scalar (i.e., only a quantity) with no rotations associated with it (therefore, no unit vectors derivative) and a second term due to rotations (where unit vectors are time-differentiated), therefore, associating (unit vectors) with the rotation vector.

since unit vectors  $\hat{K}$  and  $\hat{k}$  are aligned, the above cross-product can be performed directly.

using equation (1.48) in equation 1.47, we get:

$$\dot{\omega}_s = \dot{\omega}_2 \hat{j} - \omega_1 \omega_2 \hat{i} + \dot{\omega}_1 \hat{K} \quad (1.49)$$

*Nomenclature*

## 2 Impulse, Torque and Angular Momentum for a system of particles

### 2.1 Linear Impulse and Momentum

Newton's second law for a single particle can be expressed as:

$$F_{ext} = ma = m \frac{d}{dt}v \quad (2.1)$$

for a system of particles,

$$\sum F_{ext} = ma = m \frac{d}{dt}v \quad (2.2)$$

Linear impulse is a accumulation of forces acting on each of the particle, this can be found using integral:

$$\int_{t_1}^{t_2} \sum F_{ext} dt = \int_{v_1}^{v_2} mdv \quad (2.3)$$

simplifying the above equation:

$$\sum \int_{t_1}^{t_2} F_{ext} dt = mv_2 - mv_1 \quad (2.4)$$

or in a standard form, this can be expressed as

$$mv_1 + \sum \int_{t_1}^{t_2} F_{ext} dt = mv_2 \quad (2.5)$$

#### 2.1.1 Problem on Impulse - Momentum

Consider figure 2.1.

## Nomenclature

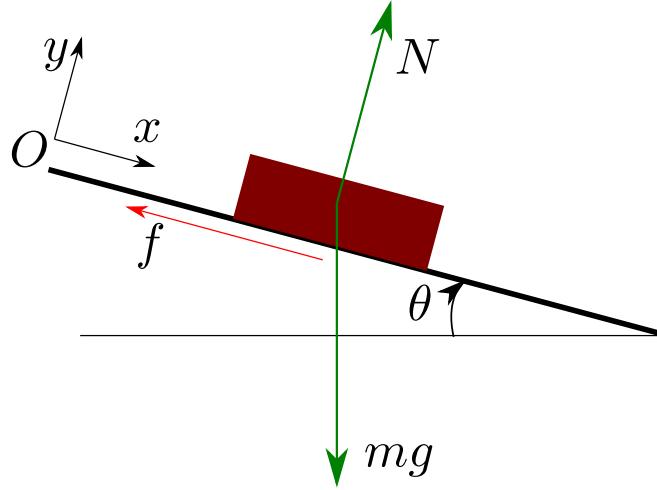


Figure 2.1: Impulse - Momentum Problem

summation on the external forces leads to

$$\sum F_x = (mg \sin \theta - \mu mg \cos \theta) = k \quad (2.6)$$

Let initial velocity  $v_{x1} = 0$  at time  $t = 0$ . equation (2.6) is a constant in time, therefore it can be expressed using a single constant character  $k$ . Using impulse-momentum equation given by equation (2.5) and substituting for  $\sum F_{ext}$ , we get:

$$\int_0^t k dt = mv_{2x} \implies kt \Big|_0^t = mv_{2x} \implies v_{2x} = \frac{kt}{m} \quad (2.7)$$

## 2.2 Momentum for system of particles

For a system of particles, with  $i$  number of particles, the center of mass of all the system of particles can be expressed by  $m_T$ . Using the center of mass definition, the momentum of all the particles can be expressed as:

$$m_T v_{G/O} = \sum_i m_i v_{i/O} \quad (2.8)$$

taking time-derivative:

$$m_T \dot{v}_{G/O} = \sum_i m_i \frac{d}{dt} v_{i/O} \implies m_T \dot{v}_{G/O} = \sum F_{i/ext} \quad (2.9)$$

to determine impulse, taking the time integral on both sides of the equation:

$$\int_{t_1}^{t_2} \sum F_{i/ext} dt = int_{t_1}^{t_2} m_T \frac{d}{dt} v_{G/O} dt \implies \int_{t_1}^{t_2} \sum F_{i/ext} dt = m_T \left[ v_{G/O} \Big|_1 - v_{G/O} \Big|_2 \right] \quad (2.10)$$

### 2.2.1 Problem on momentum on system of particles

Consider figure 2.2.

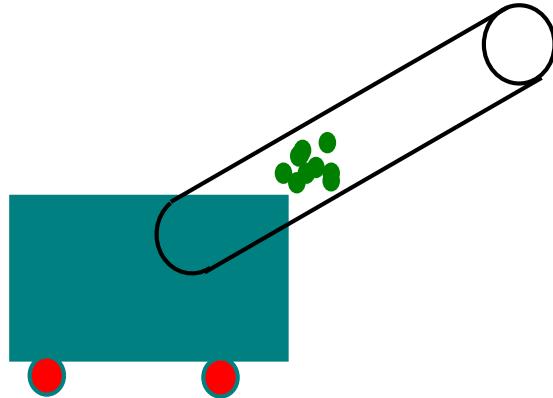


Figure 2.2: Momentum problem on system of particles

For the system of particles (green) inside the canon, the reaction force acting on the canon due to an explosive denoting the particles can be determined using the momentum equation for the system of particles given by equation (2.10).

Let the total mass of the shot (system of particles) is  $10\text{kg}$ , length of the barrel  $7\text{m}$  and the exit velocity of the shot  $215\text{m/s}$ .

As the shot was idle initially, the initial velocity is  $v_0 = 0\text{m/s}$ , final velocity  $v_f = 215\text{m/s}$ , we can find average velocity as

$$v_{avg} = (v_f - v_0)/2 \implies v_{avg} = 107.5\text{m/s} \quad (2.11)$$

further average velocity is also:

$$v_{avg} = \frac{\text{distance}}{\Delta t} \implies \Delta t = \frac{7\text{m}}{107.5\text{m/s}} = 0.065\text{s} \quad (2.12)$$

using the momentum equation (2.10):

$$\int_{t_1}^{t_2} \sum F_{i_{ext}} dt = \frac{10\text{kg}}{9.81\text{m/s}^2} (215\text{m/s}) \implies F_{i_{ext}} \Delta t = \frac{10\text{kg}}{9.81\text{m/s}^2} (215\text{m/s}) \quad (2.13)$$

which simplifies to:

$$F_{i_{ext}} = \frac{10\text{kg}}{9.81\text{m/s}^2 * \Delta t} (215\text{m/s}) = 3.65\text{kN} \quad (2.14)$$

## 2.3 Angular Momentum of a particle

Angular momentum is defined as the moment of the momentum. Consider figure 2.3, let particle B subjected to forces  $f_B$ , B is also translating so it has a momentum given

## Nomenclature

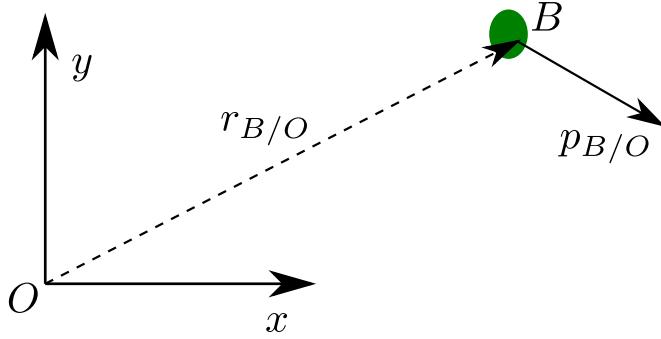


Figure 2.3: Angular momentum of B w.r.t O

by  $p_{B/O}$ . **Note:** Linear momentum is always expressed w.r.t the inertial frame. From the definition of angular momentum,

$$h_{B/O} = r_{B/O} \times p_{B/O} \quad (2.15)$$

torque is defined as the rate of change angular momentum as net total of all the external torques acting on the system,

$$\frac{d}{dt}(h_{B/O}) = \sum \tau_{B/O} \quad (2.16)$$

Lets consider an arbitrary point A, if we compute angular momentum of B w.r.t A as

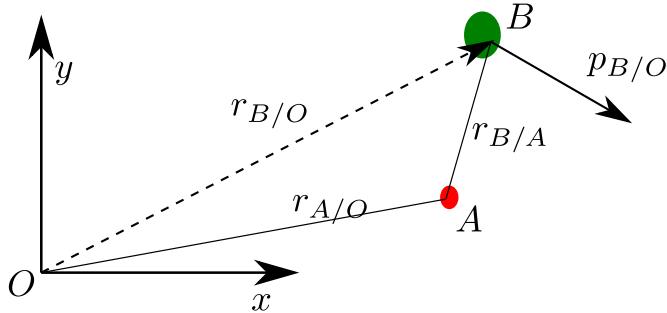


Figure 2.4: Angular momentum of B w.r.t A

shown in figure 2.4, the linear momentum is always w.r.t the inertial frame

$$h_{B/A} = r_{B/A} \times p_{B/O} \quad (2.17)$$

torque required for this system in order to change the angular momentum can be determined as follows:

Let the arm length between A and B be fixed (like in Robotics) such that there is no translation between B and A. In this case, the time derivative can be calculated as follows:

$$\sum \tau_{B/A} = \frac{d}{dt}(r_{B/A} \times p_{B/O}) = r_{B/A} \times \frac{d}{dt}p_{B/O} \quad (2.18)$$

Note that equation (2.18) is of the following form

$$\sum \tau_{B/A} = r_{B/A} \times f_B = r_{B/A} \times \frac{d}{dt} p_{B/O} = Q \quad (2.19)$$

in this case, the net effect of all the forces acting at B results in change in momentum i.e.  $p_{B/O}$ . Further from mathematics,  $Q$  is of the form:

$$Q = a \times \frac{d}{dt} b = \frac{d}{dt} (a \times b) - \frac{d}{dt} a \times b \quad (2.20)$$

using equation (2.20), equation (2.19) can be expressed as

$$\sum \tau_{B/A} = \frac{d}{dt} (r_{B/A} \times p_{B/O}) - \frac{d}{dt} r_{B/A} \times p_{B/O} \quad (2.21)$$

the terms of equation (2.19) are solved as follows

$$\frac{d}{dt} (r_{B/A} \times p_{B/O}) = \frac{d}{dt} (h_{B/A}) \quad (2.22)$$

$$r_{B/A} = r_{B/O} - r_{A/O} \implies v_{B/A} = v_{B/O} - v_{A/O} \quad (2.23)$$

using equations ((2.22) and (2.23)) and substituting in equation (2.19), we get the following

$$\sum \tau_{B/A} = \frac{d}{dt} h_{B/A} - \frac{d}{dt} (v_{B/O} - v_{A/O}) \times p_{B/O} \quad (2.24)$$

further, in equation (2.24), both  $v_{B/O}$  and  $p_{B/O}$  are pointing in the same direction as the linear momentum is always pointing the same direction as the linear velocity, the cross product of these will result in a zero, leaving behind only the cross product  $v_{A/O} \times p_{B/O}$  as follows

$$\sum \tau_{B/A} = \frac{d}{dt} h_{B/A} + \frac{d}{dt} v_{A/O} \times p_{B/O} \quad (2.25)$$

equation (2.25) is the expressing relating the net effect of torques and change in angular momentum at B w.r.t to an arbitrary point A. Equation (2.25) can now be further simplifies based on the application on the following cases:

- case 1: when  $v_{A/O} = 0$ , i.e, point A is always fixed
- case 2: when  $v_{A/O}$  and  $p_{B/O}$  are parallel, happens in cases when A is actually the center of mass for either the system of particles or for that of a rigid body

In both such cases, equation (2.25) reduces to

$$\sum \tau_{B/A} = \frac{d}{dt} h_{B/A} \quad (2.26)$$

equation (2.26) is the most simplified form of torque and angular momentum relationships, this form is most used for problems related to motion of wheels in robotics and automotive applications.

## Nomenclature

### 2.3.1 Angular momentum and torque problem for a particle

Consider figure 2.5, where B is a ride which can go round and round with angular velocity  $\omega = \dot{\theta}$  as well as translate with velocity  $\dot{r}$ , let both these velocities are constant. In this problem, as there are both the velocity terms  $\dot{\theta}$  and  $\dot{r}$ , B experiences both the centripetal accelerations due to term  $r\dot{\theta}^2$  and Coriolis acceleration due to the term  $\dot{r}\dot{\theta}$ . Further, as angular momentum is expressed as  $r_{B/O} \times p_{B/O}$ , as  $r$  is changing, the angular momentum is changing. Additionally, as  $v_{B/O}$  is changing due to  $\dot{r}$ , whenever,  $dot{r}$  changes direction, linear momentum also changes direction, therefore,  $p_{B/O}$  is also changing.

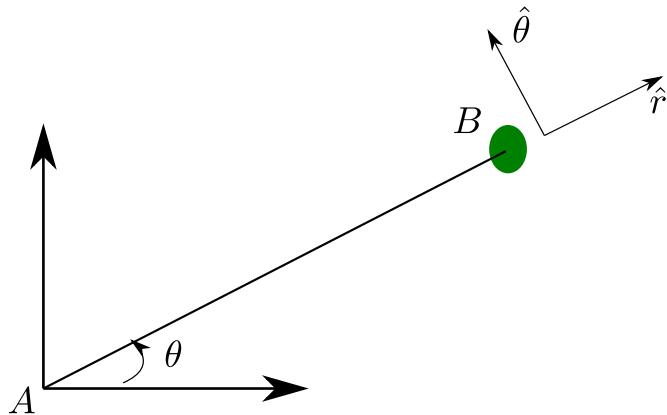


Figure 2.5: Angular momentum problem

#### 2.3.1.1 Angular momentum

velocity of the particle B is expressed as

$$v_{B/O} = \dot{r}\hat{r} + r\dot{\theta}\hat{\theta} \quad (2.27)$$

angular momentum is expressed using equation (2.15):

$$h_{B/A} = r_{B/A} \times p_{B/O} = r\hat{r} \times m(\dot{r}\hat{r} + r\dot{\theta}\hat{\theta}) \quad (2.28)$$

simplifying the above equation we get:

$$h_{B/A} = mr^2\dot{\theta}\hat{k} \quad (2.29)$$

#### 2.3.1.2 Torque

This is the case, when  $v_{A/O} = 0$ , therefore, using equation (2.26)

$$\sum \tau_{B/A} = \frac{d}{dt}h_{B/A} = \frac{d}{dt}(mr^2\dot{\theta}\hat{k}) = 2mr\dot{r}\dot{\theta}\hat{k} + mr^2\ddot{\theta}\hat{k} \quad (2.30)$$

there are two terms in the above equation, which can be defined as follows. The torque equation can be expressed as

$$\sum \tau_{B/A} = r \times F = rr \hat{r} \times (2mr\dot{r}\dot{\theta}\hat{\theta} + mr^2\ddot{\theta}\hat{\theta}) \quad (2.31)$$

the term  $2mr\dot{r}\dot{\theta}\hat{\theta}$  is the Coriolis force and the term  $mr^2\ddot{\theta}\hat{\theta}$  is due to the Euler's acceleration. In order to find centripetal acceleration, (which is not done in this problem) as the centripetal acceleration is acting the  $\hat{r}$  direction, the sum effect of forces along  $\hat{r}$  direction has to be performed.

## 2.4 Angular momentum and Torque for rigid bodies

Using equation (2.25), for rigid bodies, the torque is expressed as

$$\sum \tau_{B/A} = \frac{d}{dt} H_{/A} + \frac{d}{dt} v_{A/O} \times P_{G/O} \quad (2.32)$$

where  $H_{B/A}$  and  $P_{G/O}$  are the angular momentum and linear momentum of a rigid body. The cases used for simplification are also same in this case, therefore, reducing the equation (2.32) as follows

$$\sum \tau_{B/A} = \frac{d}{dt} H_{/A} \quad (2.33)$$

### 2.4.1 Torque and angular momentum Problem 2

Consider the system as shown in figure 2.6. Let  $r_{B/A} = rr\hat{r}$ . The velocity of B can be expressed in polar coordinates as

$$v_{B/O} = v_{A/O} + v_{B/A} + \omega \times r_{B/A} = 0 + \dot{r}\hat{r} + \dot{\theta}\hat{k} \times rr\hat{r} \quad (2.34)$$

Assuming  $\dot{r}$  to be constant, the term  $\dot{r}\hat{r} = 0$ , therefore, reducing the above equation to

$$v_{B/O} = r\dot{\theta}\hat{\theta} \quad (2.35)$$

using the above equation, the linear momentum  $p_{B/O}$  can be expressed as

$$p_{B/O} = mv_{B/O} = m(r\dot{\theta}\hat{\theta}) \quad (2.36)$$

further angular momentum w.r.t A can be expressed as (again  $r_{B/A} = r_{B/O}$ )

$$h_{B/A} = r_{B/A} \times p_{B/O} = rr\hat{r} \times mr\dot{\theta}\hat{\theta} = mr^2\dot{\theta}\hat{k} \quad (2.37)$$

## Nomenclature

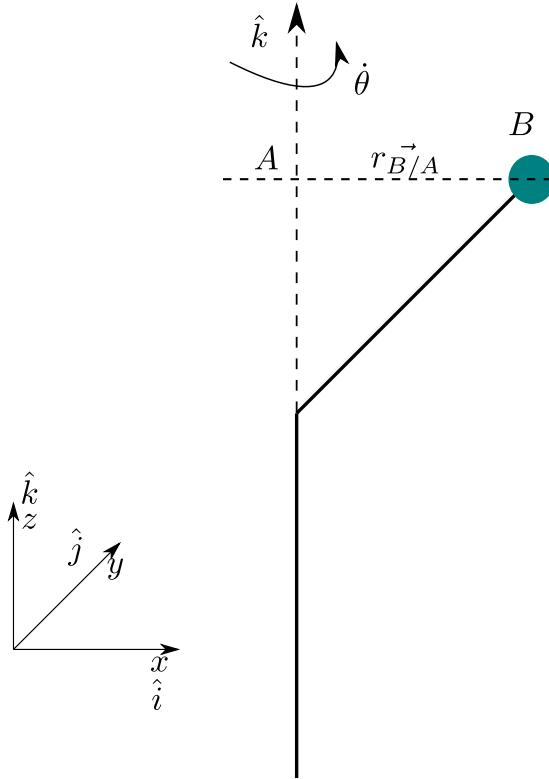


Figure 2.6: Problem 2 on Torque and Angular Momentum

This is a case when the velocity  $v_{A/O} = 0$  in equation (2.25), therefore, using equation (2.26)

$$\tau_{B/A} = \frac{d}{dt} h_{B/A} = \frac{d}{dt} (mr^2 \dot{\theta} \hat{k}) = 2mr\dot{r}\dot{\theta} \hat{k} + mr^2 \ddot{\theta} \hat{k} \quad (2.38)$$

again as  $\dot{r} = 0$ , the above equation reduces to

$$\tau_{B/A} = mr^2 \ddot{\theta} \hat{k} \quad (2.39)$$

writing the above torque equation in the form  $\tau = r \times F$

$$\tau_{B/A} = r_{B/A} \times F = r\hat{r} \times (mr^2 \ddot{\theta} \hat{\theta}) \quad (2.40)$$

therefore, the above force comes from the Euler's acceleration acting in the direction of  $\hat{\theta}$ .

Consider now another case, where the reference axis for calculating the distance is chosen as shown in figure 2.7

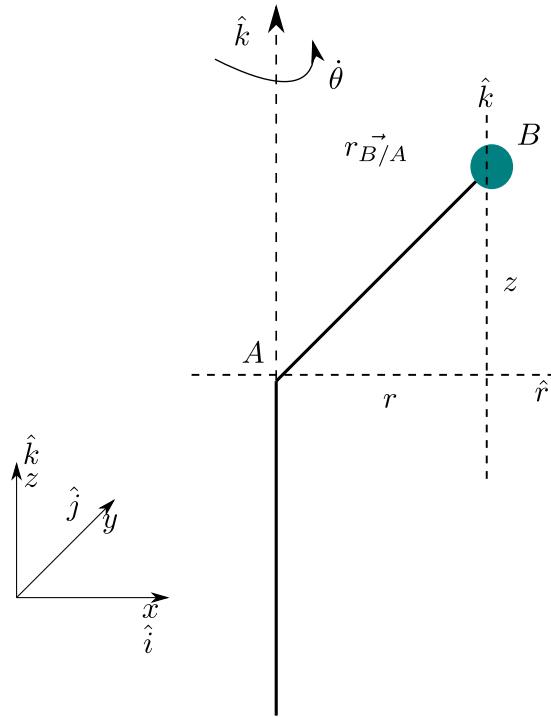


Figure 2.7: Problem 2 on Torque and Angular Momentum, case 2

in this case now the vector  $r_{B/A} = rr\hat{r} + z\hat{k}$ . Starting from velocity of B

$$v_{B/O} = v_{B/A} + \omega \times r_{B/A} = \frac{d}{dt} (rr\hat{r} + z\hat{k}) + \dot{\theta}\hat{k} \times (rr\hat{r} + z\hat{k}) = r\dot{\theta}\hat{\theta} \quad (2.41)$$

linear momentum is now given by

$$p_{B/O} = mv_{B/O} = mr\dot{\theta}\hat{\theta} \quad (2.42)$$

further angular momentum is now expressed as

$$h_{B/A} = r_{B/A} \times p_{B/O} = (rr\hat{r} + z\hat{k}) \times (mr\dot{\theta}\hat{\theta}) \quad (2.43)$$

simplifying the above equation, we get

$$h_{B/A} = mr^2\dot{\theta}\hat{k} - mrz\dot{\theta}\hat{r} \quad (2.44)$$

again using equation (2.26) as in the earlier case

$$\tau_{B/A} = \frac{d}{dt} h_{B/A} = \frac{d}{dt} (mr^2\dot{\theta}\hat{k} - mrz\dot{\theta}\hat{r}) \quad (2.45)$$

simplifying the above equation

$$\tau_{B/A} = 2mrr\dot{\theta}\hat{k} + mr^2\ddot{\theta}\hat{k} - mrz\dot{\theta}\hat{r} - m\dot{r}z\dot{\theta}\hat{r} - mrz\ddot{\theta}\hat{r} - mrz\dot{\theta}\dot{\hat{r}} \quad (2.46)$$

## Nomenclature

again  $\dot{r} = \dot{z} = 0$  in the above expression, therefore,

$$\tau_{B/A} = mr^2\ddot{\theta}\hat{k} - mrz\ddot{\theta}\hat{r} - mrz\dot{\theta}\dot{\hat{r}} \quad (2.47)$$

where  $\dot{\hat{r}} = \dot{\theta}\hat{\theta}$ , therefore, the above equation is now expressed as

$$\tau_{B/A} = mr^2\ddot{\theta}\hat{k} - mrz\ddot{\theta}\hat{r} - mrz\dot{\theta}^2\hat{\theta} \quad (2.48)$$

writing the above torque equation in the form  $\tau = r \times F$

$$\tau_{B/A} = r_{B/A} \times F = (r\hat{r} + z\hat{k}) \times (mr\ddot{\theta}\hat{\theta} - mr\dot{\theta}\hat{\theta} - mr\dot{\theta}^2\hat{\theta}) \quad (2.49)$$

therefore, from the above equation, the following forces can be defined

1.  $mr\ddot{\theta}\hat{\theta}$  Eulerian Acceleration
2.  $mr\ddot{\theta}\hat{\theta}$  Force applied by the mechanical link for structural stability (bending moment on the direction parallel to  $\hat{\theta}$ )
3.  $mr\dot{\theta}^2\hat{\theta}$  again this is the force applied by the mechanical link for structural stability (bending moment on the direction parallel to  $\hat{r}$ )

**Note:** From this case of dynamics problem, it can be shown therefore, that angular momentum equations can not only be used to determine the dynamics of the system, but also its static behavior, provided that the axis of reference are chosen appropriately for the kind of static information to be determined. This depends on the axis and the coordinates to be chosen in a way that the problem solution is forced to find all the acting forces onto the system.

## 2.5 Exercise Problem

Consider the figure 2.8. A mass  $m$  is placed on a plate which can be tilted by an angle  $\theta$ . The angular velocity of the plate is constant  $\omega = 3\text{rad/s}$ . The block of mass  $m$  slips when  $\theta = 50^\circ$ . Derive EOM's for this problem.

### 2.5.1 Chose coordinates

Because this problem has 1D rotation, polar coordinates can reduce the complexity of the problem. The length along which the block translates if chosen along direction  $\hat{r}$  and for  $\hat{\theta}$  along the angular displacement of the plate. There is only one coordinate for the whole of the system (block + plate), where translating mass has coordinate  $\hat{r}$ , rotating plate has coordinate  $\hat{\theta}$  and the whole system is constrained along  $z$  direction.

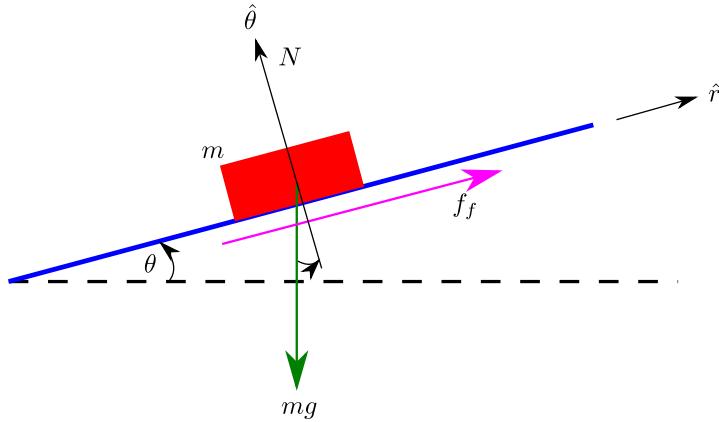


Figure 2.8: Exercise Problem on Inclined plane and polar coordinates

### 2.5.2 DOF

Among the three coordinates from the polar coordinates  $\hat{r}$ ,  $\hat{\theta}$  and  $\hat{k}$ , translation along  $\hat{k}$  is constrained, therefore, the system only has 2 DOF. For each of these 2 DOF's EOM can be derived for  $\hat{r}$  and  $\hat{\theta}$ .

### 2.5.3 Deriving EOM

For the motion along  $\hat{r}$  and  $\hat{\theta}$ , using equation (1.28) to find the accelerations along  $\hat{r}$

$$a_{B/O} = a_{A/O} + (\ddot{r}\hat{r} + \ddot{z}\hat{k}) + 2\dot{r}\dot{\theta}\hat{\theta} + r\ddot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \quad (2.50)$$

in this equation the terms with  $\ddot{z}$  and  $\ddot{\theta}$  are zero, therefore, reducing the above equation to:

$$a_{B/O} = a_{A/O} + \ddot{r}\hat{r} + 2\dot{r}\dot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \quad (2.51)$$

further point  $A$  the pivot point of the plate is also fixed, therefore the equation is reduced to

$$a_{B/O} = \ddot{r}\hat{r} + 2\dot{r}\dot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \quad (2.52)$$

using Newtons second law, the forces along  $\hat{r}$  can be expressed as

$$\sum F_{\hat{r}} = ma_{B/O} \cdot \hat{r} = m [\ddot{r} - r\dot{\theta}^2] \hat{r} \quad (2.53)$$

the forces along  $\hat{\theta}$  can be expressed as

$$\sum F_{\hat{\theta}} = ma_{B/O} \cdot \hat{\theta} = 2m\dot{r}\dot{\theta}\hat{\theta} \quad (2.54)$$

in order to determine the external forces on the system along  $\hat{r}$  direction, using figure 2.8,

$$\sum F_{\hat{r}} = -mg\sin\theta\hat{r} - f_f = -mg\sin\theta\hat{r} + \mu mg\cos\theta\hat{r} \quad (2.55)$$

## Nomenclature

where  $f_f$  is the frictional force acting in the direction opposite to the motion and is equal to the product of normal force  $N$  and the co-efficient of friction  $\mu$ .

Using the same analogy, along  $\hat{\theta}$  direction,

$$\sum F_{\hat{\theta}} = 0 \quad (2.56)$$

because there are no external torque applied to the system. Using equation (2.55) with equation (2.53), the EOM along  $\hat{r}$  and  $\hat{\theta}$  can be expressed as

$$[-mg\sin\theta + \mu mg\cos\theta] \hat{r} = m [\ddot{r} - r\dot{\theta}^2] \hat{r} \quad (2.57)$$

using this above equation of motion, any quantity can be solved for in the equation, given there is enough information about the rest of the other quantities.

For EOM along  $\hat{\theta}$ , the EOM is simply

$$\sum F_{\hat{\theta}} = 2m\dot{r}\dot{\theta}\hat{\theta} = 0 \quad (2.58)$$

this is true as in general, when a torque is applied to the system, the Coriolis force will accelerate the block along  $\hat{r}$ . Because this torque is zero, the block cannot be accelerated, it can also be shown that angular momentum is conserved (which is zero in this case, i guess) as Coriolis force is zero.

### 3 Fictitious Forces using Free Body Diagrams

The concept of Fictitious forces is something that should not be used in general practice to determine the required dynamics of the given problem statement. Fictitious forces are not actual forces, instead put in place so as to give meaning to accelerations that generate due to the dynamics of the problem while under a certain motion. For example, centrifugal (or centripetal) accelerations that are experienced by the body under circular motion is given a meaning using the concept of fictitious forces as in a way this acceleration acts as a force in order to keep the dynamics of the problem stable under a certain motion. In this case, centripetal force (or centrifugal) force keeps the body stable while it is cornering (or going around circles). It is important to always remember that these fictitious forces do not exist in reality, however, the accelerations such as centripetal accelerations do appear from the type of geometry and dynamics of the system.

The meaning to such fictitious forces is given using free body diagrams. To understand this in more detail consider figure 3.1 of a 1D rotation problem which is similar to the problem discussed in section 2.4.1. However, here the mass on the link is fixed firmly such that the motion along  $\hat{r}$  direction is constrained as follows

$$\dot{r} = \ddot{r} = 0 \quad (3.1)$$

further, the link is allowed to rotate in a variable rotational rate such that

$$\omega = \dot{\theta}\hat{k} \quad (3.2)$$

$$\dot{\omega} = \ddot{\theta}\hat{k} \quad (3.3)$$

are allowed.

## Nomenclature

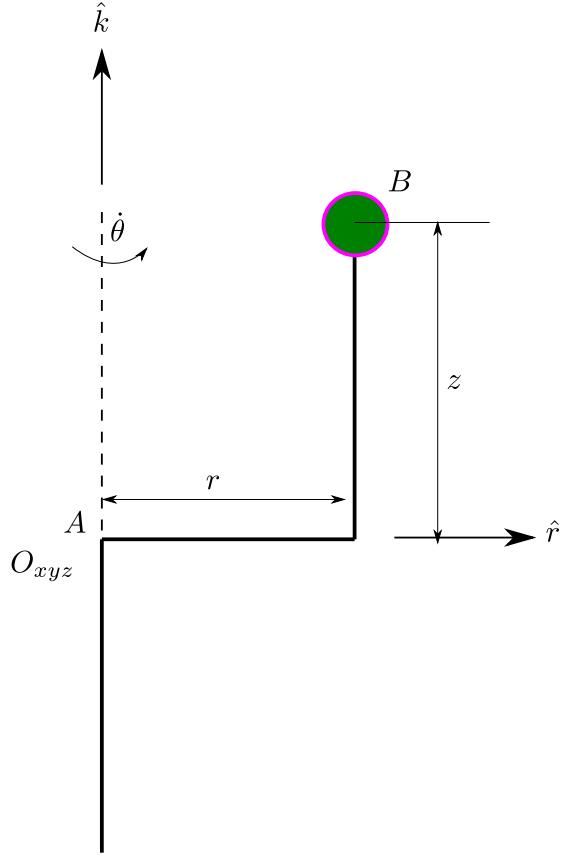


Figure 3.1: Rotation problem to describe fictitious forces

In order to find the accelerations on the system, consider equation (1.28):

$$a_{B/O} = a_{A/O} + (\ddot{r}\hat{r} + \ddot{z}\hat{k}) + 2\dot{r}\dot{\theta}\hat{\theta} + r\ddot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \quad (3.4)$$

here \$a\_{A/O} = 0\$, \$\ddot{r} = 0\$, \$\ddot{z} = 0\$, therefore, reducing the above equation to:

$$a_{B/O} = r\ddot{\theta}\hat{\theta} - r\dot{\theta}^2\hat{r} \quad (3.5)$$

using Newtons second law, the forces along \$\hat{r}\$ and \$\hat{\theta}\$ can be expressed as

$$\sum F_r = -mr\dot{\theta}^2\hat{r} \quad (3.6)$$

$$\sum F_\theta = mr\ddot{\theta}\hat{\theta} \quad (3.7)$$

clearly, the system is stable and it seems like that due to these accelerations there are forces acting on the mass \$B\$ such that these forces keep the mass stable under rotation, by keeping it in a stable circular motion. If we consider these accelerations to be labeled as forces, then we can use free body diagrams in order to explain and name them as some kind of force. Using the free body diagram as shown in figure 3.2

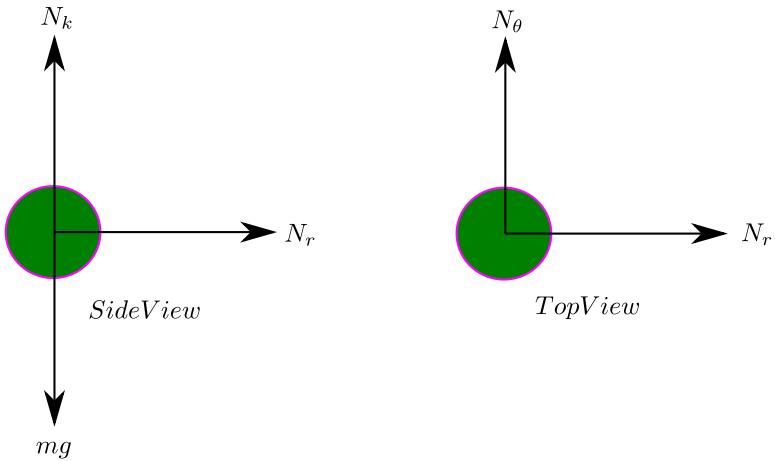


Figure 3.2: Naming fictitious forces using FBD's

Lets assume that the forces acting along the three polar coordinate directions are labeled as shown in figure 3.2, using the FBD and equation (3.6) it can be said that

$$\sum F_r = N_r = -mr\dot{\theta}^2\hat{r} \quad (3.8)$$

or that

$$\sum F_r = N_r + mr\dot{\theta}^2\hat{r} = 0 \quad (3.9)$$

which is true as there are no external forces that are acting on the system along  $\hat{r}$  direction. Note that using FBD, we considered that there is a force acting in the system  $N_r$ , which is not true as there is no force acting there. In essence,  $N_r = 0$ , so there is no force to cancel out the force due to centripetal acceleration  $-r\dot{\theta}^2\hat{r}$ . In other words, centripetal force  $-mr\dot{\theta}^2\hat{r}$  exists only because we considered  $N_r$  in the first place and we have multiplied mass to the centripetal acceleration  $-r\dot{\theta}^2\hat{r}$  to treat it as a force instead of acceleration.

It is always confusing to explain the fictitious forces, only because they don't exists in reality. What exists is only the accelerations. In some physics problems (very few old ones), sometimes people use fictitious forces in their systems to model the system easily. It becomes confusing when seen from the perspective of dynamics itself, as there is no explanation for the existing of this forces in theory.

**Note:** From torque and angular momentum equations, we would get torques required to keep the system stable, same way as we got fictitious forces required to keep the system stable when we use FBD's to name such fictitious forces and torques. In practice, it is highly recommended to avoid using the concept of fictitious forces all together.

*Nomenclature*

# 4 Deriving Equations of Motion

## 4.1 Method for deriving EOMs

1. Assign coordinates by selecting the type of system (Cartesian or polar)
2. Determine DOF
3. For each DOF there is a EOM
4. Determine the accelerations using the standard formulae from either Cartesian or Polar coordinates forms
5. Draw FBDs
6. From FBDs we get  $\sum F_{ext}$ , the  $ma_{B/O}$  part is derived from step 4
7. If there are moments in the system use formulae from angular momentum and torques to construct  $\sum \tau = dh_B/dt$

## 4.2 Exercise 1

Consider a system as shown in figure 4.1.

### 4.2.0.1 Assigning coordinates

Because this is a rotation problem, choosing polar coordinates for this system. Choosing  $\hat{r}$  along the length of the pendulum with positive side pointing away from the pendulum,  $\hat{\theta}$ , positive to CCW direction of pendulum's rotation.

### 4.2.1 DOF

Constraints on the system:

1.  $z = \dot{z} = \ddot{z} = 0$
2.  $\dot{r} = \ddot{r} = 0$

there is only one possible motion along  $\hat{\theta}$  with the pendulum pivoted to the ceiling, EOM of this problem should be constructed according to

$$\sum \tau_{ext} = \frac{d}{dt} h_{B/A} \quad (4.1)$$

## Nomenclature

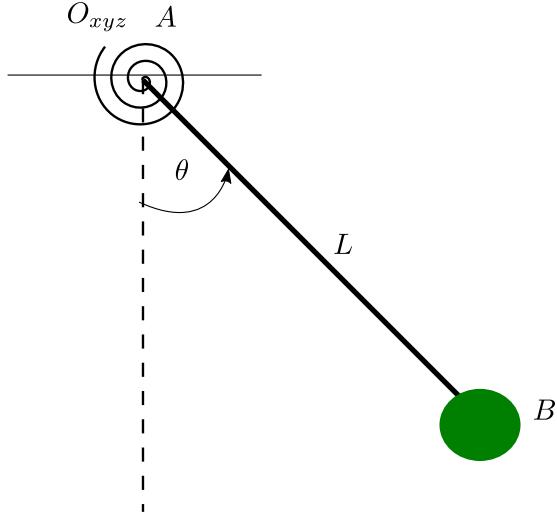


Figure 4.1: Pendulum attached to a torsional spring

### 4.2.2 Determine angular momentum the system

Angular momentum pendulum  $B$  pivoted to point  $A$  is expressed using equation

$$h_{B/A} = r_{B/A} \times p_{B/O} \quad (4.2)$$

with  $r_{B/A} = L\hat{r}$ . Determining  $p_{B/O}$ :

$$p_{B/O} = mv_{B/O} \quad (4.3)$$

$v_{B/O}$  in polar coordinates is expressed as

$$v_{B/O} = \dot{r}\hat{r} + r\dot{\theta}\hat{\theta} = L\dot{\theta}\hat{\theta} \quad (4.4)$$

substituting  $v_{B/O}$  in the equation for  $p_{B/O}$ :

$$p_{B/O} = mL\dot{\theta}\hat{\theta} \quad (4.5)$$

finally, substituting  $p_{B/O}$  into the equation for  $h_{B/A}$ :

$$h_{B/A} = L\hat{r} \times (mL\dot{\theta}\hat{\theta}) = mL^2\dot{\theta}\hat{k} \quad (4.6)$$

### 4.2.3 External Forces and Moments

In order to determine External Forces and Moments, FBDs need to be drawn for the system. As this system has two bodies (considered particles in this case), two FBDs need to be drawn as shown in figures 4.2 and 4.3

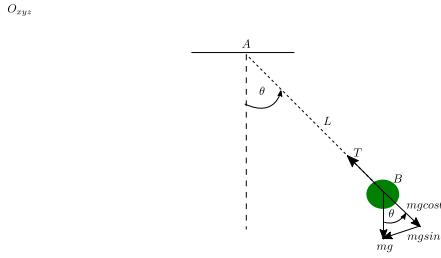


Figure 4.2: FBD for pendulum BOB

The pendulum's BOB experiences tension  $T$  from the link, this force needs to be taken into consideration.

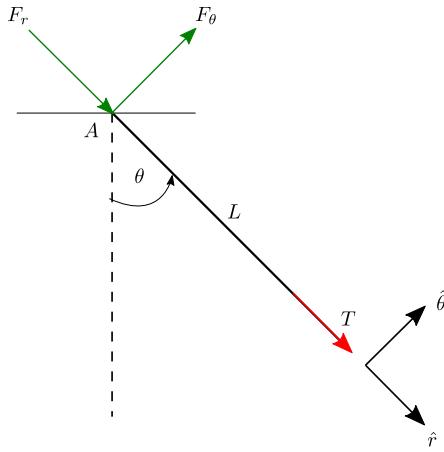


Figure 4.3: FBD for pendulum Link

The same tension  $T$  is experienced by the link in the opposite direction, this force needs to be taken into consideration. Finally, also the reaction forces from the pivot along the  $\hat{r}$  and  $\hat{\theta}$  should be considered as well.

Since in this problem, the only interest is to find EOM on  $\hat{\theta}$ , consider external moments acting about A:

- moment due to  $mgsin\theta$  as  $mgl sin \theta$  in the direction opposite to  $\hat{\theta}$ , acting about  $\hat{k}$
- moment due to spring force, consider linear with  $K_t$  as the spring constant:  $K_t\theta$  in the direction opposite to  $\hat{\theta}$ , acting about  $\hat{k}$

there are no other moments produced at point A. therefore,

$$\sum \tau_A = -mgl sin \theta \hat{k} - K_t \theta \hat{k} \quad (4.7)$$

## Nomenclature

### 4.2.4 Complete EOM

writing both sides of the EOM

$$-mglsin\theta\hat{k} - K_t\theta\hat{k} = \frac{d}{dt} (mL^2\dot{\theta}\hat{k}) \quad (4.8)$$

solving the above equation for  $\ddot{\theta}$ ,

$$\ddot{\theta} = \frac{-K_t\theta - mglsin\theta}{ml^2} \quad (4.9)$$

## 4.3 Exercise 2 - self solved

Consider 1D rotation and 1D translating object as shown in figure 4.4.

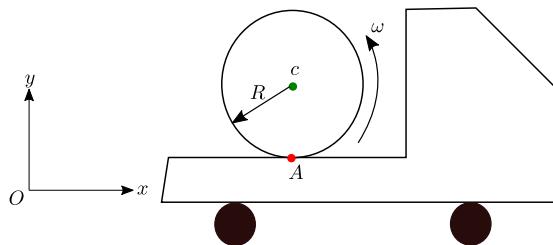


Figure 4.4: FBD for pendulum Link

### 4.3.1 Choosing coordinates

As this problem is a 1D rotation and 1D translating problem, polar coordinates would suit best for this problem. Choosing  $\hat{r}$  positive along the trucks forward motion and  $\hat{\theta}$  positive CCW to cylinders roll. Note that the cylinder will move only when the truck is either accelerating or decelerating.

### 4.3.2 DOF

There are only two possible motions for the cylinder translation  $r$  and rotation  $\theta$ , there will be 2 EOMs. Since this is a rotation problem, equation can be formed of the form

$$\sum \tau_{ext} = \frac{d}{dt} h_{B/A} \quad (4.10)$$

### 4.3.3 Determining angular momentum

The focus is only on the cylinder, using polar coordinates, the velocity of point A is to be determined. Here the cylinder can only roll about G, therefore, point A is considered as a point fixed w.r.t to a translating point G. using the velocity equation

$$v_{A/O} = \dot{r}\hat{r} + r\dot{\theta}\hat{\theta} \quad (4.11)$$

where  $\dot{r}$  is the translating velocity of point G.

$$p_{A/O} = mv_{A/O} = m(\dot{r}\hat{r} + r\dot{\theta}\hat{\theta}) \quad (4.12)$$

further angular momentum about G

$$h_{A/G} = r_{A/G} \times p_{A/O} = R\hat{r} \times m(\dot{r}\hat{r} + r\dot{\theta}\hat{\theta}) \quad (4.13)$$

As there in this case A is translating w.r.t some fixed frame, torque equation of the following form needs to be considered

$$\sum \tau_{ext} = \frac{d}{dt}h_{A/G} + \frac{d}{dt}v_{G/O} \times p_{A/O} \quad (4.14)$$

### 4.3.4 Determining external moments

Consider the case when the truck is decelerating, there is a moment produced by the frictional forces about G, which is expressed as

$$\sum \tau_G = -f_f R\hat{k} \quad (4.15)$$

### 4.3.5 Finishing EOM

$$\begin{aligned} -f_f R\hat{k} &= \frac{d}{dt}[R\hat{r} \times m(\dot{r}\hat{r} + R\dot{\theta}\hat{\theta})] + \frac{d}{dt}(\dot{r}\hat{r}) \times m(\dot{r}\hat{r} + R\dot{\theta}\hat{\theta}) \\ -f_f R\hat{k} &= mR^2\dot{\theta}\hat{k} + (\ddot{r}\hat{r} + \dot{r}\dot{\theta}\hat{\theta}) \times m(\dot{r}\hat{r} + R\dot{\theta}\hat{\theta}) \\ -f_f R\hat{k} &= mR^2\dot{\theta}\hat{k} + \ddot{r}R\dot{\theta}\hat{k} - \dot{r}^2\dot{\theta}\hat{k} \end{aligned}$$

Solving the above for  $\ddot{r}$ :

$$\ddot{r} = \frac{\dot{r}^2\dot{\theta} - mR^2\dot{\theta} - f_f R}{R\dot{\theta}} \quad (4.16)$$

## 4.4 Exercise 3

Consider a translation problem as shown in figure 4.5.

## Nomenclature

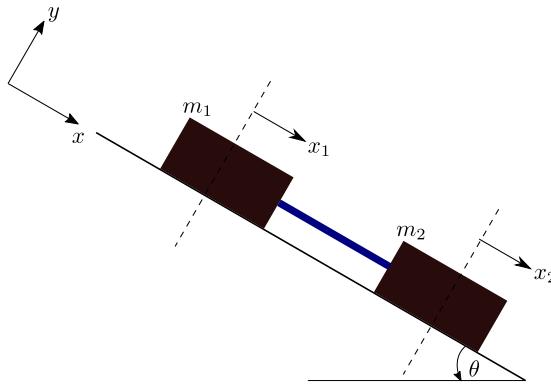


Figure 4.5: Two sliding masses

### 4.4.1 Choosing coordinates

There are two translating coordinates in this system from the 2 mass, choosing Cartesian coordinate system as shown in figure 4.5.

### 4.4.2 DOF

consider constraints

- $y_1 = y_2 = 0$
- $z_1 = z_2 = 0$
- $\omega_{x1} = \omega_{y1} = \omega_{z1} = \omega_{x2} = \omega_{y2} = \omega_{z2} = 0$

When in Cartesian coordinates, the following formula for DOF can be used

$$DOF = 6m + 3n - C \quad (4.17)$$

where \$m\$ is the number of bodies, \$n\$ is the number of particles (have only 3 translation DOF) and \$C\$ is the number of constraints. In our case, there 10 constraints as listed above along with another constraint due to **Holonomic Constraint** between the two bodies connected though a link such that  $x_1 = x_2$ . Therefore, allowing for only one DOF. Because of this Holonomic Constraint, only one EOM is possible for this system.

### 4.4.3 Determining external forces on the system

Consider figures 4.64.7 which shows the FBDs for the two bodies.

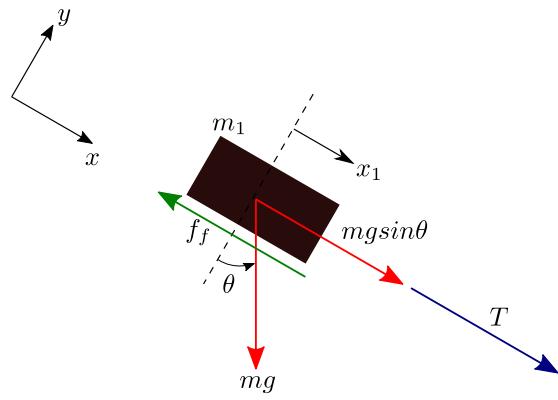


Figure 4.6: FBD mass 1

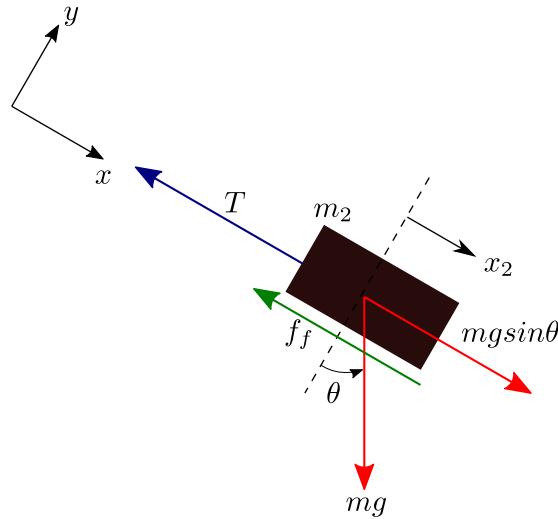


Figure 4.7: FBD mass 2

Since  $x_1 = x_2$ , consider mass 1 for finding the EOM along x:

$$\sum F_{x1} = T + mg \sin \theta - \mu mg \cos \theta \quad (4.18)$$

#### 4.4.4 Determining accelerations along x

There are no rotations in this problem, there is only one component for acceleration along x as  $\ddot{x} = \ddot{x}_1 = \ddot{x}_2$ . Therefore, only absolute velocity term and no relative velocity terms and terms due to time differentiation of unit vectors.

#### 4.4.5 Finishing EOM

$$T + mg \sin \theta - \mu mg \cos \theta = m \ddot{x}_1 \quad (4.19)$$

## Nomenclature

solving for  $\ddot{x}_1$

$$\ddot{x}_1 = \frac{1}{m} (T + mgsin\theta - \mu mgcos\theta) \quad (4.20)$$

## 4.5 Exercise 4

Consider figure 4.8.

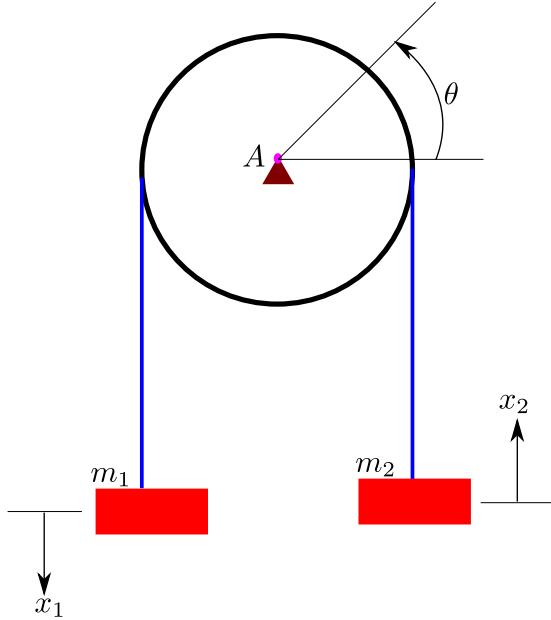


Figure 4.8: Two mass suspension system

### 4.5.1 Coordinates

Coordinates are chosen in Cartesian as shown in figure 4.8.

### 4.5.2 DOF

Using the formula  $DOF = 6 \cdot m + 3 \cdot n - C$ , there are 3 masses, no particles. Consider constraints

1.  $x_1 = z_1 = y_1 = z_2 = y_2 = z_3 = y_3 = 0$
2.  $M_{x1} = M_{y1} = M_{x2} = M_{y2} = M_{z2} = M_{x3} = M_{y3} = M_{z3} = 0$

Additionally, there is a holonomic constraint such that,  $x_1 = x_2$ . Therefore, leading to 2 DOFs. The EOM should be formed as

$$\sum F_{ext} = ma_{x1/O} \quad or \quad \sum F_{ext} = ma_{x2/O} \quad (4.21)$$

$$\sum M_{ext} = \frac{d}{dt} h_{A/O} \quad (4.22)$$

### 4.5.3 External forces

Consider the FBDs of the two masses as shown in the figure 4.9.

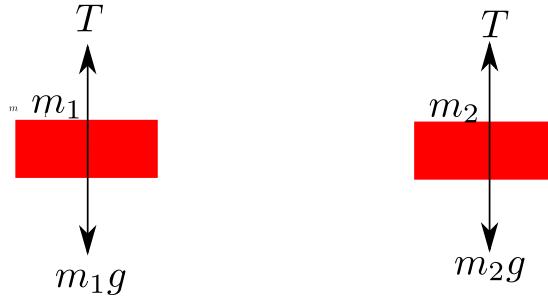


Figure 4.9: FBD of two mass

Because of all the constraints, the possible acceleration is as follows

$$\sum F_{x1} = ma_{x1/O} = mx\ddot{x}_1 \quad \text{or} \quad \sum F_{x2} = ma_{x2/O} = mx\ddot{x}_2 \quad (4.23)$$

### 4.5.4 External moments

Consider the FBD of the wheel as shown in figure 4.10.

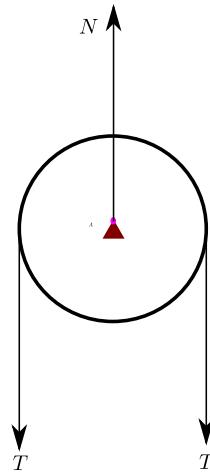


Figure 4.10: FBD of wheel

If the masses  $m_1 = m_2$  is considered, then the tension on the string on both sides will be the same, therefore, canceling out the moments by  $T$  about A. If in case  $m_1 \neq m_2$ , then the moment produced will be greater on the side of higher mass, this will lead to

## Nomenclature

the heavier mass hitting the ground or unless it is stopped using a stopper. However, lets consider that  $m_1 \neq m_2$  so that the problem can be solved for  $\dot{\theta}$  and  $\ddot{x}$ . External moments acting about A due to  $m_1 \neq m_2$  can be expressed as

$$\sum \tau_{/A} = (m_1 g R - m_2 g R) \hat{k}_1 \quad (4.24)$$

However, this is not completely true, as in Newtonian Mechanics, string tension  $T$  is considered as internal to the system. So for the system as a whole, in this problem, there are no external moments. In this problem, it can be considered that there is a moment applied at A, which is equal to the the equation mention above.

### 4.5.5 Finishing the EOMs

Consider the wheel, the angular momentum about A is expressed as

$$\begin{aligned} h_{A/O} &= r_{m1/A} \times p_{m1/O} + r_{m2/A} \times p_{m2/O} \\ &= R \hat{j}_1 \times m_1 \dot{x}_1 \hat{i}_2 + R \hat{j}_1 \times m_2 \dot{x}_2 \hat{i}_3 \\ &= -m_1 R \dot{x}_1 \hat{k}_1 - m_2 R \dot{x}_2 \hat{k}_1 \\ &= -(m_1 + m_2) R \dot{x} \hat{k}_1 \end{aligned}$$

in the above equation  $\dot{x}_1 = \dot{x}_2 = \dot{x}$  due to the constraint the non-holonomic constraint, which results due to the holonomic constraint  $x_1 = x_2$

$$\begin{aligned} (m_1 g R - m_2 g R) \hat{k}_1 &= -\frac{d}{dt} (m_1 + m_2) R \dot{x} \hat{k}_1 \\ &= -(m_1 + m_2) R \ddot{x} \end{aligned}$$

solving the above equation for  $\ddot{x}$ :

$$\ddot{x} = -\frac{g(m_1 - m_2 R)}{(m_1 + m_2)} \quad (4.25)$$

# 5 Mass Moment of Inertia

In this chapter the following assumptions are made

1. Consider a body that rotates about the center of mass
2. Use reference frames attached to the bodies

Consider the following figure 5.1. The body frame and the reference frames are labeled as shown in figure 5.1. Consider a piece of mass  $m_i$  which is at a distance  $r_{i/A}$  from A. Angular momentum of this mass  $m_i$  can be expressed as

$$h_{i/A} = r_{i/A} \times p_{i/O} \quad (5.1)$$

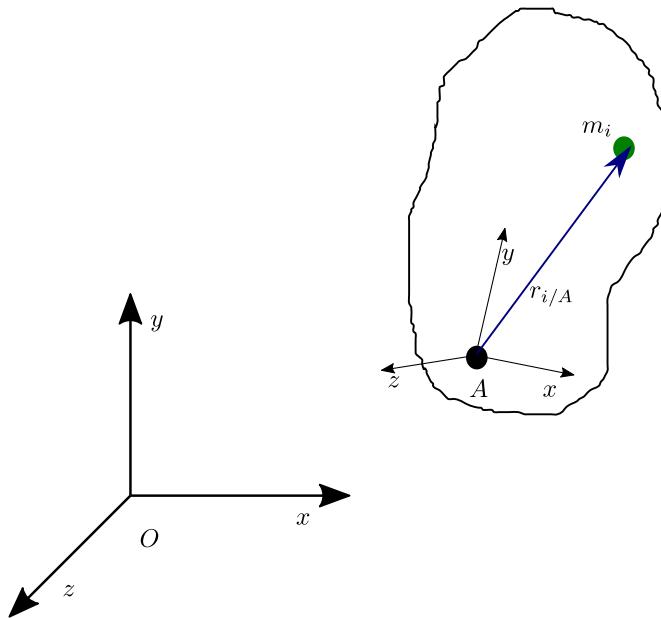


Figure 5.1: Finding mass moment of inertia (MMI)

This problem is similar to the one shown in figure 5.2. The problem can now be looked into using a specific known geometry. Since so far in this book, dynamics of only particles were discussed, this section serves as a starting point for the discussion on dynamics of rigid bodies. Here starting from simple known geometry to lay down foundational concepts, which can be extended later for specific applications of defining mass moment of inertia (MMI).

## Nomenclature

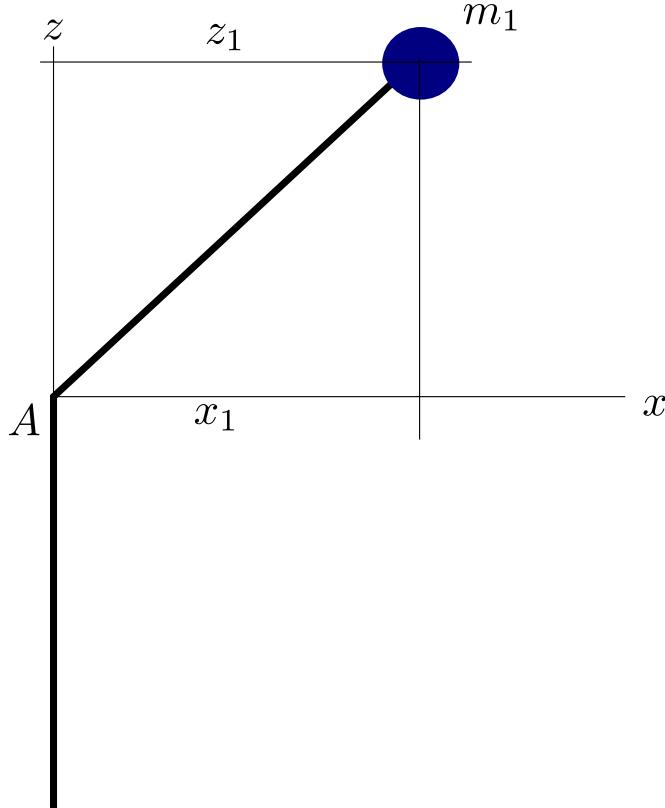


Figure 5.2: Similar problem on finding MMI

using this geometry, the angular momentum of the mass 1 attached can be found and grouped as follows:

$$\begin{aligned} h_{1/A} &= r_{1/A} \times p_{1/O} = r_{1/A} \times mv_{1/O} \\ &= (x_1 \hat{i} + z_1 \hat{k}) \times m_1 x_1 \omega \hat{j} \\ &= m_1 x_1^2 \omega \hat{k} - m_1 x_1 z_1 \hat{i} \end{aligned}$$

in the above equation, it can be see that the term  $m_1 x_1^2 \omega \hat{k} = h_z$ , angular momentum w.r.t *z* axis, similarly,  $m_1 x_1 z_1 \hat{i} = h_x$ . If the rigid body had a component along the  $\hat{j}$  direction, then angular momentum  $h_y$  would also be reflected in the above equation. Therefore, itseems that while find angular momentum and associated torques of a rigid body under circular motion is always accompanied by the components of angular momentum along all the three coordinate directions *x*, *y* and *z*. Using this analogy, in general, it can be said that the angular momentum of a rigid busy is made up as follows

$$H_{/A} = r_{/A} \times P_{/O} \quad (5.2)$$

in equation (5.2), for rigid bodies, each term is now denoted using a capital letter (except for the radial vector). Also notice that for each of the term  $H_{/A}$  or  $P_{/O}$  the

quantities are not referenced to any point, because for rigid bodies it is always reference to the center of mass, and the body as a whole.

**Note:** It can be noted here that for all rigid body problems the velocity  $v_{/O}$  always contains the part only due to  $\omega \times r_{/O}$ , as the distance between the particles in rigid bodies is always fixed, eliminating terms of relative velocities.

Equation (5.2) can be proved by considering the rigid body as a system of particles with a center of mass  $m_T$  and solving for the angular momentum equations from there

$$H_{/A} = \sum_i r_{i/A} \times p_{i/O} = \sum_i m_i r_{i/A} \times (\omega \times r_{i/A}) \quad (5.3)$$

which will result in

$$H_{/A} = H_x \hat{i} + H_y \hat{j} + H_z \hat{k} \quad (5.4)$$

where

$$\begin{aligned} H_x &= I_{xx}\omega_x + I_{xy}\omega_y + I_{xz}\omega_z \\ H_y &= I_{yx}\omega_x + I_{yy}\omega_y + I_{yz}\omega_z \\ H_z &= I_{zx}\omega_x + I_{zy}\omega_y + I_{zz}\omega_z \end{aligned}$$

or can be written conveniently using matrix form:

$$\begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (5.5)$$

## 5.1 Principal axes

Every rigid body has axes through it such that the rotating about this axes does not produce any unbalanced torques on any other axes. However, care should be taken to note that the unbalanced torque does not in essence pertain to vibrations that result from center of mass being shifted from the axis of rotation. In this case of vibration, there is no unbalanced torque produced along any other axis of rotations, therefore, even though there is a vibration on the selected axis, it can be chosen as principal axis as it does not produce any unbalanced torques on any other axes.

Consider equation 5.5, it can be seen that the only way to avoid producing any unbalanced torques on any other axes is making the off-diagonal elements to zero. Such that, angular momentum is only on one of these following axes  $H_x$ ,  $H_y$ ,  $H_z$ , when torque is applied on any of these axes.

The key thing to remember about principal axes is that they should be chosen in a way so as to make the off-diagonal elements go to zero.

There are some simple ways to find principal axes using geometry

## Nomenclature

1. If there is an axis of symmetry then that axis is a principal axis - circular objects have axis of symmetry about the center of circle. Axis of symmetry: An axis about which any point / object is equally reflected exactly across the axis as shown in figure 5.3

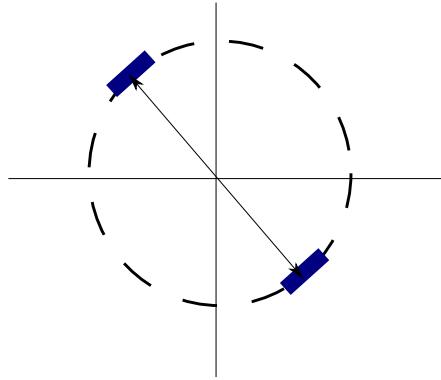


Figure 5.3: Axis of symmetry for rotating objects

2. If there is one plane of symmetry, then there is a principal axis perpendicular to it. Let it pass through center of mass  $G$
3. If there are two orthogonal planes of symmetry, their intersection is a principal axis. Principal axes are the property of an object.

## 5.2 Angular momentum of rigid bodies

Extending the equations from the previous section, angular momentum for rigid bodies is expressed in matrix form by:

$$H_{/A} = [I_{/A}] [0//0//\omega_z] = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (5.6)$$

In equation form, angular momentum for rigid bodies is expressed as:

$$H_{/A} = H_x \hat{i} + H_y \hat{j} + H_z \hat{k} \quad (5.7)$$

In the case when the  $I$  matrix has non-zero components along its off-diagonal elements, then angular momentum non-zero about the non-rotating axis, therefore, producing unbalanced torques along these non-rotating axes. This creates an imbalance of moments in the system. These off-diagonal elements are called products of inertia and are the cause for producing un-balanced torques about the non-rotating axis, as described in the following section 5.3.

**Note:** The one important factor for calculating  $H_{/A}$  for rigid bodies, is that here inertia matrix  $I$  for rigid bodies have to be considered.

## 5.3 Products of inertia

Consider figure 5.4, where the mass of an object is placed asymmetrically about the rotating axis.

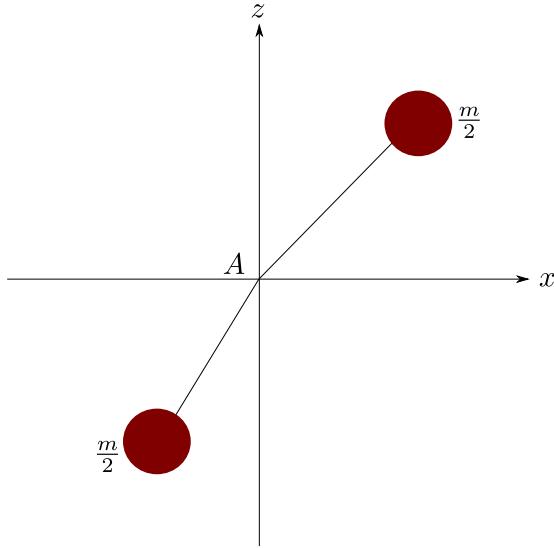


Figure 5.4: Mass placed asymmetrically about the rotating axis

$\mathbf{I}$  matrix can be computed using various standard methods described in dynamics texts, it will result in the following:

$$\mathbf{I} = \begin{bmatrix} mz_1^2 & 0 & -mz_1 z_2 \\ 0 & m(x_1^2 + z_1^2) & 0 \\ -mx_1 z_1 & 0 & mx_1^2 \end{bmatrix} \quad (5.8)$$

the rotation is only along  $\omega_z$ , therefore:

$$\mathbf{H}_{/A} = \mathbf{I} \begin{bmatrix} 0 \\ 0 \\ \omega_z \end{bmatrix} = \begin{bmatrix} -mx_1 z_1 \omega_z \\ 0 \\ mx_1^2 \omega_z \end{bmatrix} = \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} \quad (5.9)$$

torque about A van be expressed as:

$$\tau_A = \frac{d}{dt} \mathbf{H}_{/A} = \begin{bmatrix} -mx_1 z_1 \dot{\omega}_z \\ 0 \\ mx_1^2 \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (5.10)$$

It can be seen from the above expression, that due to off-diagonal elements in  $\mathbf{I}$ , there is a component of torque produced along the non-rotating  $x$  axis. Therefore, products of inertia are the cause for producing un-balanced torques about the non-rotating axis.

## Nomenclature

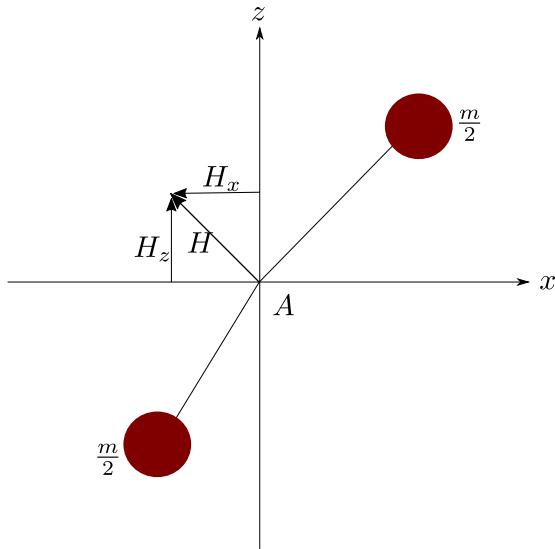


Figure 5.5: Products of inertia results in  $\mathbf{H}_{/A}$  on non-rotating axis

Because  $\mathbf{H}_{/A}$  is a vector, using vector addition, it can be shown using figure 5.5, how  $\mathbf{H}_{/A}$  produces along the non-rotating axis. As  $\mathbf{H}_{/A}$  is not pointing along the axis of rotation, therefore, it causes an imbalance.

## 5.4 Exercise 1: Cart and pendulum problem

Consider figure 5.6

here, the problem is being solved considering that the pendulum is a rigid body with a diagonal  $\mathbf{I}$  matrix, so that the angular momentum is calculated for a rigid body.

### 5.4.1 Coordinates

Positive direction of cart is chosen as  $x_1$ , with coordinate  $X_1 : \{x_1, y_1, z_1\}$ . Pendulum angle is chosen  $\theta$  about the vertical axis drawn straight from the point point A, and measured positive CCW. Further a coordinate  $X_2 : \{x_2, y_2, z_2\}$  is chosen for the pendulum as shown in figure 5.6. There are three possible distinct motions in this system, which have to be qualified using DOF and constraints.

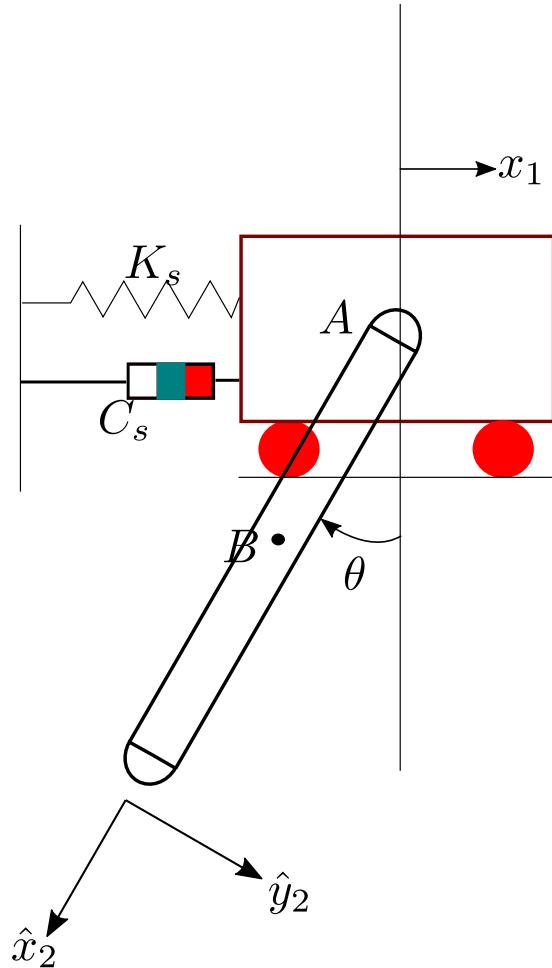


Figure 5.6: Cart and pendulum

### 5.4.2 DOF

Listing the constraints:

1.  $y_1 = z_1 = 0$
2.  $M_{x1} = M_{y1} = M_{z1} = 0$
3.  $x_2 = z_2 = 0$
4.  $M_{x2} = M_{y2} = 0$

there another holonomic constraint between  $\theta$  and  $\hat{y}_2$ , such that  $\hat{y}_2 = f(\theta)$ , therefore,  $C = 10$ ,  $DOF = 12 - 10 = 2$ . The possible coordinates are  $\theta$  for the pendulum's position and  $x_1$  for the carts position. The EOMs should be formed as:

$$\sum F_{x1} = ma_{A/0} \quad (5.11)$$

## Nomenclature

$$\sum M_{/A} = \frac{d}{dt} \mathbf{H}_{/A} + v_{A/O} \times P_{B/O} \quad (5.12)$$

(5.13)

### 5.4.3 External forces

Using FBD for the pendulum, the following as shown in figure 5.7 can be produced. Apart from the components of  $m_p g$  acting through  $G$ , reaction forces acting at A needs

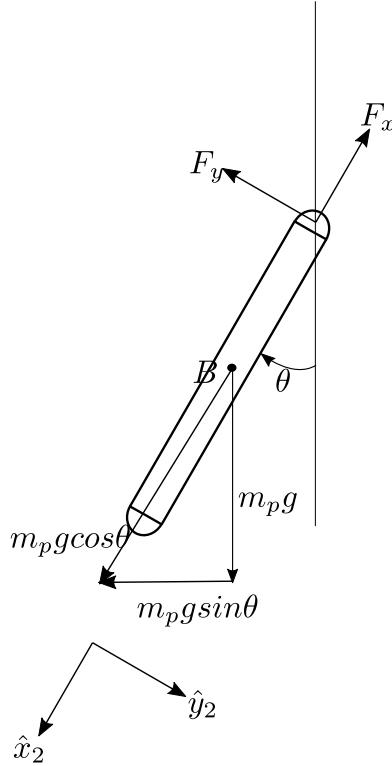


Figure 5.7: FBD of pendulum

to be considered. **Note:** In figure 5.7, the orientation of force  $m_p g \sin \theta$  is wrongly shown, it should be point in the direction parallel to  $\hat{y}_2$ . Also the direction of  $\theta$  is shown wrong, in figure it is pointing to negative  $\theta$ .

Using figure 5.7, moments about A can be expressed as (consider  $l$  as the length of the pendulum):

$$\sum M_{/A} = -m_p g \frac{l}{2} \sin \theta \hat{k}_1 \quad (5.14)$$

finally, for forces along  $\hat{x}_1$ , consider the FBD for the cart as shown in figure 5.8. using figure 5.8, balancing forces along  $\hat{x}_1$  as:

$$\sum F_{x1} = -K_s x - C_s \dot{x} - F_x \sin \theta + F_y \sin\left(\frac{\pi}{2} - \theta\right) \quad (5.15)$$

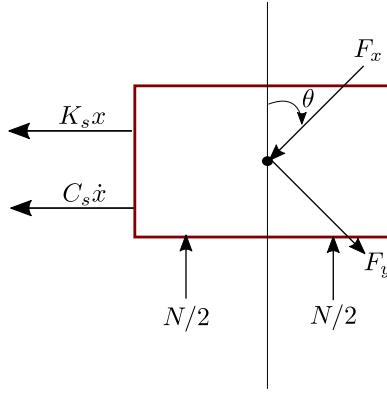


Figure 5.8: FBD of cart

where  $\theta$  is the angle measure from the vertical line drawn straight from the pivot point A.

**Note:** As the focus is only to find  $\ddot{x}_1$ , the forces that are acting only at the cart body need to be considered. Forces acting at the pendulum need not be considered, as by including reaction forces  $F_x$  and  $F_y$ , the effect of pendulum has already been considered.

## 5.5 Finishing EOM

we need to compute

$$\sum M_{/A} = \frac{d}{dt} \mathbf{H}_{/A} + v_{A/O} \times P_{B/O} \quad (5.16)$$

the left side of the equation is already been computed, the right side of the equation is computed as follows:

$$v_{B/O} = v_{A/O} + v_{B/A} + \omega \times r_{B/A} = \dot{x}\hat{i}_1 + 0 + \dot{\theta}\hat{k}_1 \times \frac{l}{2}\hat{x}_2 \quad (5.17)$$

which is simplified to:

$$v_{B/O} = \dot{x}\hat{i}_1 + \frac{l}{2}\dot{\theta}\hat{j}_2 \quad (5.18)$$

the linear momentum can therefore be calculated as:

$$P_{B/O} = m \left( \dot{x}\hat{i}_1 + \frac{l}{2}\dot{\theta}\hat{j}_2 \right) \quad (5.19)$$

the angular momentum about A can be expressed as:

$$H_{/A} = \mathbf{I}_A \begin{bmatrix} 0 \\ 0 \\ \omega_z \hat{k}_1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3}ml^2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \omega_z \hat{k}_1 \end{bmatrix} = \frac{1}{3}ml^2\dot{\theta}\hat{k}_1 \quad (5.20)$$

## Nomenclature

finally, computing  $\frac{d}{dt}\mathbf{H}_{/A} + v_{A/O} \times P_{B/O}$ :

$$\frac{d}{dt}\mathbf{H}_{/A} + v_{A/O} \times P_{B/O} = \frac{1}{3}ml^2\ddot{\theta}\hat{k}_1 + \dot{x}_1\hat{i}_1 \times m \left( \dot{x}_1\hat{i}_1 + \frac{l}{2}\dot{\theta}\hat{j}_2 \right) \quad (5.21)$$

there needs to be a relationship established between  $\hat{i}_1$  and  $\hat{j}_2$ , by working out the orientations of the unit vectors, it can be found that:

$$\begin{aligned}\hat{i}_2 &= \sin\theta\hat{i}_1 + \cos\theta\hat{j}_1 \\ \hat{j}_2 &= -\sin\theta\hat{j}_1 + \cos\theta\hat{i}_1\end{aligned}$$

using this in the above expression and simplifying the result, leads to,

$$\frac{d}{dt}\mathbf{H}_{/A} + v_{A/O} \times P_{B/O} = \frac{1}{3}ml^2\ddot{\theta}\hat{k}_1 - m\sin\theta\dot{x}_1\dot{\theta}\frac{l}{2}\hat{k}_1 \quad (5.22)$$

Equating both sides of the EOM,

$$-m_p g \frac{l}{2} \sin\theta \hat{k}_1 = \frac{1}{3}ml^2\ddot{\theta}\hat{k}_1 - m\sin\theta\dot{x}_1\dot{\theta}\frac{l}{2}\hat{k}_1$$

solving the equation for  $\ddot{\theta}$ :

$$\ddot{\theta} = \frac{m_p \sin\theta \dot{x}_1 \dot{\theta} \frac{l}{2} - m_p g \frac{l}{2} \sin\theta}{\frac{1}{3}m_p l^2} = \frac{\left(\sin\theta \frac{l}{2}\right) \left(\dot{x}_1 \dot{\theta} - g\right)}{\frac{1}{3}l^2} \quad (5.23)$$

EOM for  $\ddot{x}_1$  is expressed as (using equation (5.15)):

$$-K_s x - C_s \dot{x} - F_x \sin\theta + F_y \sin\left(\frac{\pi}{2} - \theta\right) = m\ddot{x}_1 \quad (5.24)$$

solving the above equation for  $\ddot{x}_1$ :

$$\ddot{x}_1 = \frac{1}{m_c} \left[ -K_s x - C_s \dot{x} - F_x \sin\theta + F_y \sin\left(\frac{\pi}{2} - \theta\right) \right] \quad (5.25)$$

where  $m_c$  is the mass of the cart and  $m_p$  is the mass of the pendulum. Equations (5.23) and (5.25) form the EOMs for this pendulum-cart system.

In equation (5.25), there are reaction forces  $F_x$  and  $F_y$ , that needs to be determined, in order for the system description to be complete. Since  $F_x$  and  $F_y$  are the reaction forces generated due to pendulum's motion,  $F_x$  and  $F_y$  can be found using figure 5.7. Equating forces along  $\hat{x}_2$ ,

$$\sum F_{x2} = -F_x + m_p g \cos\theta = m a_{B/O} \hat{x}_2 \quad (5.26)$$

$a_{B/O}$  from the expression above needs to be calculated,

$$a_{B/O} = a_{A/O} + {}^A a_{B/A} + [2\omega_{/O} \times v_{B/A}] + [\dot{\omega}_{/O} \times r_{B/A}] + [\omega_{/O} \times (\omega_{/O} \times r_{B/A})] \quad (5.27)$$

from the expression of  $a_{B/O}$ , the terms  ${}^A a_{B/A} = 0$  as B is fixed relative to A. Therefore, reducing the expressed for  $a_{B/O}$  to as follows

$$a_{B/O} = a_{A/O} + [2\omega_{/O} \times v_{B/A}] + [\dot{\omega}_{/O} \times r_{B/A}] + [\omega_{/O} \times (\omega_{/O} \times r_{B/A})] \quad (5.28)$$

additionally, the term  $v_{B/A}$  needs to be found,

$$v_{B/A} = {}^A v_{B/A} + \omega_{/O} \times r_{B/A} = 0 + \dot{\theta} \hat{k}_1 \times \frac{l}{2} \hat{x}_2 = \frac{l}{2} \dot{\theta} \hat{j}_2 \quad (5.29)$$

substituting  $v_{B/A}$  in expression for  $a_{B/A}$ ,

$$a_{B/O} = \ddot{x}_1 \hat{x}_1 + \left[ 2\dot{\theta} \hat{k}_1 \times \frac{l}{2} \dot{\theta} \hat{y}_2 \right] + \left[ \dot{\theta} \hat{k}_1 \times \frac{l}{2} \hat{x}_2 \right] + \left[ \dot{\theta} \hat{k}_1 \times \left( \dot{\theta} \hat{k}_1 \times \frac{l}{2} \hat{x}_2 \right) \right] \quad (5.30)$$

simplifying the above terms,

- $\left[ 2\dot{\theta} \hat{k}_1 \times \frac{l}{2} \dot{\theta} \hat{y}_2 \right] = -l\dot{\theta}^2 \hat{y}_2$
- $\left[ \ddot{\theta} \hat{k}_1 \times \frac{l}{2} \hat{x}_2 \right] = \frac{l}{2} \ddot{\theta} \hat{y}_2$
- $\left[ \dot{\theta} \hat{k}_1 \times \left( \dot{\theta} \hat{k}_1 \times \frac{l}{2} \hat{x}_2 \right) \right] = -\frac{l}{2} \dot{\theta}^2 \hat{x}_2$

Note that in equation (5.30), there is a term with  $\hat{x}_1$ , which needs to be translated to  $\hat{x}_2$  and  $\hat{y}_2$ . Performing the unit vector orientations carefully, it can be seen that the following relationship is true:

$$\hat{x}_2 = \sin\theta \hat{x}_1 + \cos\theta \hat{y}_1 \quad (5.31)$$

$$\hat{y}_2 = -\sin\theta \hat{y}_1 + \cos\theta \hat{x}_1 \quad (5.32)$$

$$(5.33)$$

the above relationship can be expressed in form of rotation matrix as follows:

$$\begin{bmatrix} \hat{x}_2 \\ \hat{y}_2 \\ \hat{z}_2 \end{bmatrix} = \begin{bmatrix} \sin\theta_s & \cos\theta_s & 0 \\ \cos\theta_s & -\sin\theta_s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{z}_1 \end{bmatrix} = {}_{X1}^{X2}\mathbf{R} \begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{z}_1 \end{bmatrix} \quad (5.34)$$

where in  ${}_{X1}^{X2}\mathbf{R}$ , subscript  $X1$  represents the base frame and  $X2$  represents the follower frame.  $\theta_s$  represents the angle measure from the vertical to  $F_x$  as shown in figure 5.8. Therefore,  ${}_{X1}^{X2}\mathbf{R} = {}^T [{}_{X2}^{X1}\mathbf{R}]$ , which is expressed as,

$$\begin{bmatrix} \hat{x}_1 \\ \hat{y}_1 \\ \hat{z}_1 \end{bmatrix} = \begin{bmatrix} \sin\theta_s & \cos\theta_s & 0 \\ \cos\theta_s & -\sin\theta_s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_2 \\ \hat{y}_2 \\ \hat{z}_2 \end{bmatrix} \quad (5.35)$$

## Nomenclature

therefore, the term  $\ddot{x}_1 \hat{x}_1 = \ddot{x}_1 \sin\theta_s \hat{x}_2 + \ddot{x}_1 \cos\theta_s \hat{y}_2$ , substituting this in equation (5.30),

$$a_{B/O} = \ddot{x}_1 \sin\theta_s \hat{x}_2 + \ddot{x}_1 \cos\theta_s \hat{y}_2 - l\dot{\theta}^2 \hat{y}_2 + \frac{l}{2}\ddot{\theta}\hat{y}_2 - \frac{l}{2}\dot{\theta}^2 \hat{x}_2 \quad (5.36)$$

equating equation (5.36) with equation (5.36), the reaction forces can be found

$$-F_x + m_p g \cos\theta = m_p \left( \ddot{x}_1 \sin\theta_s - \frac{l}{2}\dot{\theta}^2 \right)$$

solving for  $F_x$ ,

$$F_x = m_p \left( \frac{l}{2}\dot{\theta}^2 + g \cos\theta - \ddot{x}_1 \sin\theta_s \right) \quad (5.37)$$

## 5.6 Exercise 2: Translating System

Consider a translating system as shown in figure 5.9.

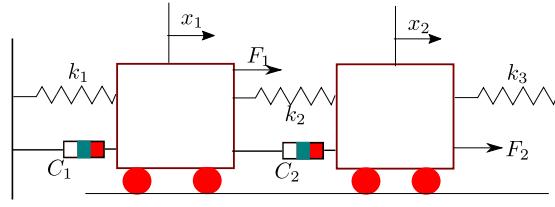


Figure 5.9: Translating masses

### 5.6.1 Coordinate selection

Choosing coordinates are quite straight forward in this problem. As shown in figure 5.9, both the masses  $m_1$  and  $m_2$  are allowed to translate only along  $x_1$  and  $x_2$  directions.

### 5.6.2 DOF

Since there are only two independent coordinate motions possible, there should be  $C = 2 \cdot 6 - 2 = 10$  constraint placed in the system. This can be explained as follows

- $y_1 = y_2 = z_1 = z_2 = 0$
- $M_{x1} = M_{x2} = M_{y1} = M_{y2} = M_{z1} = M_{z2} = 0$

Therefore, the EOMs should be of the form

$$\sum F_{x1} = ma_{x1/O} \quad (5.38)$$

$$\sum F_{x2} = ma_{x2/O} \quad (5.39)$$

$$(5.40)$$

### 5.6.3 External forces

In order to determine the external forces on the system, each of the individual masses needs to be considered. Consider figure 5.10.

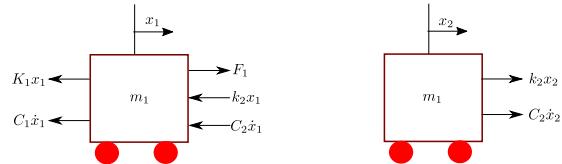


Figure 5.10: FBD of mass 1

*Nomenclature*

## **Part II**

# **Classical Control Theory**



# 6 Preliminaries

## 6.1 Euler's Identity

Euler's number  $e$  is a special number that is used in Euler's identity, it was originally discovered by Jacob Bernoulli from the problem of compound interest. Consider the formula for compound interest:

$$A = P \left(1 + \frac{r}{n}\right)^{nt} \quad (6.1)$$

where,

- A is the amount that is returned after interest
- P is the principle
- r is the interest rate
- n is the number of interest compounded per unit time
- nt is the total time of interest

Jacob Bernoulli wanted to know how much A would be maximized by making the interest maximum to 100%. He observed that no matter how much times the number of intervals  $n$  was used, A never crossed a maximum value of 2.72 for a P of 1. The exact number is 2.71828182 with more decimals following, Euler named this special number  $e$  after his own name.

One of the ways to calculate this special number  $e$  using only basic mathematical operations (addition, subtraction, multiplication, division) is by using infinite series.  $e$  can be calculated as follows:

$$\begin{aligned} e &= 1 = 1.00 \\ &\approx 1 + \frac{1}{1!} = 2.00 \\ &\approx 1 + \frac{1}{1!} + \frac{1}{2!} = 2.50 \\ &\approx 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} = 2.666 \\ &\approx 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} = 2.708 \end{aligned}$$

## Nomenclature

$$\approx 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} = 2.716$$

As more terms are added into the infinite series, more accurate  $e$  can be calculated. The advantage of infinite series is that it is easier to predict the next value of  $e$  by adding factorials into the series. As a number,  $e = 2.716$  from the infinite series can be written as  $e^1$ , in general  $e$  can be raised to any number  $x$  and therefore, the series can be written in the follow form for a general case:

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots \quad (6.2)$$

It can be seen that a smooth function (easily predictable) can be defined for  $e$  using infinite series, other functions such as  $\sin(x)$  and  $\cos(x)$  are also smooth functions that can be easily determined using infinite series. For  $\sin(x)$  and  $\cos(x)$  the following series can be defined:

$$\sin(x) = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (6.3)$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} \dots \quad (6.4)$$

it seems that the Euler's number  $e$  and the sum of  $\sin(x)$  and  $\cos(x)$  have some similarities except from the alternating signs that comes from the sum of  $\sin(x)$  and  $\cos(x)$  as shown below:

$$\sin(x) + \cos(x) = 1 + \frac{x}{1!} - \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} - \frac{x^6}{6!} - \frac{x^7}{7!} \dots \quad (6.5)$$

The difference is in the sign of the following terms:

$$-\frac{x^2}{2!} - \frac{x^3}{3!} - \frac{x^6}{6!} - \frac{x^7}{7!} \dots$$

The idea is to multiply these terms with some number in such a way that when that number is square it produces a negative equal to  $-1$ . Such a number does not exits in reality, but because this number has to be included into the mathematics in order to further solve a problem, an imaginary number is invented such that  $i^2 = -1$ . This imaginary number was invented because this problem was occurring quite frequently in mathematics and such a number had to be imagined. Now with the invention of an imaginary number the infinite series for  $e^x$  is now defined for  $e^{ix}$ , the following series will be produced:

$$e^{ix} = 1 + \frac{ix}{1!} + \frac{(ix)^2}{2!} + \frac{(ix)^3}{3!} + \frac{(ix)^4}{4!} + \frac{(ix)^5}{5!} + \frac{(ix)^6}{6!} + \frac{(ix)^7}{7!} \dots \quad (6.6)$$

Now,

- $i^2 = -1$
- $i^3 = i^2 \times i = -i$

From equation 6.6:

$$e^{ix} = 1 + \frac{ix}{1!} - \frac{x^2}{2!} - \frac{ix^3}{3!} + \frac{x^4}{4!} + \frac{ix^5}{5!} - \frac{x^6}{6!} - \frac{ix^7}{7!} \dots$$

Rearranging  $i$  terms and all other terms:

$$e^{ix} = \left(1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!}\right) + i \left(\frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!}\right) \quad (6.7)$$

comparing equation (6.7) with equations (6.3) and (6.4), it can be seen that:

$$e^{ix} = \cos(x) + i\sin(x) \quad (6.8)$$

Equation (6.8), is called the **Euler's Formula**. A special case of Euler's formula leads to **Euler's Identity**, which occurs when  $e^{ix} = e^{i\pi}$ . Consider equation (6.8), since  $\cos(\pi) = -1$  and  $\sin(\pi) = 0$ , Euler's formula now reduces to:

$$e^{i\pi} = -1 + i0 \implies e^{i\pi} + 1 = 0 \quad (6.9)$$

Equation (6.9), relates four mathematical concepts in one formula which are:

- Euler's number  $e$
- cosine and sine terms
- Irrational number  $\pi$
- Complex number  $i$

## 6.2 Linear differential equations

The input/output behavior of a system can be described using linear differential equation:

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_n y = b_1 \frac{d^{n-1} u}{dt^{n-1}} + \dots + b_n u \quad (6.10)$$

where,  $y$  and  $u$  are number of inputs and outputs of the system respectively and  $a_k$  and  $b_k$  are real constant coefficients. For all such linear differential equations, the solution of the differential equation (6.10) is expressed as a linear combination of **null solution** and **particular solution** as

$$y(t) = y_n(t) + y_p(t) \quad (6.11)$$

## Nomenclature

### 6.2.1 Null Solution

the null solution is the function of system variables  $y(t)$  (when input to the system is zero) and particular solution is the function of inputs to the system  $u(t)$ . The null solution  $y_n$  is derived from its characteristic equation, the null solution is then a linear combination of the roots of the characteristic equation. Therefore, if the roots of the characteristic equation are  $s_k$ , it forms that

$$y_n(t) = \sum_{k=1}^n C_k e^{s_k t} \quad (6.12)$$

The characteristic equation of an differential equation (such as equation (6.10) when RHS is zero) with the roots  $s_k$  is given as

$$a(s) = s^n + a_1 s^{n-1} + \dots + a_n = 0 \quad (6.13)$$

The solution depends on the number of roots and the number of roots depends on the order of the characteristic equation (6.13). Using the derived roots of characteristic equation 6.13 ( $s_k$ ), the terms  $\sum_{k=1}^n e^{s_k t}$  of the null solution can be found. The parameter  $C_k$  is dependent on the initial conditions of the system, they are included there to include the IC's.

#### 6.2.1.1 Deriving characteristic equation for LDE

Lets consider the solution of the LDE 6.10 is given by a smooth sinusoidal function

$$y(t) = e^{st} \quad (6.14)$$

substituting equation (6.14) into LDE 6.10 (inputs are ignored to find characteristic equation)

$$s^k e^{st} + a_1 s^{k-1} e^{st} \dots + a_n e^{st} = 0 \implies (s^k + a_1 s^{k-1} \dots + a_n) e^{st} = 0 \quad (6.15)$$

since  $e^{st}$  cannot be a zero, there the term  $s^k + a_1 s^{k-1} \dots + a_n$  should diminish to zero. Therefore, the characteristic equation of 6.10 is given by

$$s^k + a_1 s^{k-1} \dots + a_n = 0 \quad (6.16)$$

The roots of this characteristic equation (6.16) depends on the order of the LDE. For second order systems, the characteristic equation becomes quadratic and is expressed by

$$s^2 + a_1 s + a_2 = 0 \quad (6.17)$$

with the roots  $s_{1,2}$  therefore being quadratic.

### 6.2.2 Particular Solution

The process of finding the particular solution can be visualized using the figure 6.1.

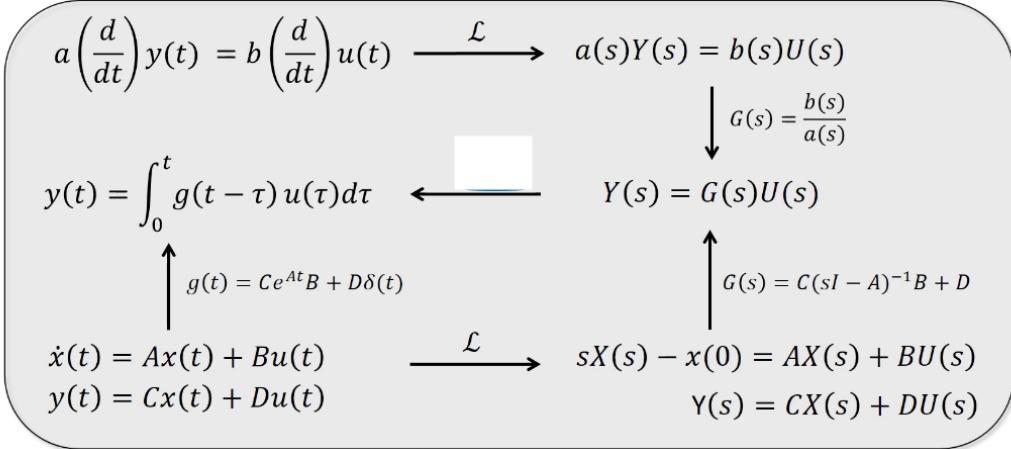


Figure 6.1: Flow of solution to LDE to and from Laplace Transforms

As a particular solution depends on the input, let's consider a sinusoidal input such that  $u(t) = e^{st}$ , where  $s$  is a complex number  $s = \sigma + j\omega$  and  $\sigma$  is chosen such that the input signal is a damped signal. An investigation can be performed to check if there is a unique particular solution of the form

$$y(t) = G(s)u(t) \implies y(t) = G(s)e^{st} \quad (6.18)$$

Equation (6.18) is chosen such that the solution  $y(t)$  is some function  $G(s)$  of the input signal  $u(t)$ . The idea is to determine this unknown function  $G(s)$ , the following steps deals with determining  $G(s)$ .

Substituting equation (6.18) into the differential equation (6.10) and writing its characteristic equation gives:

$$(s^n + a_1s^{n-1} + \dots + a_n)G(s)e^{st} = (b_1s^{n-1} + \dots + b_n)e^{st} \quad (6.19)$$

From equation (6.19), it can be seen that  $G(s)$  can be determined as the ratio of two polynomials on LHS and RHS which here are labeled as  $b(s)$  and  $a(s)$  respectively

$$G(s) = \frac{b_1s^{n-1} + \dots + b_n}{s^n + a_1s^{n-1} + \dots + a_n} = \frac{b(s)}{a(s)} \quad (6.20)$$

Equation (6.20) is called the **Transfer function**. A Transfer function is the ratio of output to the input of a system.

Now, the solution to the differential equation is expressed as the linear combination of equations (6.12) and (6.18)

$$y(t) = \sum_{k=1}^n C_k e^{s_k t} + G(s)e^{st} \quad (6.21)$$

## Nomenclature

the first term in equation (6.21) is due to the IC's and the second term is due to the inputs via transfer function.

The **Transfer function**  $G(s)$  can be determined using Laplace Transform. Notice, however, in this example we did not use LT to find the particular solution so no inverse LT can be performed to determine the transfer function  $G(s)$ . In this case therefore, the step of writing the particular solution as a convolutional integral term in time domain cannot be done.

The solution to LDE using figure 6.1 is given here to indicate that particular solutions contain transfer function and this can be found easily using Laplace transforms instead.

### 6.2.3 Laplace transform

The transfer function as given by equation (6.20) can be determined using Laplace transform. The LT for a function  $f(t)$  written in time domain is generally defined in  $s$ -domain as

$$F(s) = \mathcal{L}\{f(t)\} = \int_0^\infty f(t)e^{-st}dt \quad (6.22)$$

Where  $f(t)$  is transformed into  $s$ -domain using the above definition. Here  $s$  is called a Laplace variable and is complex in nature. For a differential in time-domain, in LT the equivalent is multiplication with Laplace variable  $s$  and subtraction with initial conditions such that

$$\mathcal{L}\left\{\frac{df(t)}{dt}\right\} = sF(s) - f(0) \quad (6.23)$$

If the system is originally at rest at  $t = 0$  such all IC's are zero, then the transfer function of a differential equation (6.10) can be expressed as the ratio of outputs to inputs

$$Y(s) = \frac{b_1 s^{n-1} + \dots + b_n}{s^n + a_1 s^{n-1} + \dots + a_n} U(s) = \frac{b(s)}{a(s)} U(s) = G(s)U(s) \quad (6.24)$$

Laplace transform follows the **convolution theory** such that the solution from the product of two equations in time domain is individual terms taken in Laplace domain as

$$\mathcal{L}\left\{\int_0^t g(t-\tau)f(t)d\tau\right\} = \mathcal{L}\{g(t)\}\mathcal{L}\{f(t)\} = G(s)F(s) \quad (6.25)$$

using the above result, the solution to equation (6.24) (replace  $F(s)$  with  $U(s)$ ) can be determined easily for any input signals  $U(s)$  as the following relationships holds good in the Laplace domain:

$$\mathcal{L}\left\{\int_0^t g(t-\tau)u(t)d\tau\right\} = G(s)U(s) \quad (6.26)$$

The transfer function  $G(s)$  can actually be used to express the systems response to an arbitrary input signal.

**Note:** the term  $g(t)$  in the convolutional integral is the impulse response and  $u(t)$  is the input signal. Therefore, using convolutional integral the solution  $y(t)$  can be convoluted using the solution of  $g(t)$  to any input signal  $u(t)$ .

### 6.3 Example of solving a LDE

Lets consider a LDE given as

$$\dot{x} = ax + bu, \quad x(0) = x_0 \quad (6.27)$$

#### Null Solution

For null solution we have to first find the characteristic equation, assuming a solution with a smooth function such that  $x(t) = Ce^{st}$ , the characteristic equation can be found by substituting  $x(t) = Ce^{st}$  into equation (6.27) and considering terms with  $u = 0$

$$\dot{x} - ax = 0 \implies Cse^{st} - Cae^{st} = 0 \implies Ce^{st}(s - a) = 0 \implies s - a = 0 \quad (6.28)$$

the root of characteristic equation is  $s = a$ , substituting this in equation  $x(t) = Ce^{st}$

$$x_n(t) = Ce^{at} \quad (6.29)$$

where  $C$  can be found from IC

$$x_n(0) = Ce^{a0} = x_0 \implies C = x_0 \quad (6.30)$$

which completes the null solution

$$x_n = x_0 e^{at} \quad (6.31)$$

#### Particular Solution

The particular solution can be found using the convolutional integral (6.26), in order to use the convolutional integral we first need the impulse response. The impulse response can be in-turn written using inverse LT of the TF

$$g(t) = \mathcal{L}G(s)$$

starting from LT of equation (6.27)

$$\mathcal{L}\{\dot{x} = ax + bu\} \implies sX(s) - x_0 - aX(s) = bU(s) \quad (6.32)$$

further transfer function needs that the IC's are zero,

$$X(s)(s - a) = bU(s) \implies G(s) = \frac{X(s)}{U(s)} = \frac{b}{s - a} \quad (6.33)$$

the solution in Laplace domain can be written as

$$X(s) = G(s)U(s) \quad (6.34)$$

## Nomenclature

finally,  $g(t)$  can be found from inverse LT to write the solution  $x(t)$  in time-domain

$$\mathcal{L}^{-1}\{G(s)\} = \mathcal{L}^{-1}\left\{\frac{b}{s-a}\right\} = be^{at} \quad (6.35)$$

the complete particular solution due to an input  $u(t)$  is therefore derived from the convolutional integral

$$x_p = \int_0^t g(t-\tau)u(\tau)d\tau = \int_0^t be^{a(t-\tau)}u(\tau)d\tau \quad (6.36)$$

where  $\tau$  is a constant in time when  $u$  is applied. The complete solution to equation (6.27) is therefore given by

$$x(t) = x_n(t) + x_p(t) = x_0e^{at} + \int_0^t be^{a(t-\tau)}u(\tau)d\tau \quad (6.37)$$

## 6.4 Basic definitions of Linear system

A system is linear with respect to its inputs and outputs if and only if (iff) ***superposition*** hold. Principle of superposition says that if there are two inputs  $u_1$  and  $u_2$  which produce outputs  $y_1$  and  $y_2$  respectively, then the total response of the system from these two inputs can be determined as a linear combination of two outputs as

$$y = au_1 + bu_2 \quad (6.38)$$

It also says that the amplitudes  $a$  and  $b$  do not matter, if we scale inputs by factors  $a$  and  $b$  respectively, the outputs also scale individually as shown in the equation above. A ***linear time-invariant*** system is independent of time such that the coefficients  $(a_k \ b_k)$  of the differential equation (6.10) are constant. If the coefficients become dependent on time, then equation (6.10) become time-variant (ie., the response changes as time of application of inputs changes).

## 6.5 Solution to differential of $e^{\lambda t}$

The value of  $e^{\lambda t}$  where  $\lambda$  is a scalar, can be found using infinite series and can be expressed as the sum

$$e^{\lambda t} := \sum_{k=0}^{\infty} \frac{(\lambda t)^k}{k!} \quad (6.39)$$

taking the time derivative

$$\frac{d}{dt} e^{\lambda t} = \frac{d}{dt} \left( \sum_{k=0}^{\infty} \frac{(\lambda t)^k}{k!} \right) \quad (6.40)$$

Note that the infinite series can be expanded from second to the infinite term as follows

$$e^{\lambda t} = 1 + \sum_{k=1}^{\infty} \frac{(\lambda t)^k}{k!}$$

using the above result, the derivative of constant can be removed from the original equation, leaving behind the derivative from second to the infinite term. Also note that the derivative  $\frac{d}{dt}((\lambda t)^k)$  is of the form  $\frac{d}{dt}(x^n) := nx^{n-1}$

$$\frac{d}{dt}e^{\lambda t} = \lambda \sum_{k=1}^{\infty} \frac{k(\lambda t)^{k-1}}{k!} \quad (6.41)$$

the term  $k$  in the numerator cancels the term  $k$  in the denominator leaving behind the factorial  $(k - 1)!$ , the differential now becomes

$$\frac{d}{dt}e^{\lambda t} = \lambda \sum_{k=1}^{\infty} \frac{(\lambda t)^{k-1}}{(k - 1)!} \quad (6.42)$$

if we substitute a new coordinate  $l$  such that  $l = (k - 1)$ , the infinite series becomes

$$\frac{d}{dt}e^{\lambda t} = \lambda \sum_{l=0}^{\infty} \frac{(\lambda t)^l}{l!} \quad (6.43)$$

which again is the definition of the infinite series  $e^{\lambda t}$ , therefore, the solution to the differential now becomes

$$\frac{d}{dt}e^{\lambda t} = \lambda e^{\lambda t} \quad (6.44)$$

## 6.6 Solution to differential of an exponential matrix $e^{\mathbf{A}t}$

### 6.6.1 Defining $e^{\mathbf{A}t}$

If  $\mathbf{A}$  is to be diagonal, then it is defined as

$$e^{\mathbf{A}t} = \mathbf{E}\Lambda\mathbf{E}^{-1} \quad (6.45)$$

where,

$$\lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \quad (6.46)$$

Using the definition above for  $\mathbf{A}$ , the exponential with matrix  $\mathbf{A}$  can be defined as

$$e^{\mathbf{A}t} = \sum_{k=0}^{\infty} \frac{1}{k!} (\mathbf{E}\Lambda\mathbf{E}^{-1}t)^k \quad (6.47)$$

## Nomenclature

simplifying the above equation

$$e^{\mathbf{A}t} = \sum_{k=0}^{\infty} \frac{1}{k!} (\mathbf{E})^k (\mathbf{\Lambda})^k (\mathbf{E}^{-1})^k t^k \quad (6.48)$$

Apparently the terms  $\mathbf{E}$  and  $\mathbf{E}^{-1}$  can be factored out leaving the terms  $\mathbf{\Lambda}^k t^k$  inside the summation

$$e^{\mathbf{A}t} = \mathbf{E} \left( \sum_{k=0}^{\infty} \frac{1}{k!} (\mathbf{\Lambda}t)^k \right) \mathbf{E}^{-1} \quad (6.49)$$

further,

$$e^{\mathbf{A}t} = \mathbf{E} \left( \sum_{k=0}^{\infty} \frac{1}{k!} \begin{bmatrix} (\lambda_1 t)^k & 0 \\ 0 & (\lambda_2 t)^k \end{bmatrix} \right) \mathbf{E}^{-1} \quad (6.50)$$

Now the terms inside the summation is the definition of exponential for every diagonal element, therefore,

$$e^{\mathbf{A}t} = \mathbf{E} \begin{bmatrix} e^{\lambda_1 t} & 0 \\ 0 & e^{\lambda_2 t} \end{bmatrix} \mathbf{E}^{-1} \quad (6.51)$$

Therefore, it can be said that ***exponent of a diagonal matrix*** is a ***exponent of diagonal elements***.

### 6.6.2 Time derivative of exponential matrix $e^{\mathbf{A}t}$

using the definition

$$e^{\mathbf{A}t} = \sum_{k=0}^{\infty} \frac{1}{k!} (\mathbf{E} \mathbf{\Lambda} \mathbf{E}^{-1} t)^k \quad (6.52)$$

time-derivative is expressed as

$$\begin{aligned} \frac{d}{dt} e^{\mathbf{A}t} &= \frac{d}{dt} \left( \sum_{k=0}^{\infty} \frac{1}{k!} (\mathbf{E} \mathbf{\Lambda} \mathbf{E}^{-1} t)^k \right) \\ &= \frac{d}{dt} \left( \mathbf{E} \left( \sum_{k=0}^{\infty} \frac{1}{k!} (\mathbf{\Lambda} t)^k \right) \mathbf{E}^{-1} \right) \\ &= \frac{d}{dt} \mathbf{E} \left( \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \sum_{k=1}^{\infty} \frac{1}{k!} (\mathbf{\Lambda} t)^k \right) \mathbf{E}^{-1} \\ &= \sum_{k=1}^{\infty} \frac{1}{(k-1)!} \mathbf{E} \begin{bmatrix} (\lambda_1 t)^{k-1} & 0 \\ 0 & (\lambda_2 t)^{k-1} \end{bmatrix} \mathbf{E}^{-1} \\ &= \mathbf{A} e^{\mathbf{A}t} \end{aligned}$$

### 6.6.3 State Transition Matrix

The solution from section 6.6.2 that the matrix exponential behaves same like a scalar exponential is central in Control theory and appears so often in LTI systems, it is given its own name and called as a ***State Transition Matrix*** and is denoted using a special symbol as shown,

$$e^{\mathbf{A}(t-t_0)} = \Phi(t, t_0) \quad (6.53)$$

This helps establish the following results, suppose a time derivative of a matrix exponential is to be determined, as it is known that a matrix exponential behaves same like a scalar exponential, it can be said that,

if the time-derivative of a scalar exponential is given as follows

$$\frac{d}{dt} e^{at} = ae^{at} \quad (6.54)$$

then the time-derivative of a matrix exponential is given as follows

$$\frac{d}{dt} e^{At} = \frac{d}{dt} \Phi(t, t_0) = A\Phi(t, t_0) \quad (6.55)$$

and also it can be proven that,

$$\Phi(t, t) = \mathbf{I} \quad (6.56)$$

where  $\mathbf{I}$  is the Identity matrix. As substituting  $t_0 = t$  will make the matrix exponential  $e^0 = 1$ .

### 6.6.4 Solutions to free and forced response of a system using results from State Transition Matrix

#### 6.6.4.1 Null Solution

As the solution to  $\frac{d}{dt} e^{At}$  is given by

$$\frac{d}{dt} e^{At} = \frac{d}{dt} \Phi(t, t_0) = A\Phi(t, t_0) \quad (6.57)$$

comparing (6.57) with a state-space model,

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u$$

$\mathbf{A}\mathbf{x}$  forms the solution when there are no inputs to the system. Equation (6.57) denotes the null solution to the time-derivative of a State Transition Matrix.

## Nomenclature

### 6.6.4.2 Particular Solution

In mathematics, the solution to the time-derivative of a State Transition Matrix with a given input  $u$  ( $\dot{x} = Ax + Bu$ ) is expressed as

$$x(t) = \Phi(t, t_0)x(t_0) + \int_{t_0}^t \Phi(t, \tau)\mathbf{B}u(\tau)d\tau \quad (6.58)$$

where  $\tau$  is a dummy integral variable used to plug in any required value of  $t$ . The term with integral is called constitutional integral term. The plausibility that (6.58) is a unique solution to the time-derivative of a State Transition Matrix with an input  $u$  can be checked using **Initial Condition check** and **Dynamics check**.

**Initial Condition check** Initial Condition check can be done by substituting  $t = t_0$ , therefore (6.58) becomes

$$x(t_0) = \Phi(t, t_0)x(t_0) + \int_{t_0}^{t_0} \Phi(t_0, \tau)\mathbf{B}u(\tau)d\tau \quad (6.59)$$

where  $\Phi(t_0, t_0) = \mathbf{I}$  as established earlier and the term with integral goes to zero. Leaving behind the solution

$$x(t) = \mathbf{I}x(t_0) \quad (6.60)$$

### Dynamics check

## 6.7 Newton's Law's for rotating bodies

Stating Newton's Law's for rotating bodies: Starting from linear momentum, it is described as

$$\mathbf{P}_{/O} = m\mathbf{v}_{G/O} \quad (6.61)$$

Where  $\mathbf{P}_{/O}$  is the linear momentum expressed with reference to frame-O (Inertial frame). Linear momentum is always expressed in inertial frame-O,  $v_G$  is the velocity of center of mass of the body. Further, from Newton's second law it can be seen that, the law states the rate of change of  $\mathbf{P}_{/O}$  as

$$\sum \mathbf{F} = \frac{d}{dt}\mathbf{P}_{/O} \quad (6.62)$$

However, the angular momentum itself is expressed as the moment of the moment, that is

$$\mathbf{H}_{/A} = \sum_i r_{i/A} \times p_{i/O} \quad (6.63)$$

where  $\mathbf{H}_{/A}$  denotes angular momentum with reference to frame-A,  $r_{i/A}$  denotes the distance vector of body  $i$  with reference to frame-A's origin. Because angular momentum unlike linear moment can be expressed with reference to any point of frame. So

choosing an arbitrary frame  $A$  so as to include any effects of off-the axis loading in the rotary system. Further, (6.63) can be solved and expressed for solid bodies as

$$\mathbf{H}_{/A} = \mathbf{H}_{/G} + r_{G/A} \times \mathbf{P}_{/O} \quad (6.64)$$

the above equation can be written in pseudo-code as ***Angular momentum w.r.t any point A = Angular momentum w.r.t G + momentum due to off the axis loading of the rigid body from its 'G'***

Here Frame-A is any moving frame w.r.t Inertial Frame-O, therefore, there will always be a term that calculates the angular momentum of  $G$  w.r.t this moving frame. Alternatively, angular momentum can be calculated using matrix formula

$$\mathbf{H}_{/A} = [I_{/A}] \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (6.65)$$

However, using equation (6.64) the use of parallel-axis theorem to calculate the inertia tensor  $[I_{/A}]$  can be avoided.

Finally, the net torques acting on the system can be calculated using Newton-Euler's equations for rotating rigid bodies

$$\sum \mathcal{T}_{/A} = \frac{d}{dt} \mathbf{H}_{/A} + \mathbf{v}_{A/O} \times P_{/O} \quad (6.66)$$

Similar, to angular momentum, if torques are to be calculated w.r.t to a moving Frame-A, then there will be an additional term to include the torques due to this moving frame.

## 6.8 Harmonic Oscillator

This section describes mathematical modeling of harmonic oscillator both its free and forced response. The forced response is

*Nomenclature*

# 7 System Modeling

"A dynamic system is a system that change or evolve over time [MT19a]. Dynamic systems can be represented generally by either **First Order** or **Second Order** differential equations. In general a system can be represented in a state equation form as:

$$[\dot{x}] = f([x(t)], [u(t)], t) \quad (7.1)$$

where,

- $\dot{x}$  is a vector of dynamics (rate of change) of the states of the system
- $x(t)$  is the vector of states of the system
- $u(t)$  is the vector of inputs to the system

From the initial conditions of the states  $x_0$  at  $t = 0$ , and the history of the inputs  $u(t)$  the next possible states value after every time step can be determined both analytically or through numerical integration.

## Assumptions

A nonlinear time dependent function is generally difficult to solve mathematically, so two assumptions are made in order to simplify the problem statement.

1. If  $f$  does not depend on time explicitly such as that  $\dot{x} = f(x(t), u(t))$  instead of  $\dot{x} = f(x(t), u(t), t)$ , then the system is said to be time-invariant. The state variables  $x(t)$  and the control inputs  $u(t)$  may still be dependent on time. Also for time-invariant systems, the parameters or the coefficients of the function  $f$  are constant [MT19a].
2. An explicit function in mathematics is a function that can be written in the form:  $y = f(x)$  such as  $y = 5x^3 - 3$ , in case of time-invariant functions which can be written in a form  $y = e^{at}$  which is not a time-explicit function.  $a$  is a coefficient that is a constant, there are time-varying coefficients such as  $\sin \alpha$ . When the systems are not dependent on time explicitly, physically their behavior does not change with time, such that once the step response of a spring-mass system reaches its steady-state value, the response wont change after a certain amount of time has been passed.
3. Systems can be assumed a linear behavior over a certain range of their operation. In such a case, the differential equation that represents the dynamics can be expressed in a linear matrix form  $\dot{x} = Ax + Bu$ .

## 7.1 Three phase method of modeling

A three phase method is employed for modeling a system using mathematical relationships as follows

1. Structuring
  - a) Divide complex system into subsystems
  - b) Establish the input output relationship between various subsystems that can be connected together such as using TF's of every subsystem block
2. Use first principles from physics
  - a) Conservation laws: Conservation laws are the balance equations established using the first principles of physics, most of the commonly used balance equations from physics are as follows
    - i. Force balance (Newton's Laws for linear motion)
    - ii. Torque balance (Newton's Laws for angular motion)
    - iii. Voltage balance (Kirchhoff's voltage law)
    - iv. Current balance (Kirchhoff's current law)
    - v. Volume flows (Fluid flow modeling)
  - b) Constitutive relations: These relations relate variables of different units with each other or some quantity such as
    - i. Ohm's Law (Relates voltage across an electrical component to the current flowing)
    - ii. Ideal gas law (Relates relationship between pressure, volume and temperature of gas molecules)
    - iii. Hooke's Law (Relates relationship between displacement and force of a spring)
    - iv. Air resistance (between velocity of the air and force exerted by it on a cross sectional area)
3. Equations put together in a standardized way for analysis, simulation and design
  - a) Here using State-space method, as it is a modern design concept and has advantages modeling complex systems and their control design over classical control methods
  - b) State-space method uses the following method in general
    - i.  $\dot{x} = f(x, u, d)$ , the evolution of a state variables is a function of the states  $x$  itself, the inputs  $u$  and disturbances to the system  $d$

- ii. the output  $y = h(x, u, d)$  is another function of the states  $x$ , the inputs  $u$  and disturbances while measuring  $d$
- iii. An easy way to choose state variables is to ask: What is the variable in formed equations from phase II that is changing (Hint: Check the variables that are written in time derivatives and start from there). As state-space model is not unique, any other new chosen variable would change model and express the new model with reference to new variable.

In order to illustrate the above modeling method consider a mechanical system of a simple mass-spring system.

#### **7.1.0.1 Phase I: Subsysteming and choosing inputs and outputs**

For the first phase the input can be recognized as  $F(t)$  and output as  $x(t)$ , therefore the subsystem is a system that transform the input  $F(t)$  to displacement  $x(t)$ .

#### **7.1.0.2 Phase II: Mathematical modeling**

In the second phase the following questions are asked

- What is it that changes?
- What is causing this change?

For the case of spring-mass system, both these questions are answered using Newton's Law which relates the motion (effect) in terms of forces (cause) acting on the system using

$$\sum F_{net} = m\ddot{x} \implies F_{ext} - F_{int} = m\ddot{x} \implies F_{ext} - F_{spring} = m\ddot{x} \quad (7.2)$$

Finally, the constitutive relationships for the spring force needs to be established, here using Hooke's law

$$F_{spring} = Kx \quad (7.3)$$

The equations formed using physics can be put together, which forms a compact form of system description expressed mathematically

$$F_{ext} - Kx = m\ddot{x} \quad (7.4)$$

In the third phase, these equations has to be put together (in this case using a state-space model) so as to render its analysis, simulation and design of components (redesigning or control design).

## Nomenclature

### 7.1.0.3 Phase III: Writing State-space form

Looking at equations of motion formed using Phase II, it can be seen that the variable that is changing and the one that has time derivative term in equation (7.4) is  $\ddot{x}$ . State-space forms are always to be written in first order differential or difference equations form. Therefore, further structuring is required using new variables so as to split a second order ODE to a system of two first order ODE's.

$$y = x \quad (7.5)$$

$$\dot{y} = \dot{x} \quad (7.6)$$

The state derivatives can now be expressed as

$$\frac{dy}{dt} = \dot{x} \quad (7.7)$$

$$\frac{\dot{y}}{dt} = \ddot{x} = \frac{F_{ext}}{m} - \frac{Ky}{m} \quad (7.8)$$

It can be seen that equation (7.8) is written in the form of  $\dot{x} = f(x, u, d)$  where the derivative of state-variable is a function of the states  $x$  itself, the inputs  $u$  and disturbances to the system  $d$  (in this case there is no  $d$ ).

## 7.2 Model verification and validation

Verification is the process of determining that the model implemented is accurate. That is, it is a check if the implementation (software, computer code) represents the mathematical model accurately according to the developers conceptual description and the solution to the model

Validation is the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model (check to see if all use case scenarios are matched)

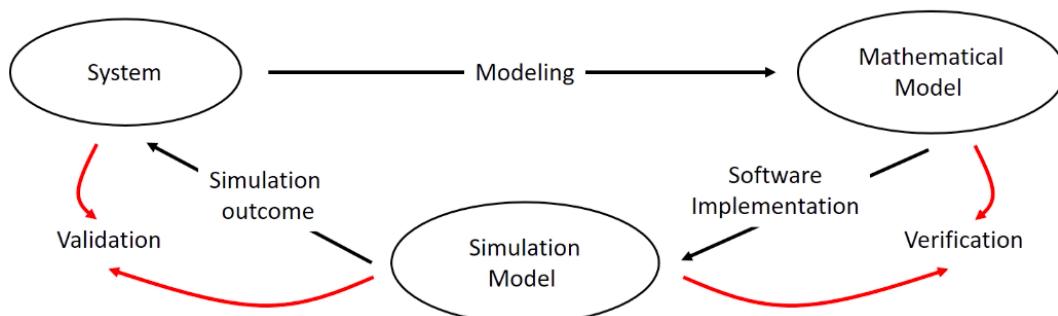


Figure 7.1: Model verification and validation definition

## 7.3 Mathematical Modeling

In mathematical modeling of the dynamic systems, there are two approaches. State space approach and Laplace transform (or writing the transfer function) of the system.

### 7.3.1 State Space Representation

In state space approach the linear time-invariant (constant coefficient) first order differential equation can be written as a linear sum of matrices containing the dynamic states and the control inputs to the system.

For continuous linear time-invariant (LTI) systems, the standard form of state-space representation is [MT19a]:

$$\dot{x} = Ax + Bu \quad (7.9)$$

$$y = Cx + Du \quad (7.10)$$

- $\dot{x}$  is the system dynamics
- if  $u(t)$  is input to the system
- Any measured output should also be a function of  $x(t)$  plus  $u(t)$ .
- Matrix  $A$  defines the system matrix, that is the coefficient of all system states.
- Similarly, matrix  $B$  represents the coefficients of all system inputs.
- Equation  $y(t)$  is the output of the system that is of control importance.
- Matrix  $C$  is used to choose the states that are required for control, other non-important states from the controls point can be ignored.
- Matrix  $D$  is used to choose the inputs that directly influence the outputs, otherwise  $D$  is taken as zero matrix.

## 7.4 Transfer function representation

With transfer function representation, the linear ordinary differential equation representing the dynamics of the states can be transformed into a linear algebraic equation. Laplace transform is conducted to transform a function  $f(t)$  from time domain to a  $s$  domain. A  $s$  domain is a domain of complex numbers  $s = \sigma \pm j\omega$  which are plotted on the complex plane of real and complex numbers. The transfer is performed through the function

$$\int_0^t e^{-st} f(t) dt$$

## Nomenclature

. A  $s$  domain is a plane of complex numbers, where the solution of the transfer function can be plotted and studied easily. It so happens that for a system such as a mass-spring with a damper, the real plane values control the damping and the imaginary values control the frequency of the system respectively.  $s$  domain is not a transformation to a frequency domain rather to a domain of complex numbers that are easier to evaluate and visualize.

*Transfer functions model only linear systems.*

*More detailed mathematical derivations are provided in one note notes.*

## 7.5 General form of Second order System

Consider a PT-2 system such as a mass, spring and a damper system which can be written in the form as:

$$m\ddot{x} + k\dot{x} + cx = 0 \quad (7.11)$$

division by  $m$  on both sides:

$$\ddot{x} + \frac{k}{m}\dot{x} + \frac{c}{m}x = 0$$

Following substitutions can be made using system parameters as given below:

$$\begin{aligned} \frac{k}{m} &= 2\delta \\ \frac{c}{m} &= \omega_n^2 \end{aligned}$$

where  $\delta$  is called the damping coefficient (how fast the oscillations damp out?) and  $\omega_n$  is called the natural or Eigen frequency of the undamped system (how fast does the undamped system oscillate?). Therefore, the above equation becomes:

$$\ddot{x} + 2\delta\dot{x} + \omega_n^2 x = 0 \quad (7.12)$$

Further, a new term damping ratio can be defined as  $\zeta = \delta/\omega_n$  with  $\delta$  the damping coefficient of the system. Therefore, the above equation can be re-written as:

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2 x = 0 \quad (7.13)$$

Using equation (7.13), general form of PT-2 non-homogeneous systems can be expressed as:

$$\ddot{y}(t) + 2\zeta\omega_n\dot{y}(t) + \omega_n^2 y(t) = \omega_n^2 u(t) \quad (7.14)$$

Once a general form of PT-2 system is obtained then in the literature there are various ways to solve them using methods such as Laplace transform as described in section 8.4.3, Algebraic (Eigen-Value analysis) as described in section 8.4.1 and Numerical (Matlab). All these methods are aimed primarily at determining the modes of motion that the PT-2 system exhibits and thereby determine its stability.

## 7.6 Non-Canonical Systems

Non-canonical systems are the ones that do not fit into the domain of first and second order systems. They are generally of the following types:

1. Higher order systems (more than two poles)
2. Nonlinear systems
3. Systems with zeros ( $s$  terms in the numerator)

### 7.6.1 Higher order systems

In general the response of the higher order systems can be considered as the sum of first and second order responses. The exact time responses can be determined using partial fraction expansion in Laplace domain. The poles themselves can have only real roots or complex conjugates, with such a limited possibility it is easier to then map the responses of the higher order systems with the responses learnt from such roots in the first and the second order systems. For example a higher order system can be expressed in Laplace domain as follows:

$$\frac{K}{(s+a)(s^2 + \zeta\omega_n s + \omega_n^2)} = \frac{A}{s+a} + \frac{B}{s^2 + \zeta\omega_n s + \omega_n^2} \quad (7.15)$$

Additionally, most of the higher order system responses can be approximated with either the first or the second order responses. This is either because of dominant roots of first or second order type which would dominate the response of the system (approximating the overall behavior) or because of the system behavior in terms of component or subcomponent level where the states exhibit the response of canonical systems (as in the case of a vacuum tube which is very nonlinear in its response in the component levels but linear in its higher subsystem level).

#### 7.6.1.1 Model reduction

This section provides some ideas on model reductions of higher order systems. One of the techniques used in model reduction is to split the higher order terms into simple components of first and second orders working as components or subsystems for the entire system of the higher order. An example:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{20}{(s+2)(s+10)} \quad (7.16)$$

Such a higher order system can be replaced with a components of first order systems connected in series such as:

$$\frac{20}{(s+2)(s+10)} = \frac{2}{s+2} \cdot \frac{10}{s+10} \quad (7.17)$$

## Nomenclature

this is not partial fraction expansion of transfer function (required to solve for the coefficients) and cannot be performed using Matlab, rather this is just a simple splitting of the subsystem into components ***connected in series***. Also, this operation cannot be performed on electrical circuits though they are connected in series as circuit have cascading effects (the later components load onto the former components). This system now has two poles at  $s = -10$  and  $s = -2$  in which case the response of the system can be determined as settling time using these two poles as  $e^{-10t}$  and  $e^{-2t}$  ( $e^{\sigma t}$  would be a possible analytical solution for a first order ODE). Also noting that the system has only real poles therefore, it signifies damping in the system. Here the component with response  $e^{-10t}$  is much quicker than component with  $e^{-2t}$  as the response will be equal to  $t_s = \sigma/10$  and  $t_s = \sigma/2$  respectively.

Using Matlab it can be seen that the response time of the system becomes quicker with increasing the damping ratio  $\zeta$ . As  $\zeta$  is doubled and quadrupled from  $\zeta = 0.2$  to  $\zeta = 0.4$  and  $\zeta = 0.8$ , though the damping the higher the response times would also become more quicker which also doubles and quadruples as given by the real part of the poles from  $s_{1,2} = -1$  to  $s_{1,2} = -2$  and  $s_{1,2} = -4$ . Therefore, higher damping ratios  $\zeta$  will make the system responses much quicker this will greatly reduce the period of the oscillation and therefore it will not be comforting experience when used as shock absorbers in vehicles. Another thing to note is that the damping ratio  $\zeta$  does not change the steady state step response at all, it only changes the settling time  $t_s$  taken by the system to reach the steady state as shown in the figures (7.2, 7.3 and 7.4).

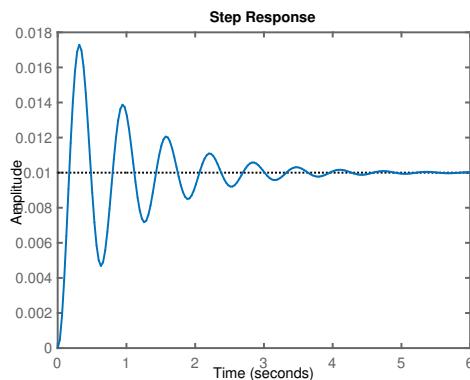


Figure 7.2: Step response at  $\zeta = 0.2$

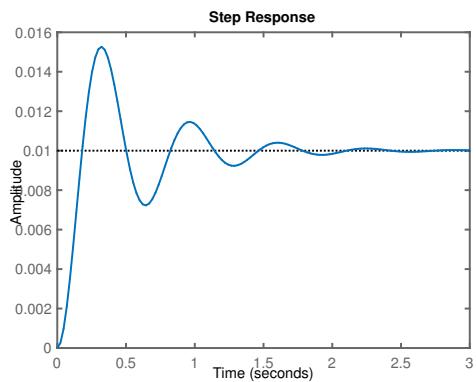


Figure 7.3: Step response at  $\zeta = 0.4$

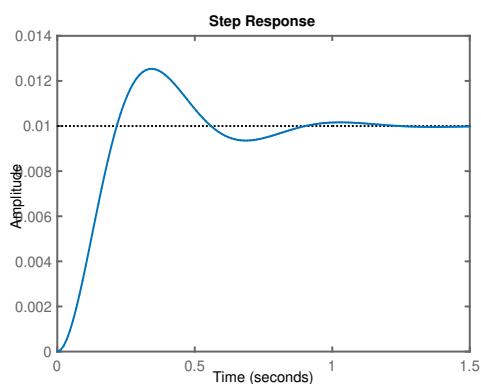


Figure 7.4: Step response at  $\zeta = 0.8$

## Nomenclature

Going back to the example given by equation (7.17), the responses of the individual components can be plotted using Matlab as shown in figures 7.5 and 7.6:

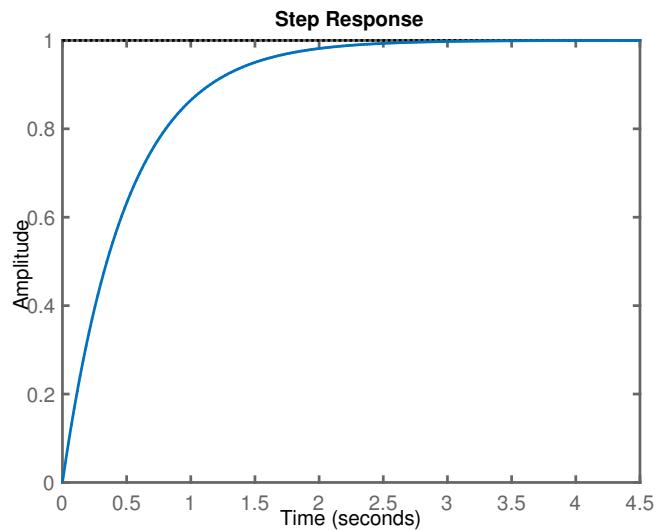


Figure 7.5: Step response for pole at  $s = -2$

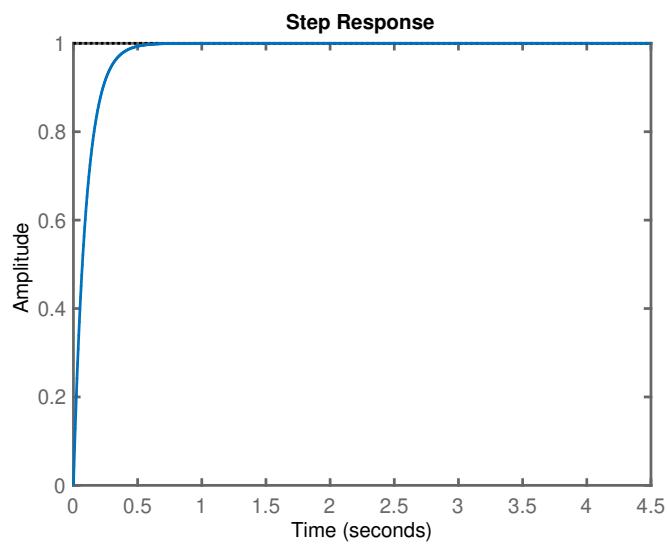


Figure 7.6: Step response for pole at  $s = -10$

It can be seen clearly from figures (7.5 and 7.6) that the response of the system is much higher at higher damping ratios and this does not change the step response of the system itself. Generally since the response of component with response  $e^{-2t}$  will be much slower it will dominate the overall system behavior as this makes the entire system as slow as response  $e^{-2t}$  is. Consider figure 7.7, which shows the response of the whole system with both the poles  $s_{1,2} = \{-2, -10\}$ . It can be seen from figure 7.7

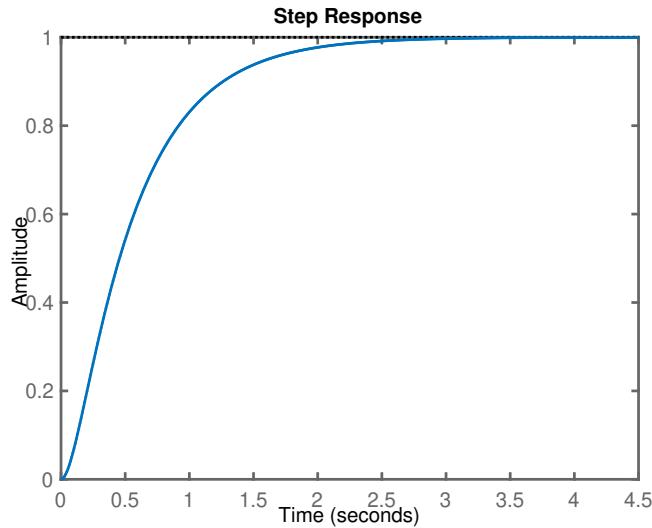


Figure 7.7: Step response for whole system with both the poles at  $\{-2, -10\}$

that the response of the system as a whole and the response of the system with only one component with pole at  $s = -2$  is almost the same. Therefore, such techniques can be used to reduce the size of a higher order model by approximating the behavior to a lower order systems.

In such a case the response of the component with response  $e^{-10t}$  have almost no influence on the system, however, care has to be taken to include its steady state contribution which can be done just by using a static gain  $K$  in Simulink. The size of  $K$  depends on the value of the step function (DC gain from the faster component) that goes as an input to the slower component. For example if the step response (of a faster component) is of magnitude 1, then a static gain of  $K = 1$  will be given as input to the slower component. In general the gain that is needed to replace into the reduced system can be found using ***final value theorem***. Consider the system given by equation (7.17), the gain of the system can be given by:

$$G(s) = \frac{20}{(s+2)(s+10)} \approx G_{Red}(s) = \frac{K}{s+2} \quad (7.18)$$

here gain  $K$  has to be set such that  $G(s)$  and  $G_{Red}(s)$  are both the same.  $K$  or DC gain of a system is defined as ***the steady-state response of the system for a DC input***

## Nomenclature

(constant input) such that:

$$k_{dc} = \lim_{s \rightarrow 0} s Y(s) \quad (7.19)$$

where

$$Y(s) = G(s)U(s) = \frac{20}{(s+2)(s+10)} \frac{1}{s}$$

therefore,

$$k_{dc} = \lim_{s \rightarrow 0} s \frac{20}{(s+2)(s+10)} \frac{1}{s} = \lim_{s \rightarrow 0} G(s) = G(0)$$

$k_{dc}$  for the original system given by equation (7.17) can be found therefore as follows:

$$G(0) = \frac{20}{(0+2)(0+10)} = 1 \quad (7.20)$$

A  $k_{dc} = 1$  will produce a steady-state value of  $G(0) = 1$  for the system. Therefore, the DC gain of the reduced system should be  $k_{dc} = 1$ , such that:

$$G_{Red}(s) = \frac{1}{s+2} \quad (7.21)$$

## 7.6.2 Effect of zeros

**Zeros** are the values of  $s$  that make the numerator of the transfer function  $G(s)$  go to zero. They are the counterparts of poles. Their effect on the poles can be studied using partial fraction expansion in Matlab using the command  $[C,p] = \text{residue}(20,[1 12 20])$ .

### 7.6.2.1 Effect 1: Contribution of each pole to the transfer function

Consider the original system with no zeros and its partial fraction expansion:

$$G(s) = \frac{20}{(s+2)(s+10)} = \frac{2.5}{s+2} - \frac{2.5}{s+10} \quad (7.22)$$

now with zeros induced into the system the partial fraction expansion:

$$G(s) = \frac{s+20}{(s+2)(s+10)} = \frac{2.25}{s+2} - \frac{1.25}{s+10} \quad (7.23)$$

zeros effectively change the coefficients of the transfer function ie., they change the contribution of the poles on  $G(s)$ .

### 7.6.2.2 Effect 2: Increases the overshoot

Consider the explanation given in figure 7.8:

# Effect of Zeros

## Interpretation #2

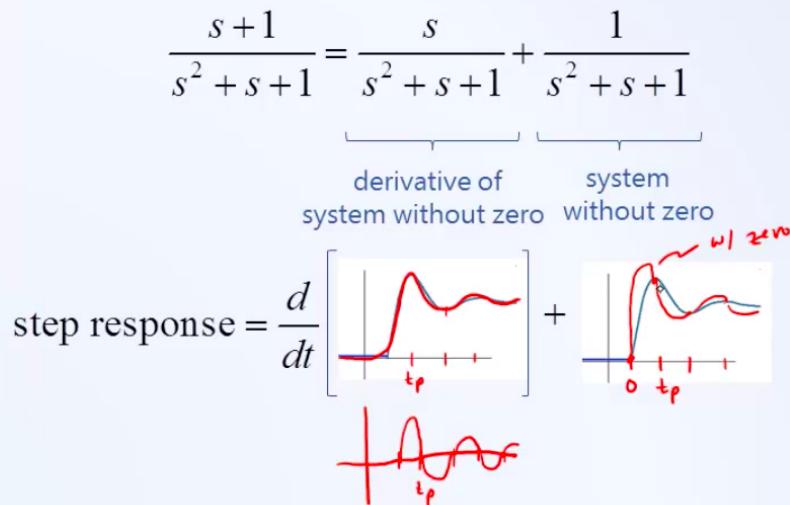
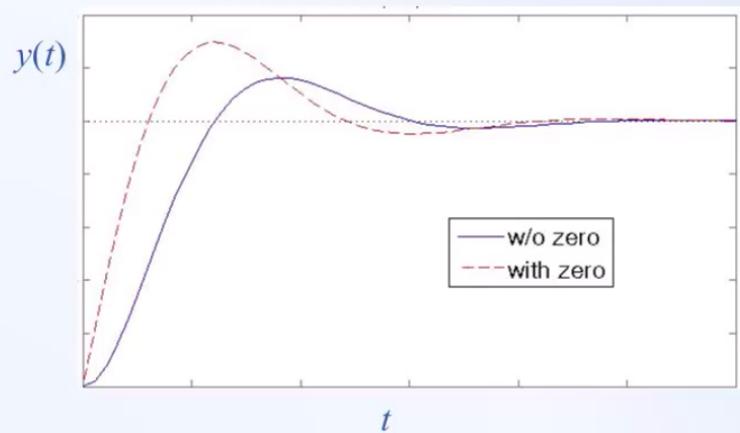


Figure 7.8: Zeros increase the overshoot of the system

As multiplication of  $s$  to a transfer function is considered derivative we have a first term in the equation that the derivative of the step response. Adding the derivative part will carry the values of the derivatives to the step response which will increase the overshoot of the system as shown in figure 7.8. Further consider figure 7.9, in which it is clearly shown that zeros increase the overshoot and reduce the response time. This in-turn will induce more oscillations into the system:

## Effect of Zeros



- Zeros tend to have the effect of increasing overshoot and decreasing rise/peak time

Non-Canonical Systems

Figure 7.9: Zeros increase the overshoot of the system

### 7.6.2.3 Effect 3: Non-minimum Phase Behavior

If the zero has a *positive real root*, then it will induce into a system a response which will initially move in the reverse direction. Such a behavior is called a non-minimal phase behavior as exhibited by a bicycle when given an impulse torque to its steering.

### 7.6.2.4 Effect 4: Canceling poles that are near them

## Effect of Zeros

- An exception to the preceding rules is when a zero is near a pole

$$\tilde{H}(s) = \frac{s+z}{s+p} H(s)$$

If  $z = p$ , then pole and zero cancel exactly and  $\tilde{H}(s) = H(s)$

If  $z \approx p$ , then effect of pole and zero are minimal and  $\tilde{H}(s) \approx H(s)$

- Example:

$$\frac{\cancel{s+1}}{(s+1.1)(s+10)} \approx \frac{1}{s+10}$$

Non-Canonical Systems

Figure 7.10: Zeros which are close to poles cancel them

## Nomenclature

### 7.6.3 Nonlinear Systems

In most of the cases the solution to the nonlinear systems are only possible numerically. Numerical approximation of the solutions do not yield the conceptual understanding that is possible with analytical solutions. One way to have an analytical solution is to use **Model reduction techniques** in order to simplify (approximate) the nonlinear system into a linear system in a neighborhood about an operating point. Such as the linearizing functions that are used around an operating point in mathematics. Which defines the approximation of a nonlinear function  $f(x)$  about an OP  $\bar{x}$ :

$$\frac{df}{dx} \approx \frac{f(x) - f(\bar{x})}{\Delta x} \quad (7.24)$$

$f(x)$  is linearized using the above relation as:

$$f(x) \approx f(\bar{x}) + \frac{df}{dx} \Delta x \quad (7.25)$$

Lienarization is possible only near the small range of neighborhood just near the OP. As the system deviates from the OP the approximation made by the lienarization deviates rapidly. A more detialed analysis on linearization is shown in chapter 21.

# 8 System Analysis

From the mathematical models, system analysis is performed in order to determine their behavior in time and frequency domains. Because control systems are then used on these dynamic states of the system to control their stability, steady-state error, tracking etc., among various others once the behavior in time and frequency domains are established.

## 8.1 Time-domain analysis

Time-domain analysis is response of the state of the dynamic system in time when given an input. The history of the response of the dynamic state due to the input is plotted as the time-response of the system.

As dynamic systems are modeled using ODE's, their response can be determined using integration over a certain period of time either analytically or numerically. The time response of a LTI system consists of **transient response** and **steady-state response** due to an input to the system. These responses correspond to the null solution and the particular solution of the governing differential equation [MT19b].

## 8.2 Frequency domain analysis

LTI systems have a property such that for any signaled input to the system, the output is also with the same frequency but could vary in phase and magnitude. This difference in phase and magnitude is the function of the input frequency and hence such response corresponds to the frequency response of the system [MT19b].

In order to find the frequency response of the system, first the transfer function of the system is derived. Then using the complex variable  $s$ , the values of the plant transfer function is derived at every frequency starting from (0 to  $\infty$ ). Here,  $s$  is given by  $s = \sigma \pm j\omega$ ,  $\sigma$  is the real part of the root and indicates the damping in the system,  $j\omega$  terms that lie on the imaginary axis indicates the frequency of the system's oscillation. Therefore, by substituting  $G(s) = G(j\omega)$  ( $s = j\omega$ ), a frequency response of the system can be studied. If  $G(s)$  is the open-loop transfer function of the system and  $\omega$  is the vector of input frequencies, then the frequency response can be evaluated for every  $\omega$  [MT19b].

With the transfer function  $G(s)$ , the frequency response of the function  $G(s)$  can be derived for every  $s = j\omega$  and a plot of output  $G(j\omega)$  versus input frequencies  $\omega$  is

## Nomenclature

plotted which is called **Bode Plot**. Since  $s$  is in complex plane both the magnitude and the phase of output frequencies  $G(j\omega)$  can be plotted using **Bode Plot**. As well as the position of  $G(j\omega)$  in the complex plane can be studied using **Nyquist Diagrams** [MT19b].

## 8.3 Analysis of first order systems

A first order system hereby called PT-1 systems can be expressed in their most general form as:

$$\tau \dot{y}(t) + y(t) = k_{dc}u(t) \quad (8.1)$$

Where  $\tau$  and  $k_{dc}$  are the parameters of the system or the constant coefficients of ODE.  $k_{dc}$  is the amount of scaling of the systems output for a constant input.  $\tau$  defines the amount of time required for the system to obtain 63.2% of the transient dynamics that the system undergoes. It is easier to determine the solutions of the equation (8.1) in algebraic form, (8.1) is written in the Laplace domain as:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{k_{dc}}{\tau s + 1} \quad (8.2)$$

Again, where,  $\tau$  and  $k_{dc}$  are the parameters that completely define the response characteristics of a first order system and are called time constant and DC gain respectively. A time response of a PT-1 system is described using time signals such as step or impulse as inputs.

From equation (8.2), the denominator values of  $s$  for which the equation goes to infinity is called the pole (values) of the system given by equation (8.2). In equation (8.2), we have only one possible value of  $s$  at  $s = -1/\tau$ . Such a system with one pole is also defined as a first order system. Similarly a second order system will have two poles [Hil13a, t.2.05].

### 8.3.1 Impulse response

Consider a unit impulse as input to the first order system. The Laplace transform of the impulse signal is given by  $L(s) = 1$ . Therefore, the first order system with an impulse input is expressed as:

$$Y(s) = G(s)U(s) = \frac{k}{\tau s + 1} \cdot 1 \quad (8.3)$$

the inverse Laplace of the equation (8.3) will express the solution of the equation of the differential equation (8.1) with an unit impulse input:

$$y(t) = \frac{k}{\tau} e^{-\frac{t}{\tau}} \quad (8.4)$$

The above equation is the null solution of the differential equation (8.1) given as  $y = Ce^{at}$  with initial condition  $C = k/\tau$  and  $a = -1/\tau$ . From the solution (8.4) it can be inferred that for an unit impulse, the response of the system at time  $t = \infty$  is to settle down to zero with an exponential decay. From a numerical solution generated in MATLAB, the response decays to zero as shown in figure 8.1.

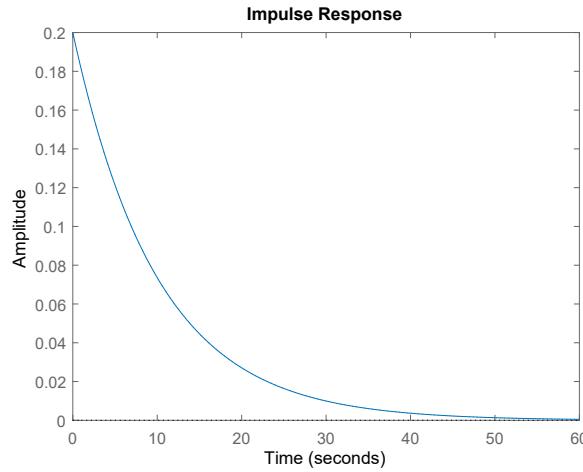


Figure 8.1: Impulse response of a first order system

### 8.3.2 General first order responses

A PT-1 system is first expressed in time-constant form as given below:

$$\tau \frac{dy(t)}{dt} + y(t) = Ku(t) \quad (8.5)$$

where  $y(t)$  and  $u(t)$  are the state and inputs of the system respectively. The behavior of the system given by the above equation is described for  $t > 0$  for various inputs  $u(t)$  and / or initial conditions. The behavior is focused on how the output  $y(t)$  changes with time constant  $\tau$ .

A time constant  $\tau$  is defined for a more general system described by constant coefficients:

$$a \frac{dy(t)}{dt} + by(t) = cu(t) \quad (8.6)$$

writing the above equation in time-constant form is to write the equation with coefficients of  $y(t) = 1$ . In such a case, the following operation is performed:

$$\frac{a}{b} \frac{dy(t)}{dt} + y(t) = \frac{c}{b} u(t) \quad (8.7)$$

## Nomenclature

therefore, the time-constant is defined by the ratio  $a/b$ , using the constant coefficients of PT-1 system. The ratio  $a/b$  appears along with time variable  $t$  in the solution of PT-1 systems and therefore serves as a factor to split and measure time, which is used to measure the system output at those factored times.

Similarly, the gain of the system is defined using the ratio  $c/b$ , as this appears as the coefficient of  $u(t)$ , it is the scaling of  $y(t)$  for the given input  $u(t)$ . Therefore, the ratio  $c/b$  is defined the gain of the system.

### 8.3.2.1 $u(t) = 0$ , non-zero initial condition

$$\tau \frac{dy}{dt} + y = 0 \implies y(t) = y(0)e^{-t/\tau} \quad (8.8)$$

The solution shown above can be verified using LT:

$$\implies \tau[sY(s) - y(0)] + Y(s) = 0 \implies [s\tau + 1]Y(s) - \tau y(0) = 0 \quad (8.9)$$

Rearranging the above equation:

$$Y(s) = \frac{\tau y(0)}{s\tau + 1} = \frac{y(0)}{s + 1/\tau} \implies y(t) = y(0)e^{-t/\tau} \quad (8.10)$$

These responses are characterized primarily by time-constant, which are the results derived from the solution:

1. if  $t = \tau$ ,  $\implies y(t) = y(0)e^{-1} = 0.37y(0)$
2. if  $t = 2\tau$ ,  $\implies y(t) = y(0)e^{-2} = 0.14y(0)$
3. if  $t = 3\tau$ ,  $\implies y(t) = y(0)e^{-3} = 0.05y(0)$
4. if  $t = 4\tau$ ,  $\implies y(t) = y(0)e^{-4} = 0.02y(0)$

summary of responses:

time	e	response
$t = \tau$	$e^{-1}$	37 % of initial condition
$t = 2\tau$	$e^{-2}$	14 % of initial condition
$t = 3\tau$	$e^{-3}$	5 % of initial condition
$t = 4\tau$	$e^{-4}$	2 % of initial condition

Table 8.1: Table summarizing PT-1 response for non-zero initial conditions

### 8.3.2.2 Step response

The object is to find the solution to the differential equation  $y(t)$  as given by the equation (8.1). This system with a unit step input  $u(t) = 1$  can be expressed in the Laplace domain by multiplying the input signal as:

$$Y(s) = G(s)U(s) = \frac{k}{\tau s + 1} \frac{1}{s} \quad (8.11)$$

Taking the inverse Laplace transform of equation (8.11), the algebraic solution of the ODE given by equation (8.1) can be expressed as:

$$y(t) = k - ke^{-t/\tau} \quad (8.12)$$

The solution of the PT-1 system should have both null and particular solutions. The term with exponential function was initially assumed response of a homogeneous system. Therefore, the first term is the particular solution and the second term is the null solution of the ODE for PT-1 systems. In general, the terms of the null solution will impose the dynamics of the states in the system which decay after a while leaving behind the steady-state impact of the particular solution. The influence of the particular solution remains in the system as long as the input remains active without changing. Further the stability of the system can be analyzed using **BIBO** or using **Eigen value Analysis**. Both the analyses lead to the same conclusion of system stability that do not blow up while operation. A BIBO stability can be performed numerically using MATLAB by choosing system parameters  $\tau$  and  $k_{dc}$  and a plot is generated as shown in figure 8.2. Further analysis on system parameters can be made numerically such as

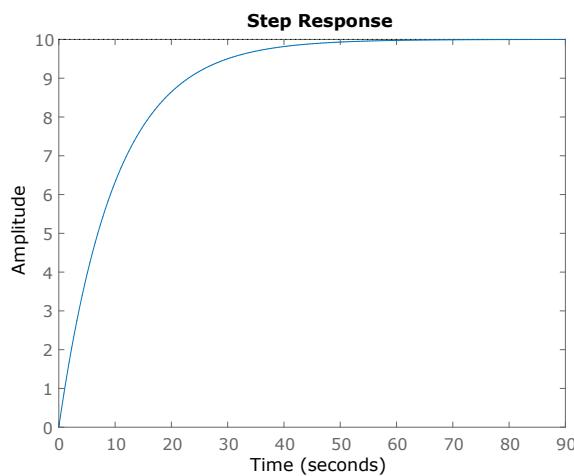


Figure 8.2: Step response of a first order system

the DC gain can be inferred from the steady-state value reached by the system. For a unit step input, the DC gain is given by the step-up value at which the output of the

## Nomenclature

system's response settles down, in this case,  $k = 5$  which is evident from the system value chosen to generate this plot. The time constant  $\tau$  can be inferred from plot and equation (8.12), for time  $t = \tau$ :

$$y(t) = k(1 - e^{-1}) = 0.632k \quad (8.13)$$

therefore, at one time constant  $\tau$ , the systems response settles down to about 63.2% of the steady-state vale. Generally, a steady-state response is fixed when this value reaches 98% of steady-state value, which from the plot and equation (8.12) can be inferred as at time  $t = 4\tau$ :

$$y(t) = k(1 - e^{-4}) = 0.98k \quad (8.14)$$

Summary of PT-1 systems step response:

time	e	response
$t = \tau$	$(1 - e^{-1})$	63 % of steady-state response
$t = 2\tau$	$(1 - e^{-2})$	84 % of steady-state response
$t = 3\tau$	$(1 - e^{-3})$	95 % of steady-state response
$t = 4\tau$	$(1 - e^{-4})$	98 % of steady-state response

Table 8.2: Table summarizing step response of PT-1 system

### 8.3.2.3 step response with non-zero initial condition

For a linear system, the response due to two different inputs can be determined using principle of superposition. Since we already now the response from step input and non-zero initial condition, they both can just be added together. Therefore, from equations (8.10) and (8.12), we have:

$$y(t) = y(0)e^{-t/\tau} + K(1 - e^{-t/\tau}) = (y(0) - K)e^{-t/\tau} + K \quad (8.15)$$

The term  $(y(0) - K)e^{-t/\tau}$  in equation (8.15) gives the exponential decay from the IC and ss-response  $K$  from step input.

The above response can also be derived using LT, by putting both LT's for IC and step-function into the equation of PT-1 system:

$$[s\tau + 1]Y(s) - \tau y(0) = \frac{K}{s} \quad (8.16)$$

then solving the above equation for  $Y(s)$  and doing inverse LT, the solution can be derived.

Observations made from the solution: Consider the response given by equation (8.15), the following observations can be made:

1. ss: depends on the gain  $K$  and the input of the system. Therefore, ss =  $K$
2. Dynamics: depend on how fast the exponential goes to zero, this depends solely on the time constant  $\tau$
3. The term  $(y(0) - K)$  in equation (8.15) defines the distance from IC to the ss

The response by changing  $\tau$ , let  $(y(0) - K) = d$ :

time	d	y(t)	response
$t = 0$	$y(0) - K$	$y = K + d = y(0)$	At the initial state
$t = \tau$	$0.37(y(0) - K)$	$y = K + 0.37d$	37 % deviation from ss
$t = 2\tau$	$0.14(y(0) - K)$	$y = K + 0.14d$	14 % deviation from ss
$t = 3\tau$	$0.05(y(0) - K)$	$y = K + 0.05d$	5 % deviation from ss

Table 8.3: Table summarizing response of PT-1 due to step input and IC

From the list and the able given above, it seems that for PT-1 systems, with step input and non-zero IC, the system starts from IC and ends at ss. Where IC =  $Y(0)$  and ss =  $K$ .

consider an example PT-1 system:

$$2 \frac{dy}{dt} + 6y = 4u \quad (8.17)$$

writing in time-constant form:

$$\frac{1}{3} \frac{dy}{dt} + y = \frac{2}{3}u \quad @ \quad y(0) = -1 \quad (8.18)$$

Before, generating the plot, the following observations can be made by comparing to the standard response:

- $\tau = 1/3$  and  $K = 2/3$
- at  $t = 0$  response starts at  $y(0) = -1$  at a distance  $d = y(0) - K$  and ends at ss =  $K = 2/3$  when  $t > 3\tau$
- for a ss of 5 %, simulation time should run upto  $t = 3\tau$

## 8.4 Stability analysis of PT-2 systems

Once a model of a system has been established, it may be required to analyze the model in order to determine the response of the system, in this case in the time domain. In the perspective of a control design, the control system is often designed to improve the stability of the system, improve the speed of the response or the steady-state error, or prevent oscillations. Therefore, it is necessary to determine the dynamic properties of the system from the system model. In this section, stability is defined formally for linear and non-linear systems (such as a bicycle).

### 8.4.1 Stability in linear systems

For defining the important concepts in stability analysis, a simple second order linear and homogeneous system is considered as described below:

$$\ddot{y} + B\dot{y} + Cy = 0 \quad (8.19)$$

For a special case (roots of characteristic equation lie in complex  $s$  plane) such that the solution of the above ODE is given by  $y = e^{st}$ , the above ODE can be re-written into its characteristic algebraic form by substituting  $y = e^{st}$ :

$$s^2 + Bs + C = 0 \quad (8.20)$$

the roots of the above equation are given by:

$$\{s_1, s_2\} = \frac{-B \pm \sqrt{B^2 - 4C}}{2} \quad (8.21)$$

and the null (homogeneous) solution of the equation (8.20) can be expressed as the linear combination of roots:

$$y(t) = C_1 e^{s_1 t} + C_2 e^{s_2 t} \quad (8.22)$$

Alternatively, writing equation (8.19) in a general state equation matrix form (two first orders):

$$\frac{d}{dt} \begin{bmatrix} y \\ y' \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -C & -B \end{bmatrix} \begin{bmatrix} y \\ y' \end{bmatrix} \quad (8.23)$$

alternatively, the above equation can be realized in the form:

$$\frac{d}{dt} \mathbf{Z} = [\mathbf{A}] [\mathbf{Z}] \quad (8.24)$$

where  $\mathbf{Z}$  is the state matrix  $[y \ y']^T$ ,  $\mathbf{A}$  is called the companion matrix (also system matrix for the dynamic states  $\mathbf{Z}$ ), matrix  $\mathbf{A}$  can be used to find the Eigen-Values of  $\mathbf{A}$ :

$$\{\lambda_1, \lambda_2\} = \frac{-B \pm \sqrt{B^2 - 4C}}{2} \quad (8.25)$$

and the null solution of the second order differential equation:

$$y(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t} \quad (8.26)$$

Both the null solutions predicted by the roots  $\{s_1, s_2\}$  and Eigen-Values  $\{\lambda_1, \lambda_2\}$  give the solution of the differential equation (8.19), however, with Eigen-values it is much easier to reach at the null solution. From the roots  $\{s_1, s_2\}$  or the Eigen-Values  $\{\lambda_1, \lambda_2\}$ , following cases can be drawn about the stability of the second order system:

Cases	Condition	Eigen-Values
Over-damped	$(B^2 > 4C)$	Negative distinct Roots
Critically damped	$(B^2 = 4C)$	Negative repeated roots
Under-damped	$(B^2 < 4C)$	Complex conjugate roots
Undamped	$B = 0$	Imaginary Roots

Since the characteristic equation (8.20) is quadratic in nature, the roots will be of complex in nature. With complex roots it can be said that the imaginary part will indicate the frequency of systems oscillations and the real part will indicate the damping (or growth based on parameter  $B$ ) of the system. In case of undamped system, even the BIBO stability can not be said to be achieved completely as the system will be oscillating with an undamped magnitude (high risk of resonance). Therefore, the BIBO and the asymptotic stability of the system can only be defined in the cases where there is a damping [Hil13a].

In terms of BIBO stability the stability of the PT-2 system can be roughly described using the values of the real part and can be described as follows:

- Pure oscillations: Real part of poles is zero.
- Exponential decay (BIBO stability): Real part are negative.
- Exponential growth (unstable): If at-least one of the Real part is positive.

**Asymptotic stability:** The PT-2 system is considered to be asymptotically stable if and only if, the state  $x \rightarrow 0$  as  $t \rightarrow \infty$  [Hil13a, t.8.45].

**BIBO stability:** A system is said to be BIBO stable if for any bounded inputs the system outputs remain bounded ie., the system output does not blow up.

For linear systems, using these two stability criterion the stability of PT-2 system can be established.

### 8.4.2 Stability Analysis in Non-linear systems

With linear systems, if the system is already in the stable condition, then regardless of the initial conditions, the system will get back to its stable position. However,

## Nomenclature

a non-linear system is stable only if the perturbations lie within a certain range of neighborhood. Lets consider a non-linear autonomous system (time-invariant system) expressed as a first order system dependent on velocity:

$$\dot{x} = f(x) \quad (8.27)$$

The equilibrium states can be defined for a non-liner system if it satisfies the condition:

$$\dot{x} = f(x_e) = 0 \quad \forall t > t_0 \quad (8.28)$$

where  $x_e$  is the point at which the function  $f(x) = 0$  and  $x_e = \text{constant}$ . The neighborhood is defined as: Given perturbation  $\delta > 0$ , a state vector  $x(t)$  is said to be in the neighborhood  $B_b(x_r(t))$  of the state  $x_r(t)$  if:

$$\|x(t) - x_r(t)\| < \delta \quad (8.29)$$

then,

$$x(t) \in B_\delta(x_r(t)) \quad (8.30)$$

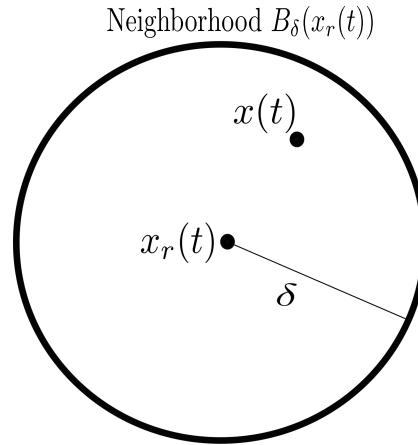


Figure 8.3: Neighborhood of a non-linear system stability

**Lagrange stability** is defined as: the state  $x(t)$  is said to be Lagrange stable (or bounded) relative to  $x_r(t)$  if there exists a  $\delta > 0$  such that:

$$x(t) \in B_\delta(x_r(t)) \quad \forall t > t_0 \quad (8.31)$$

**Lyapponov stability** of non-linear system is defined as: the state  $x(t)$  is said to be Lyapponov stable relative to  $x_r(t)$  if for each  $\epsilon > 0$  there exists a  $\delta(\epsilon) > 0$  such that:

$$x(t_0) \in B_\delta(x_r(t_0)) \quad (8.32)$$

$$x(t) \in B_\epsilon(x_r(t)) \quad \forall t > t_0 \quad (8.33)$$

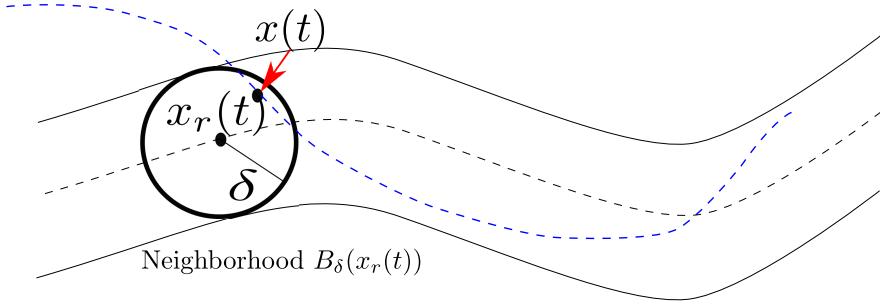


Figure 8.4: Lagrange Stability

Unlike Lagrange stability, in Lyaponov stability lets to define another neighborhood  $B_\epsilon(x_r(t))$  such that perturbations in the previous neighborhood  $B_\delta(x_r(t_0))$  would lead to  $B_\epsilon(x_r(t))$ , hence, unlike Lagrange stability which does not depend on the initial conditions (but on the physics of the problem) Lyaponov stability narrows down the initial states that could lead to stability in the motion in a later state. Also, unlike in the linear systems, the non-linear systems would lead to convergence just because the system is in the stable condition. The boundary  $B_\epsilon(x_r(t))$  could be very close to  $B_\delta(x_r(t_0))$  and the system would be stable in the  $\epsilon$ . For numerical simulation, it is required that the simulations are run for long enough time to prove that the stability remains within the conditions described by the Lyaponov conditions.

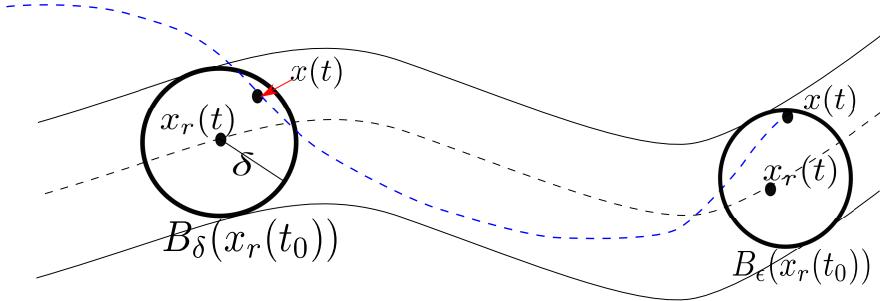


Figure 8.5: Lyaponov Stability

Once Lyaponov stability has been defined, an ***Asymptotic stability*** can be extended and defined as: the state  $x(t)$  is asymptotically stable relative to  $x_r(t)$  if  $x(t)$  is Lyaponov stable and there exists a  $\delta > 0$  such that:

$$x_0(t) \in B_\delta(x_r(t_0)) \text{ then,} \quad (8.34)$$

$$\lim_{t \rightarrow \infty} x(t) = x_r(t) \quad (8.35)$$

Therefore, in case of asymptotic stability, the boundary due to  $\epsilon$  goes to zero and the boundary  $B_\epsilon(x_r(t))$  does not exists anymore, so now the motion  $x(t)$  converges to the reference  $x_r(t)$ . This is an ideal condition to converge to from controls perspective.

## Nomenclature

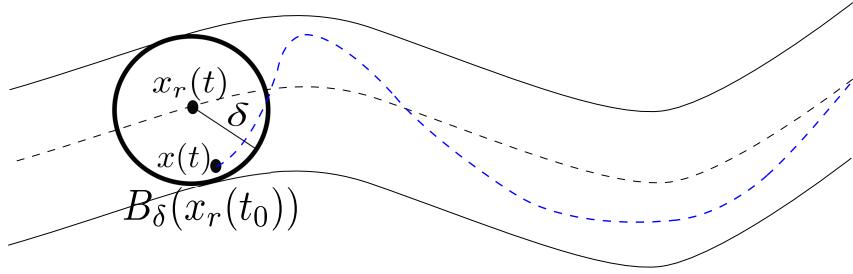


Figure 8.6: Asymptotic stability

Certain dynamic systems have shown to converge to this stability point in their natural modes of operation. A bicycle is one such example where under a certain range of velocities the weave motion of the bicycle converges to asymptotic stability by damping (not physically damping like in friction) the weave motion towards a zero lean and steering rates.

### 8.4.3 Solving / Analysing PT-2 using Laplace transform

Laplace transform of the PT-2 system can be expressed as:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (8.36)$$

the poles of the denominator in equation (8.36) can be found:

$$s_{1,2} = \frac{-2\zeta\omega_n \pm \sqrt{4\zeta^2\omega_n^2 - 4\omega_n^2}}{2} = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$$

with poles  $s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$ , the stability analysis of the system can be made based on the numerical values of the roots. Only  $\zeta$  is a variable and  $\omega_n$  is a constant also  $\zeta$  can be controlled as a variable as the system frequency  $\omega$  varies. Therefore,  $\zeta$  now becomes the fundamental variable for the behavioral changes in the motion as follows:

---

$\zeta = 0$	poles are Im	undamped
$\zeta < 1$	poles are complex	under-damped
$\zeta = 1$	repeated real poles	critically damped
$\zeta > 1$	distinct real poles	over damped

---

#### 8.4.3.1 Determining the system behavior

Various system behaviours can be determined using the results from the previous analysis and they are important which define the PT-2 systems according to their behavior.

## Second-Order Systems

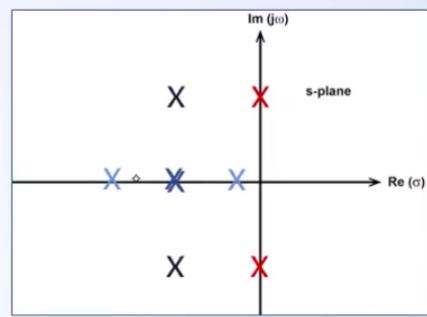
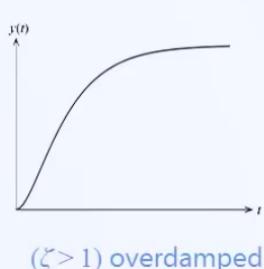
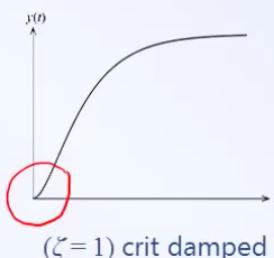
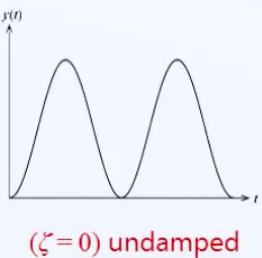


Figure 8.7: Poles in the PT2 system

Using an underdamped system response various system behaviors can be defined as with underdamped systems all the system behavioral characteristics are clearly visible. Using Laplace transforms with a step function the solution of PT-2 system can be found using inverse Laplace as shown in the red box in figure 8.8. Several important properties highlighted by the solution obtained in figure 8.8 can be seen visually as shown in figure 8.9.

*Nomenclature*

Second-Order Systems  $G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$

- Underdamped step response ( $u(t) = 1(t)$ )

$$U(s) = \mathcal{L}[1(t)] = \frac{1}{s}$$

$$Y(s) = G(s) \cdot U(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \cdot \frac{1}{s} = \frac{\omega_n^2}{(s + \sigma)^2 + \omega_d^2} \cdot \frac{1}{s}$$

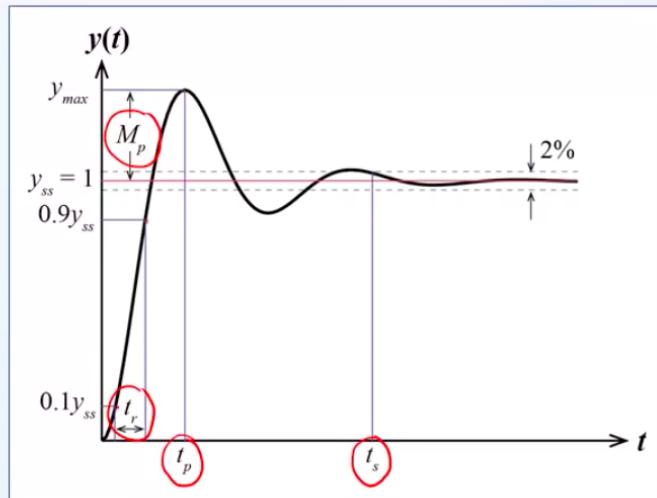
$$= \frac{1}{s} - \frac{s + \sigma}{(s + \sigma)^2 + \omega_d^2} - \frac{\sigma}{\omega_d} \frac{\omega_d}{(s + \sigma)^2 + \omega_d^2}$$

$$y(t) = \mathcal{L}^{-1}[Y(s)] = 1 - e^{-\sigma t} \cos \omega_d t - \frac{\sigma}{\omega_d} e^{-\sigma t} \sin \omega_d t$$

Figure 8.8: Solution to PT-2 system with step input using Laplace transform

## Second-Order Systems

$$y(t) = 1 - e^{-\sigma t} \left( \cos \omega_d t + \frac{\sigma}{\omega_d} \sin \omega_d t \right), t \geq 0$$



Several important properties for specifying a system come from its step response

Figure 8.9: Important behavioral properties seen in PT-2 systems

These important properties can be listed as follows:

1. Delay time
2. Rise time
3. Settling time
4. Peak time
5. Maximum Overshoot

the definitions and analytical deduction of the properties are given in figures (8.10, 8.11, 8.12 and 8.13).

## Second-Order Systems

- Delay time: ( $t_d$ ) time required for response to reach half the final value for the first time
- Rise time: ( $t_r$ ) time required for response to rise from 10% to 90% (or 0% to 100%)
  - Wouldn't want to use 0% to 100% definition for overdamped systems

Second-Order Systems

## Second-Order Systems

- Settling time: ( $t_s$ ) time required for response to reach and stay within 2% of final value
- Approximated from the exponential decay, calculated value is often conservative

$$y(t) = 1 - e^{-\sigma t} (\cos \omega_d t + \frac{\sigma}{\omega_d} \sin \omega_d t), t \geq 0$$

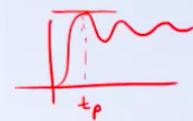
since  $e^{-4} \approx 0.02$ ,  $t_s = 4\tau \approx \frac{4}{\sigma}$

$\sigma t \approx 4$

Figure 8.11: Settling time seen in PT-2 systems

Settling time is approximated by roughly determining the time required to settle down the state dynamics by 2% at the steady state. The sinusoid inside the brackets will produce oscillations all the time unless they are completely damped out as  $t \rightarrow \infty$ . However by controlling the exponential value to reach 2%, the output from the sinusoids can be controlled within 2%. Therefore, from the analogy that  $e^{-4} \approx 0.02$ , the settling time  $t_s$  is approximated to  $4/\sigma$ .

## Second-Order Systems



$$y(t) = 1 - e^{-\sigma t} \left( \cos \omega_d t + \frac{\sigma}{\omega_d} \sin \omega_d t \right), t \geq 0$$

- Peak time: ( $t_p$ ) time required for response to reach first peak of the overshoot

$$\begin{aligned} \dot{y}(t) &= \sigma e^{-\sigma t} \left( \cos \cancel{\omega_d} t + \frac{\sigma}{\omega_d} \sin \omega_d t \right) - e^{-\sigma t} \left( -\omega_d \sin \omega_d t + \sigma \cos \cancel{\omega_d} t \right) \\ &= \underbrace{e^{-\sigma t} \left( \frac{\sigma^2}{\omega_d} + \omega_d \right)}_{\text{const} = 0} \sin \omega_d t = 0 \end{aligned}$$

- $\sin \omega_d t = 0$  for first time when  $\omega_d t = \pi \Rightarrow$

$$t_p = \frac{\pi}{\omega_d}$$

Figure 8.12: Peak time seen in PT-2 systems

## Second-Order Systems



$$y(t) = 1 - e^{-\sigma t} \left( \cos \omega_d t + \frac{\sigma}{\omega_d} \sin \omega_d t \right), t \geq 0$$

- Maximum overshoot: ( $M_p$ ) maximum peak value measured from ss value (often as %)

$$\begin{aligned} \text{at } t = t_p, \quad & y\left(\frac{\pi}{\omega_d}\right) = 1 + M_p \\ &= 1 - e^{-\sigma \frac{\pi}{\omega_d}} \left( \cos \varphi_d \frac{\pi}{\omega_d} + \frac{\sigma}{\omega_d} \sin \varphi_d \frac{\pi}{\omega_d} \right) \\ \Rightarrow 1 + M_p &= 1 + e^{-\sigma \frac{\pi}{\omega_d}} = 1 + e^{-\zeta \omega_n \frac{\pi}{\sqrt{1-\zeta^2}}} \\ \Rightarrow M_p &= e^{-\zeta \pi / \sqrt{1-\zeta^2}} \end{aligned}$$

Second-Order Systems

Figure 8.13: Maximum overshoot seen in PT-2 systems

## Nomenclature

### 8.4.4 Poles determine the system behavior

Consider figure 8.14, from the poles placed in the complex plane the following details can be extracted. The distance of the poles on the real axis is given by the real part of the roots and the distance of the poles on the imaginary axis is given by the Im part of the poles. The radius of the poles from the origin is defined by the natural frequency of the system  $\omega_n$  and the angle between pole radius  $\omega_n$  and real axis is given by  $\beta$ .

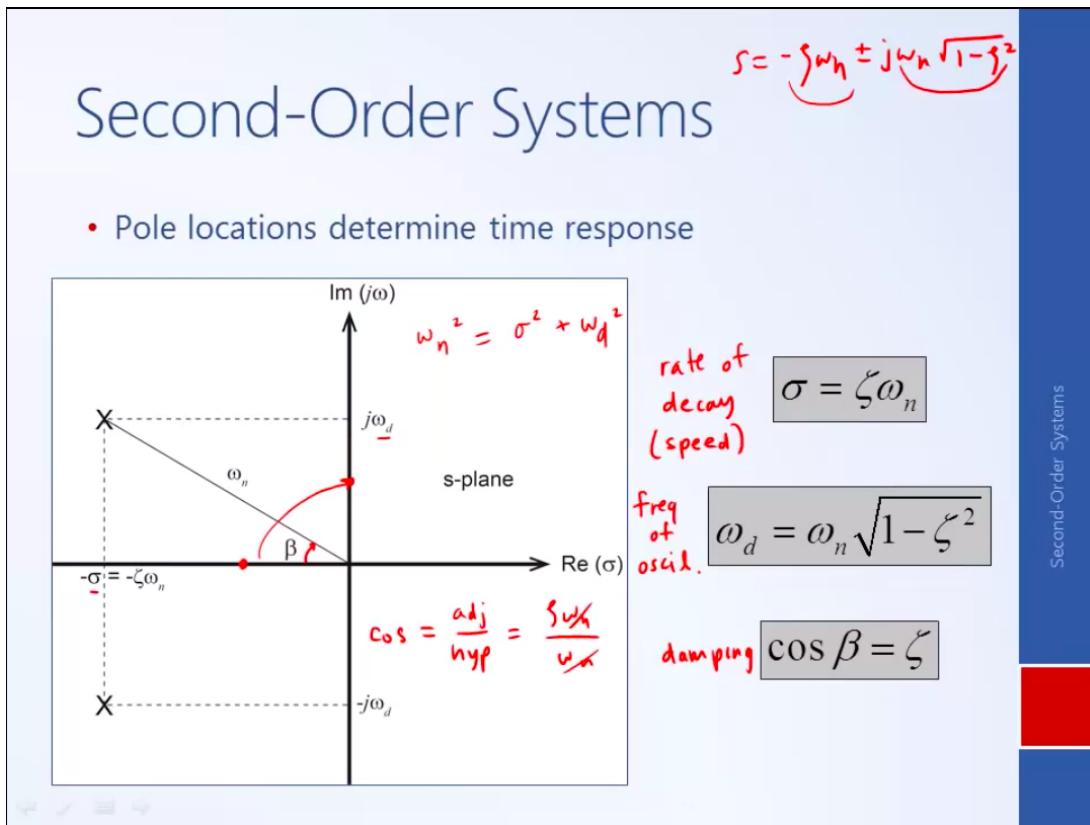


Figure 8.14: Poles determine the system behavior

From the relationship  $\cos \beta = \zeta$  it can be said that the angle  $\beta$  determines the damping in the system such as when  $\cos \beta = \pi/2$  then damping is zero and the system oscillates freely. When  $\cos \beta = 0$ , the poles lie exactly on the real axis and also lie on one another (repeated poles), making the system critically damp. Therefore, as the angle  $\cos \beta$  increases towards  $\pi/2$ , the damping in the system decreases and vice-versa.

### 8.4.5 Solving / Analysis PT-2 using characteristic polynomial

The general form of PT-2 using a characteristic polynomial is given by equation (8.20) (the coefficients in various forms may vary but they don't change the solution or the behavior as they are constants and treated differently in different approaches). From

in the table 8.4.1, the roots of the polynomial can be used to define the system's damping and therefore its behavior. For the first analysis, consider case (1) in the table 8.4.1 for roots  $B^2 > 4C$  leading to definite negative roots which overcome the oscillations from the Im roots and therefore make the system over damped.

With the roots of the characteristic equation (8.20) given by  $\{s_1, s_2\}$ , the general solution of equation (8.20) can be expressed as the linear sum of the solution of each roots as given by:

$$y(t) = Ae^{s_1 t} + Be^{s_2 t} \quad (8.37)$$

where  $A$  and  $B$  are constants which are determined using initial conditions which is same as equation (8.22) just with different coefficients. Now equation (8.22) gives the general solution for a homogeneous PT-2 given by (8.20). For the sake of analysis, let's use a step function and perform a time-domain analysis of PT-2 systems. Now equation (8.19) can be written for a step input as:

$$a\ddot{y} + b\dot{y} + c = 1 \quad (8.38)$$

the 1 on RHS signifies the external step input, with initial conditions  $x(0) = \dot{x}(0) = 0$ . Therefore now the general solution to equation (8.38) is given by:

$$y(t) = Ae^{s_1 t} + Be^{s_2 t} + \frac{1}{c} \quad (8.39)$$

$A$  and  $B$  can be solved using initial condition's using  $y(t)$  and  $\dot{y}(t)$  leading to:

$$\begin{aligned} \left(\frac{s_2}{s_1} - 1\right)B &= \frac{1}{c} \\ A &= \frac{-s_2}{s_1}B \end{aligned}$$

therefore, equation (8.39) becomes:

$$y = \frac{1}{c} \left[ 1 + \left( \frac{1}{s_1 + s_2} (s_1 e^{s_2 t} - s_2 e^{s_1 t}) \right) \right] \quad (8.40)$$

$c$  can be determined at the steady-state such that at steady state  $\ddot{y} = \dot{y} = 0$  which leads to  $c = 1$  in equation (8.19).

#### 8.4.6 Numerical solution for PT-2 systems using Matlab

For linear systems, Matlab has many inbuilt functions that can be used directly to determine the various system properties given in section 8.4.3.1. It is only necessary to remember these properties rather than be able to solve them analytically. Control system toolbox in Matlab can be used to conveniently determine all these properties on a computer rather than solving with pen and paper as well plot results to see these properties graphically for understanding.

Using the command `linearSystemAnalyzer('step',G)`, a figure shows in Matlab all the properties of a linear PT-2 system which are evaluated numerically.

*Nomenclature*

# 9 Introduction to Control

In chapters System modeling and Analysis, systems were modeled and analyzed by their time-response behavior by using a step function. Once the system behavior is observed and its properties defined as described in section 8.4.3.1, sometimes most of the system properties such as its response may not be the desired response that one is looking for. In such cases system behavioral properties can be modified using a *control*.

## 9.1 Control Architectures

### 9.1.1 Open-loop control / Feedforward control

Simple to design in controls perspective given predefined disturbances and plant dynamics. The open loop control architecture can be visualized using block diagrams as shown in figure 9.1.

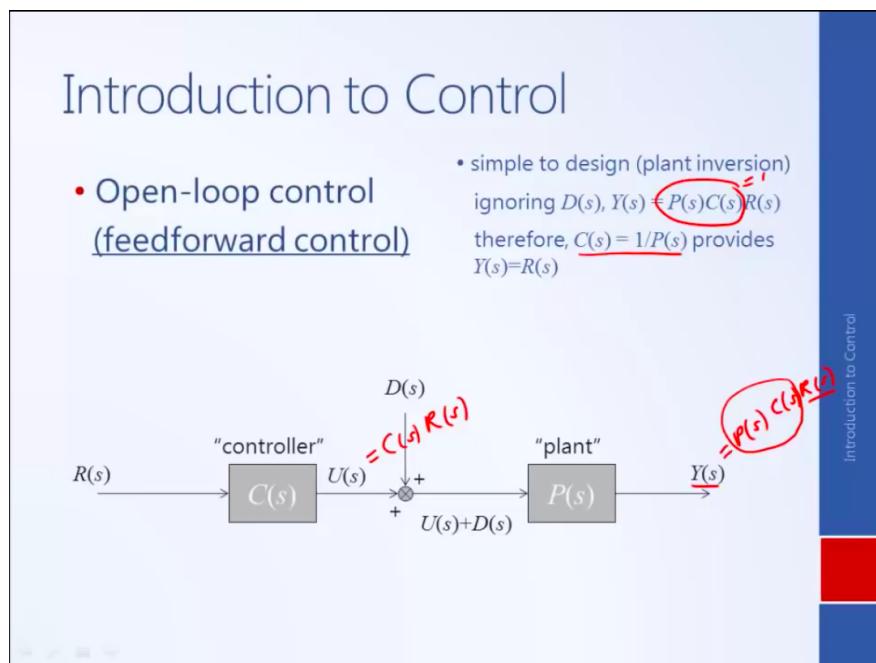


Figure 9.1: Open Loop Control Architecture

## Nomenclature

The control in the open loop architecture works as an inverted plant which can be explained as follows. The idea is to have output signal  $Y(s)$  follow the reference signal  $R(s)$ . If the disturbance signal  $D(s)$  is ignored the output signal of the entire system can be expressed as  $Y(s) = P(s)C(s)R(s)$ . If the condition that  $Y(s) = R(s)$  to be met, it follows that  $P(s)C(s) = 1$ . If  $P(s)C(s) = 1$  should be met then it follows that  $C(s) := 1/P(s)$ . Therefore, the control in an open-loop is just the plant inverted. This is the most simple form of control that can be applied just by inverting the plant dynamics and it requires no additional components such as sensors.

Obviously this type of design is not robust mainly because of the follows two reasons:

- Model errors: It is not sometimes possible to capture the dynamics of the system perfectly or the parameters of the system would change over time or there could also be some after market changed in the system that were not previously accounted for.
- The disturbances to the system cannot be pre-defined accurately as this is some unknown quantity that is being influenced from the environment.

Open-loop architecture is used in the most simplified cases where the control will not be affected from external disturbances such as in washing machines where the system is isolated from the environment and the control can do its work without any external influences. Or they are also used in conjunction with feedback in complex systems as the tracking error with open-loop can be essentially eliminated to zero.

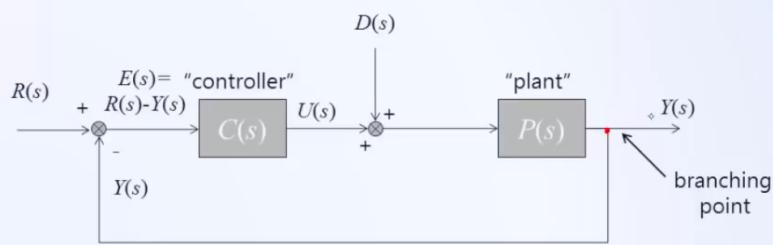
### 9.1.2 Closed Loop Architecture / Feedback control

Feedback control architecture will make the control system robust by using a feedback of the system outputs via a sensor. Feedback controller uses the differences between output and the reference signal to calculate the control action required to reduce the tracking error. In short feedback control works by providing a control action with respect to the error signal generated. The architecture is shown in figure 9.2

## Introduction to Control

- **Closed-loop control**  
(feedback control)

- more robust
- more complicated/expensive
- can cause a stable system to become unstable
- response can be slower



Introduction to Control

Figure 9.2: Closed Loop Control Architecture

One drawback of closed loop control is that it could induce instability into the system. The controller and the plant themselves may be stable on their own with negative real roots in the transfer function. But when the loop is closed the system on its whole could become unstable.

### Use of feedback: Dynamics of individual components need not be modeled

With a feedback unlike in open-loop control, the knowledge of the individual components of the system need not be known. For example consider system shown in figure 9.3, where the position of a spindle is controlled using an electric motor. It can be seen

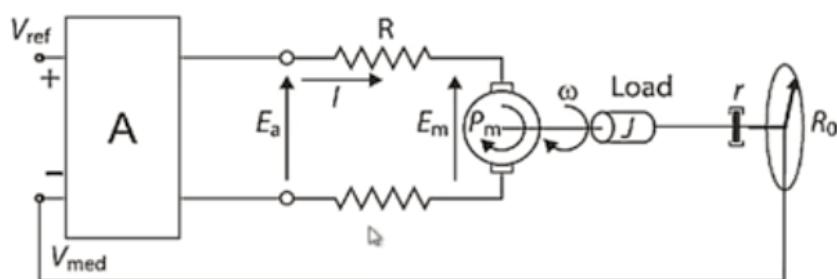


Figure 9.3: Use of Feedback

from figure 9.3, that the control on position is implemented only using the voltage to the motor as input. The individual dynamics of components such as functioning of

## Nomenclature

the motor and the load inertia themselves may not even be modeled into the system. For example using a step input the response of the system can be modeled as a whole (either as PT-1 or PT-2 or the combination of both).

### Use of feedback: Make system robust and disturbance rejection

Making a system robust is in essence successfully reject disturbance and overshadow the plant and sensor properties while controlling. For example consider a complex system as shown in figure 9.4, in order to determine what parameters the output of this system depends, a output can be determined using transfer function of the whole system:

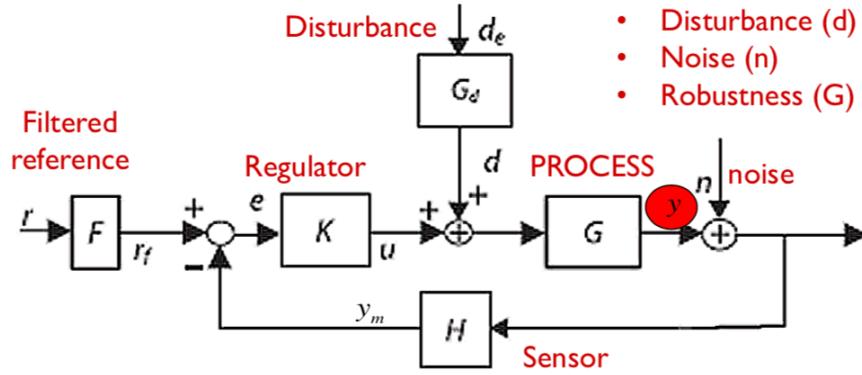


Figure 9.4: Use of Feedback: Making the system robust

$$Y(s) = F \frac{GK}{1 + GKH} r + G_d \frac{G}{1 + GKH} d_e + \frac{GKH}{1 + GKH} n \quad (9.1)$$

when the controller gain \$K\$ is made significantly large, in the above equation, the following terms would take the following values:

$$\begin{aligned} \frac{GK}{1 + GKH} &= 1 \\ \frac{G}{1 + GKH} &= 0 \\ \frac{GKH}{1 + GKH} &= 1 \end{aligned}$$

therefore, the system output equation reduces to:

$$Y(s) = Fr + n \quad (9.2)$$

From equation (9.2), it can be seen that when \$K\$ is significantly large, the response of the system becomes mostly dependent on reference TF \$F\$ with signal \$r\$ and noise signal \$n\$. The dynamics of plant \$G\$, sensor \$H\$ is removed from the equation, therefore the change in the dynamics of the plant or the sensor will have almost no impact on the control system itself. Further the equation (9.2) does not contain the term of disturbance \$G\_e\$, therefore, the effect of disturbance is also rejected by the use of feedback.

## 9.2 Block diagram reduction

Basically the stability or the response of the system depends on the poles. In order to be able to determine the poles of the entire system each individual blocks of the system should be reduced to form a single component system so that the transfer function and in-turn the poles of the system can be calculated easily.

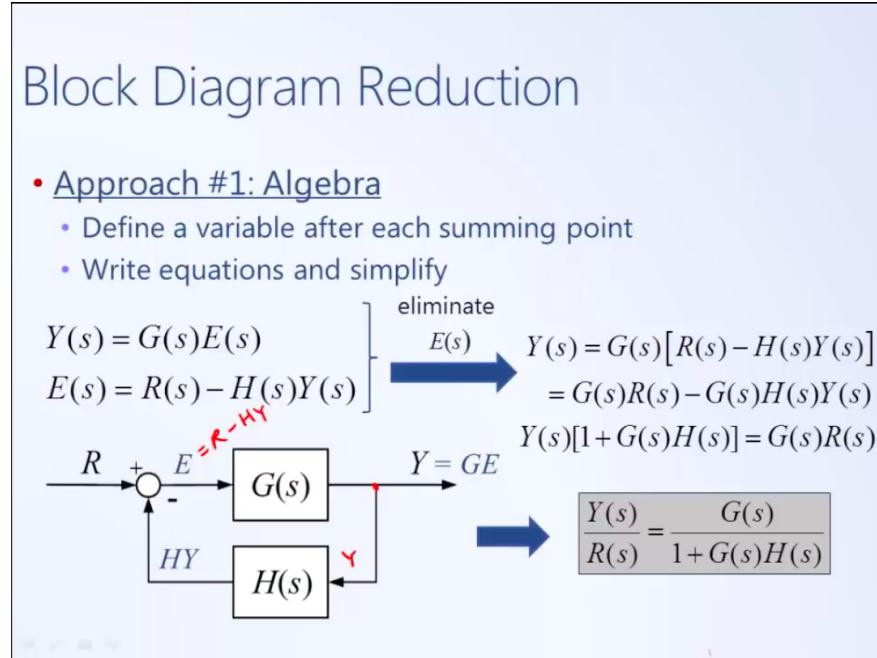


Figure 9.5: Block diagram reduction: Algebraic

## Nomenclature

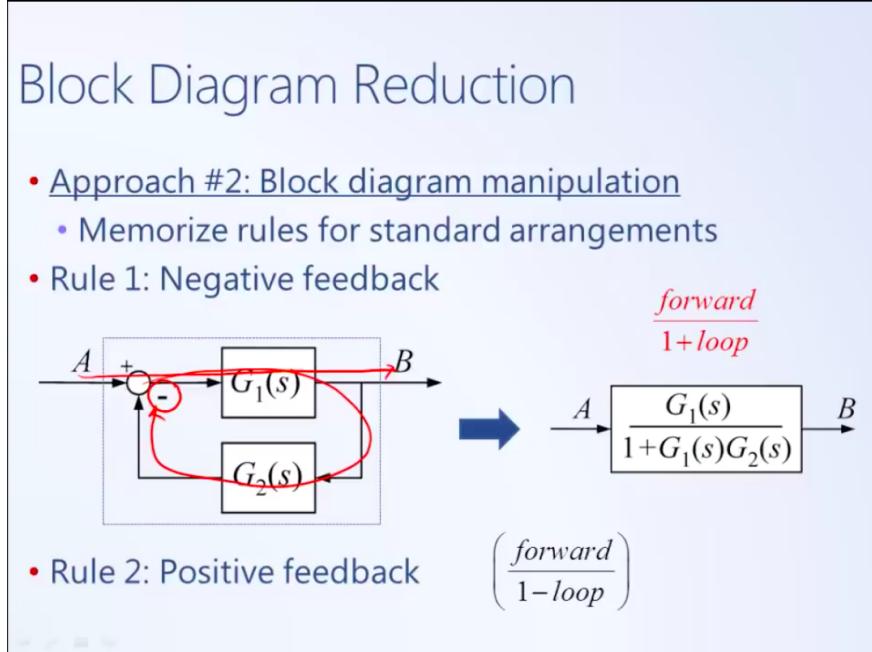


Figure 9.6: Block diagram reduction: Memorizing rules

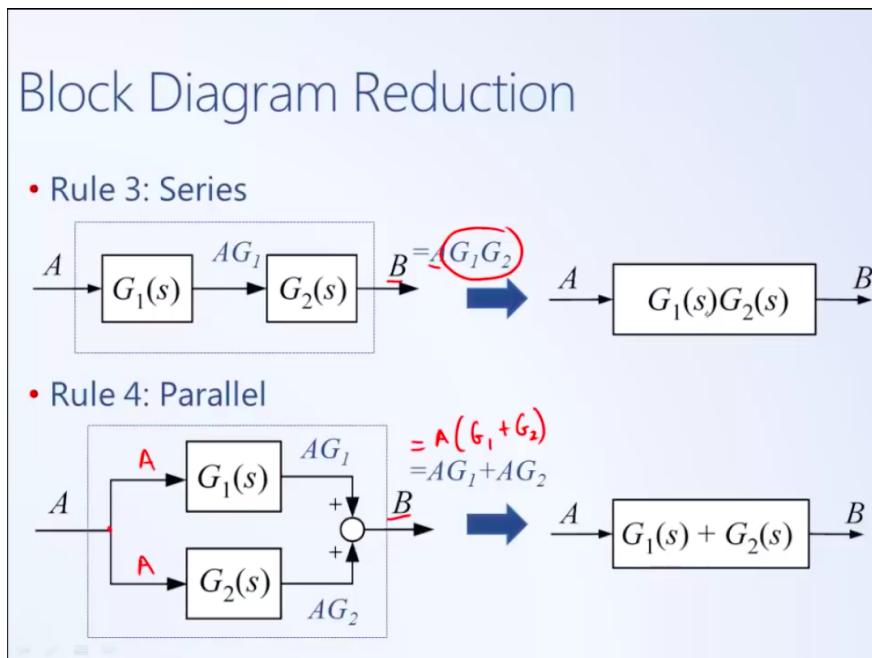


Figure 9.7: Block diagram reduction: Series and Parallel

## 9.3 Linearity and Principle of Superposition

For linear systems with multiple inputs, the effect of each of the individual signals into the system can be studied separately by ignoring all other signals into the system block diagram as in linear systems the response of the system for all the inputs can later be added up to determine the overall system response as explained in a very clear detail graphically in [Hil13b, t.41.19].

## 9.4 Cruise control example and drawbacks of proportional regulator

As described in section 9.1.2 of closed loop architecture, once the error has been determined the controller can use this error to provide some control signal to the actuator which in-turn will reduce the tracking error. Once the error has been determined there are many ways in control theory to use this error to determine the control signal in-turn. One easiest way to calculate the control signal is by using a proportional gain regulator.

### 9.4.1 Open-loop system model

The cruise control system controls the traction force  $F$  (or traction force is the control signal) and the tracking is required on the output velocity of the vehicle  $v$ . A longitudinal dynamic model of a vehicle is choose for simplicity and also noting that for a cruise control which controls only the output velocity  $v$  of the vehicle, such simplified longitudinal model is sufficient. From longitudinal model it can be said that the accelerating force that is required to maintain desired  $v$  can be found from Newtons second law as:

$$F - F_{res} = m\dot{v} \quad (9.3)$$

where  $F$  is control signal generated to accelerate the vehicle,  $F_{res}$  are the longitudinal resistance the vehicle experiences due to air drag, rolling resistance and gradient resistance,  $\dot{v}$  is the vehicle acceleration and  $m$  is the system parameter. Additionally,  $F_{res}$  can be approximated as function of  $v$  and a linear function as:

$$F_{res} = bv \quad (9.4)$$

where  $b$  is a system parameter that influences  $F_{res}$ . Equation 9.3 becomes:

$$F - bv = m\dot{v} \quad (9.5)$$

$$m\dot{v} + bv = F \quad (9.6)$$

Equation (9.6) forms the first order ODE of the longitudinal vehicle model. And the Laplace transform of equation (9.6) is:

$$msV(s) + bV(s) = F(s) \quad (9.7)$$

## Nomenclature

writing in terms of transfer function:

$$\frac{V(s)}{F(s)} = \frac{1}{ms + b} \quad (9.8)$$

writing equation (9.8) in canonical form:

$$\frac{V(s)}{F(s)} = \frac{1/b}{\frac{m}{b}s + 1} \quad (9.9)$$

where  $k_{dc} = 1/b$  which will also give the steady-state response and  $\tau = m/b$  which gives the settling time (or time constant) of the plant and  $\frac{V(s)}{F(s)}$  gives the transfer function of the plant.

### 9.4.2 Closed Loop Model

Now a controller with proportional gain  $K$  is added in series to the plant which actuates with reference signal  $R(s)$  and  $V(s)$  will be the output of the entire system as shown in figure 9.8. Since this is closed loop transfer function, the block diagram reduction is shown in section 9.2 in figure 9.6 will lead to the following system TF:

$$\frac{V(s)}{R(s)} = \frac{\frac{K}{ms+b}}{1 + \frac{K}{ms+b}} \quad (9.10)$$

simplifying the above equation leads to:

$$\frac{V(s)}{R(s)} = \frac{\frac{K}{b+K}}{\frac{m}{b+K}s + 1} \quad (9.11)$$

therefore,  $k_{dc}$  and time constant of the system are  $\frac{K}{b+K}$  and  $\frac{m}{b+K}$  respectively. If  $K$  is chosen sufficiently larger then it will dominate the terms  $k_{dc}$  and  $\tau$  such that  $k_{dc} \rightarrow 1$  and  $\tau \rightarrow 0$  which is the most ideal case for the controller leading to perfect tracking with fast response. Additionally, the pole is at  $s = \frac{-(b+K)}{m}$  and as  $K$  increases the roots become more negative and this will lead to much faster response times of the controller. With no external disturbances and noise in the system the control does the tracking perfectly as shown in figure 9.9. However, a feedback loop also induces proportional noises into the system proportional to  $K$  which has to be studied separately.

### 9.4.3 Effects of Noise with Proportional Regulator

With noise as the input signal (ignoring  $R(s)$ ) the closed loop TF of the system is expressed as:

$$\frac{V(s)}{N(s)} = \frac{-\frac{K}{ms+b}}{1 + \frac{K}{ms+b}} = \frac{-K}{ms + b + K} \quad (9.12)$$

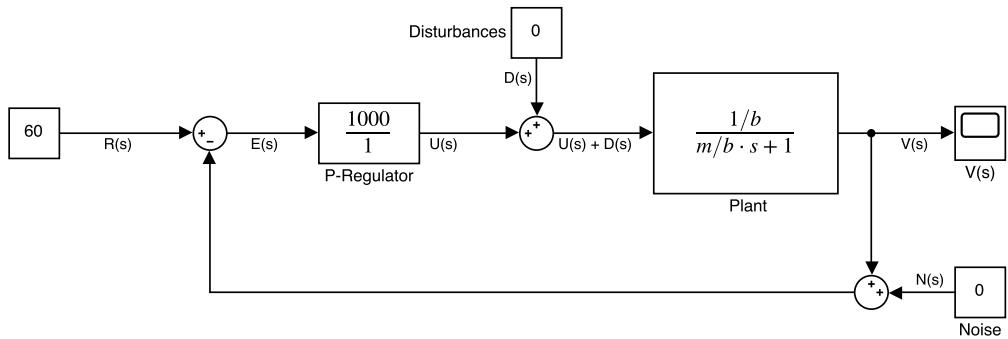


Figure 9.8: Block diagram of Cruise controller with P-type regulator

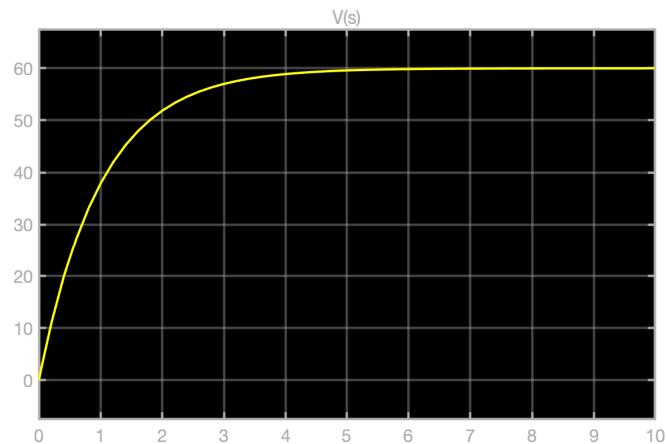


Figure 9.9: Tracking achieved by Hight Propotional controller gain

From equation 9.12m it can be seen that when  $K$  is made sufficiently large, the TF will increase towards a value of  $-1$ . A TF of  $-1$  will make the system output  $V(s)$  react to noise in the opposite to the desired value and it amplifies as  $K$  increases. Therefore, a larger  $K$  will amplify the effect of noises into the system and such cases the system will fail to track as shown in figure 9.10.

Additionally, there is also something called actuator saturation that comes from larger  $K$  ie., limit of engine dynamics, friction between tire-road interface among others which would lead to higher fuel consumption among others. Additionally a larger  $K$  will induce system oscillations. Therefore, it is necessary to avoid larger  $K$  due to noise disturbance amplifications, actuator saturation and system oscillations (due to overshoot).

*Nomenclature*

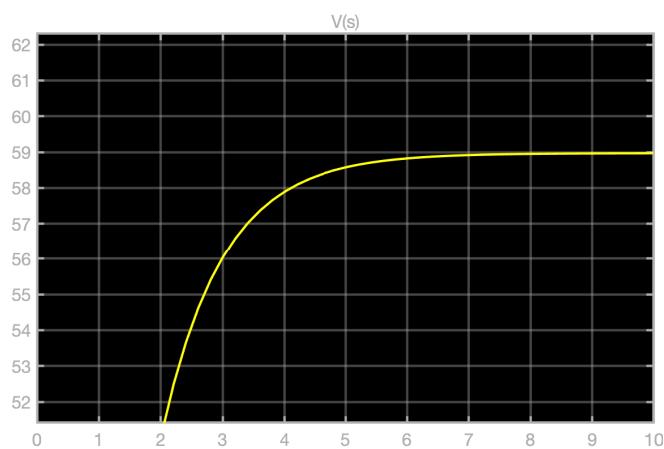


Figure 9.10: Tracking failed by high Proportional controller gain with noise

# 10 Eigen Value Analysis

Eigenvalues tells about how the given system matrix  $\mathbf{A}$  will respond or evolve in the particular direction of Eigenvectors. Consider a system matrix given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (10.1)$$

the Eigenvalues and the Eigenvectors of the system are given by

$$\lambda_1 = 1$$

$$\lambda_2 = -1$$

$$v_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

If the Eigenvectors are plotted on a simple scale along the horizontal and the vertical axis as shown in the figure 10.1

## Nomenclature

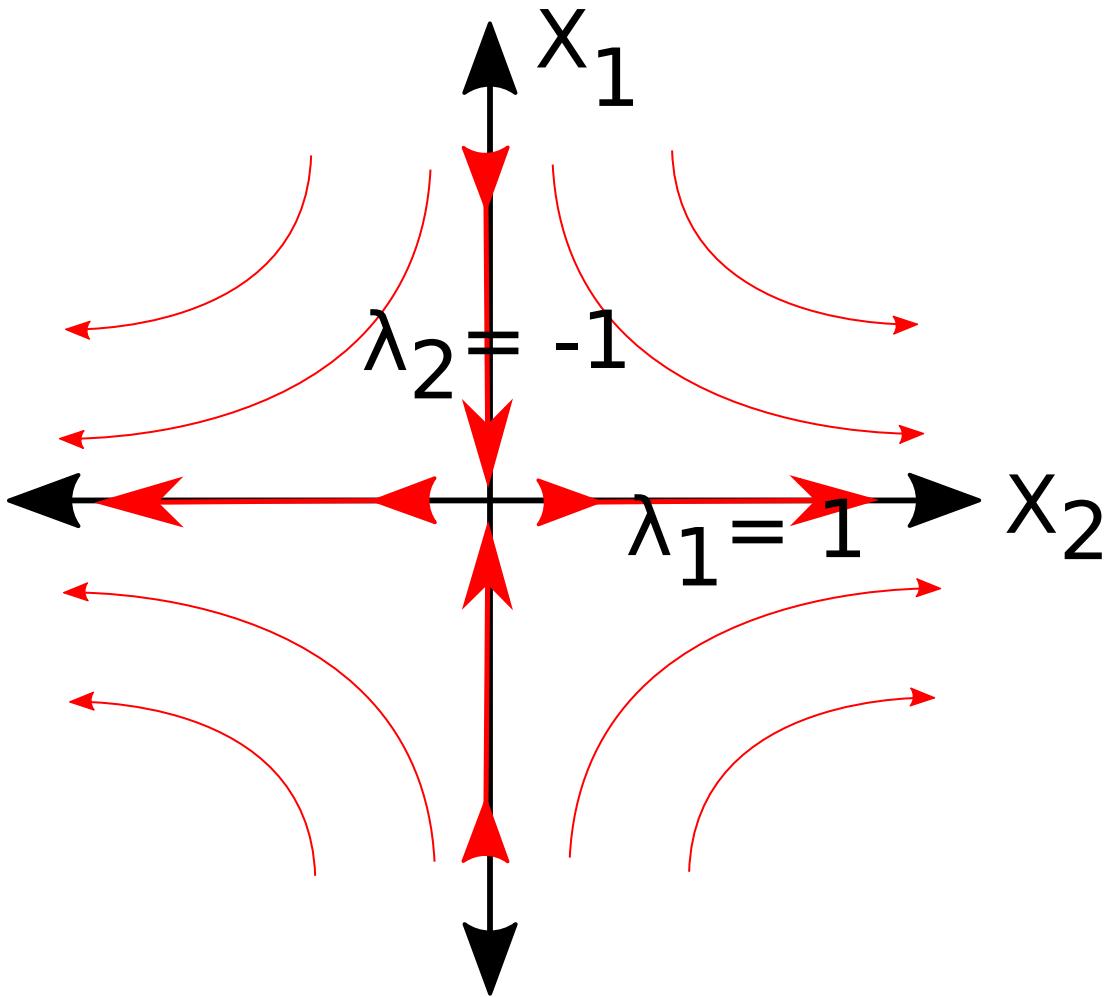


Figure 10.1: Plot of Eigenvectors and the matrix A evolution along Eigenvectors

**Note:** The axis  $x_1$  and  $x_2$  in figure 10.1 are interchanged !!!

suppose the system starts on the axis  $x_2$  where  $\lambda_2 = -1$ , the system dynamics will evolve towards the origin therefore, attenuating the dynamics and the system will be stable. However, if the system starts on axis  $x_1$  where  $\lambda_1 = 1$ , then the states will evolve towards positive side of the Eigenvector which will amplify the dynamics and therefore, the system will be unstable. At any other points, due  $\lambda_2 = -1$ , the dynamics will move downwards and due to  $\lambda_1 = 1$ , the states will evolve towards amplification, therefore the system as a whole will not be stable. One particular case, when the system starts exactly along the Eigenvector where  $\lambda = -1$  is in the case of an inverted pendulum which is perfectly aligned vertically towards the line of gravity and will remain in that position unless an external disturbance is applied.

## 10.1 How Eigen Value Analysis Works

Using the definition of Eigen value,

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v} \quad (10.2)$$

which says that  $\mathbf{A}$  is a matrix which acts as a scaling factor to the vectors  $\mathbf{v}$ , such that the scaling factor is  $\lambda$ . Now the scaling factor  $\lambda$  is acting on  $\mathbf{v}$  in order to scale this vector. How much this vector gets scaled and in which direction depends on the factor  $\lambda$ . Here  $\mathbf{v}$  is called the Eigen vector and  $\lambda$  is called the Eigen value.  
Therefore, there is now a relationship where it can be found the evolution of vector of  $\mathbf{v}$  depending on the factor  $\lambda$ .

*Nomenclature*

# 11 Classical Control System Design

## 11.1 Control Specifications

Generally engineering specifications are provided as a requirement for control systems. Specifications such as engine torque requirements with engine speeds, vehicle speed control requirements (acceleration in specified amount of time), passenger comfort etc.,. The task of the requirements engineer is to derive the control specifications in terms of engineering specifications mentioned in the documents.

Control specifications are in-terms of transient control specifications as per the properties of dynamics such as those mentioned in section 8.4.3.1 and Steady-state requirements specifications. In case of transient controller specifications, specifications are given in terms of step response such as: Settling time, peak time, Percentage overshoot etc.,. The steady-state specification are given in-terms of percentage steady-state error to be less than certain amount (ex:  $e_{ss} < 0.02$ ). The steady-state requirements can be derived using Final Value theorem.

## 11.2 Control Expectations (Performance expectations)

A control system should always satisfy the following general performance exceptions, also in the same order of importance (where the first is the most important): **General performance expectations:**

1. Stability
2. Tracking
3. Disturbance Rejection
4. Noise Filtering
5. Robustness
6. Optimality in Control

**Specific performance expectations:**

1. Object avoidance
2. Goal oriented control

## Nomenclature

### 3. Other system specific requirements

The first and the foremost important goal to be achieved by the control is to make the system stable. In some cases, using feedback may make the system unstable (see relative stability in frequency response analysis) or in any case the first objective of control is to make the system stable. Only when stability is achieved successfully, the other control objectives can be met. Stability is defined based on pole positions (BIBO and Asymptotic stability) in case of time-domain analysis and relative stability (gain and phase margins) on how far the system is from going unstable in case of frequency-domain analysis.

The next important control objective is to track the reference successfully. The control system which is stable but which can not track the reference is useless. Disturbance rejection is something that comes with choosing correct control actions (such as PID) and proper control gain  $K$ . Filtering the noise should also be an integral part of the control system.

Once the basic needs on stability, tracking, disturbance rejection and noise filtering are achieved, then it is better to further design the control system to be robust enough to continue delivering the required performance even when the dynamics of the system changes (for example when one of the rotor blades are damaged in a quad-copter). One of the ways to make the control robust is shown in figure 9.4, where by increasing the control gain  $K$  large enough, the system response is not affected either by the dynamics of plant or the sensor.

Further complex control objectives can be met when the system has satisfied all the above objectives successfully and there is a possibility of optimizing the way the control is implemented. A optimal control strategy can be implemented, generally this is an advance concept and a subject of its own. Additionally, there are certain system specific control objectives to be met such object detection and others which needs to be separately included in terms of behavior based control modeling approach.

## 11.3 Pole Placement

Pole placement is based on the theory of dynamics which states that the poles of the system determine various system dynamics and hence they can be used to manipulate the system dynamics to the required dynamic state by placing the poles at the required places in the complex plane. This in theory can be done just by manipulating the controller gain  $[K_P \ K_I \ K_D]$  so that the poles of the closed loop system meet the requirements.

Consider a closed loop PI - regulator as shown in the figure 11.1 used for automatic cruise control.

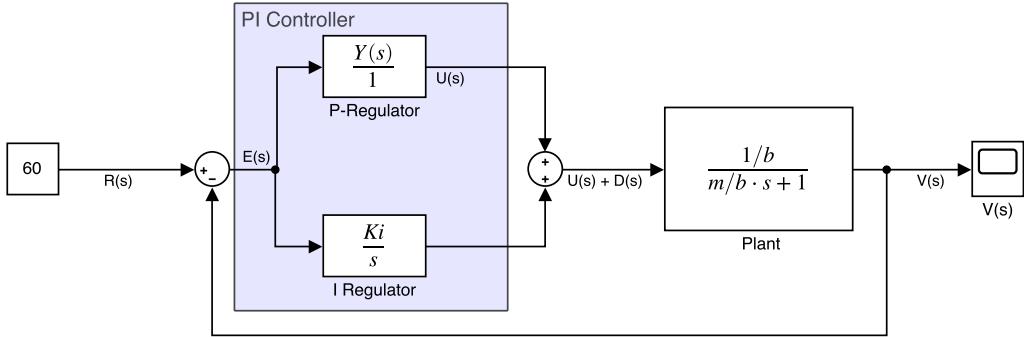


Figure 11.1: PI Controller for Automatic Cruise Control

Analytically, the TF of the entire system can be found using block diagram reduction techniques as described in section 9.2. The TF of the system can be expressed as:

$$\frac{V(s)}{R(s)} = \frac{\frac{K_P s + K_I}{s(ms+b)}}{1 + \frac{K_P s + K_I}{s(ms+b)}} \quad (11.1)$$

Simplifying above equation gives:

$$\frac{V(s)}{R(s)} = \frac{K_P s + K_I}{ms^2 + (b + K_P)s + K_I} \quad (11.2)$$

The system has two parameters  $\{K_P, K_I\}$  to vary so as to change the pole positions (2 DOF). In order to analyses the system behavior w.r.t the pole placements the poles of the system transfer function can be determined first. From equation (11.2), the denominator can be written in the canonical form so as to determine the poles:

$$s^2 + \frac{(b + K_P)}{m}s + \frac{1}{m}K_I \quad (11.3)$$

further poles can be determined by matching terms with the canonical form:

$$s^2 + 2\zeta\omega_n s + \omega_n^2$$

$$\zeta\omega_n = \sigma = \frac{(b + K_P)}{2m} \quad (11.4)$$

$$\omega_n = \sqrt{\frac{K_I}{m}} \quad (11.5)$$

where  $\sigma$  is the real part of the root of canonical second order system given by  $\{s_{1,2}\} = -\zeta\omega_n \pm j\sqrt{1 - \zeta^2}$  also written in short from as  $\{s_{1,2}\} = \sigma \pm j\omega_d$ . As described

## Nomenclature

in section 8.4.4, the location of the poles determine the time response of the system. From equation 11.5, it can be seen that the integral gain directly influences the systems natural frequency therefore, adjusting  $K_I$  will also adjust the location of poles on the real axis and in-turn adjust the systems  $\omega_n$  accordingly. Similarly, from equation (11.4), proportional gain  $K_P$  directly influences system damping, therefore, adjusting  $K_P$  will adjust the location of poles on the real axis and in-turn adjust the systems damping.

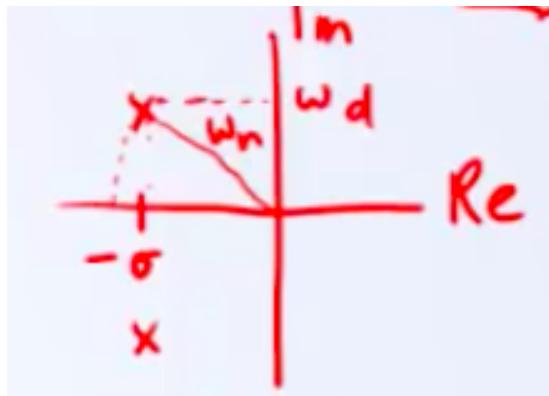


Figure 11.2: Pole location of the roots given in equations (11.4) and (11.5)

**Case I: Increasing  $K_P$  holding  $K_I$  constant:** Consider figure 11.2, if  $K_P$  is increased maintaining  $K_I$  constant, then  $\sigma$  increases and  $\omega_n$  remains constant, as  $\omega_n$  is the function of  $K_I$ . Therefore, the radius of the poles is fixed and by increasing  $\sigma$ , the poles will move towards real axis radially. As  $\sigma$  increases, the time response properties associated with  $\sigma$  also increases such as settling time  $t_s = 4/\sigma$  and peak time  $t_p = \pi/\omega_d$  the peak overshoot however decrease exponentially as given by  $M_p = e^{-\zeta\pi/\sqrt{1-\zeta^2}}$ .

$M_p$ .

**Case II: Increasing  $K_I$  holding  $K_P$  constant:** Increasing  $K_I$  holding  $K_P$  constant,  $\sigma$  will be a constant as given in equation (11.4). A constant  $\sigma$  with increasing  $K_I$  will increase the radius of the pole and the poles increase vertically maintaining the position on the real axis fixed. This increases the angle  $\cos\beta$  which will reduce damping and increase oscillations. Also a constant  $\sigma$  will keep the settling time  $t_s = 4/\sigma$  constant, even though there is increased oscillations in the system due to increasing  $\cos\beta$ .

However, in case of damped system, the increasing oscillations are seen via damped frequency of the system which also increases due to increase in  $\omega_n$  as given by the Im part of the pole  $\omega_d = \omega_n \sqrt{1 - \zeta^2}$ . Further the peak time decreases as  $t_p = \pi/\omega_d$ . The increasing oscillation can further be re-enforced by seeing that  $\zeta$  decreases as  $\omega_n$  increases  $\zeta = \sigma/\omega_n$ . As mentioned previously that  $M_p$  is inversely proportional (inverse exponential) to  $\zeta$ , in this case naturally  $M_p$  reduces as  $\zeta$  decreases.

It is important to note that these relationships do not hold perfectly for the given system as it is not in the canonical second order form due to the presence of zero. However, understanding the pole placements gives a fairly good starting point for designing the

controller gains also for this non-canonical system. Once the starting points for the pole locations been found out then a area can be chosen in the complex plane for the poles such that system would meet requirements such as faster settling time (in which case the poles could be placed from far left in the negative real axis).

## 11.4 Calculating steady-state error

Consider a system shown in figure 11.3, the objective is to find the steady-state error using final value theorem. The steady-state error can be found by writing the TF of error signal  $E(s)$  as output with reference signal  $R(s)$  as the input to the system. The forward loop in such a case is between  $R(s)$  and  $E(s)$ , essentially there are no blocks between them therefore, the TF is just 1, in case of feedback loop, the loop starts from  $R(s)$  then  $E(s)$  which continues to the two blocks of the systems and closing back at the summation blocks. Therefore, the TF can be written as:

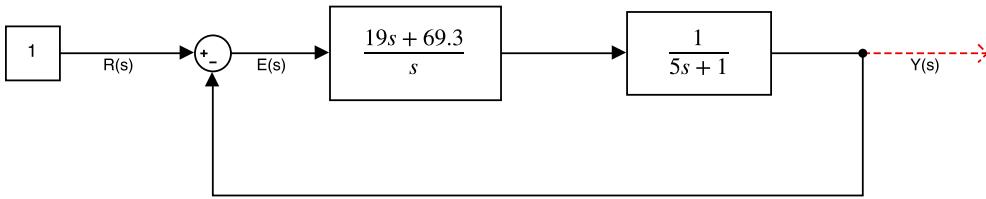


Figure 11.3: Example calculating SS error

$$G(s) = \frac{E(s)}{R(s)} = \frac{\text{forward}}{1 + \text{loop}} = \frac{1}{1 + \frac{19s+69.3}{s(5s+1)}} \quad (11.6)$$

Simplifying the above equation:

$$\frac{E(s)}{R(s)} = \frac{s(5s+1)}{s(5s+1) + 19s + 69.3} \quad (11.7)$$

From the above equation, the error can be written as:

$$E(s) = \frac{s(5s+1)}{s(5s+1) + 19s + 69.3} R(s) \quad (11.8)$$

Using Final Value Theorem:

$$e_{ss} = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} s \frac{s(5s+1)}{s(5s+1) + 19s + 69.3} \frac{1}{s^2} \approx 0.014 \quad (11.9)$$

in the above example the reference signal is assumed to be a unit ramp signal the Laplace transform of which is  $1/s^2$ .

## 11.5 PID Control

A feedback control works taking error  $E(s)$  as input and control signal  $U(s)$  as output. Therefore the TF of the PID control is expressed as:

$$C(s) = \frac{U(s)}{E(s)} = K_P + K_D s + K_I \frac{1}{s} \quad (11.10)$$

further the control action with reference to error can be visualized by writing equation (11.10):

$$U(s) = K_P E(s) + K_D s E(s) + K_I \frac{1}{s} E(s) \quad (11.11)$$

where  $K_P$  is a constant multiplied (scaling) to the error  $E(s)$ , in Laplace transform multiplication with Laplace term  $s$  makes the differentiation, therefore  $K_D s E(s)$  scales and differentiates the error signal  $E(s)$ . Finally  $K_I \frac{1}{s} E(s)$  scales and integrates the error signal  $E(s)$ . PID are most commonly used because they are easy to understand and implement as differentiation and integration are quite a common knowledge. Not only the control action is based on the current error using proportionality, but also the history of errors is taken into consideration using integral as well as the anticipation of future error is taken into consideration using a differential. Therefore, a PID though very simple is quite a sophisticated control as it considers the past trends, current trends and anticipates the future trends of the error.

In the time domain for the error which is the function of time, the PID control output can be expressed as:

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{d}{dt} e(t) \quad (11.12)$$

Similar to equation (11.10), equation (11.12) models the control action as follows. The current error is generated due to the difference between reference signal  $r(t)$  and the output signal  $y(t)$  for which the proportional control amplifies the control action by a predefined factor  $K_P$ . The time history of the accumulated errors due to loss of tracking either in the transient stage or the steady-state of the system response is added up together using an integral action which is then scaled to a predefined factor of  $K_I$ . Finally, the trend of the error is captured by taking time derivative of error  $e(t)$  which in essence captures the slope of the trending curve which is then scaled by a predefined factor of  $K_D$ . The control action is summed up from all of these three individual control action in a PID control.

### 11.5.1 Understanding PID using a canonical system

A plant of PT-2 canonical type is chosen in order to understand each terms of the PID control's effect on the system as a whole. Consider a control system with the control  $C(s)$  and plane  $P(s)$  as shown in figure 11.4. Now by choosing controller gains and

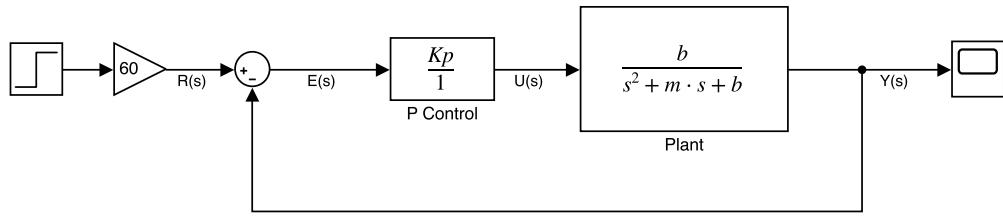


Figure 11.4: PT-2 canonical system with P-Control

simulating the system in Matlab it can be seen that the output  $Y(s)$  never tracks the reference value and there are always some steady-state error as show in figure 11.5. This can be explained by writing down the system analytically and solving for the poles and steady-state error using final value theorem. The system given in figure 11.4 can be written down as:

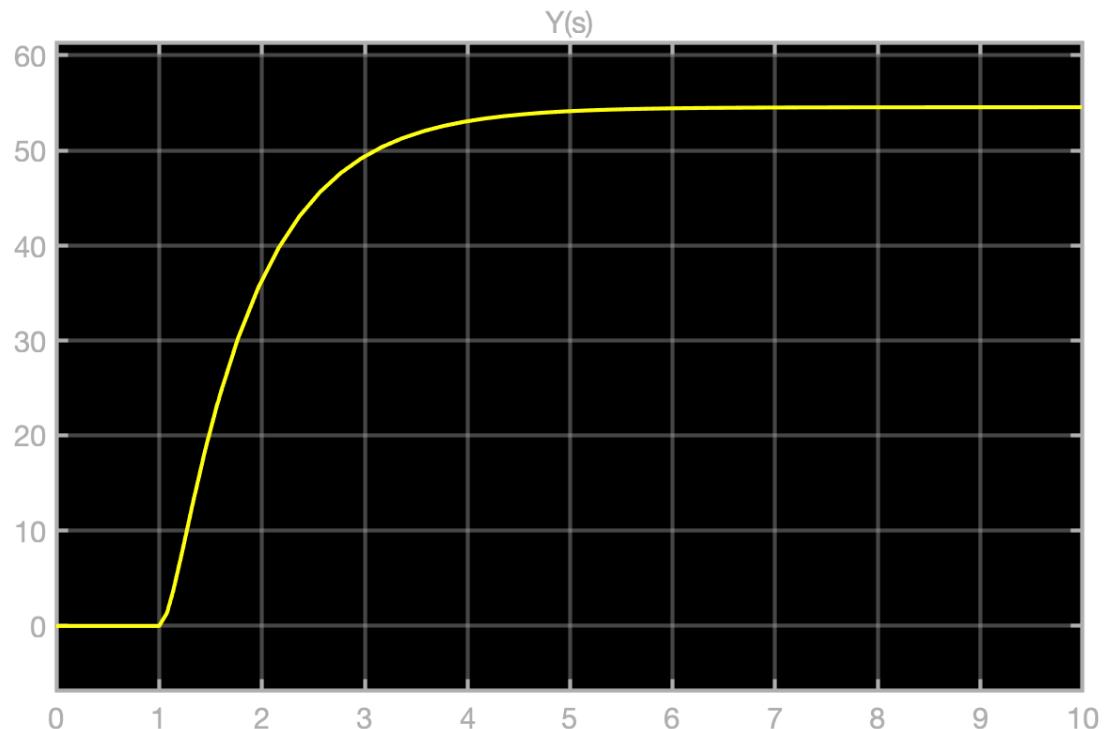


Figure 11.5: Steady-state error with P-Control

$$\frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{bC(s)}{s^2 + as + b(1 + C(s))} \quad (11.13)$$

## Nomenclature

### 11.5.1.1 P Control

Consider figure 11.4, with P control, the control signal is given by  $U(s) = K_P E(s)$  ( $C(s) = K_P$ ), therefore, equation of the system (11.13), now becomes:

$$\frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} = \frac{bK_P}{s^2 + as + b(1 + C(s))} \quad (11.14)$$

The terms in the above equation can be matched with the terms of the PT-2 canonical system in order to derive the system properties such as:

$$\omega_n^2 = b(1 + K_P) \quad (11.15)$$

$$2\zeta\omega_n = a \quad (11.16)$$

further  $\omega_n$  and  $\zeta$  are simplified as:

$$\omega_n = \sqrt{b(1 + K_P)} \quad (11.17)$$

$$\zeta = \frac{a}{\omega_n} = \frac{a}{2\sqrt{b(1 + K_P)}} \quad (11.18)$$

It can be seen from equations (11.17) and (11.18) that as  $K_P$  the control gain is increased,  $\omega_n$  increases and  $\zeta$  decreased. Increasing  $\omega_n$  and decreasing  $\zeta$  says that as  $K_P$  is increasing, the system is becoming more stiffer. Also noting that for this system the poles are shifted using only  $K_P$  therefore, it is a 1 DOF system.

#### Steady-state performance

For a unit step reference:

$$y_{ss} = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} s \frac{C(s)P(s)}{1 + C(s)P(s)} = s \frac{bK_P}{s^2 + as + b(1 + K_P)} \frac{1}{s} \quad (11.19)$$

the above equation simplifies to:

$$y_{ss} = \lim_{s \rightarrow 0} sY(s) = \frac{K_P}{1 + K_P} \neq 1 \quad (11.20)$$

From equation (11.20), it can be seen that the ratio is never equal to 1 for a unit step reference. Therefore, mathematically it is impossible to track the system perfectly using P-Control. Further from equation (11.20), it can be seen that as  $K_P$  is made large enough, the denominator of equation (11.20) is dominated by  $K_P$ , thereby leading to a ratio which is very close to 1. This is one of the ways to reduce the SS-error in the system as shown in figure 11.6.

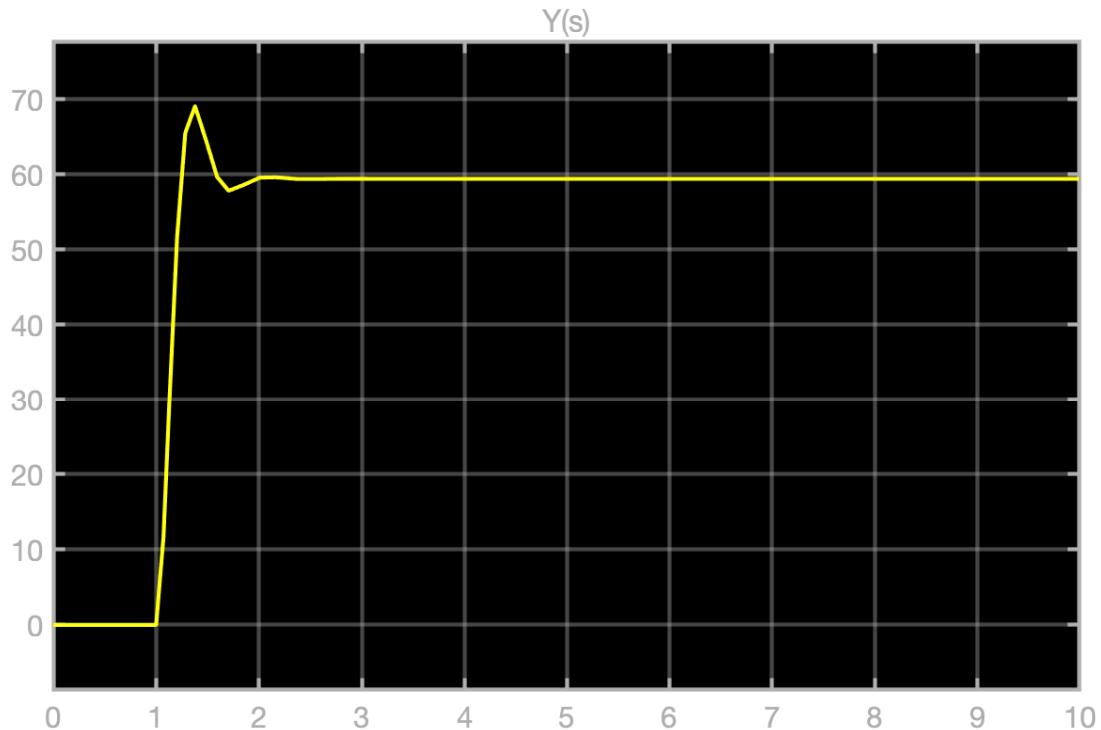


Figure 11.6: Reduced Steady-state error with P-Control

However, it can also be seen in figure 11.6, increasing  $K_P$  also increases overshoot and induces oscillations into the system which is evident from equations (11.17) and (11.18) which indicated that higher  $K_P$  would make the system stiffer and increase oscillations and overshoot (due to decreased damping).

**Tips:** Increasing the proportional gain  $K_P$  tends to scale the error (control action) to the same level. As in a P control pushing harder will accelerate (system will react more quickly for higher  $K_P$ ) the systems response more quickly it will also cause an overshoot. Increasing  $K_P$  can reduce the ss-error but not eliminate it completely as proved using final value theorem on ss-error.

### 11.5.1.2 D Control

In a proportional only control, when a proportionally large control action needs to be given to the system due to rapidly increasing dynamics, the proportional control will have to wait until the value of the error itself increases. However, in a differential control the control looks at the slope of the error which in-turn reflects the rapid change in the dynamics. The slope will have started first in any curve before the curve reaches to its maximum value following that slope. Therefore, in a differential control as the control is given proportional to the slope, a massive control action can be given to the

## Nomenclature

system even before it reaches the maximum change in state (value) as soon as the slope changes. This in essence is like predicting the future trends of the system dynamics and taking a necessary control action before hand (anticipating the future).

### Properties of derivative control

Using figure 11.8 and ignoring noise, the TF of the derivative control is expressed as  $C(s) = K_D s$ , therefore, the TF of the plant with noise as input can be written as:

$$\frac{Y(s)}{R(s)} = \frac{\frac{K_D s b}{s^2 + ms + b}}{1 + \frac{K_D s b}{s^2 + ms + b}} = \frac{K_D s b}{s^2 + (m + K_D b)s + b} \quad (11.21)$$

Comparing equation (11.21) with PT-2 canonical system, the system properties can be defined as:

$$\omega_n = \sqrt{b} \quad (11.22)$$

$$2\zeta\omega_n = m + K_D b \implies \zeta = \frac{m + K_D b}{2\sqrt{b}} \quad (11.23)$$

As can be seen from equation (11.22), the derivative control does not effect the natural frequency and in-turn the damped frequency of the system. The derivative control only effects the damping of the system as given in equation (11.23). It is a natural case of dynamic system that the derivative control can only effect the damping. As is the case with PT-2 systems, the damping is only present as a function of the first order derivative (velocity term) in the dynamic equation. The poles can be found using equations (11.22) and (11.23):

$$\sigma = \zeta\omega_n = \frac{m + K_D b}{2\sqrt{b}} \quad (11.24)$$

$$\omega_n = \omega_n \sqrt{b} \quad (11.25)$$

From the above equations it can be seen that changing  $K_D$  will effect only the real part of the pole as the imaginary part of the pole can be held constant by maintaining  $K_D$  a constant. So changing  $K_D$  will change the real part of the pole radially as the radius of the pole  $\omega_n$  will always be a constant in this case. By increasing  $K_D$ , the angle  $\cos(\beta)$  will move more closely towards the real axis un-till the poles become repetitive in case of critical damping. As a derivative control increasing damping it cuts down the overshoots in the system as the property  $M_p$  which determines the percentage overshoots is reduced exponentially due to the increase in  $\zeta$ . Further, the settling time  $t_s = 4/\sigma$  also reduces as  $\sigma$  increases. Finally, the overshoot peak time  $t_p = \pi/\omega_n$  will remain unaffected by  $K_D$  as a derivative control does not change the location of pole radially.

The action of the derivative control reducing the overshoot of the system can be seen using the system as shown in figure 11.4. It can be seen in figure 11.6, that a high  $K_P$  increases the overshoot of the system which can be controlled by adding a derivative control such that  $C(s) = K_P + K_D s$ . Figure 11.7 (produced with Matlab) shown

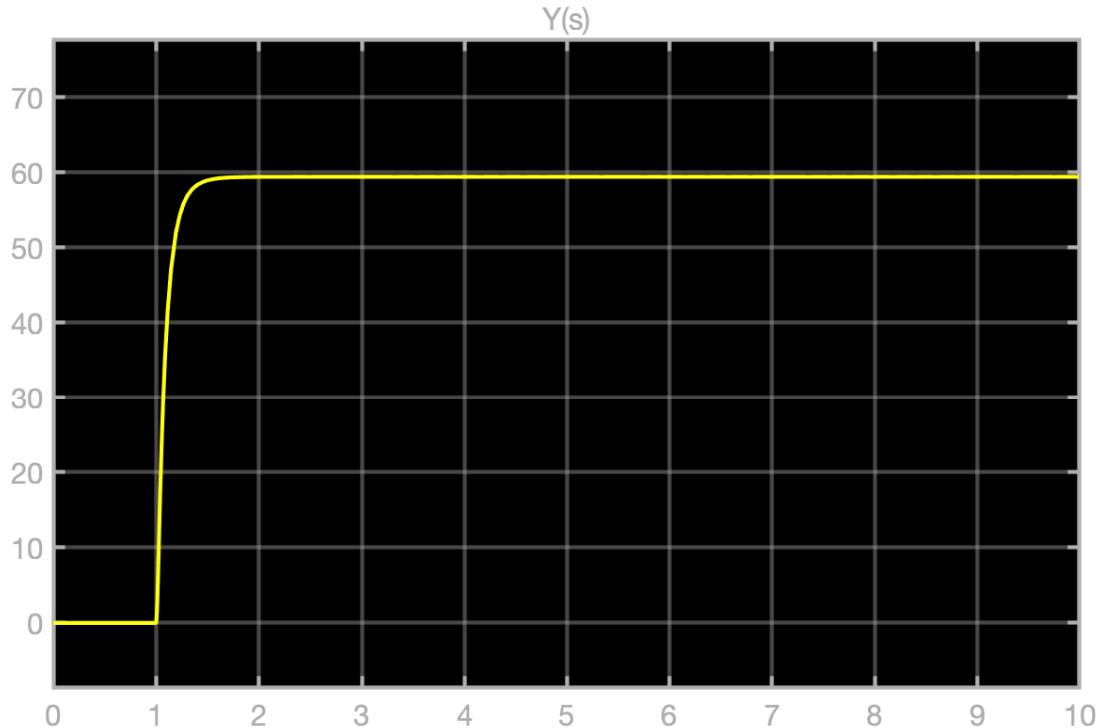


Figure 11.7: Overshoot elimination using derivative control

the action of the derivative control reducing the overshoot in the system which acts a stabilizer as it damps the oscillations in the system. While designing the controller for an industry application, it is a common practice to first start with the proportional control and then use derivative control for reducing overshoot in the system or increasing damping or similar properties.

**Tip:** A P control will only increase if the error also increases, as a D control will anticipate the trend of the error in advance by checking the slope, this anticipation will tend to add damping into the system, reducing the overshoot.

#### Noise amplification in derivative control

A D-control however cannot be used all by itself and is not generally used because of its nature to amplify the noise into the system. In order to study the effects of the noise in the system lets consider analytical solution of the system with noise as shown in figure 11.8. writing the TF when noise  $N(s)$  is input to the system:

$$\frac{Y(s)}{N(s)} = \frac{\frac{-K_D s \cdot b}{s^2 + ms + b}}{1 + \frac{K_D s \cdot b}{s^2 + ms + b}} \quad (11.26)$$

simplifying above equation:

$$\frac{Y(s)}{N(s)} = \frac{-K_D s \cdot b}{s^2 + (m + K_D b)s + b} \quad (11.27)$$

## Nomenclature

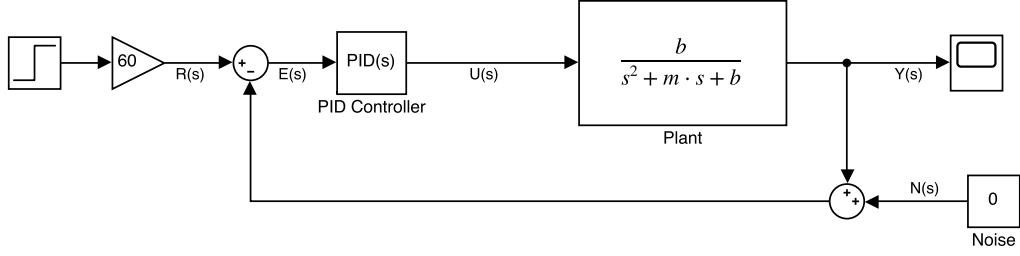


Figure 11.8: Performance of Derivative Control with noise

the steady-state error of the output can be found using FVT and a unit step function as the noise signal  $N(s)$ :

$$ss_y = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} s \frac{-K_D s \cdot b}{s^2 + (m + K_D b)s + b} \frac{1}{s} = \infty \quad (11.28)$$

Using Matlab the following simulation was produced with a derivative control 11.9. A derivative control always produces a steady-state error which increased to  $\infty$  for any kind of input noise signal. A derivative control always has this effect of amplifying noise into the system. It is for this reason that in most of the industries a derivative control is avoided as far as possible.

### 11.5.1.3 I Control

An integral control adds new features to the system which are very helpful for the control in terms of achieving a perfect steady-state response. An integral control generally increases the order of the system, so a PT-1 system with an integral control will become a PT-2 system and so on. By doing so it is intuitively easier to understand that the higher order system are generally more oscillating and have longer settling times compared to their lower counter-parts. That what the integral control does to the system, it generally increases the oscillations in the system and makes the system slower. It is generally difficult to analyses systems with integral control, an attempt is made to come up with a most simplest example. Consider a PT-1 system of the canonical form:

$$P(s) = \frac{1}{\tau s + 1} \quad (11.29)$$

with an integral control  $C(s) = K_I/s$ , the forward TF for the system can be expressed as:

$$G(s)_f = \frac{K_I}{s(\tau s + 1)} = \frac{K_I}{s^2 \tau + s} \quad (11.30)$$

the TF of the system (for a negative feedback) can be expressed as:

$$G(s) = \frac{Y(s)}{R(s)} = \frac{\text{forward}}{1 + \text{loop}} = \frac{\frac{K_I}{s^2 \tau + s}}{1 + \frac{K_I}{s^2 \tau + s}} \quad (11.31)$$

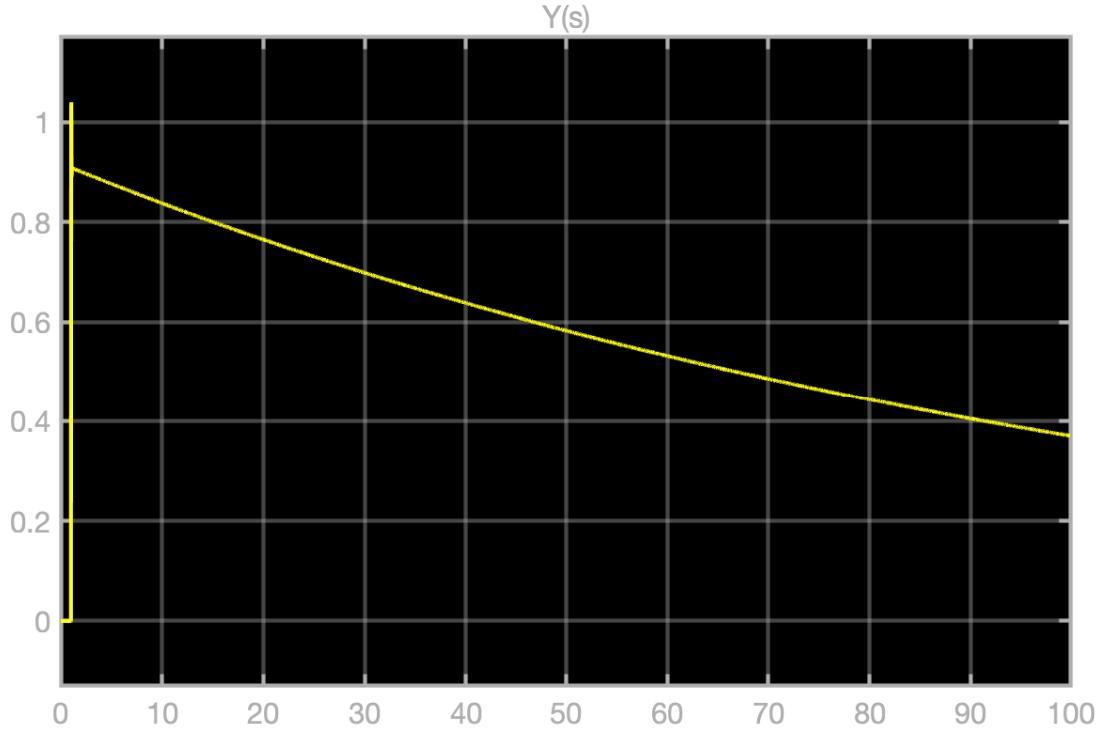


Figure 11.9: Noise amplification of derivative control into the system

simplifying the above equation:

$$G(s) = \frac{K_I}{\tau s^2 + s + K_I} = \frac{K_I}{s^2 + s/\tau + K_I/\tau} \quad (11.32)$$

It can be seen just by analyzing equations (11.29) which was the previous order of the system without integral control and the current system equation (11.32) which is now a PT-2 system. Generally a PT-1 system has only one pole at  $s = -1/\tau$  whereas a PT-2 system has two poles of complex conjugate in nature which can be found in case of equation (11.32) by comparing with the canonical PT-2 system:

$$\omega_n = \sqrt{\frac{K_I}{\tau}} \quad (11.33)$$

$$2\zeta\omega_n = \sigma \implies 2\zeta\omega_n = 1/\tau \quad (11.34)$$

there are two observations that can be made at this point, the real part of the pole  $\sigma = 1/\tau$  is a constant and will remain fixed in the complex plane. The Im part can be placed in the complex plane by changing  $K_I$  which will increase the vertical placement of the pole along the Im axis. This increases the angle  $\cos \beta$  which will increase the oscillations in the system. Therefore, it can be proved with this simple example that an

## Nomenclature

integral control will induce oscillations into the system firstly by increasing the poles in the system and also by placing the poles away by increasing angle  $\cos\beta$ .

The effect of increasing settling time increase by the integral control is complicated at this time. But according to the error build-up concept explained in [Hil13c, t.1:01:00], if the error in the system changes its sign, the control action proportional to it does not change its sign. This would lead to undoing the control action in the next cycle when the error changes the direction again. Therefore, an integral control makes the system slower to settle down compared to proportional and derivative controls.

**Tip:** The I control reduces the ss-error by increasing the control action which results due to the persistent built-up of the ss-error. This build-up gets summed up by the integrator and so will be the control action as a result of it. The I control makes the system slow because if the error signal changes the sign then due to that addition of the overall built-up error, the change in error sign will partly cancel out the previously built-up error in summation. Therefore, it will take some time for the I control to "unwind" from changing signs in error signals.

### 11.5.2 PD Control

A PD control for a second order canonical system can be used to determine the control system behavior. The PD control is expressed as  $C(s) = K_P + K_D s$ , for the system shown in figure 11.8 ignoring noise, the TF of the system with PD control can be expressed as:

$$G(s) = \frac{Y(s)}{R(s)} = \frac{b(K_P + K_D s)}{s^2 + (m + bK_D)s + b(1 + K_P)} \quad (11.35)$$

From equation 11.35, it can be seen that the derivative control term again works in conjunction with the damping term of the PT-2 canonical system. Therefore, in summary to applying individual proportional and integral controls is that  $K_P$  will change the  $\omega_n$  of the system independently {1 DOF} and  $K_D$  will only change the damping  $\zeta$  of the system independently {1 DOF}. Which leads to the fact that with a PD control now it is possible to combine the poles of the system independently for either increasing stiffness  $\omega_n$  or increase the damping  $\zeta$  independently {2 DOF} (can place the poles anywhere). Also with the PD control the steady-state error of the system remains the same as that of the ss error when using only P control as can be seen from equation (11.35). That also means that the steady-state error amplification due to noise seen by using only derivative control (equation (11.28)) will be eliminated and replaced with equation (11.20).

#### 11.5.2.1 PI Control

An integral control acts on the basis of the time history of the accumulated errors and therefore gets larger as errors get accumulated over time. An advantage of using a PI control in combination is that the integral action will lead the system to a perfect

tracking of the steady-state response. Consider a PI control for the system shown in figure 11.4 with control action

$$C(s) = K_P + K_I/s$$

, the TF of such system can be expressed as:

$$\frac{Y(s)}{R(s)} = \frac{b(K_P + K_I/s)}{s^2 + as + b(1 + K_P + K_I/s)} = \frac{b(K_P s + K_I)}{s^3 + as^2 + b(1 + K_P)s + bK_I} \quad (11.36)$$

the steady-state performance of the system can be found using FVT:

$$y_{ss} = \lim_{s \rightarrow 0} sY(s) = \lim_{s \rightarrow 0} s \frac{b(K_P s + K_I)}{s^3 + as^2 + b(1 + K_P)s + bK_I} \frac{1}{s} = \frac{bK_I}{bK_I} = 1 \quad (11.37)$$

therefore, by using an integral control in conjunction with proportional control makes the ss-error of the system zero. This is true even if  $K_I$  is small (however, a bigger  $K_I$  will be much faster).

### 11.5.2.2 Summarizing PID control actions

The summary of each term can be best visualized using the following table:

Control	Action	Pros	Cons
Proportional	Proportional action w.r.t error	Fast reponse	Increases overshoot
Differential	Anticipates error using slope	Adds damping (reduces overshoot and $t_s$ )	Amplifies noise into the system
Integral	Control builds up as erro builds up	Reduce steady-state error	Induces oscillations and makes the system slow to settle down

**Note:** This intuition is developed using only PT-2 canonical system, the results may vary for other systems. The most important thing is to check for the location od poles and zeros in the overall system after adding the controls. However, for higher order non-canonical system, model order reduction techniques could be used to reduce the system to PT-1 or PT-2. Only trial and erro method is used to determine the effect of PID control action on non-canonical systems.

## 11.6 System Type - Steady-state error

As can be seen from section 11.5.2.1, using an integral action in the cases of the canonical system type always leads to a perfect steady-state tracking, essentially bringing the

## Nomenclature

steady-state error of the system to zero. This integral action does not need to be only in the controller level but integral actions are sometimes inherent in the plant itself. Basically, it is the pole placement of the integral action. The integral control always places the poles at the origin as can be seen with the transfer function of the an I-Control  $C(s) = K_I/s$ , such a control with a PT-1 plant, the OL TF of the system:

$$G(s) = \frac{K_I}{s(\tau s + 1)} \quad (11.38)$$

it can be seen that this system has two poles, one at  $s_1 = -1/\tau$  and other at  $s = 0$  due to the pole made by the integral control. Therefore, an integral control will always make a new pole at origin when used in any kind of system. Using the above result certain system types can be formalized among control systems using the following definition.

**System Type:** The number of poles at the origin in the forward path of a *unity feedback* control system is called the system's type.

With poles either in the controller or in the plant itself. The following system types can now be defined:

**Type 0:** System with no poles at the origin

$$G(s) = \frac{s+1}{s+4} \quad (11.39)$$

**Type 1:** System with one pole at the origin

$$G(s) = \frac{s+1}{s(s+4)} \quad (11.40)$$

**Type 2:** System with two poles at the origin

$$G(s) = \frac{s+1}{s^2(s+4)} \quad (11.41)$$

**Type  $n$ :** System with one pole at the origin

$$G(s) = \frac{s+1}{s^n(s+4)} \quad (11.42)$$

A system type which includes the reference in its TF can track the signal perfectly. For ex: Type 1 will track input with unit step and Type 2 will track input with unit ramp perfectly and so on. The following table provides some conclusions using various system types with various input signal types.

	<b>Step</b> $r(t) = 1$ $R(s) = 1/s$	<b>input</b> 1	<b>Ramp</b> <b>put</b> $r(t) = t$ $R(s) = 1/s^2$	<b>in-</b> <b>put</b> $r(t) = \frac{1}{2}t^2$ $R(s) = 1/s^3$	<b>Acceleration</b> <b>in-</b> <b>put</b> $r(t) = \frac{1}{2}t^2$ $R(s) = 1/s^3$
<i>Type 0</i>	nonzero		$\infty$		$\infty$
<i>Type 1</i>	0		nonzero		$\infty$
<i>Type 2</i>	0		0		nonzero

Table 11.1: Table summarizing standard system types with standard input types

*Nomenclature*

# 12 Root Locus

Root locus method is another approach for control design apart from the algebraic approach used in pole placement method as described in section 11.3. A pole placement technique is a simple algebraic technique which is based on algebraic manipulation of control gains in order to place the poles of the system at the locations so as to achieve certain requested property of the control system. However, pole placement technique is a rather simple technique which can be used rather fairly with canonical systems as with such system the algebraic terms relating to the control gains are very much clear. For noncanonical higher order systems this process would be much difficult to see the effect gains on all poles. For such rather complex system another technique is used called as **Root Locus Method** for controller design.

**Root Locus** is a graphical tool that shows how the locations of the poles of a closed-loop TF move as a parameter (like a control gain) is varied as shown in figure 12.1. The

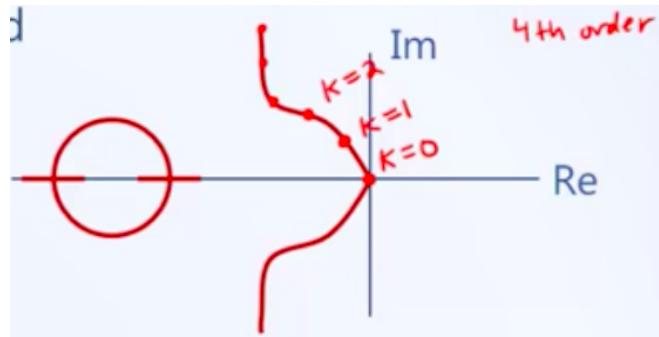


Figure 12.1: Root locus definition

above system is a 4<sup>th</sup> order system with 4 poles, therefore a complicated higher order system which would be very complex and time consuming with algebraic manipulation using pole placement method. Also with root locus it is possible to find the design control gains very quickly and it provides approximate results quickly (graphically).

## 12.1 Theory on Root Locus

The theory that describes the origin of root locus that is needed to draw the graphical form of the evolution of system poles with the increasing control gains  $K$ . This theory in itself is not very important to be known in terms of control design perspective, however, the rules that the theory lays down at the end are important to be remembered.

## Nomenclature

The root locus theory states that the evolution of the system poles with respect to control gains can be completely described only by using open-loop (OL) system poles and zeros. In order to do so two kinds of constraints are developed by establishing a relationship between the values of  $s$  and the control gain  $K$ . It should also be noted that this theory is nowadays only relevant in order to understand where the root locus of the control system poles originates and would only give a rough approximation in the paper, a better accurate numerical value of root locus can be directly found from Matlab which uses another way to find root locus numerically as described in a later section 12.1.4. Before developing in the theory it is important to define open-loop (OL), forward and close-loop (CL) transfer function of a system.

### 12.1.1 OL, forward and CL TF

Consider a system shown in figure 12.2, for which OL, forward and CL TF's of the system can be defined as follows: **Open-Loop TF:** The OL TF is the TF taken when

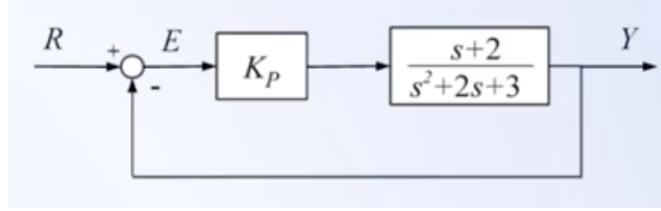


Figure 12.2: Root locus definition

the summation point is broken off and taking the TF by multiplying all the blocks in the loop just the way thinking that the summation point of the loop is like an output as shown below:

$$TF_{OL} = \frac{K_P(s+2)}{s^2 + 2s + 3} \quad (12.1)$$

**Forward TF:** The forward TF is taken considering only the forward path of the system, therefore, if the system contains sensors in the feedback loop they would be ignored as shown below (in this example there is no feedback sensor):

$$Forward_{TF} = \frac{K_P(s+2)}{s^2 + 2s + 3} \quad (12.2)$$

**Closed-Loop TF:** CL TF is defined using block reduction techniques that are used to calculate the TF of the entire system and the following formula is obtained for a negative CL system:

$$CL_{TF} = \frac{Forward_{TF}}{1 + TF_{OL}} \quad (12.3)$$

### 12.1.2 Defining the constraint equation for the root locus

Consider the CL TF for the system shown in figure 12.2:

$$G(s) = \frac{P(s)H(s)}{1 + P(s)H(s)} \quad (12.4)$$

where,

$$P(s)H(s) = \frac{K_P(s+2)}{s^2 + 2s + 3} \quad (12.5)$$

therefore, the denominator term of equation 12.4 from which the poles of the system can be determined can be written as:

$$1 + P(s)H(s) = 1 + K \frac{N(s)}{D(s)} \quad (12.6)$$

where  $K$  is taken as the general control gain of the system. Further using the above equation a relationship can be established as:

$$1 + K \frac{N(s)}{D(s)} \Rightarrow \frac{N(s)}{D(s)} = -\frac{1}{K} \quad (12.7)$$

It should be remembered that the relationship established using equation (12.7) was derived completely using the denominator of equation (12.4) so that only the description of the poles is obtained w.r.t to the control gain  $K$ . Equation (12.7) form the constraint equation that was previously described in the introduction.

The common gain of the system can be also be determined using the following way. Suppose a PID controller is used in series with a plant PT-1:

$$\begin{aligned} G(s) &= C(s)P(s) = K_P + \frac{K_I}{s} + K_D s \left( \frac{1}{\tau s + 1} \right) \\ G(s) &= \frac{K_P s + K_I + K_D s^2}{s} \left( \frac{1}{\tau s + 1} \right) \\ G(s) &= K \frac{s + K_I^* + K_D^* s^2}{s(\tau s + 1)} = K \frac{N(s)}{D(s)} \end{aligned}$$

where,  $K_I^* = K_I/K_P$  and  $K_D^* = K_D/K_P$ . This is just for an example ,the factor  $K$  could also have been chosen for  $K_D$ .

#### 12.1.2.1 Constraint 1

From equation (12.7), it should be seen that for the ratio on the  $RHS - 1/K$  is satisfied iff the value of  $s$  satisfies the ratio on the  $LHS = N(s)/D(s)$ . In other words, there are only a certain definite values of  $s$  for which the ratio  $-1/K$  is satisfied and also the vice-versa that the control gain  $K$  only satisfies the root locus for a certain values of  $s$ .

## Nomenclature

### 12.1.2.2 Constraint 2

If the constraint equation (12.7) is used to plot on the complex plane, the constraint ratio  $-1/K$  can be plotted as shown in figure 12.3. As the ratio is negative and lies

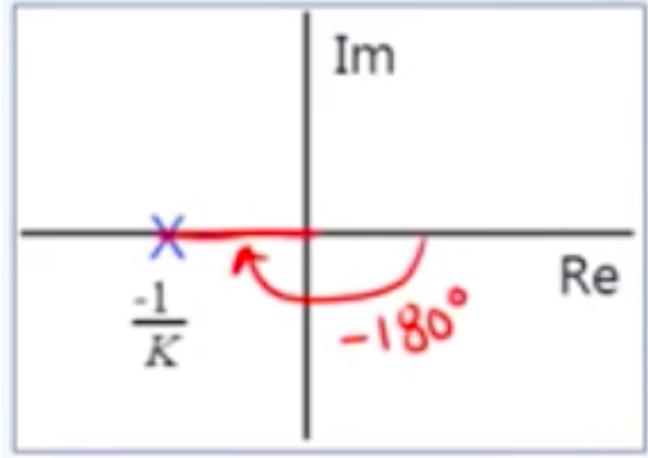


Figure 12.3: Root locus constraint ratio plot on the complex plane

on the real axis the position is plotted as shown at  $-1/K$  in figure 12.3. The position  $-1/K$  should have some magnitude and direction if it can be seen through polar coordinates just like the poles that have magnitude  $\omega_n$  and direction  $\cos \beta$  w.r.t negative real axis. Therefore, a new constraint relationship can be established now using such magnitude and angle constraints. In polar coordinates the angle is measured w.r.t the positive real axis as shown in figure 12.3, also it can be seen that these angles are repeating for every  $180^\circ$  whether it is approached from  $+\theta$  or  $-\theta$  from the positive real axis. Therefore, a new constraint can be formed using angle constraint on the ratio  $-1/K$  as given below:

$$\angle \frac{N(s)}{D(s)} = 180^\circ \pm 360^\circ n \quad (12.8)$$

where  $n = 0, 1, 2, 3, \dots$ . Further, the magnitude is quite straight-forward and it should be expressed as:

$$\left| \frac{N(s)}{D(s)} \right| = \frac{1}{K} \quad (12.9)$$

Further, using the magnitude constraints, the following conclusions can be drawn and the first of the few rules of root locus can be laid down. Generally, in a TF of the form  $\frac{N(s)}{D(s)}$ , the values of  $s$  for which the denominator goes to zero are called poles and similarly the values of  $s$  for numerator can be defined as zeros. Now we have a relationship of  $\frac{N(s)}{D(s)}$  which was originally taken from the denominator of equation 12.4 such that the poles from manipulated equation  $\frac{N(s)}{D(s)}$  alternate themselves between further values of poles and zeros as explained further.

In case  $K = 0$  in equation (12.9), the ratio  $\rightarrow \infty$ , therefore,  $\frac{N(s)}{D(s)} \rightarrow \infty$ . For this to be true  $D(s) \rightarrow 0$ . The values of  $s$  for which the denominator  $D(s) \rightarrow 0$  are called poles. If  $s$  has the value of poles, then it should have been the value without  $K$  into consideration. A pole value without  $K$  taken into consideration is also the OL pole where  $K = 0$ . Therefore, the root locus approaches the OL poles (see section 12.1.4 how this happens numerically). Similarly, for the condition that  $K \rightarrow \infty$ , the ratio  $\frac{N(s)}{D(s)} \rightarrow 0$ , for this condition  $N(s) \rightarrow 0$ . For this condition then the root locus approaches the OL zeros. So it can be seen that the root locus starts from OL poles and terminates at OL zeros (first rule of root locus). This also proves the facts that the root locus can be drawn using only the OL TF of the system, as was this mathematical (strangely weird) derivation of root locus began with.

Lastly, it should also be noted that when there are more poles than zeros, then this includes zeros (where  $\text{zeros} = \#\text{poles} - \#\text{zeros}$ ) that approach infinity (also called asymptotes).

### 12.1.3 Rules for drawing rot locus

The following rules are generally required to draw a root locus:

1. Locate the OL poles and zeros in the s-plane
2. Determine root locus on the real axis
3. Approximate the asymptotes of the root locus
4. Approximate the break-away and break-in points
5. Determine angles of departure and arrival
6. Find imaginary angle crossings

On pen and paper, it is only possible to draw a rough sketch of the root locus, it is better practice to use Matlab for all kinds of root locus calculations as this method is more suited numerically but analytically. A rough idea of how the root locus looks like can be drawn on the paper just to get an idea but the precise root locus with all its properties is only possible using Matlab.

### 12.1.4 Root locus using Matlab

Matlab uses a brute force calculation of root locus determination using numerical calculations rather using the analytical approach as described in the previous section. A numerical approach for calculating is better suited approach as it is only possible numerically to obtain a better precision than analytically drawing on tools such as graphs.

## Nomenclature

Consider the system shown in figure 12.2 again, the OL and CL TF can be expressed as:

$$G_{OL} = \frac{K_P}{s^2 + 2s + 3} \quad (12.10)$$

$$G_{CL} = \frac{K_P(s+2)}{s^2 + (2+K_P)s + (3+2K_P)} \quad (12.11)$$

In Matlab the response of the poles is calculated starting from  $K = 0$  to  $K \rightarrow \infty$  numerically such as when  $K = 0$ , the CL denominator becomes:

$$D(s)_{CL} = s^2 + 2s + 3 \quad (12.12)$$

which is equal to the OL denominator, further, with  $K = 1$  and  $K = 2$ , the CL denominator can be expressed as  $s^2 + 3s + 5$  and  $s^2 + 4s + 7$  respectively. At  $K = \infty$ ,  $D(s) = K_Ps + 2K_P = K_P(s+2)$  which is equal to the OL zeros. Therefore, it can also be proved numerically that the root locus of a control system starts with OL poles and ends at OL zeros.

## 12.2 Examples of control design with Root Locus

Steps in control design using Root Locus:

- The first thing in the control design is to translate the performance specifications into closed loop pole locations
- Draw Root Locus of the system using a simple gain  $K$
- See if gain change can meet the requirements
- If not, then add poles and / or zeros via a controller to reshape the root locus to pass through the desired closed-loop pole locations

### 12.2.1 Example: Using a simple P control

Consider the system as shown in figure 12.4. The system has a zero to it, this system however can be approximated to the properties of a PT-2 canonical system. The object is to find the control gain  $K$  such that settling time for the system is less than  $4/3s$ . Even though the system is not canonical, the control properties can be approximated using the properties from PT-2 system and the control gains found can serve as a starting point in achieving the required control from the given non-canonical system. The settling time for a PT-2 system is given by:

$$t_s \approx \frac{4}{\sigma} = \frac{4}{3} \implies \sigma = 3 \quad (12.13)$$

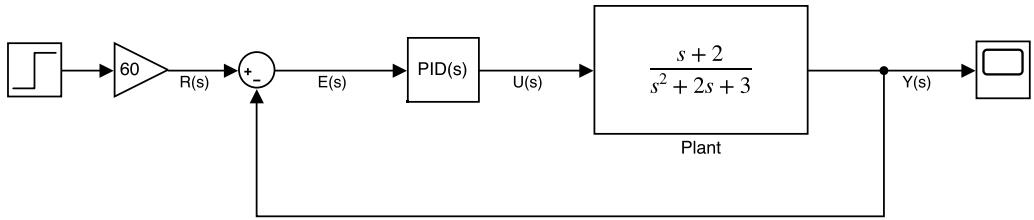


Figure 12.4: Control design using root locus (non-canonical systems)

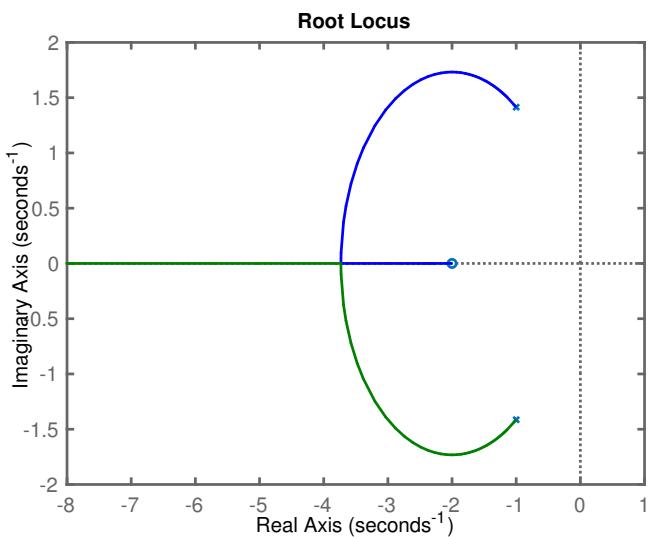


Figure 12.5: Control design using root locus (non-canonical systems)

The root locus of the system is generated using Matlab as shown in figure 12.5. As can be seen from equation (12.13) that the real part of the poles should be 3 for the condition to be met. Looking at figure 12.5, it can be seen that this condition is possible at two complex pole location (if we imagine a line drawn through  $-\sigma = -3$ ). So the requirement can be met. It can also be noted that as  $K$  is increased, the two poles proceed farther towards left of  $\sigma$ , therefore, providing a better settling time. However, it should also be noted that one of the two poles (blue line) after break-in point proceeds in the direction of increasing  $\sigma$  for higher  $K$ . There are two poles in the system, at higher  $K$  the pole with blue line will be dominant because of its slower settling time.

Because this system is not exactly a canonical system, the settling time of  $4/3$  may be achieved based on the root locus that falls under the value of  $-\sigma = -3$ . However, it is always not possible with non-canonical systems to achieve these properties accurately mainly because of the lack of insights that we have compared to the canonical systems.

## Nomenclature

Using Matlab the following system responses can be generated for a unit step function. For  $\sigma = 3$ , from Matlab it can be found that  $K = 4.07$ . The system response is found using Matlab again as shown in figure 12.6.

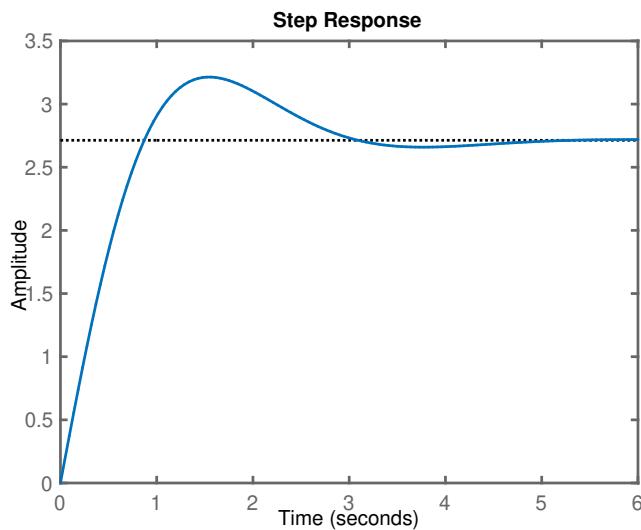


Figure 12.6: System response for  $K = 4.07$

Further increasing gain  $K = 5.46$  such that  $-\sigma = -3.73$ , the following system response was generated given in figure 12.7.

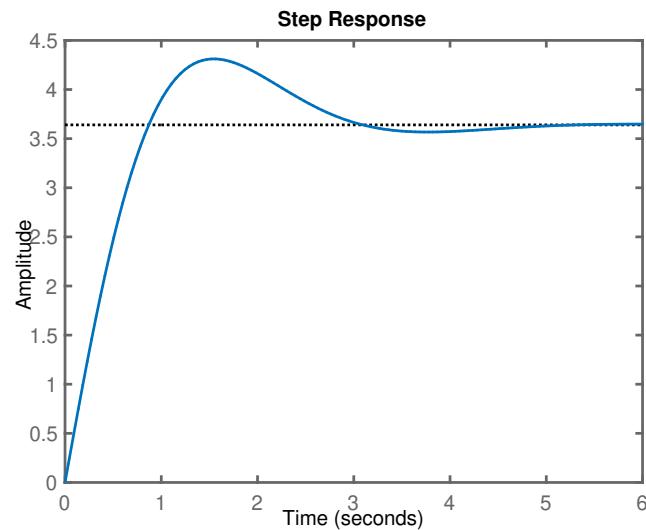


Figure 12.7: System response for  $K = 5.46$

## Nomenclature

Clearly, from both the figure (12.6 and 12.7), it can be seen that the required system response is not achieved, this is all because of a single zero that is present in the system due to which it would be not possible from the  $\sigma$  value calculated using (12.13) which is actually an approximated value for  $t_s$  of a PT-2 system.

### 12.2.2 Example: Altering root locus using I or D poles or zeros

Consider a PT-2 system in series initially with a P control as shown in figure 12.8. The object here is to achieve a peak time of less than 1 sec and overshoot less than 4 %. As

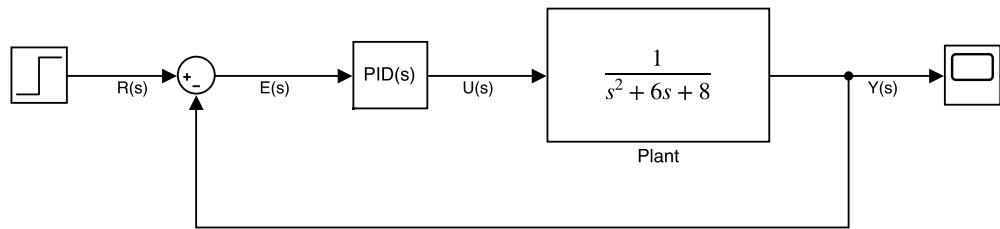


Figure 12.8: Example using root locus altering for this system

this is a PT-2 system, the system properties formulae for the PT-2 system can be used for finding frequency  $\omega_n$  and damping ratio  $\zeta$  (to find the area under root locus which actually produces desired results) :

$$t_p = \frac{\pi}{\omega_d} < 1 \implies \omega_d > \pi \quad (12.14)$$

$$M_p = e^{-\zeta\pi/\sqrt{1-\zeta^2}} < 0.04 \implies \zeta > \sqrt{\frac{\ln(0.04)^2}{\pi^2 + \ln(0.04)^2}} \approx 0.707 \quad (12.15)$$

also the angle  $\beta$  will provide the area under the root locus that of interest in this problem:

$$\cos \beta = \zeta \implies \beta = \cos^{-1} \zeta \implies \beta \leq 45^\circ \quad (12.16)$$

Plotting this root locus on Matlab with the parameters that determine the area under the root locus important  $\omega_n$   $\zeta$   $\beta$ , the root locus is as shown in figure 12.9.

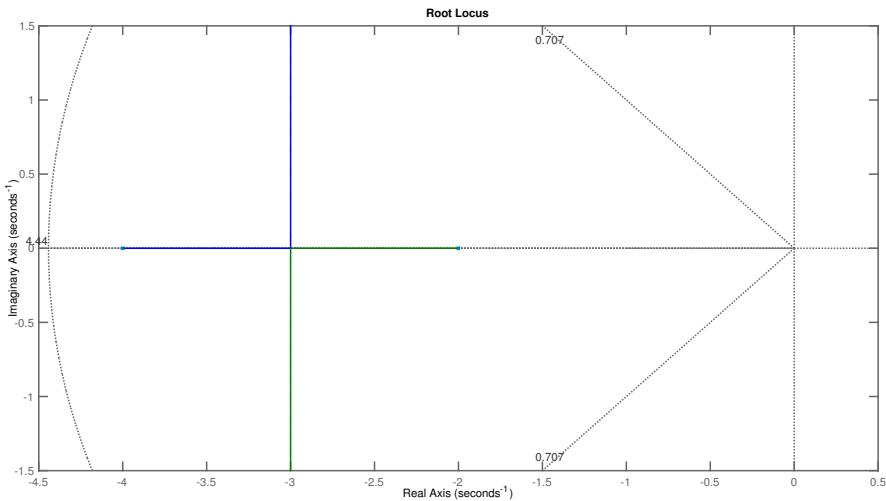


Figure 12.9: Root locus with area of control interest enveloped between semi-circle and angled lines

As can be seen from figure 12.9, the 45% angles lines are from the value  $\beta$ , the area under which  $\beta < 45\%$  (required). The semi-circle denotes the line on which  $\omega_n = 4.44\text{rad/s}$ , the area over this line is when  $\omega_n > 4.44$  (required). It can be seen that this area does not cover the existant root locus, therefore, there is a need to add either a differential or an integral control either to add poles or zeros. By observing figure 12.9, it can be seen that the poles are proceeding to  $\infty$ , the required area is to the left of the existing root locus, therefore, a first attempt would be to try to bend the poles that are pointing to  $\infty$  towards the left of the existing root locus. One way to do this is by adding a zero which will bend the poles pointing to  $\infty$  towards the left side where it is required. A D control is therefore employed in the first attempt.

Writing the PD control,

$$C(s) = K_D s + K_P = K(s + z) \quad (12.17)$$

where,  $z = K_P/K_D$ . From equation (12.17), it can be seen that the location of zero is decided by  $s = -z$ , so through an iterative approach it can be decided how far the zero location can be placed (value of  $z$ ) in order to achieve desired performance. Looking at the root locus 12.9, we have two poles, there are three positions where a simple zero can be placed as shown below.

### 12.2.2.1 Case 1: Zero to the right of two poles

Suppose the zero is placed to the right of two poles, such that  $s = -1$ . The new root locus generated with Matlab can be seen from figure 12.10.

## Nomenclature

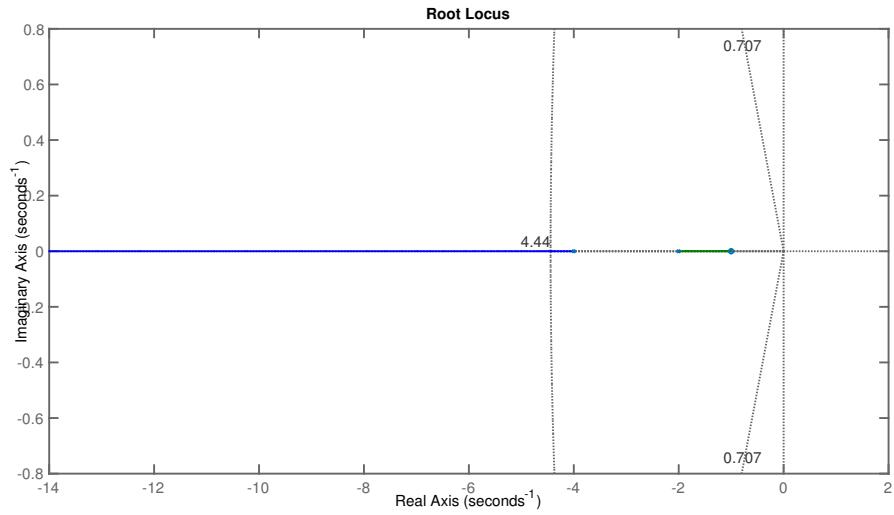


Figure 12.10: Altering root locus by placing a zero to the left of two poles

It can be seen from figure 12.10, it can be seen in this case that even though the root locus lies in the desired are (blue line), the dominant pole (green line) does not lie in the deisred region. The dominant pole will dominate the response of the system and its root locus does not lie in the requiried area for the desired control. Therefore, this case will be rejected.

### 12.2.2.2 Case 2: Zero in between two poles

For a zero to be inbetween two poles, suppose that  $s = -3$ , from Matlab the following root locus with the desired area can be generated as shwon in figure 12.11.

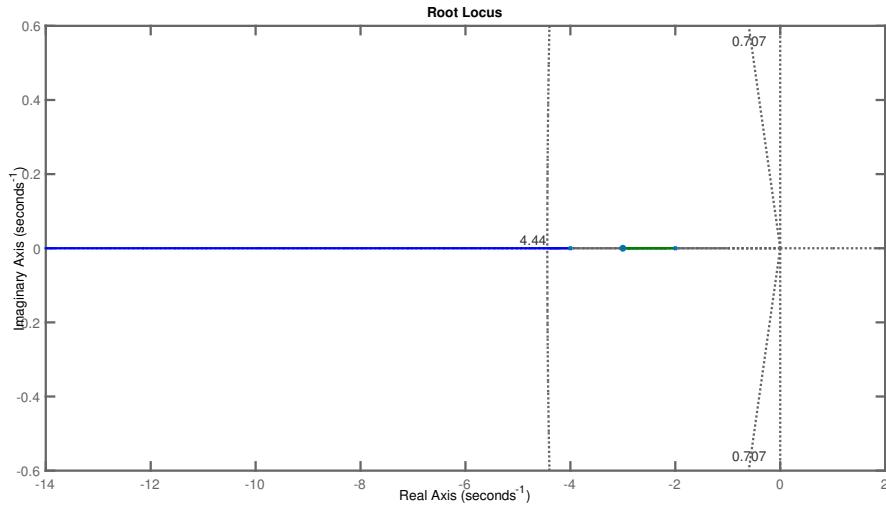


Figure 12.11: Altering root locus by placing a zero in-between two poles

Again in this case the result is same as the result of case 1;

### 12.2.2.3 Case 3: Zero to the left of two poles

For a zero to the left of two poles, suppose that  $s = -5$ , from Matlab the following root locus with the desired area can be generated as shown in figure 12.12.

## Nomenclature

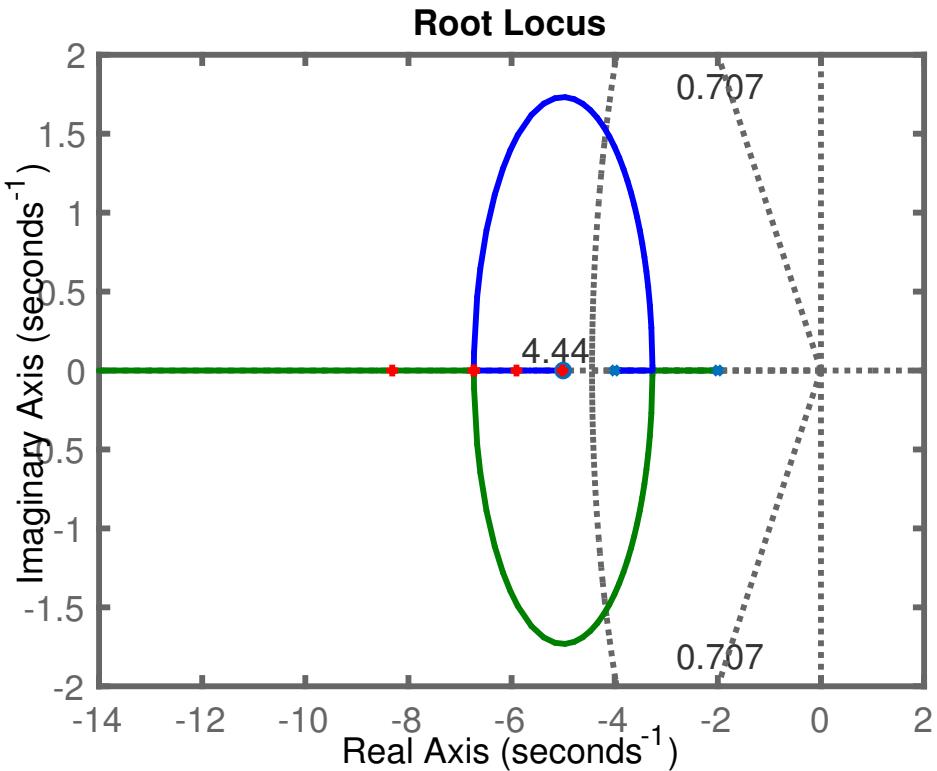


Figure 12.12: Altering root locus by placing a zero in-between two poles

Finally, with this case, the dominant pole lies inside the design area for control (shown by red dots on the blue line). Therefore, the control equation (12.17) can be set to:

$$C(s) = K(s + 5) \quad (12.18)$$

Using Matlab function `[k,poles] = rlocfind(sys)`, the system gain  $K$  for which all the control properties that match the desired requirements can be found, which is found to be at maximum gain  $K = 224.84$  therefore, the new poles will now be at  $\{s_1, s_2\} = \{225.84, 5.01\}$  for the chosen gain  $K$ . The step response of the system without any control is as shown in figure 12.13.

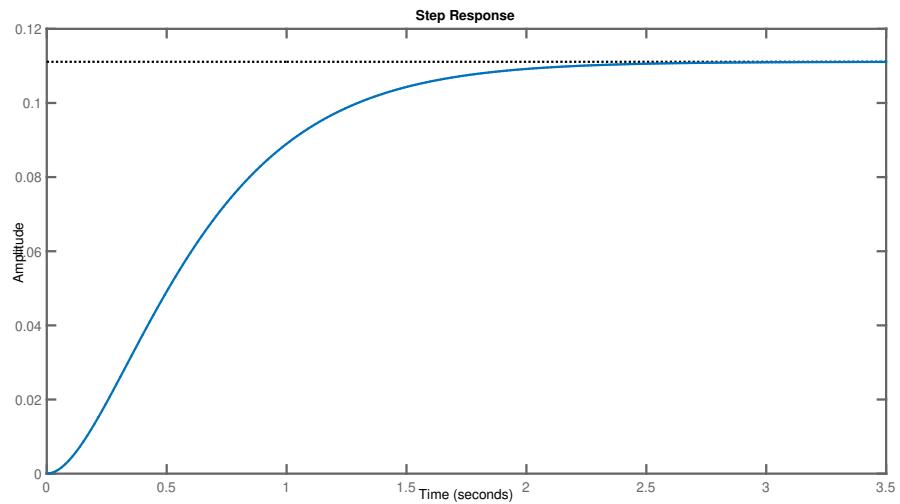


Figure 12.13: Altering root locus by placing a zero in-between two poles

It can be seen from figure 12.13, that the closed loop step response of  $G$  has a rise time greater than 1, which needs to be controlled. Therefore, now with the control design by placing a zero into the system as shown in the above description, a new step response can be plotted using Matlab as shown in figure 12.14.

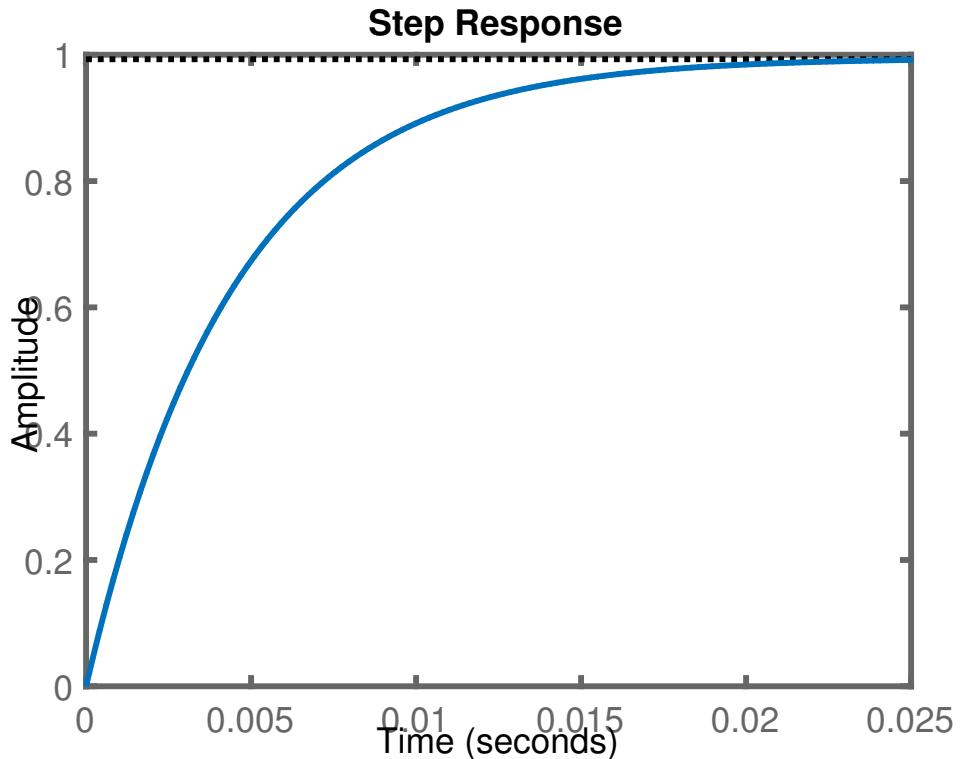


Figure 12.14: Altering root locus by placing a zero in-between two poles

As can be seen from figure 12.14, the design objective for the required control has been achieved for this system. Generally, it is not possible to achieve such a perfect control as in most of the cases the systems are noncanonical and the inputs are more random. Therefore, a different control design using frequency domain analysis of the system is employed in all such cases.

### 12.2.3 General trends of adding a zero or a pole

In general the following trends from the poles and zero positioning in the root locus can be said:

1. Adding a zero to the left of poles will make the system more stable and settle faster.
2. Adding a zero near the pole or vice-versa will cancel out their effect (pole-zero cancellation)
3. Adding a pole will tend to pull the root locus towards right which tends to make the system more unstable and much slower.

# 13 Frequency Response Analysis

So far the literature discussed in this book is concerned with the time response analysis using a step input. The time response analysis of dynamical system is limited to simple inputs such as step input or impulse input and so is the control design techniques used in pole placement and root locus methods. In general, there are complex signals given to a dynamical system and therefore a separate analysis of such complex signal has to be studied separately. In frequency response, analysis of system behavior is studied using complex repetitive signals such as sinusoidal waves. For a quick understanding of frequency response of a dynamical system, the analytical solution can be considered where there is both a transient response (due to natural system dynamics) and a particular response (due to forcing functions) as shown in figure 13.1. Where the terms

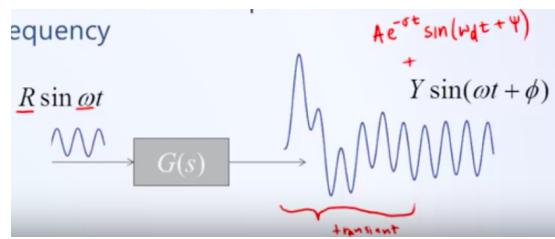


Figure 13.1: Graphical Representation of the Analytical frequency response

in black are the input and the steady-state response (particular solution) of the system. The term in red is the null solution (transient) of the system. As the null solution is a function of  $e^{-\sigma t}$ , where  $\sigma$  is a real part of the complex number. It can be seen that this null part goes to zero leaving behind only the particular solution (ss response). For a linear and a stable system (given input of a sinusoidal frequency), after the transients have been died out, the systems steady-state response will also be a sinusoidal output (particular solution) but might have a different magnitude (scaling with input) and could also have a phase shift with the input signal. In frequency response analysis, a study is conducted in order to determine such magnitude and phase angle differences between the input signal and the response of the system.

In frequency response there are only two primary quantities that have implications on the system performance:

1. Scaling: The scaling of the magnitude of the system response to the input signal.
2. Phase shift: The delay in the response of the system's response to the input signal.

## Nomenclature

Generally, these two parameters are determined using experimental data, they can also be calculated using the TF  $G(s) = G(j\omega)$ , where  $s = j\omega$ , considering the frequencies of the system only. Therefore, the magnitude is determined by  $|G(j\omega)|$  (ie., calculating the magnitude of a complex number (generally pythagorean theorem)) in which case you should get  $\omega$  of the systems frequency. Similarly, the phase shift can be determined by calculating the angle of the complex number  $\angle G(j\omega)$  from the Re axis (generally calculated using atan).

Frequency analysis can help decide system behavior on various aspects:

1. Attenuation: It could be both desired or undesired for the signal from the control system to be attenuated.
  - Desired: In cases of noise cancellation and disturbance rejections (in some cases attenuation of control signal towards external disturbances is desired).
  - Undesired: In cases where the control signals does not track the commanded reference signal.
2. Amplification: Can destabilise the system (may cause resonance)
3. Phase lag: Information can be delayed which would cause the decay in the performance or could also destabilize the system.

One of the ways to present this information is done using several different diagrams:

1. Bode Diagram (two graphs)
  - magnitude vs. frequency
  - phase vs. frequency
2. Nyquist plot (single plot)
  - magnitude vs. phase (polar)
3. Nichols chart (single plot)
  - magnitude vs. phase (rectangular)

## 13.1 Plotting bode plots

Bode plots basically describe the frequency response of a dynamical system (magnitude and phase shifts) with reference to an input signal. As described in the previous section, these magnitude and phase shift of linear and stable system determine their various behavior and therefore, the response of the system as described in the lists 13. In order to determine any of these system responses with reference to an input frequency, the magnitude and phase shift of the system response can be determined using Bode plots. There are two ways of plotting Bode plots, point by point (Matlab approach) and Line approximation approach (quick drawing method on paper).

### 13.1.1 Approach 1: Point-by-Point approach

With point by point approach, Matlab calculates the individual magnitude and phase shifts at every input frequency through a brute force method using magnitude and angle calculations of the complex numbers. In the brute force method, substitute  $s = j\omega$  into  $G(s)$  and calculate magnitude and phase for series of different input frequencies  $\omega$ . The magnitude is scaled to a log-log scale so as to incorporate the whole range of frequencies and multiplied by a factor of 20 (not sure why?) as follows:

$$M(\omega) \text{indB} = 20\log_{10}|G(j\omega)| \quad (13.1)$$

Consider a PT-1 system,

$$G(s) = \frac{1}{\tau s + 1} \quad (13.2)$$

In order to determine the magnitude of this systems response, substitute  $s = j\omega$  at every  $\omega$  and perform the complex number caculation.

$$G(j\omega) = \frac{1}{\tau j\omega + 1} \quad (13.3)$$

taking the log scale and multiplying by a factor of 20, the above equation takes the form:

$$\begin{aligned} 20\log_{10}(G(j\omega)) &= 20\log_{10}\left(\frac{1}{\tau j\omega + 1}\right) \\ &= 20\log_{10}(1) - 20\log_{10}(j\omega + 1) \\ &= 0 - 20\log_{10}(j\omega) - 20\log_{10}(1) \\ &= -20\log_{10}(j\omega) \end{aligned}$$

Phase shift can be determined calculating angle of the complex numbers:

$$\phi(\omega)\text{indeg} = \angle G(j\omega) \quad (13.4)$$

Alternatively, the magnitude and phase angle can be also determined using geometry:

$$|G(j\omega)| = \sqrt{\operatorname{Re}(G(j\omega))^2 + \operatorname{Im}(G(j\omega))^2} \quad (13.5)$$

$$\angle G(j\omega) = \tan^{-1}\left(\frac{\operatorname{Im}(G(j\omega))}{\operatorname{Re}(G(j\omega))}\right) \quad (13.6)$$

### 13.1.2 Straight line approximations

This approach is a rough sketch of the freqeucy response hand drawn in order to derive a quick apprximation of the response. This is derived using standard results for each of the terms such as constants, poles or zeros at origin or at other places and summing

## Nomenclature

up their results. This method is good for understanding quickly the systems response before hand and improves analytical understanding.

It is important to know the most important standard results of various Bode plots from different components such as integrator, differentiator, gain, poles and zero in order to draw conclusions from the given Bode plot about the system. Studying system responses from different components helps to understand the system components responsible for such Bode plot generation during system identification.

In this straight line approximation technique, the response of individual components is first determined on the Bode plot and then each of the individual responses is added together using some mathematical properties. For example consider a system given by its TF:

$$G(s) = \frac{10s}{s + 10} \quad (13.7)$$

this system can be decomposed into its individual components such as:

$$G(s) = (10)(s) \left( \frac{1}{s + 10} \right) \quad (13.8)$$

where 10 is the gain,  $s$  is the differentiator and  $1/(s + 10)$  is the pole of the system. Since the magnitude response is plotted on the logarithmic scale as  $dB$  on Bode plots, the summation property of the magnitude responses can be described as follows:

$$\text{magnitude} = |G_1(j\omega)G_2(j\omega)| = |G_1(j\omega)||G_2(j\omega)| \quad (13.9)$$

$$\implies \log(|G_1(j\omega)||G_2(j\omega)|) = \log(|G_1(j\omega)|) + \log(|G_2(j\omega)|) \quad (13.10)$$

From equation (13.10), it can be seen that on the logarithmic scale, the response of each component can be added linearly for the entire system as given by equation (13.10).

In case of phase response, the phase is not plotted on the logarithmic scale, rather the mathematical properties of complex numbers is used as described below:

$$\text{phase} = \angle G_1(j\omega)G_2(j\omega) = \angle G_1(j\omega) + \angle G_2(j\omega) \quad (13.11)$$

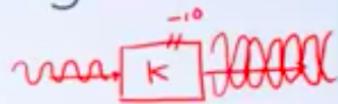
The above equation (13.11) holds good as two complex numbers when multiply each other the result is given by their linear sum. Therefore, this mathematical proofs are used in order to construct a Bode plot for straight line approximations.

### 13.1.2.1 Bode plot of Gain $K$

When the component is a simple gain  $K$ , then the magnitude will only be scaled up by the value of  $K$ . The phase will depend on the sign of  $K$ . If  $K$  is positive, then phase =  $0^\circ$  or when  $K$  is negative, then the phase will be  $180^\circ$  as shown in figure 13.2. The magnitude will remain same whether  $K$  is positive or negative, only the phase would change.

## How to Plot a Bode Diagram

- Need a library of components
- Constant gain ( $K$ )



Sketching Bode Diagrams

Figure 13.2: Bode plot of gain  $K$

### 13.1.2.2 Bode plot of Differentiator $s$

The magnitude response of the Differentiator can be determined by substituting values of  $s = j\omega$  in the TF  $G(s)$ . For a case when the number is purely imaginary,  $s = j\omega$ , as the value of  $\omega$  increases, the value on the Im axis also increases ( $j$  is to only denote that the value is on the Im axis). Because the magnitude of  $j\omega$  is increasing there should be a slope to this line, in straight line approximation this slope is considered constant and therefore, the magnitude increases linearly. Further, at  $\omega = 1$  rad/s,  $|G(j\omega)| = 20\log(1) = 0dB$  at every multiple of  $\omega$  to 10 rad/s (a decade), the magnitude will be  $|G(j\omega)| = 20\log(10) = 20dB$ . Therefore, the slope will be a constant (linear) of  $20dB/decade$  as shown in figure 13.3. Similarly, for every factor of  $\omega$  in 10 rad/s, the slope will be continue backward with a slope value of  $-20dB/decade$ , therefore, the magnitude at 0.1 rad/s will be  $-20dB$  as shown in figure 13.3. Further, the phase of the differentiator will always be  $90^\circ$  as  $s = j\omega$  will always be on the positive of Im axis (negative  $\omega$  does not exist physically, there could be a sign added due to convention) which makes  $90^\circ$  with the positive Re axis as shown in figure 13.3. Another intuition for the phase plot is as follows, if the input sign wave is sinusoidal of sine wave type,

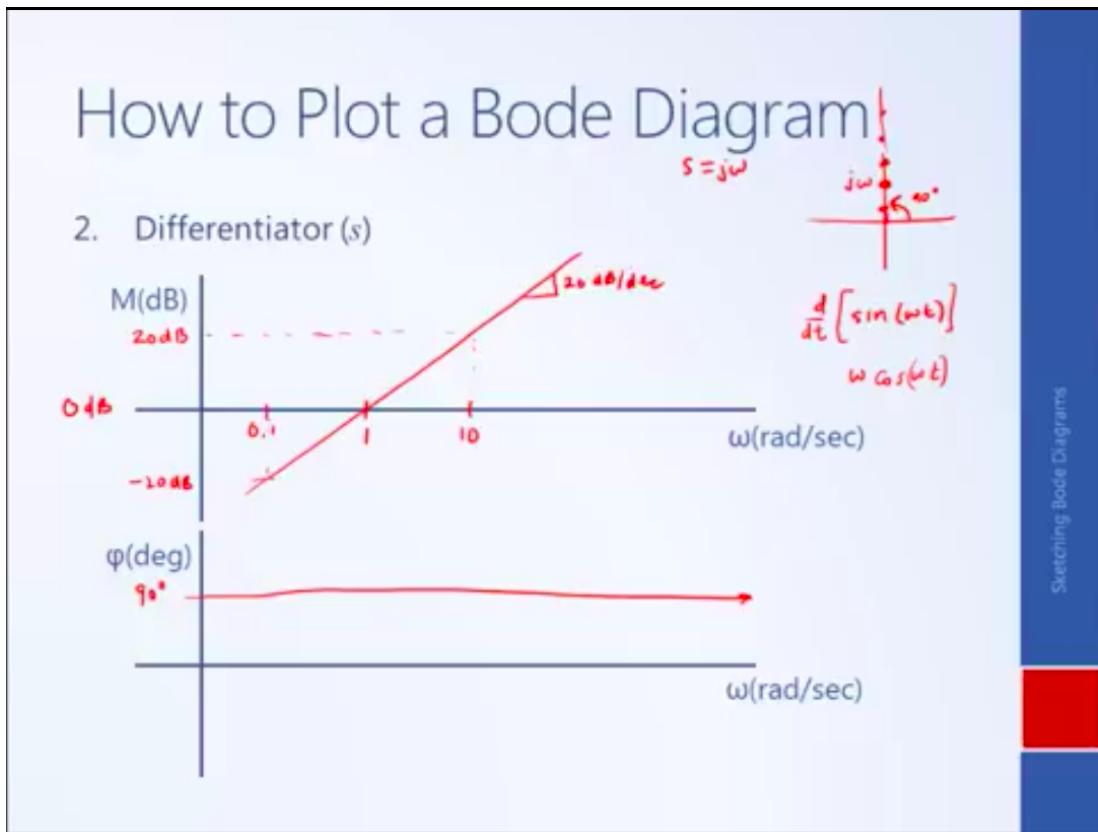


Figure 13.3: Bode plot of Differentiator  $s$

the differentiator will differentiate this wave as:

$$\frac{d}{dt} \sin(\omega t) = \omega \cos(\omega t) \quad (13.12)$$

As cosine wave is always leading the sine wave by  $90^\circ$ , in this it is lagging as cosine is positive, the phase leads by  $90^\circ$  to the input sine wave. Also, note that the magnitude of the response increases as  $\omega$  increases as given by equation (13.12) ( $\omega$  is multiplied with cosine wave).

### 13.1.2.3 Bode Plot of Integrator $1/s$

Again, the Bode plot of integrator can be determined by substituting  $s = j\omega$ . In case of complex numbers, generally when  $j$  is in the denominator, it is canceled out as:

$$\frac{1}{s} = \frac{1}{j\omega} = \frac{1}{j\omega} \frac{j\omega}{j\omega} = \frac{j\omega}{-\omega^2} = -\frac{j}{\omega} \quad (13.13)$$

From the above equation, it can be seen that as  $\omega$  increases, the value of the magnitude is plotted as the ratio  $-1/\omega$  on the Im axis as shown in figure 13.4. As  $\omega$  increases,

the ratio  $-1/\omega$  decreases and reached the origin of the Im axis at higher values of  $\omega$ . Therefore, the slope in this case is negative and is of value  $-20dB/decade$  as shown in figure 13.4. The magnitude cross-over point can be found numerically when  $\omega = 1$  rad/s.  $|G(j\omega)| = 20\log(-1) = 0dB$ .

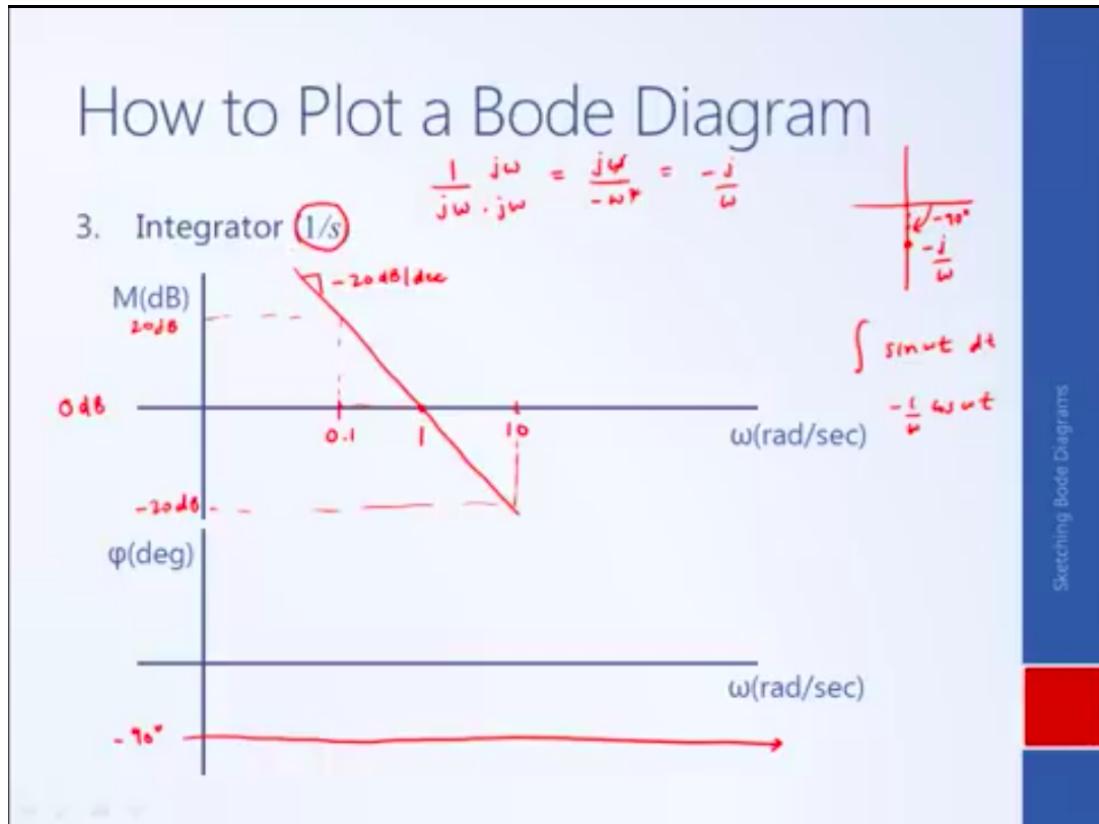


Figure 13.4: Bode plot of Integrator  $1/s$

## Nomenclature

Another intuition can be made in time-domain as  $1/s$  is an integrator in time-domain, for an input frequency of sine wave, the input sine function as a function of  $\omega \sin(\omega t)$  is integrated as follows:

$$\int \sin(\omega t) = -\frac{1}{\omega} \cos(\omega t) \quad (13.14)$$

The negative cosine wave is always lagging by  $90^\circ$  from the input sine wave. Also ,it can be seen that the magnitude is initially negative and increases towards the origin as per the ratio  $-1/\omega$ . Another intuition from the above equation is that at low frequencies the magnitude of the system response if high, this results in high DC gain  $K_{dc}$  of the system at low frequencies. Such high gain at low frequencies provides the sufficient system gain to over-come the steady-state error.

### 13.1.2.4 Bode Plot for simple Zero

For a case of simple zero the TF is written in the form  $(Ts + 1)$ , this is called the Bode form. The magnitude and phase plots are determined by substituting  $s = j\omega$ . At low frequencies  $Ts$  is very small and 1 will be dominating, therefore the magnitude and phase response in Bode plot will be a flat line similar to a simple gain  $K$  as shown in figure 13.5. At higher frequencies (higher vales of  $\omega$ ), the term  $Ts$  will be dominating and the multiplication of  $s$  term in Laplace will result in differentiation in time-domain. Therefore, the magnitude and phase response in Bode plot will be similar to that of a differentiator  $s$  as described previously. The magnitude will increase with a positive slope of  $20dB/decade$  and the phase will be leading with  $90^\circ$  angle to the input as shown in figure 13.5. In between these two lines the intermediate phase is a straight line approximation of a straight line connecting those two lines.

The cut-off point when the magnitude crosses over the  $0dB$  line is given by the ratio  $1/T$  (I don't know why). For the phase plot, the slope starts at a decade earlier when  $\omega = 1/T$  and ends a decade later after  $\omega = 1/T$  as shown in figure 13.5.

### 13.1.2.5 Bode Plot for simple Pole

Similarly, in the case of a simple pole, the magnitude and phase response is determined by substituting  $s = j\omega$ . The TF of Bode form of a simple pole is expressed as:

$$SimplePole = \frac{1}{Ts + 1} \quad (13.15)$$

substituting  $s = j\omega$ :

$$\frac{1}{Ts + 1} = \frac{1}{Tj\omega + 1} \quad (13.16)$$

at lower input frequencies, the term 1 will dominate the function and the output TF will just be 1 and the magnitude in  $dB$  will be 0. The magnitude response is therefore, similar to a simple gain  $K$ . The phase similarly, behaves like a simple gain with phase  $\phi = 0^\circ$  as in this case the simple gain is always positive. At higher frequencies the

$s + 3$

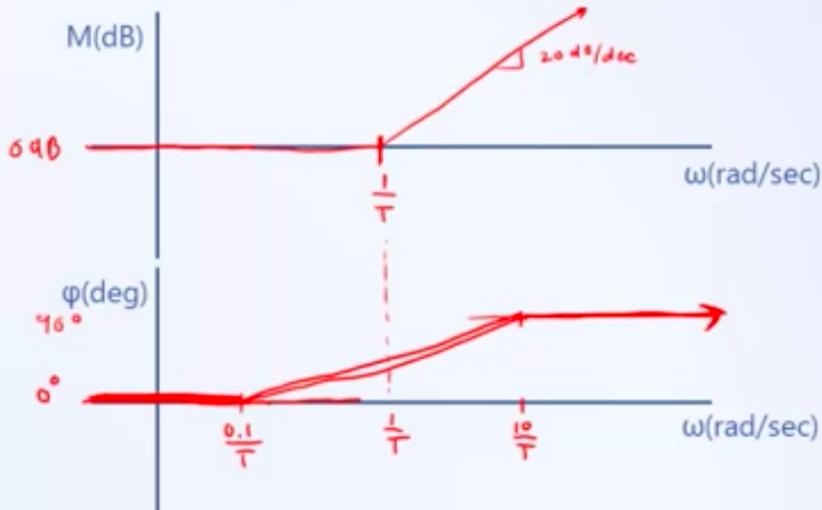
## How to Plot a Bode Diagram

### 4. Simple zero ( $Ts + 1$ )<sup>\*</sup>

$T(i\omega) + 1$

for small  $\omega \approx 1$

for large  $\omega \approx s$



Sketching Bode Diagrams

Figure 13.5: Bode plot of simple zero ( $Ts + 1$ )

term  $1/Ts$  will dominate and therefore, the output TF acts like an integrator in the time-domain. As it is with integrators, the magnitude will decrease with a negative slope of  $-20\text{dB}/\text{decade}$  and the phase will have a constant lag of  $-90^\circ$  with the input frequency  $\omega$  as shown in figure 13.6. The intermediate line between a constant gain term 1 and an integrator term  $1/Ts$  is approximated as a straight line. Similar to the differential plot, the break-off frequency  $\omega$  at which the magnitude crosses-over the 0dB line is at  $1/T$  and the slope starts and ends a decade earlier  $0.1/T$  and a decade later  $10/T$  respectively as shown in figure 13.6.

**Important notes:** As was seen in this section, that a simple pole acts like an integrator at higher frequencies, the effect of integrator eliminating ss-error seems like should be evident at higher frequencies. At lower frequencies therefore an additional integrator is added so that the effect of the integrator is already evident at lower frequencies eliminating ss-error.

**Important notes:** Unlike time-domain response (step input), in frequency response the effects of non-dominant poles (poles with faster settling times) can be clearly seen in the Bode plots. As the effects of such poles are evident at input  $\omega$  of higher frequencies. In time-domain analysis, the input frequency is very low and the effects of poles with faster settling times is not captured. As at higher frequencies the speed (frequen-

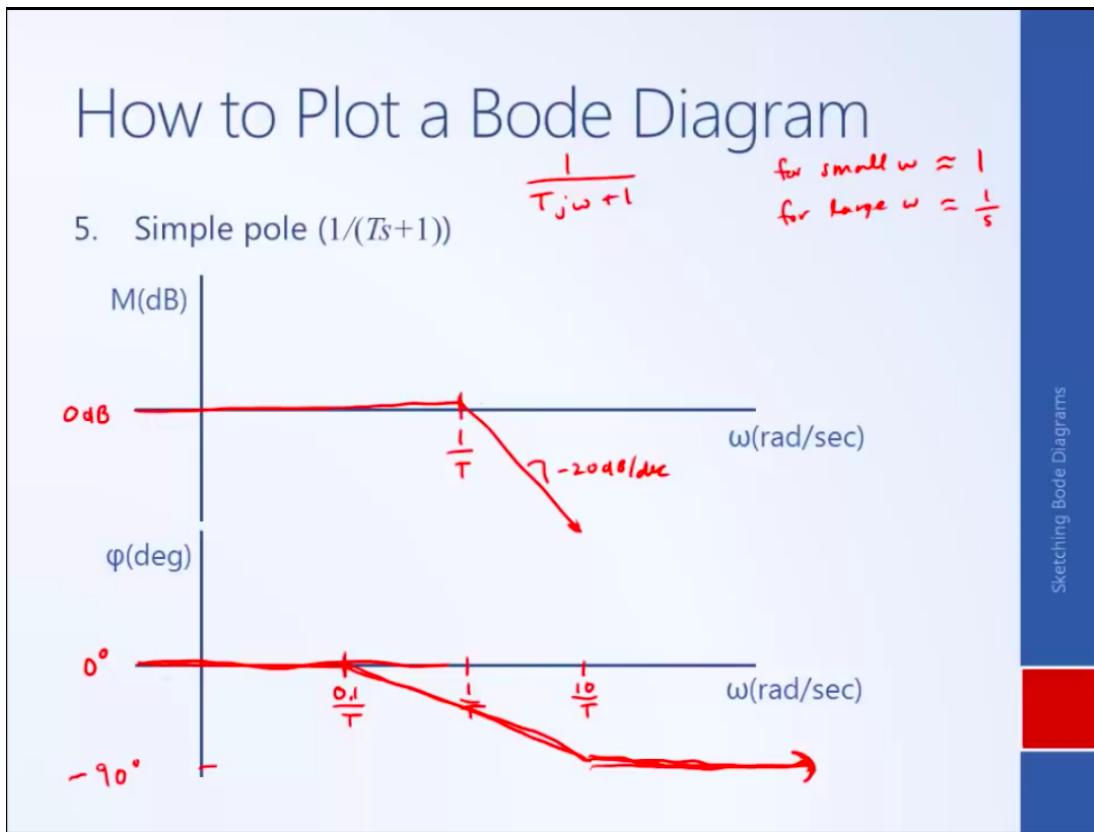


Figure 13.6: Bode plot of simple pole  $1/(Ts + 1)$

cies) of the input waves become faster the effect of the faster poles start to show up as a result of high speeds (frequencies).

Approach for drawing Bode plots:

1. Put the components (gain, diff, int, zeros and poles) in the Bode form
2. Sketch the straight line approximations of the components on the Bode plot
3. Add the graphs
4. Try to approximate the curves

Step 4 is optional as an accurate Bode plot is always possible with Matlab.

## 13.2 Stability analysis from Frequency response

From time-domain analysis, it was found that based on the position of poles from the imaginary axis, the systems stability could be defined whether it is stable or unstable or marginally stable (such as in the case of defining BIBO stability). Further assigning

degree of stability such as asymptotic stability (where the system is perfectly stable and all oscillations will damp out to zero), another degree of stability term is used in the frequency domain analysis called ***relative stability***.

Relative stability is the degree of stability of the system determined knowing the positions of the poles from the imaginary axis. As the systems with poles located far away from the imaginary axis is much more stable, damping out the oscillations quickly and have faster settling times than the systems that have their poles located closer to the imaginary axis.

Additionally, to the system stability definitions derived in the time-domain analysis using the pole positions, stability of the system can be defined using frequency based analysis as well. It turns out that the stability is defined in much better way using frequency response of the system.

### 13.2.1 Relative Stability: How far is the system from going unstable

The stability of closed-loop system can also be determined from the open-loop frequency response. Consider a closed-loop system as shown in figure 13.7.

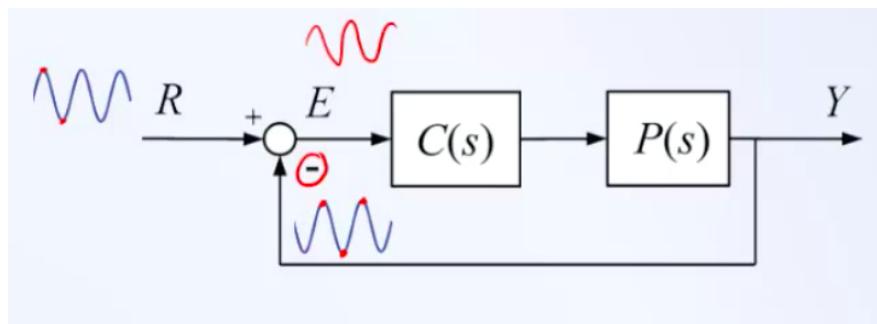


Figure 13.7: A Closed-loop system with input as frequency

## Nomenclature

The input to the system is in form of frequency, therefore, the system would react from the error signal that is also of the shape of the input signal.

**Note:** When summing two frequencies, the resulting frequency is of the same shape as the first frequency.

If the open-loop response of the system has a frequency that is completely out of phase ( $\phi = 180^\circ$ ), this will lead to addition of the two signals and therefore, the magnitude of the error is more than 1 (1 is the magnitude of the reference signal). If the open-loop attenuates the signal, then this is not a problem as the magnitude of error though greater than 1 will start to subside. However, if the open-loop amplifies the signal, then the feedback would add and amplify the error signal again and again (error signal grows exponentially) causing the actuator to saturate at its peak and the system as a whole going unstable.

The above theory can be visualized mathematically taking two sinusoids  $Ae^{-\sigma t} \sin(\omega t)$  and  $Be^{-\sigma t} \sin(\omega t + \phi)$ . Where  $Ae^{-\sigma t}$  is the amplitude (envelope) of the sine wave  $\sin(\omega t)$  given as a variable of time  $t$ . In case of  $B = A$ . Because of the summation (let the amplitude  $A = B = 1$ ), the two signals become:

$$\sum e(t) = e^{-\sigma t} \sin(\omega t) - e^{-\sigma t} \sin(\omega t - \phi) \quad (13.17)$$

$$= 2e^{-\sigma t} \sin(\omega t) \quad (13.18)$$

**Note:** The above equation was constructed by me and is not an accurate mathematical description of error signal at the summation point. The above result is generated using  $\phi = \pi$ , the  $180^\circ$  phase lag.

As can be seen from equation (13.18), the error signal becomes double the reference signal instead of going to zero (this would be the case when the signal was in phase, then the summation becomes negative in equation (13.18)). This error signal  $r(t) = 2e^{-\sigma t} \sin(\omega t)$  if amplified in the system (due to gain  $K+$ ), then it will be added and amplified again and again in the loop (exponential increase). Such a phenomena is also called **constructive interference**.

When the OL response of the system is out-of-phase from the reference signal, then the error would actually add up as given by the previous equation instead of having the difference between them. The signal can then be amplified or attenuated by the controller based on the gain  $K$ . So how far can the control gain be given before the system goes unstable due to error signal going from attenuation to amplification?

### 13.2.1.1 Defining Relative stability

With the theory laid down in the previous section, relative stability can be defined based on how far the open-loop frequency response of the system is to phase lag of  $180^\circ$  and a magnitude of 1.

Specifically, terms can be defined that are useful for defining relative stability: **Gain Margin  $G_m$** : Says when the OL response is at a phase lag of  $180^\circ$ , how much gain can be given to the system before it starts to amplify? Specifically,  $G_m$  is the distance

from the magnitude of  $1(0dB)$  at the frequency where  $\phi = -180^\circ$  (phase cross-over frequency).

**Phase Margin  $\phi_m$ :** Says, when we are at the OL-response magnitude of  $1$  or  $(0dB)$ , then how far is the OL-response phase from the  $-180^\circ$  out-of-phase frequency. Specifically,  $\phi_m$  is the distance from the phase of  $-180^\circ$  at the frequency where  $M = 0dB$  (gain cross-over frequency). Both the gain and the phase margins can be understood from the graphical figure shown in 13.8.

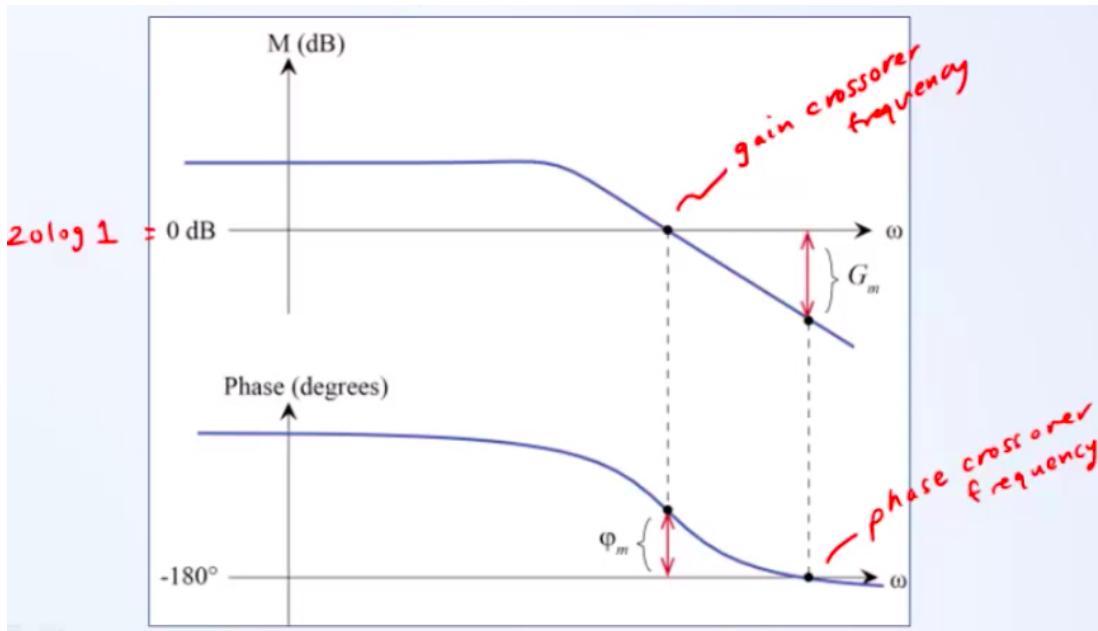


Figure 13.8: Gain and phase margins on Bode plot

Also, it should be noted from figure 13.8, that both the gain and phase margins should be positive in order for the system to be stable. As can be seen in figure 13.8, at phase cross-over frequency, the magnitude is under  $0dB$ , if gain is made negative the magnitude will cross over or at least reach  $0dB$  therefore, crossing over from attenuation to amplification that might cause resonance and actuator saturation (undesired behavior). Also, in case of phase, the phase is already in the negative region same as the magnitude any more negative phase will tend the frequency (OL) towards  $-180^\circ$  (out-of phase).

As can be seen from figure 13.8, when the OL system response is out-of-phase, then the OL magnitude is attenuated and is less than  $0dB$ . Therefore, the errors signal should also be less than the reference signal. This is the case for the given system gain  $K$ . If the control gain  $K$  is changed such a way that the OL magnitude becomes  $0dB$  then the error signal will now becomes double of the reference signal as given by equation (13.18). From here, the systems response would be added and multiplied by the feedback, causing the system to go unstable. Now, the margin of control gain

## Nomenclature

$K$  that we can alter such that OL system magnitude reaches  $0dB$  is called the **gain margin**. The similar analogy is true for **phase margin**.

In order to visualize the effect of  $K$  in eating up the gain margin  $G_m$ , consider the figure 13.9. In Bode plots when two components are connected in series their system

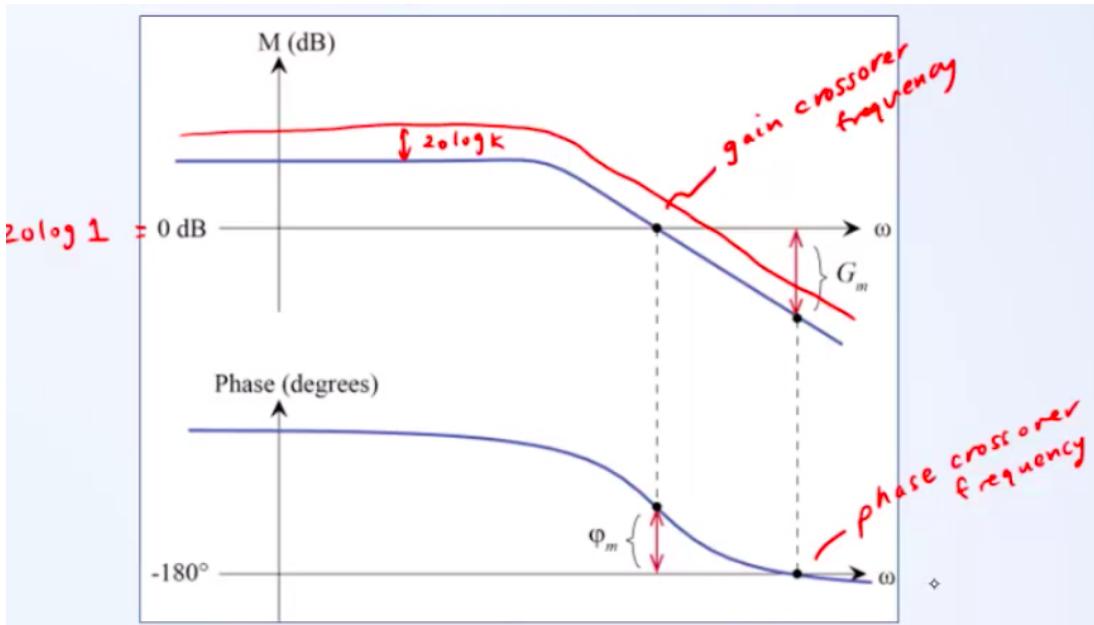


Figure 13.9: Effect of  $K$  on relative stability

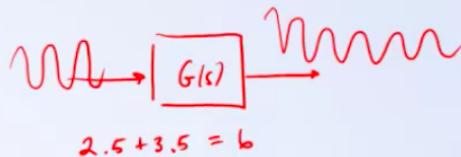
response is added up in the Bode plot. Here, the plant  $G(s)$  is in series with the controller  $C(s)$  which has a gain  $K$ . Depending on the numerical value of  $K$ , the response due to  $K$  on Bode plot is a straight line given by  $20\log_{10}K$  (adding  $20\log_{10}K$  with the blue line) which would amplify or attenuate the OL response magnitude. In case changing  $K$  attenuates the OL response magnitude, such a scenario is shown in figure 13.9. Therefore, in this case  $K$  eats up  $G_m$ , the limit beyond which the system is unstable.

## 13.3 Calculation of gain and phase shifts

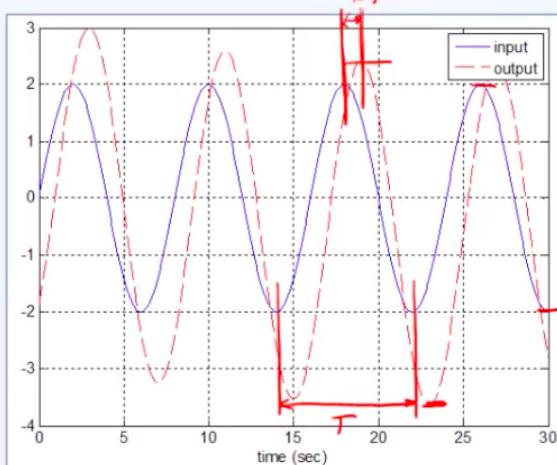
For a given systems response, the gain amplification / attenuation along with the phase shift is determined at its **steady-state response**. As an example, the following calculation is performed in order to determine the response parameters (magnitude and phase shift) as shown in figure 13.10.

Once the response (attenuation/amplification and phase) is determined for one such input frequency, the process is repeated at every input frequency and all the data points are plotted on the **Bode Plot**.

## Example



- Determine the magnitude and phase of a system that has the following steady-state output to the given input



$$T = 8 \text{ sec} \Rightarrow \omega = \frac{2\pi}{T} = \frac{2\pi}{8 \text{ sec}} = \frac{\pi}{4} \text{ rad/sec}$$

$\approx 0.785 \text{ rad/sec}$

$$\text{mag} = \frac{Y}{R} = \frac{3}{2} = 1.5$$

$\Rightarrow 20 \log 1.5 \approx 13.5 \text{ dB}$

phase lag =  $-2 \text{ sec}$  (where  $T = 8 \text{ s}$ )

phase lag  $\approx -\frac{2}{8} = -\frac{1}{4}$  of a cycle

1 cycle =  $360^\circ$

$\phi \approx -70^\circ$

Figure 13.10: Determining Magnitude and Phase using Bode Plots

### 13.3.1 System Identification using Bode Plots

In another example shown in figure 13.11, the TF of the system is determined using only the Bode plots (open-loop Bode Plot) of the system. This example is performed in order to establish various parameters that can be identified using only the Bode plots or Frequency response of the system. The following method is generally employed in **System Identification** methods. In Matlab however, there is this in-built program which does all this automatically, however, it is very important to understand the analytical basis of system identification method described for identifying control systems.

The first step is to determine the presence of integrators and differentiators in the system by examining the low frequency response. Since an integrator or the differentiator will always produce a negative or a positive slope respectively right from the beginning, it is easier to notice their presence. In our case as shown in figure 13.11, this is not the case as the line is flat at low frequencies and the slope at medium to high frequencies is due to the poles. Therefore, this system has no integrators or differentiators or in other words, there are no poles or zeros at the origin.

The second step is to determine the DC gain  $K_{dc}$  of the system. The DC gain is the

## Nomenclature

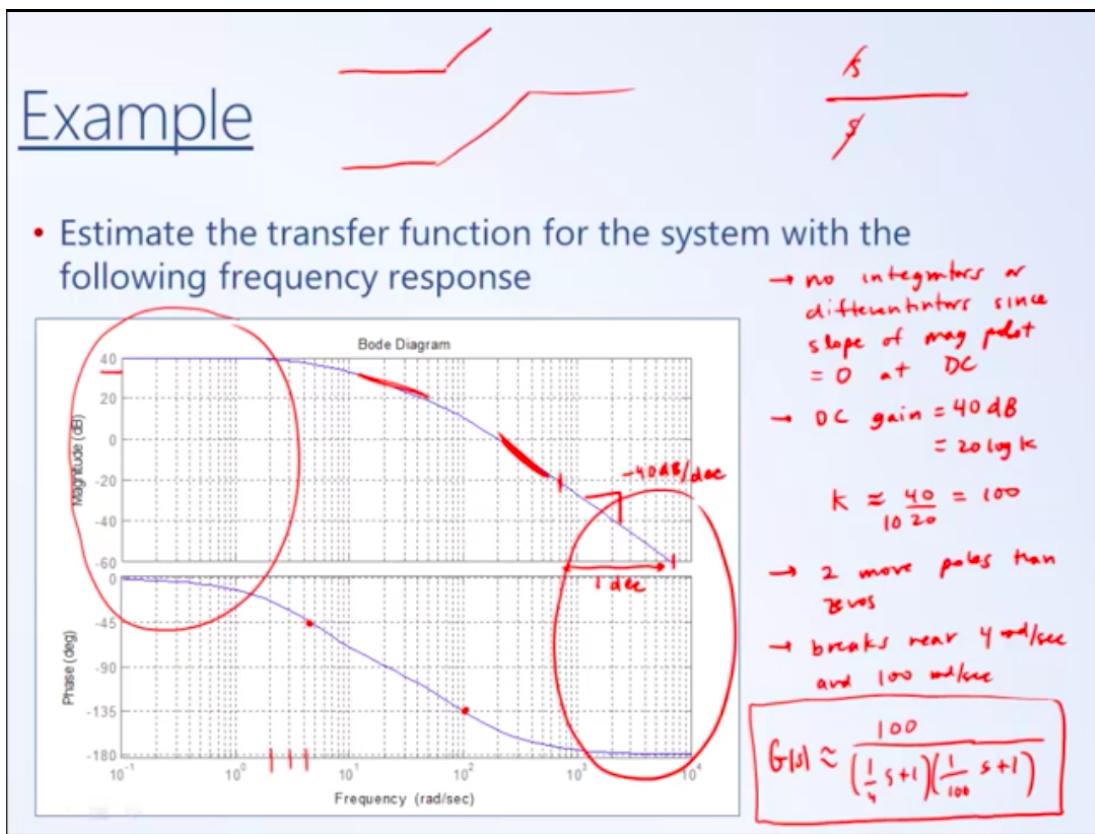


Figure 13.11: system Identification using Bode Plot

ss-response of the system at low frequencies with a constant input (step input). From the low frequency response of the system, it can be seen that the DC gain is 40dB. From this the DC gain can be expressed in the non-logarithmic scale by taking anti-log (anti-log of base 10 is 10 raise to power) as

$$40dB = 20\log K \implies K = 10^{40/20} = 100$$

The third step is to determine the number of poles and zeros. The relative number of poles in the system that are more than zeros can be determined using the high frequency response in the Bode plots. That is, the number of poles that are more than zeros. From the high frequency response shown in figure 13.11, it can be seen that there are two poles in the system more than zeros which can be verified using both the magnitude and phase response. From Magnitude response the ss-response has a slope of  $-40dB/decade$ , which is the case for two poles as each poles produce a slope of  $-20dB/decade$ . Additionally, from the phase plot it can be seen that the phase shift at higher frequencies has shifted to about  $-180^\circ$ . As is the case with each pole the phase shifts to about  $-90^\circ$ . Sometimes, it is also said that there are zeros also in the numerator but they have been canceled out by the poles in the denominator. The

cancelled out poles and zeros could be either at origins (integrators and differentiators) or simple poles and zeros (non origin). Either way, as the zeros cancel out, using Model Reduction technique it could be said that since their effect is not present, there is no need to include their dynamics and cancel them again later.

The fourth step is to determine the location of the poles. It is rather difficult to determine this manually using Bode plots, however, a rough estimation can be provided. Using Matlab the precise locations of the poles can be determined. From the magnitude plot the breaks points of the poles where the response starts to slope (as described in section 13.1.2.5) due to the presence of poles. It can be seen that the break point ( $1/T$ ) can roughly be noticed at 4 rad/s where magnitude slope is roughly  $-20\text{dB/decade}$  and the mid point  $45^\circ$  of the phase plot occurs at 4 rad/s. Similarly, the midpoint of phase plot for the second pole where the phase goes from  $-90^\circ$  to  $-180^\circ$  is clearly at 100 rad/s. Also noting that the magnitude plot slopes roughly to  $-40\text{dB/decade}$  from this point. Writing the poles in the bode form and using the DC gain determined previously, the following TF can be expressed for the given Frequency response:

$$G(s) = \frac{K_{dc}}{(s + z_1)(s + z_2)} = \frac{100}{(\frac{1}{4}s + 1)(\frac{1}{100}s + 1)} \quad (13.19)$$

## 13.4 Nyquist Diagram

A Nyquist diagram visualizes the information of the frequency response of the open-loop system on a single plot unlike in Bode plots where different plots are used. A Nyquist plot is useful for feedback loop analysis and design. Unlike in Bode plots, the gain and phase margins of a complex number  $G(j\omega)$  is not considered separately in Nyquist plots. Instead, the complex number is taken as given and a plot of  $G(j\omega)$  is generated on the complex plane, for all real values of  $\omega$ . However, in the long run, a plot of  $-\omega$  is also generated to complete the sketch.

For calculating the a simple complex number for a vector of  $\omega$ , the following approach is used: **Inverting a complex number**

$$G(j\omega) = \frac{1}{j\omega + 1} = \frac{1}{j\omega + 1} \frac{j\omega - 1}{j\omega - 1} = \frac{j\omega - 1}{-\omega^2 - 1} = \frac{1 - j\omega}{1 + \omega^2}$$

However, when the calculation for simplifying the complex number is complex, then magnitude and phase equations derived in Bode plots can be used as given by equations (13.5) and (13.5) respectively. Then the complex number can be determined by combining magnitude and phase values calculated from equations (13.5) and (13.5). For  $m$  the magnitude and  $\phi$  the phase of the complex number, the complex number itself can be calculated using either the **phasor form** or the **component form**. **phasor form**:

$$\text{complexnum} = me^{j\phi} \quad (13.20)$$

**component form:**

$$\text{complexnum} = m\cos\phi + j\sin\phi \quad (13.21)$$

## *Nomenclature*

In case of using computing softwares not capable of handling complex numbers, component form is more suitable and in case of analytical calculations, phasor form is more suitable as it is easier with multiplication and division.

### **13.4.1 An example of plotting Nyquist plot for a simple system**

Consider a open-loop of a simple system given by:

$$G(s) = \frac{1}{s + 1} \quad (13.22)$$

The complex numbers can be calculated for each of the input  $\omega$  using either of the two methods described above and a table can be constructed containing vectors of input  $\omega$  and the calculated complex number as shown below: A Nyquist plot generated for the

$\omega$	$G(j\omega)$
0.1	0.99 - 0.99j
0.5	0.8 - 0.4j
1	0.5 - 0.5j
2	0.2 - 0.4j
10	0.01 - 0.1j

Table 13.1: Table showing  $G(j\omega)$  calculation for Nyquist plot

complex numbers given in table 13.1, is plotted using Matlab as shown in figure 13.12

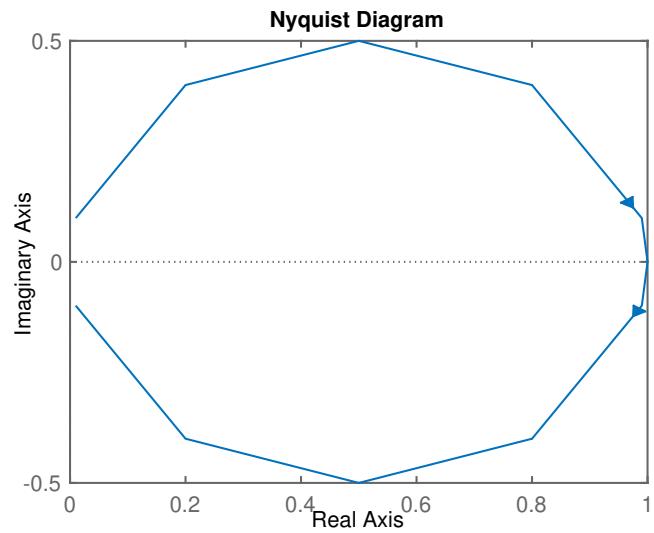


Figure 13.12: Plotting example of complex numbers in Nyquist plot

Figure 13.12 also contains  $-\omega$  shown in the upper half plane.

*Nomenclature*

# 14 Frequency Response for Controller Design

Before designing a controller, it is always preferred to express the relationship between time domain and the frequency domain. This is because ultimately the response of any system is achieved as response as a function of time. This step is done mainly by comparing step response of the system and relating parameters (such as settling time, speed of response and overshoot) in the frequency response using Bode plots. The Bode plots are then used to design the controller based on gain and phase margins available. Finally, other aspects of the controller can be decided such as Robustness and Optimality.

## 14.1 Relationship between domains

The response relationship between time and frequency response is established using ***Open-loop frequency response*** (Bode plots) and ***Closed-loop step response***. For the purpose of analysis, consider a PT-2 system of Type 0 (no integrator) described as:

$$P(s) = \frac{1}{s^2 + 0.4s + 1} \quad (14.1)$$

The step response and frequency response of the system is as plotted using Matlab in figures 14.1 and 14.2 respectively.

## Nomenclature

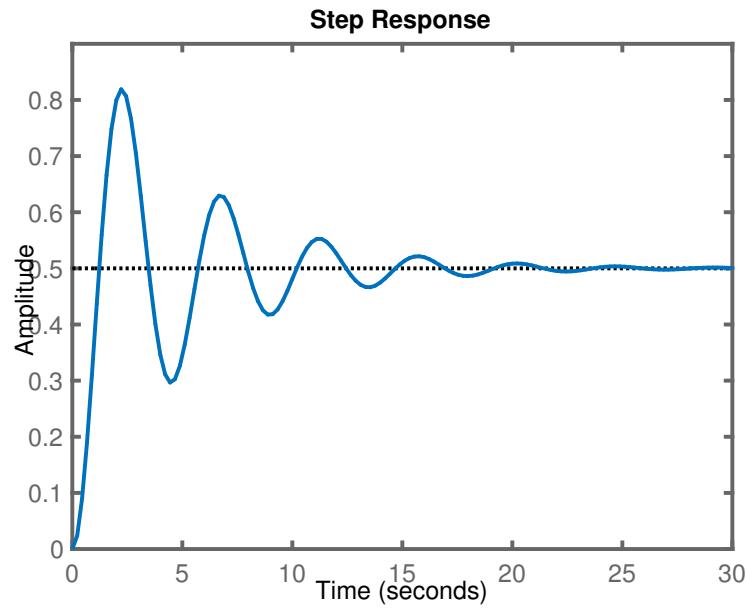


Figure 14.1: Step Response

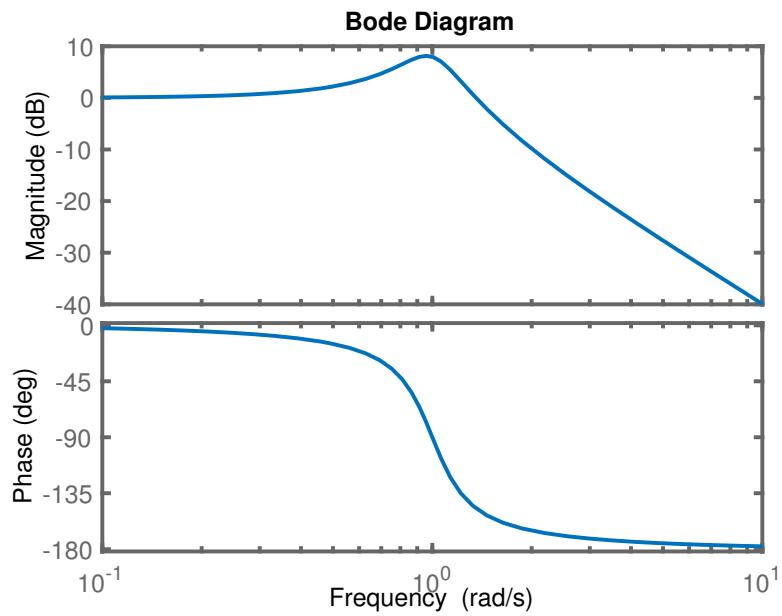


Figure 14.2: Frequency Response

Generally, the close-loop time response is related to the open-loop frequency response for various parameters described earlier. For the case of a Type 0 system, the ss-error

(closed-loop) is found using final value theorem:

$$G_{CL}(s) = \frac{E(s)}{R(s)} = \frac{\text{forward}}{1 + \text{loop}} = \frac{1}{1 + \frac{K}{s^2 + 0.4s + 1}} = \frac{s^2 + 0.4s + 1}{1 + K} \quad (14.2)$$

In the forward path, there are no blocks between  $R(s)$  and  $E(s)$ , in the loop there are blocks of  $C(s) = K$  and  $P(s)$ . The final value theorem is defined as follows:

$$e_{ss} = \lim_{s \rightarrow 0} sE(s) \implies \lim_{s \rightarrow 0} s \frac{s^2 + 0.4s + 1}{1 + K} R(s) = \lim_{s \rightarrow 0} s \frac{s^2 + 0.4s + 1}{1 + K} \frac{1}{s} \quad (14.3)$$

$$e_{ss} = \frac{1}{1 + K} \quad (14.4)$$

Looking at frequency response shown in figure 14.2, the steady-state response in time-domain is due to an DC input (constant) at low frequency. Therefore, the magnitude response at lower frequencies will determine the ss-error from Bode plots. From figure 14.2, at low frequencies the magnitude is 0dB or 1, therefore, there is a perfect tracking of the system w.r.t the input. At medium frequencies the magnitude amplifies indicating resonance and further at higher  $\omega$ , the magnitude attenuates. An attenuated magnitude at higher  $\omega$  says that the system is not capable of tracking the input magnitude at higher input speeds ( $\omega$ ), therefore, this system does not have fast dynamics to take care of the faster inputs.

The speed of the system (rise time, peak time), that is the time required by the system to react to faster inputs is determined using the Bode plot magnitude response at higher frequencies as described earlier. As the input frequencies speed up in rad/s, the system as shown in figure 14.2 cannot keep up the magnitude. It essentially, attenuates the response as a result, therefore, if an input has to be given for the system, a faster input will not be followed by the system due to its slow dynamics. The rise time and peak times of this system will be larger. This is only the qualitative analysis so far, as we are not looking into exact numbers in terms of rise and settling times.

In terms of the overshoot in step-response, this behavior can be related at medium frequencies in Bode plot. It can be seen that the system overshoots (amplifies) at the region of medium frequencies. A step input fed into the system even only once will show such a behavior of medium frequency input just for one cycle. Therefore, this system as shown in figure 14.2, shows tendencies of amplification in Bode plot, this amplification is related to the overshoot behavior in the time domain.

Summarizing the above results, the step response and frequency response of a system can be correlated for various parameters as below:

## 14.2 Design with Frequency Response

Once the relationships have been established between the time and the frequency response of the system, a controller is designed using the Bode plots. The controller

## Nomenclature

<b>Low frequencies</b>	Relates the ss-error of step response
<b>Medium frequencies</b>	Relates the overshoot of step response
<b>High frequencies</b>	Relates the speed (rise time and settling time) of step response

## Relation Between Domains

- OL Bode indicates CL time response properties
  - Steady-state performance:
    - Larger gain at DC reduces steady-state error
    - DC gain of a type 0 system can be identified by the magnitude at small frequencies
    - Slope of magnitude at small freq indicates system type
  - Transient performance:
    - Smaller resonant peak and larger phase margin indicate smaller overshoot
    - Larger gain crossover frequency indicates faster response



Figure 14.3: Overview of time and frequency relationships

is design to produce required effects in the time domain because ultimately the time response of a system is achieved in the real world. Such as its peak time (fastness), settling time and ss-error. The following steps can be laid down while designing a controller using frequency response:

1. Analyse time response behavior and determine the parameters that needs to be improved
2. Relate this parameters on the Bode plot
3. Add a controller to change the shape of the frequency response so that it ultimately brings about the changes required in the time domain. The open-loop

(OL) response of the system is considered for plotting Bode plots mainly because of two reasons:

- The OL response of a system determines its CL stability, as the magnitude and the phase shifts are evident in the OL, the CL then used the error generated by this OL response to implement a control.
- The OL Bode plot linearly adds the controller into the magnitude and phase plots as  $G_{OL} = C(s)P(s)$ , taking the log will add the products of the TF as seen in section 13.1.2.1. From the CL TF this linear addition of the controller is not possible as the TF will now become  $G_{CL} = CP/1 + CP$ , therefore, the log will not add the controller linearly anymore.

4. The process of adding control gain and checking with the parameters is iterative mainly for two reasons:

- Magnitude and Phase plots are dependent of each other, changing one changes the response of another
- For non-canonical systems, the system parameters (settling time, overshoot etc.,) cannot be defined analytically, at this point the analysis will only be quantitative, therefore requires an iterative approach.

Consider the system shown in figure 14.1 and 14.2, if we decided that the systems has to be made much faster, reduce its overshoot and improve the ss-error. From the Bode plot figure 14.2, the following changes has to be made:

- Make system faster: Push the magnitude response (gain cross over frequency) in Bode plot to the right
- Reduce the overshoot: Particularly, the system phase margin has to be pushed upwards so that it has larger phase margin. By doing so, the constructive interference will be reduced at the summation point of the feedback in turn reducing the resonance of the system output. Also, the phase margin has to be pushed upwards for the new gain margin that was achieved in earlier step so that the constructive interference is reduced at that point.
- Reduce the ss-error: ss-error is always reduced when the DC gain of the system is large enough (in P control a higher gain is given, I control does this by itself as it adds the pat errors). Essentially, the magnitude response in the Bode plot should be at low frequencies should be given a larger DC gain.

### 14.2.1 Design with a lead compensator

A lead compensator has a TF of a type:

$$C(s) = K \frac{s + z}{s + p} \quad (14.5)$$

## Nomenclature

It has a simple zero and a pole and the zero is much smaller than a pole ( $z \ll p$ ). Therefore, the effect of zero will be dominating (at-least at higher frequencies when pole becomes more dominant). In essence, in a lead compensator, the effect of zero is tried to hit first in the system. In order to appreciate the effect of a lead compensator in the system, plot its Bode plot as given in figure 14.4. The TF of lead compensator is initially written in Bode form as given below:

$$C(s) = K \left( \frac{z}{p} \left( \frac{\frac{1}{z}s + 1}{\frac{1}{p}s + 1} \right) \right) \quad (14.6)$$

For the purpose of generating a plot in Matlab random values of  $1/z$  and  $1/p$  are chosen in this case the values were chosen such that the break frequencies occur at  $\omega = 2\text{rad/s}$  for zero and  $\omega = 3\text{rad/s}$  for the pole (this was done by comparing the plants bode plot).

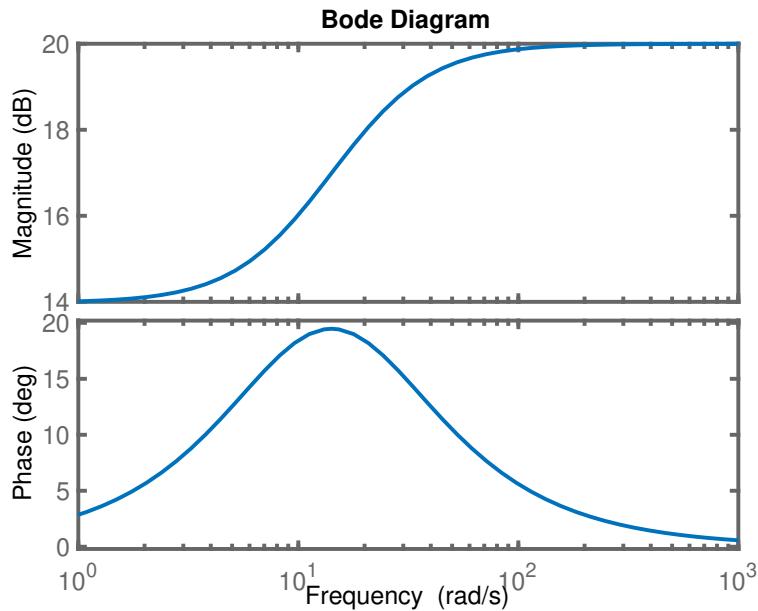


Figure 14.4: Bode plot of the lead compensator

In order to understand the behavior of Lead compensator consider a straight-line approximation Bode plot as given in figure 14.5.

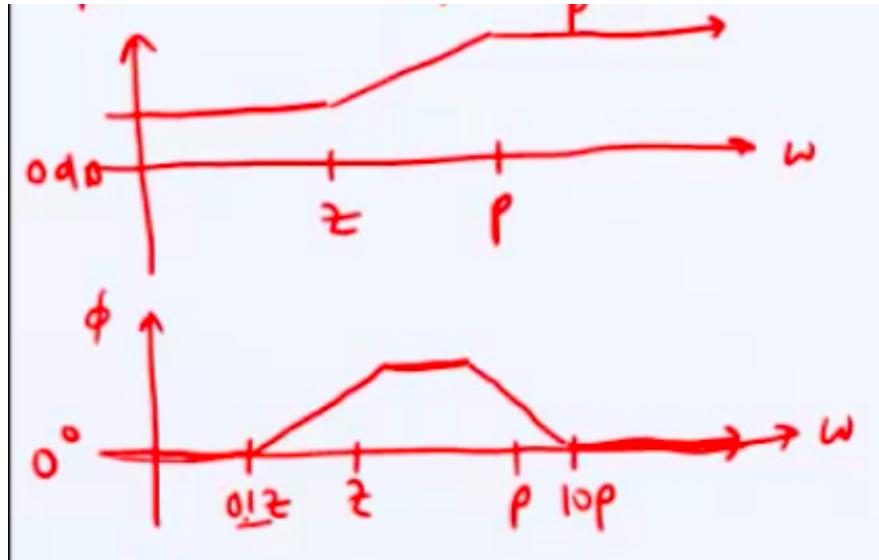


Figure 14.5: Bode plot of the lead compensator

At low frequencies until the effect of zero hits the magnitude plot at  $\omega = 1/z$ , the magnitude is essentially a straight line with value  $20\log K$ . At cut-off frequency  $\omega = 1/z$ , the effect of zero hits in and the magnitude will increase with a slope of  $20dB/decade$ , this should apparently go on increasing until the maximum  $\omega$  but because of effect of pole hitting off at  $\omega = 1/p$ , the magnitude will now be influenced by the effect if pole with a decreasing slope  $-20dB/decade$ . Since the effect if the both the slopes now interact and the overall effect will be to cancel the slope and make the magnitude a flat line after cut-off  $\omega = 1/p$  as shown in figure 14.5.

In case of phase plots, at lower frequencies until the cut-off  $\omega = 0.1z$  (a decade earlier when the slope starts), phase is zero. After which phase starts to increase such that the phase becomes constant of  $90^\circ$  at  $\omega = 10z$  (a decade later). Additionally, since the effect of pole starts to hit off at  $\omega = 0.1p$ , the phase becomes flat even earlier but not at  $\omega = 10z$ . However, after  $\omega = 10z$ , the effect of zero will run out as it should have become a constant line of  $90^\circ$ . The effect of pole to lag the phase continues until it reaches the cut-off  $\omega = 10p$  so a lag in phase is seen further untill the point of  $\omega = 10p$  after which the phase again becomes flat after the effect runs out at  $\omega = 10p$ .

## Nomenclature

Plotting both the Bode plots of both the plant and controller, the following figure 14.6 can be produced.

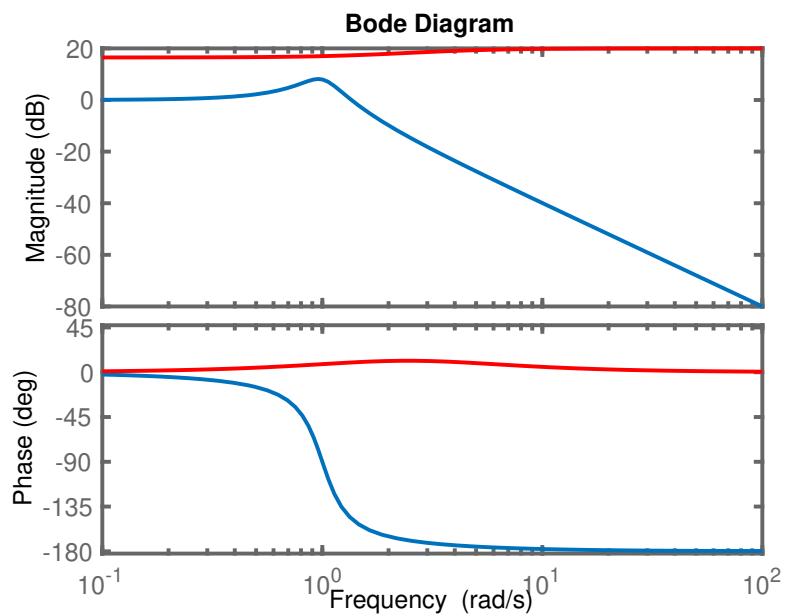


Figure 14.6: Bode plot of the lead compensator and plant

Further the effect of addition of a controller to the plant results in the following FrRe of the system in Bode plot (given by the yellow line) as shown in figure 14.7.

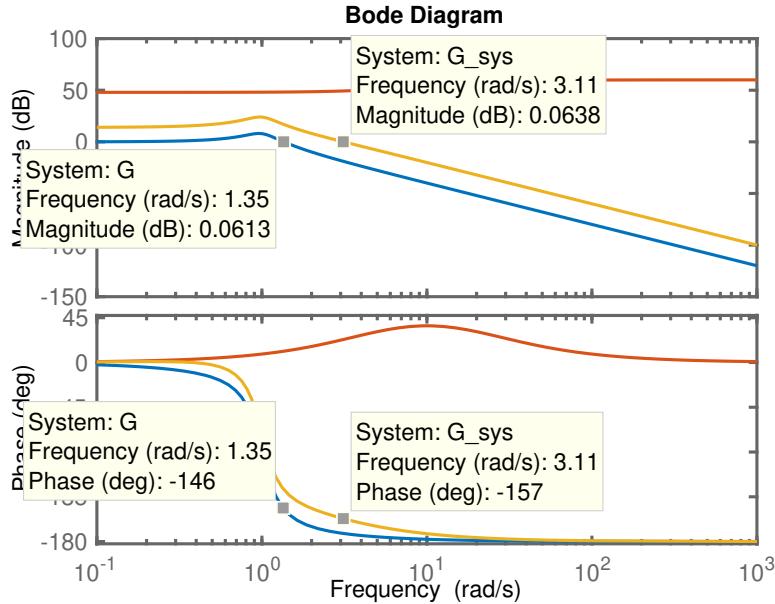


Figure 14.7: Bode plot of plant with LC

It can be seen from figure 14.7, (Yellow line) by using the LC quantitatively the objectives have been achieved by increasing the DC gain, pushing the gain margin has been pushed to the right and the  $P_m$  has been pushed upwards to reduce the constructive resonance which reduces the resonance of the system.

## 14.2.2 Design with a PID controller

### 14.2.2.1 Design with a PD regulator

In case a PD regulator is used in place of a LC, the behavior of the system can be determined using the Bode plot of  $C(s)$ . Writing the  $C(s)$  in the Bode form, the following Bode plots can be generated:

$$C(s) = K_D s + K_P = K(s + z) \implies Kz \left( \frac{1}{z} s + 1 \right) \quad (14.7)$$

where  $z = K_P/K_D$ . In comparison to a LC given by equation (14.5), a PD control add only a zero. Consider the following Bode plot of a PD control as shwon in figure 14.8.

Since a PD control only adds a zero, after the cut-off  $\omega$ , the effect of the zero will kick-in and the MgRe increases with a positive slope of  $20\text{dB}/\text{decade}$ . As there is

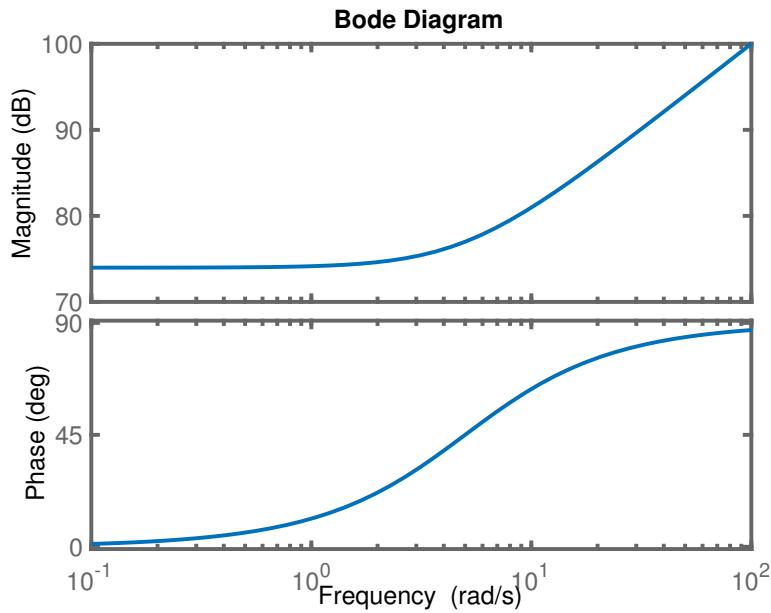


Figure 14.8: Bode plot of plant with LC

no pole, this increase will continue till infinity  $\omega$ . It is evident from figure 14.8, at higher frequencies the MgRe amplifies to infinity and therefore using a PD control will induce the constructive resonance thereby increasing the resonance and making the system unstable. Also, if the input to the PD control is the noise (even though the error is attenuating), it will amplify the magnitude of noise and make the system unstable. In general, noise of high frequency input to the system and a derivative type regulator always amplifies signals that are of high frequency.

### 14.3 Comparison of Frequency based control design with Time based control design

#### 14.3.1 Advantages of Frequency based control design

1. Provides rich source of information starting from a constant DC input to frequencies of any order
2. ss-error information cannot be derived from time based design (pole placements), pole placements would determine the settling time, damping and other dynamic characteristics of the system but do not determine the ss-error. However, in Frequency based design, using Bode plots the MgRe of the system can be determined at any frequency.

3. Settles the ambiguities from root locus (higher-order dynamics, effect of zeros):  
From root locus the effect of poles placed at far left in the Re plane do not show immediate effects in terms of time response, which is clearly visible in the Bode plots. Also, the effects of zeros in the MgRe and phase response is clearly seen in the Bode plots. So FrRe clarifies the effects of each terms (poles and zeros) of any system (lower or higher order systems).
4. Good for experimental derivation (System Identification): Once the Bode plot of a system has been generated, the effects of each of the poles and zeros presents in the system can easily be identified which thereby effect the response properties such as gain cross-over and phase cross-over  $\omega$ ,  $G_m$ ,  $P_m$  and others.

### **Advantages of Time response based control design**

- The time response of a system directly gives the behavior of the systems response in the time domain, which is ultimately required to be controlled. However, in most of the cases when the inputs are vary random with alternating frequencies, a qualitative comparison as perform in section 14.1 has to be done in order to relate the time-domain characteristics with FrRe and the design the controller.

*Nomenclature*

# 15 Introduction to Robust Control

Robustness defines how well a system will perform in presence of:

- Measurement noise: The noise in the system has to be filtered out. This in general needs to be implemented in the control system itself and the control action has to be free from the noises coming from the measurements.
- Disturbances: The control system should work robustly even when the disturbances to the system are random and irregular.
- Model Uncertainty: The control should continue to perform the required action even if there are changes in the system parameters or the system parameters were not correctly defined during the control design. Change in system parameters during operation would include for example a breakage of a rotor blade of a quad-copter.
- Time delays: There are time delays from the signals coming from the sensors to the control, such time delays should also not effect the quality of the control action.

In general, most of the listed characteristic required for implementing a robust control are inter-linked to each other and therefore, one characteristic cannot be improved by reducing another characteristic. For example it is not possible to improve both the noise reduction and effect of disturbances in terms of the control action. Therefore, a trade-off between various characteristics of robustness is always present.

## 15.1 Robustness in a MIMO system

Consider a MIMO system as shown in figure 15.1.

The multiple inputs to the system are the following:

- Reference  $R(s)$
- Disturbance  $D(s)$
- Noise  $N(s)$

## Nomenclature

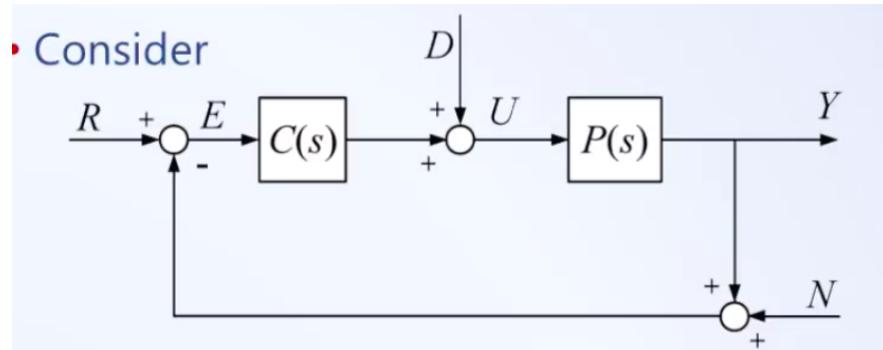


Figure 15.1: MIMO system with noise and disturbance

The effect on the system with each of these inputs can be determined by writing the TF (output/input) of the system with each of these signals as the input. For input with  $R(s)$ , the TF can be expressed as:

$$G_{yr} = \frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} \quad (15.1)$$

For the input with  $D(s)$ , the following TF can be expressed:

$$G_{yd} = \frac{Y(s)}{D(s)} = \frac{P(s)}{1 - (-P(s)C(s))} = \frac{P(s)}{1 + P(s)C(s)} \quad (15.2)$$

with input  $N(s)$ , the TF can be expressed as the following:

$$G_{yn} = \frac{Y(s)}{N(s)} = \frac{-C(s)P(s)}{1 - (-C(s)P(s))} = \frac{-C(s)P(s)}{1 + C(s)P(s)} \quad (15.3)$$

It can be seen that TF's  $G_{yr} = -G_{yn}$ , there is a direct relationship between the TF's with inputs  $R(s)$  and  $N(s)$ . As magnitude attenuation is generally controlled using the control gain  $K$ , changing  $K$  to attenuate  $G_{yn}$  will also attenuate  $G_{yr}$ . Additionally, it can also be proved that

$$G_{yr} = -G_{yn} = \frac{P(s) - G_{yd}(s)}{P(s)}$$

Therefore, all the TF's have a relationship with each other, so changing the system gain  $K$  will effect the changes in all these TF's. This relationship between all the TF's indicate that a trade-off between all these TF's have to be implemented depending on the robustness requirements.

### 15.1.1 Attempting a control design

**Attempt 1:**  $C \rightarrow \infty$

When the control gain  $K$  is increased thereby increasing the control effort  $C(s)$ , the following effects can be seen by each of the inputs. From equation (15.2), the disturbance will go to zeros as the denominator of equation (15.2) will be a large number and  $N_r$  will be a small number. On the other hand equation (15.3) for the noise will tend to  $-1$ . This will cause the noise amplitude to be fully transferred to the system by shifting the phase as well. So  $C \rightarrow \infty$  will not solve the problem.

#### Attempt 2: $C \rightarrow 0$

From equation (15.2),  $G_{yd} \rightarrow P$ . This means that the CL system will act for the disturbance in the same way as a plant would react to the signal of same input frequency. Also, from equation (15.3),  $G_{yn} \rightarrow 0$  (zero divided by any number is zero).

From both these cases, it can be seen that either by attenuating noise or the disturbance, the other quantity gets transferred into the system. So both these quantities are dependent and they cannot be attenuated independently from the system.

### 15.1.2 One solution from frequency response

There is one solution to this problem of dependencies. From FrRe it can be seen that the system behaves differently for different input  $\omega$ . Also, based on the knowledge of input signals  $D(s)$  and  $N(s)$ , it is seen that  $N(s)$  mostly appear in high frequency input and  $D(s)$  as low frequency inputs to the system. Additionally, using the knowledge of attenuation of each  $D(s)$  and  $N(s)$  as described in the previous section, it can be seen that  $D(s)$  attenuates for larger  $C(s)$  and  $N(s)$  attenuates for lower  $C(s)$ . The control strategy considering these two parameters (high gain at low  $\omega$  and vice-versa), can be laid as shown in the following figure 15.2.

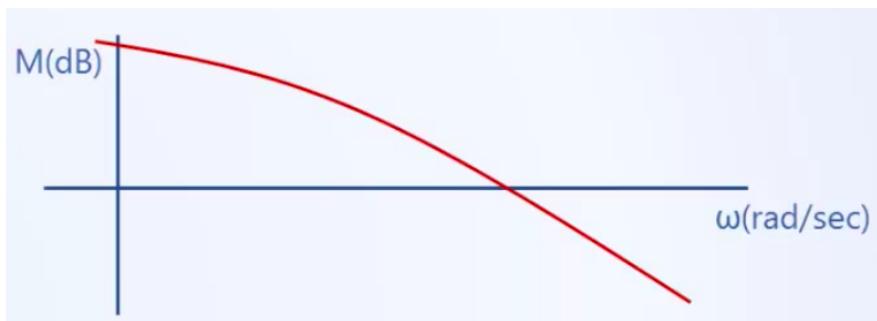


Figure 15.2: Control strategy for noise and disturbance attenuation

Also noting that this kind of control strategy is always desirable. As the system tends to be most uncertain at higher  $\omega$ . Most of the time it is also difficult to record the phase change at higher  $\omega$  among other model uncertainties. Attenuation would be the best possible control strategy at such regions of uncertainties.

## 15.2 Pure Time Delay (Transport Lag)

Another aspect in Robustness apart from Disturbance rejection and Noise canceling is to cope-up with the time-delays. Time delays are inherent in real systems. Systems such as sensors and actuators will have time delays due to the following reasons:

- Discrete Sampling: The signal is read only at the end of the sampling time, so the signal would be old by the sampling time when the controller reads it.
- Computation time: Some of the computations would be time consuming and the controller would be reading the old computed values by the time the computation is completed.
- Network Delays: Most common time delays in automotive applications especially in a CAN bus. Due to this the controller would be reading old values.

### Pure Time Delay (Transport Lag)

- Recall, delay in time domain is multiplication by an exponential in the Laplace domain
$$\mathcal{L}[f(t-T)] = e^{-sT} F(s)$$
s = j\omega
- Using Euler's identity
$$e^{-j\omega T} = \cos \omega T - j \sin \omega T$$
- Therefore,
$$\text{mag} = |e^{-j\omega T}| = \sqrt{(\cos \omega T)^2 + (-\sin \omega T)^2} = 1$$

$$\text{phase} = \angle e^{-j\omega T} = \tan^{-1} \left( \frac{-\sin \omega T}{\cos \omega T} \right) = -\omega T$$

Figure 15.3: Time Delay theory 1

From figure 15.3, it can be seen that the magnitude of the time delay  $T$  is always 1, a unit magnitude. However, its only the phase that changes w.r.t to the frequency of the input signal  $\omega$  and the frequency lags as  $\omega$  increases as  $\phi = -\omega T$ . Therefore, in Bode plot, time-delay can be added as a block with magnitude of 1 and  $\phi = -\omega T$ .

Adding the magnitude and phase due to time-delay on the Bode plot, the new characteristic of the system can be determined. (Consider figure 15.4) This is done simply

by adding the Bode plots of the system without delay (Blue line) and with delay. The red line is the result of adding Bode plot of the system without delay and with delay. The time delay with a constant magnitude  $20\log 1 = 0dB$  and phase that varies as per  $\phi = -\omega T$  (converted to degrees,  $T$  is time-delay in s) are plotted in the OL TF of the Bode plot as

$$OL = T(s)P(s)$$

$$\angle OL = \angle T(s) + \angle P(s)$$

, where  $T(s)$  is the block (TF) like a control block  $C(s)$  which is connected in series to the plant  $P(s)$ . The result is as shown in figure 15.4.

## Pure Time Delay (Transport Lag)

- Therefore, time delay doesn't affect the magnitude plot, but does add phase lag (tends to destabilize a system)
- Converting the phase to degrees,

$$\angle e^{-j\omega T} = -57.3\omega T$$

### Example

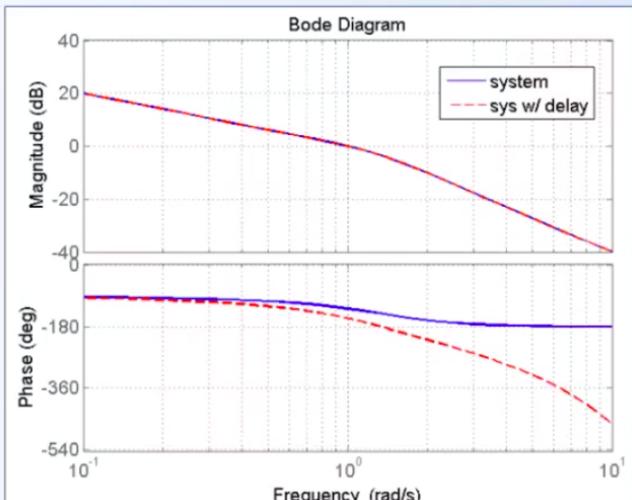


Figure 15.4: Time Delay theory 2

*Nomenclature*

# 16 Fundamentals of Electric circuits

In this chapter, a method is defined that is used to develop mathematical models to capture the dynamic behavior of electrical systems.

## 16.1 Basics of Electricity

A particle can either have a positive charge or a negative charge. Like charges repel and unlike charges attract each other. This is because charged particles creates an electric field as shown in figure 16.1 and an electric field in turn exerts a force on charged particles as shown in figure 16.2.

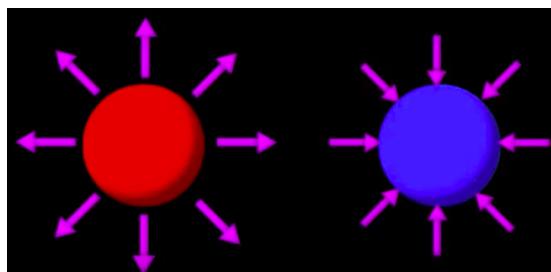


Figure 16.1: Charged particles creates electric fields

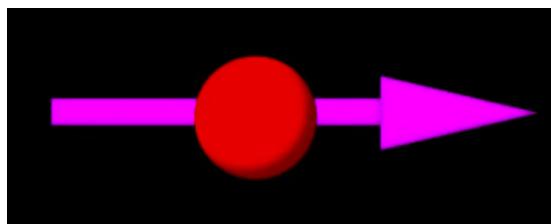


Figure 16.2: Electric fields exert force on charged particles

When there is an electric field between two points, it is said that there is a *voltage* between these two points. A voltage difference in practice creates this electric field between two points in space through use of devices such as batteries. The electric field then exerts a force on the charged particles that move around because of it. The number of flow of the charged particles is called the *current* in the circuit.

## Nomenclature

A voltage difference between any two points essentially acts like a potential difference due to height in a gravitational field. Where the height will be the voltage and the gravitational field will be electric field. With one difference that in case of electric circuits higher voltage difference will cause a stronger electric field and vice-versa. The current is a flow-rate of charged particles and therefore can be denoted by  $I = dq/dt = \dot{q}$ . Voltage is the potential energy every charged particle and is denoted using either  $V$  or  $e$  in this notes. Therefore, if we need to get the total power from the circuit the charge flow-rate can be multiplied with the energy of the each charge ( $e$ ).

$$P = \dot{q}e$$

**Tip:** The circuit already has the charged particles in it, a battery only creates an electric field (by imposing voltage between two points) which pushes the charged particles along the circuit. Also in the olden times, current was thought to be the flow of positive charged particles, however, it has been found that it is the electrons that flow in the circuit.

Only a moving charged particle creates a **magnetic field**. And the magnetic field is exerted only on other moving charged particles. A charged particle moving in a loop creates a magnetic field that loops from north pole to the south pole of the charged particle which is same as a single electric charge spinning. A **magnet** is an object where all such spinning charged particles are all moving collectively in the same direction. Therefore, creating a magnetic field that loops from north pole to the south pole. The property of a charged particle to create both electric and magnetic fields is called **electro-magnetic** property, this principle is used in design of electro-magnetic sensors and actuators (DC Motor).

## 16.2 Modeling Electrical System

Similar to mechanical systems, electrical systems are of three basic types:

1. Resistance elements – Dampers elements
2. Capacitance elements – Spring elements
3. Inductance elements – Inertia elements

### Resistance Elements

Resistance elements are modeled using Ohm's law which states that the voltage drop across the resistor as follows (here current is denoted using  $i$ ):

$$e = iR \tag{16.1}$$

### Capacitance element

The voltage across a capacitor is expressed using the definition of capacitance, which says that the capacitance of a capacitor is equal to the charge stored by the capacitor per unit of voltage supplied to it. Therefore, capacitance  $C$  can be defined as:

$$C = \frac{q}{e} \quad (16.2)$$

the voltage equation can then be written as:

$$e = \frac{1}{C}q \quad (16.3)$$

Capacitor stores potential energy just like a spring, when voltage is applied to capacitor it holds a charge inside which is then released to the circuit after removing the voltage across the capacitor.

### **Inductance element**

An inductor acts like an inertia elements of a mechanical system. As inertia of a object due to its mass or shape resists a change in the direction of the motion, here a magnetic field does this to an electrical circuit. Magnetic fields are generated in an electrical circuit when there is a change (magnitude or direction) in electric field. These magnetic fields acts like an inertia to the change in electric field. An electric in this sense is the magnitude of current  $i$  and its direction giving in essence the nature of electric field. Similar to a rotating mass which tries to maintain the rotation in the same magnitude and direction. A magnetic field which will be generated again when the current  $i$  in the circuit changes its magnitude or direction. This magnetic field will then provide resistance to the electric field (current  $i$  and its direction) to be changed. Also noting that whenever there is no change in the electric field, the magnetic field is not generated and the inductor will just behave like a wire allowing all the current to flow through it.

Inductance as its is related to a phenomenon like inertia in mechanical systems is something which stores kinetic energy of the electric circuit. The voltage across an inductor is expressed as a function of change in  $i$ :

$$e = L \frac{di}{dt} \quad (16.4)$$

The above equation can be verified using two observations:

- When there is a change in  $i$ , the voltage drop occurs as the function of the change (slope  $di/dt$ )
- When there is no change (slope = 0), then voltage drop across inductor is zero and it behaves just like a wire

## 16.3 Principles of electrical systems

Once the fundamental principle of basic forms of electrical components has been established, the established laws of electrical circuits are used to model the behavior of the dynamical system. There laws are called **Kirchhoff law's** which in essence are like Newton's laws required to model the dynamics of mechanical system. There are two law's of Kirchhoff for electrical circuits:

### 16.3.1 Kirchhoff's Laws

**Current Law** States that the sum of currents  $i$  entering any node should be exactly equal to the sum of  $i$ 's leaving that node.

**Loop Law** States that the sum of voltages around the loop should be zero

Using these two laws any electric circuit can be modeled. In case of electric circuits it is rather quite simple to model then as there are only two laws that govern their dynamic behavior. Unlike in mechanical systems there are laws differently stated and for liner and rotational motions, further getting complex when there is addition of constraints into the system.

### 16.3.2 Modeling electrical circuits

#### Resistors in series

The voltage applied across resistors in series drops after every resistor as it is a potential energy of the charge which drops off at different levels of energy states. At each energy state that is in this case a resistor the voltage drop across all such elements can be described using Ohm's law such that  $e_i = iR_i$ . Further, applying Kirchhoff's laws, as the current entering the node should be equal to the current leaving the node, the current across every resistor is same. This is also evident from Ohms law  $e = iR$ , as  $R$  changes, voltage drop  $e$  changes accordingly and  $i$  is a constant in this equation.

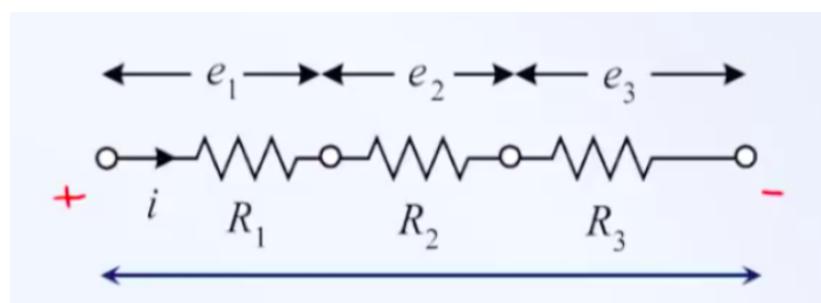


Figure 16.3: Resistors in series

Only using the first law, the dynamic behavior can be modeled as:

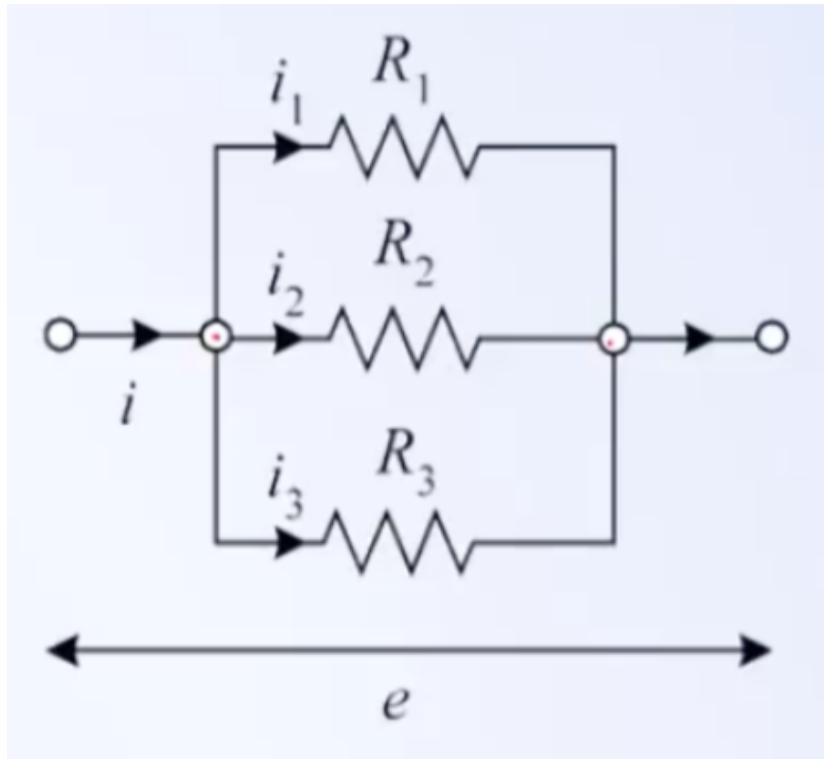


Figure 16.4: Resistors in parallel

$$e = iR_1 + iR_2 + iR_3 = i(R_1 + R_2 + R_3) \quad (16.5)$$

### Resistors in parallel

Only using the first law the system can be modeled again as with resistors connected with series, the current splits at the node where the connections split into three. Therefore, now the voltage drop across different resistors is given by  $i_1R_1$ ,  $i_2R_2$  and  $i_3R_3$ . However, the voltage across each resistor is same, therefore the system can be modeled as follows:

$$\sum i = i_1 + i_2 + i_3 = \frac{e}{R_1} + \frac{e}{R_2} + \frac{e}{R_3} \quad (16.6)$$

### 16.3.3 Modeling DC converter

There are two types of DC converters, voltage amplifiers (boost converters) and voltage attenuators (buck converters). The boost converter is as shown in figure ???. There are two states in the circuit, ON state and OFF state. The idea is to use the flow of the current in the circuit such a way that when the current flows into the battery that needs to be charged at higher voltage gets this higher voltage due to the flow of current in the circuit. In ON states when the switch is ON, the current only flows in the first loop through the inductor, as inductor will try to hold the current in the circuit when

## Nomenclature

the switch in is the OFF state, the second loop is activated where now the initial current  $i \neq 0$ . The task is to find out using Kirchhoff's law for circuits whether the voltage has been stepped up due to such non zero initial current into the circuit.

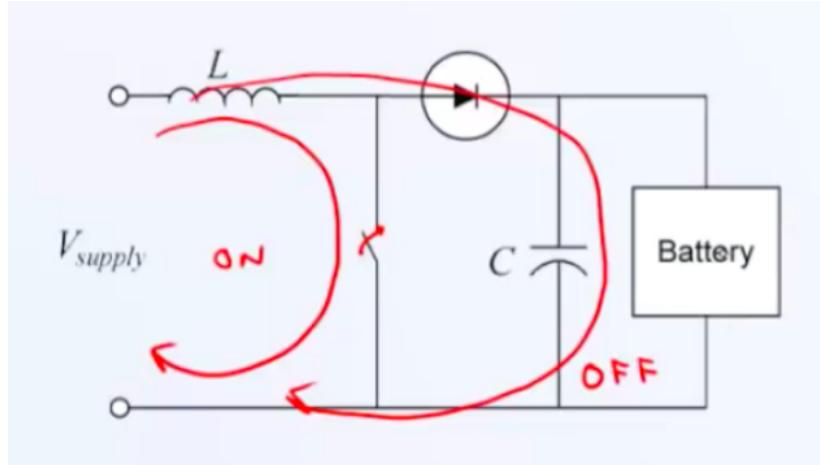


Figure 16.5: Schematic of a boost converter

### ON State

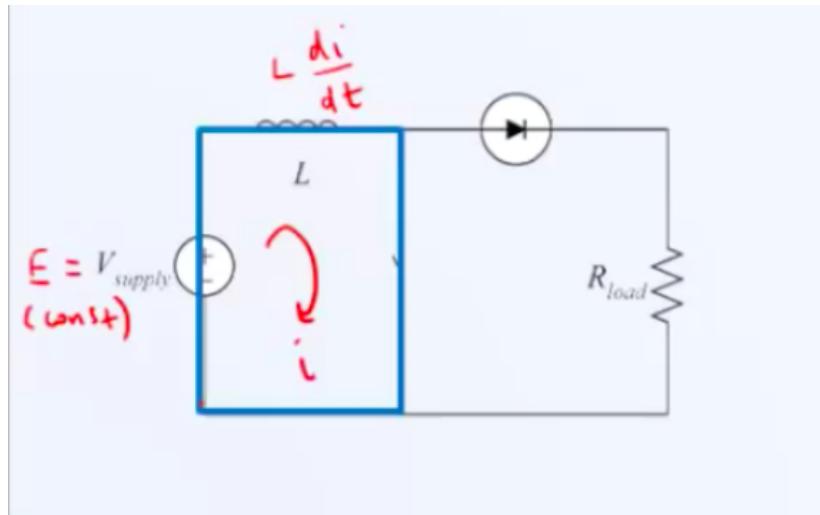


Figure 16.6: Boost converter CKT in ON mode

Using second law and applying at a point (generally any point can be chosen in the ckt):

$$V_{\text{supply}} - L \frac{di}{dt} = 0 \quad (16.7)$$

Applying LT for the above equation:

$$\frac{E}{s} - L(sI(s) - i(0)) = 0 \quad (16.8)$$

As  $i(0) = 0$ , the initial condition,

$$\frac{V}{s} - L(sI(s)) = 0$$

the current  $I(s)$  is:

$$I(s) = \frac{E}{L} \frac{1}{s^2} \quad (16.9)$$

inverse LT on the above equation:

$$i = \frac{E}{L}t, \quad t \geq 0 \quad (16.10)$$

### OFF state

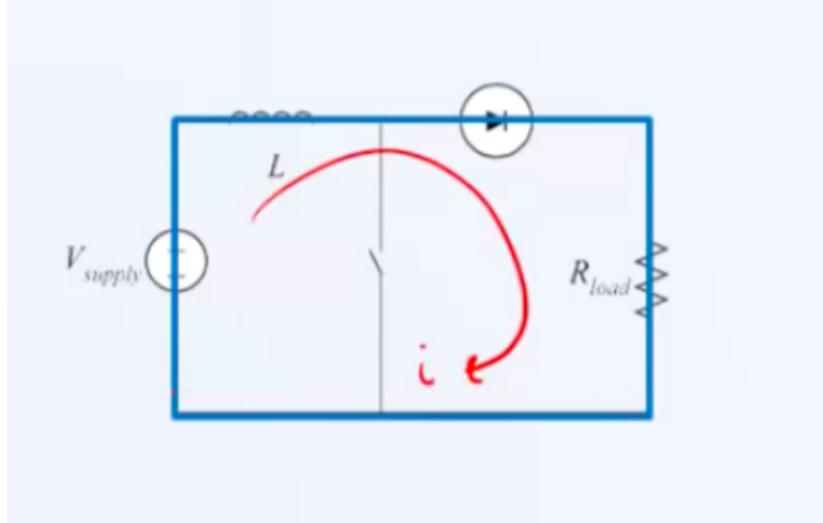


Figure 16.7: Boost converter CKT in OFF mode

$$V_{supply} - L \frac{di}{dt} - R_{load}i = 0 \quad (16.11)$$

applying LT:

$$\frac{E}{s} - L(sI(s) - i(0)) - R_{load}I(s) = 0 \quad (16.12)$$

the current in the circuit is then expressed as:

$$I(s) = \frac{E}{s(sL + R_{load})} + \frac{Li(0)}{sL + R} \quad (16.13)$$

Also noting the voltage step up by solving the equation of OFF state in terms of  $E$ :

$$E = \frac{I(s)}{L}s^2 + I(s)R_{load}s - Li(0)s \quad (16.14)$$

## Nomenclature

It can be seen from equation 16.14, there are three terms out of which the first two terms contribute to the voltage drop across the ckt elements and the third term which actually contributes to the voltage step-up due to its negative sign. And the amount of voltage that it adds is the quantity of the previous current value  $i(0)$ .

From the current equation (16.13), solving the equation using partial fractions and taking inverse Laplace, the following form can be obtained:

$$i(t) = \frac{E}{R_{load}} - \frac{E}{R_{load}}e^{-\frac{R_{load}}{L}t} + i(0)e^{-\frac{R_{load}}{L}t}, \quad t \geq 0 \quad (16.15)$$

Therefore, from equations (16.10) and (16.15), the circuit current during ON and OFF states can be summarized as follows:

### ON State

$$i = \frac{E}{L}t, \quad t \geq 0$$

### OFF State

$$i(t) = \frac{E}{R_{load}} + \left( i(t_s) - \frac{E}{R_{load}} \right) e^{-\frac{R_{load}}{L}t}$$

During the ON state the current built up in the circuit which helps to step-up the voltage from the supply voltage as described in the above two equations. The behavior of which can be described using the plots as shown in figure 16.8

When the this ON and OFF states are switched at a certain frequency, the following output current at the battery charging can be observed as shown in figure 16.9

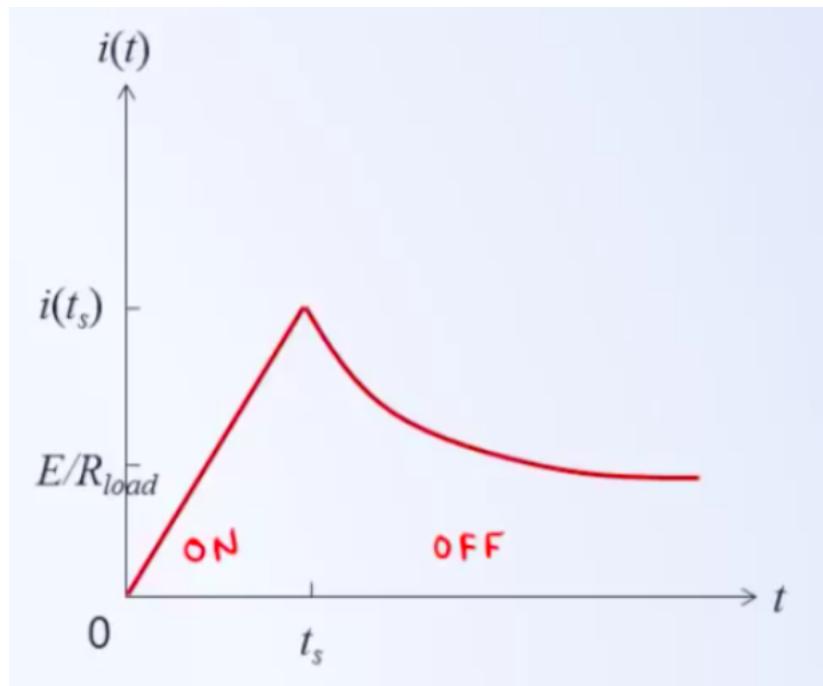


Figure 16.8: Performance of boost converter

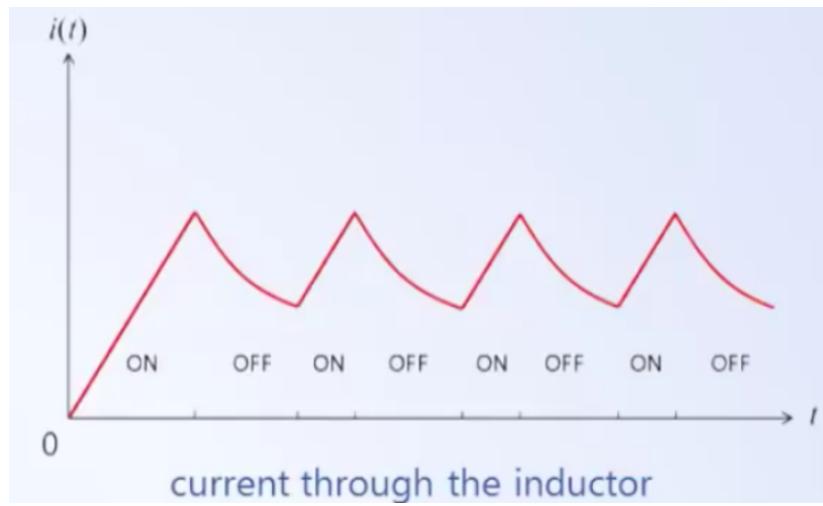


Figure 16.9: Charging frequency of the boost converter

The number of charging cycles can be seen as shown in figure 16.10

The variations seen in the charging cycles are generally smoothed out using a capacitor which acts like a low pass filter providing a smooth voltage across the battery.

## Nomenclature

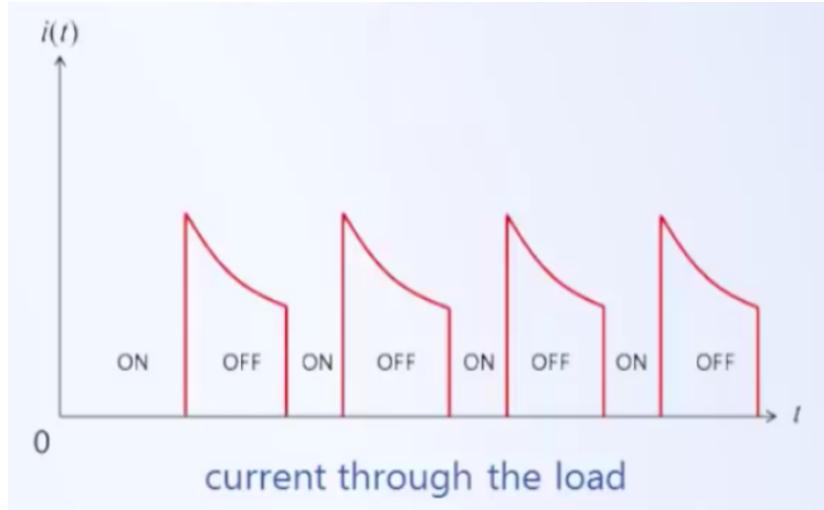


Figure 16.10: Boost converter charging cycles

## 16.4 Modeling circuits in Laplace Domain

While modeling circuits in Laplace domain, a quantity can be defined using the ratio between the input voltage and the output current and this quantity is called the complex impedance as described in the following equation:

$$Z(s) = \frac{E(s)}{I(s)} \quad (16.16)$$

Complex impedance is treated like resistance and therefore can be defined using Ohm's law and using impedance equivalent circuits can be described in series and parallel like we used in resistors. However, impedance is a property of an electrical element that varies with frequency and for each of the electrical element its complex impedance can be described as follows.

### Resistor

$$\mathcal{L}[e = iR] \implies E(s) = I(s)R \implies Z(s) = \frac{E(s)}{I(s)} = R \quad (16.17)$$

In case of resistor impedance is a constant at any input frequencies.

### Capactor

$$\mathcal{L}[e = \frac{1}{C} \int idt] \implies E(s) = \frac{1}{C} \frac{I(s)}{s} \implies Z(s) = \frac{E(s)}{I(s)} = \frac{1}{Cs} \quad (16.18)$$

Therefore, in case of capacitors the impedance reduces at higher frequencies.

### Inductors

$$\mathcal{L}[e = L \frac{di}{dt}] \implies E(s) = LsI(s) \implies Z(s) = \frac{E(s)}{I(s)} = sL \quad (16.19)$$

Therefore, in case of inductors the impedance increases with the increasing frequency.

### 16.4.1 TF example of voltage divider

Consider the following circuit as shown in figure 16.11. In this example a method using TF is used to model the circuit instead of time domain modeling done previously using Kirchhoff's Laws.

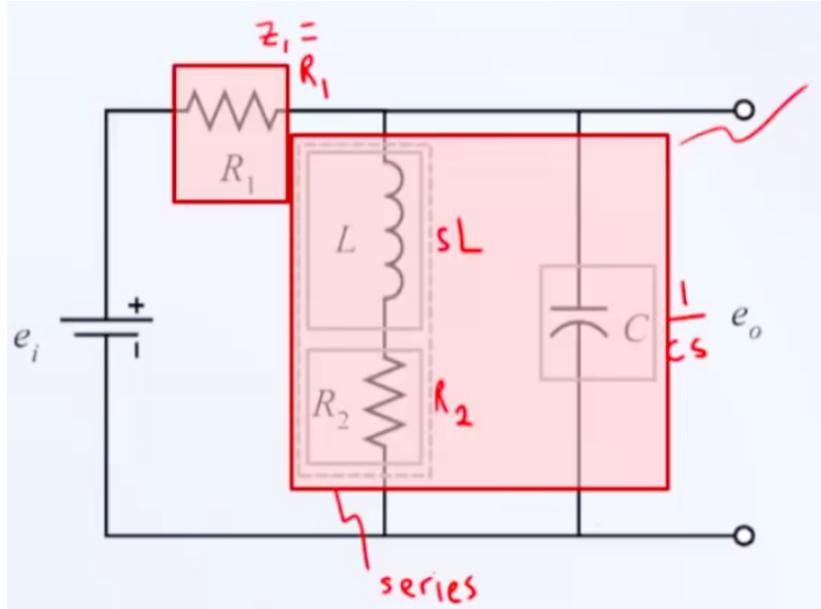


Figure 16.11: Example for circuit modeling using TF

The two components  $L$  and  $R_2$  are in series, as impedance has the same rules as resistors the impedance of these two components can be added up as

$$sL + R_2$$

further the components in series are parallel to the capacitor, therefore using resistors in parallel formula and calling the whole thing as  $Z_2$  as:

$$Z_2 = \left[ \frac{1}{sL + R_2} + \frac{1}{1/Cs} \right]^{-1}$$

Solving the above equation leads to

$$Z_2 = \frac{sL + R_2}{R_1 CL s^2 + (CR_1 R_2 + L)s + R_1 + R_2}$$

Now the system can be seen as a two component system as shown in figure 16.12  
Using the figure 16.12, the TF of the system can be written as described, the output voltage is only the voltage across  $Z_2$ , therefore, the output voltage is defined using resistance law as

$$E_0(s) = I(s)Z_2(s)$$

## Nomenclature

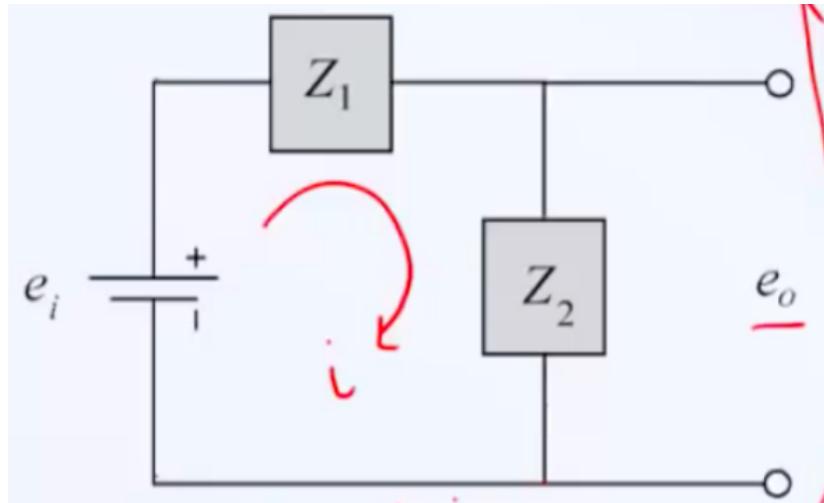


Figure 16.12: System modeled in the form representing a voltage divider

similarly, the input voltage is the voltage across the two components \$Z\_1\$ and \$Z\_2\$ and is described as:

$$E_i(s) = Z_1(s)I(s) + Z_2(s)I(s)$$

\$I(s)\$ cancels out from the equation and substituting \$Z\_1\$ and \$Z\_2\$ in the TF and solving the TF leads to:

$$G(s) = \frac{sL + R_2}{R_1CLs^2 + (CR_1R_2 + L)s + R_1 + R_2} \quad (16.20)$$

# **17 Advanced control architectures**

## **17.1 Two degrees of freedom control**

*Nomenclature*

## **Part III**

# **Model Based Design of Control Systems**



# 18 Introduction

## 18.1 Adaptive cruise control example

- Specification: Maintain constant velocity against road gradients
- Control Signal: Throttle angle
- Disturbance: Road-gradient
- Output: Velocity

The relationship between control input  $u$  and system output  $v$  can be modeled using Newtons laws as described below:

$$\frac{dv}{dt} = -\frac{a}{m}v + \frac{b}{m} \quad (18.1)$$

Adding disturbance to plant model given by equation (18.1) the following system model can be described:

$$\frac{dv}{dt} = -\frac{a}{m}v + \frac{b}{m} - d \quad (18.2)$$

where  $d = g\sin(\alpha)$  and  $\alpha$  is the road-gradient.

### 18.1.1 Open-loop control design

The system model given by equation (18.1) can be expressed in LT from as:

$$G(s) = \frac{V(s)}{U(s)} = \frac{b/m}{s + a/m} \quad (18.3)$$

In order to determine the OL control input, an assumption is made such that  $\alpha = 0$ , the control input is then:

$$0 = -\frac{a}{m}v + \frac{b}{m} \implies u = \frac{a}{b}r \quad (18.4)$$

where  $r = v$ , as velocity is supposed to follow the reference.  $u$  is LT will be:

$$U(s) = \frac{a}{b}R(s) \quad (18.5)$$

## Nomenclature

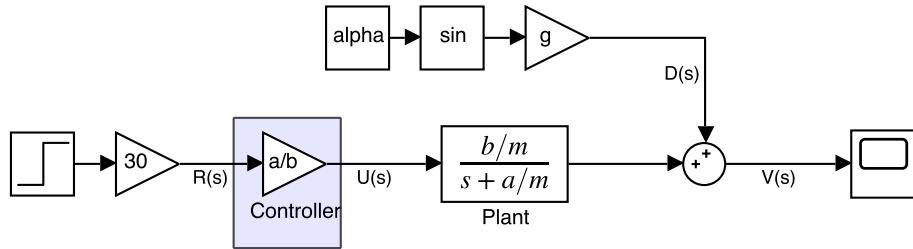


Figure 18.1: ACC in OL control

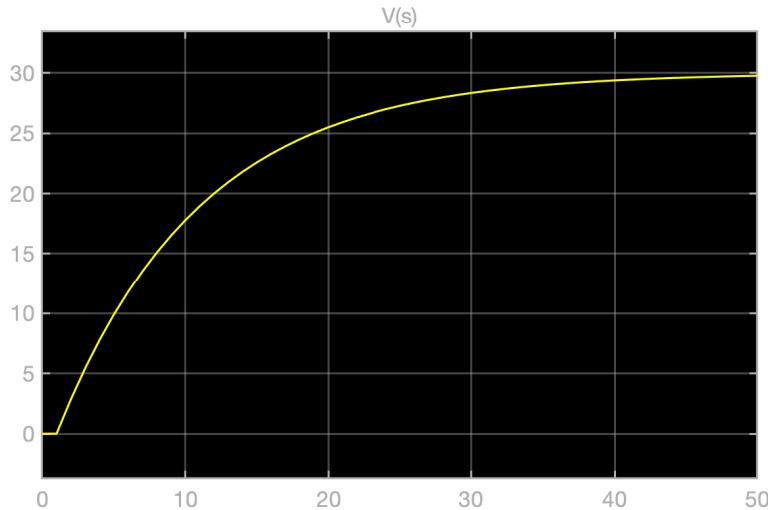


Figure 18.2: ACC in OL control

As described in section 9.1.1, the OL control signal is inverse of the plane  $P(s)$ , from equation (18.5) it can be seen that at steady-state  $s \rightarrow 0$ ,  $P(s) = b/a$ . The system model with control input is as shown in figure 18.1. The response of the system shown in figure 18.1 is shown in figure 18.2. It can be seen that for no  $D(s)$ , the system follows  $R(s)$ . Generally, OL system cannot track changes in road-gradient and changes in system parameters, since the control signal given by (18.5) eliminates system parameters, the control has become robust in my case for any change in road-gradient and/or the system parameter  $m$ . However, if other system parameters ( $a$ ,  $b$ ) are not predefined in the control, the system would not follow the reference properly. Also note that from figure 18.2, the system is quite sluggish, as it is very slow to settle for a given input  $R(s)$ . These two problems could be solved using a feedback controller.

## 18.2 Model Based Control Design general methodology

Control design technology based on Model Based Design is very important in Automotive development. Some of the examples are Power-train control, Engine control, ABS and more advanced controls used in Autonomous driving such as Lane assist, Tracking etc.. The following general methodology is used in control system design in Automotive applications:

1. Control objectives (specification)
  - a) Asks questions on what should be established using a control
  - b) Requires knowledge on system dynamics as well as disturbances, so that a valid performance could be achieved and not fight the system against its own
  - c) Kinds of control objectives:
    - i. **Qualitative objectives:** Objectives that are listed as some kind of improvements that are not exactly specified (achieving as good results as possible) such as some improvement in fuel-consumption, minimize battery energy consumption etc.,
    - ii. **Quantitative objectives:** Objectives that are well-defined such as achieving desired response times
2. Description of system or a plant
  - a) Asks questions on level of system abstraction required (component level, subsystem level or plant level etc.,)
  - b) Modeling can be done in various ways such as mathematical or measurements (black-box) etc.,
3. Controller Design
  - a) Up until plant model, the system is on the HW side, from controller design onward, the SW side of the system developments starts
  - b) Asks questions on type of control design such as OL or CL etc.,
  - c) Several design methods are employed such as classical methods (PID) or modern methods using state-space methods
  - d) Choosing control parameters is most important which will determine the performance of the system. mostly methods such as trial and error, design methods from control theory and optimization methods are employed
4. Analyze the performance: To check if the desired results are achieved, methods that are used are:
  - a) Analysis

## *Nomenclature*

- b) Simulation
- c) Experiments (Testing)

## **18.3 Control design methods**

### *Classical control design (PID)*

- 1. Works well for simple systems
- 2. Do not require good mathematical knowledge of the systems as the controller parameters can be designed using trial and error methods
- 3. A good control theory knowledge can also be helpful for better design but are rather intensive
- 4. Due to their large trial and error iterations this method is not ideal for system with multiple input and outputs (MIMO)

### *State-space method*

- 1. can easily handle large scale systems (MIMO)
- 2. tuning can be formed as an optimization problem and are available as systematic tools that are pre-developed
- 3. they are also easier to implement
- 4. on the other hand state-space control design requires good mathematical knowledge

## **18.4 State-space control design methodology**

State-space methods for a control design is a very powerful technique for a control design because they are very systematic and not based on theoretical intuition required in classical control design. This method is best suited for large scale systems such as an Automotive but are seldom taught in a formalized way in most of the control courses.

A state-space is a dynamic modeling of a system using first order differential or difference equations. The state-space methodology can be visualized using the figure 18.3.

The what of state-space model is the description of the system dynamics using first order differential or difference equations, here state variables are used that essentially are sufficient to describe the evolution of the system. They are regarded as the memory

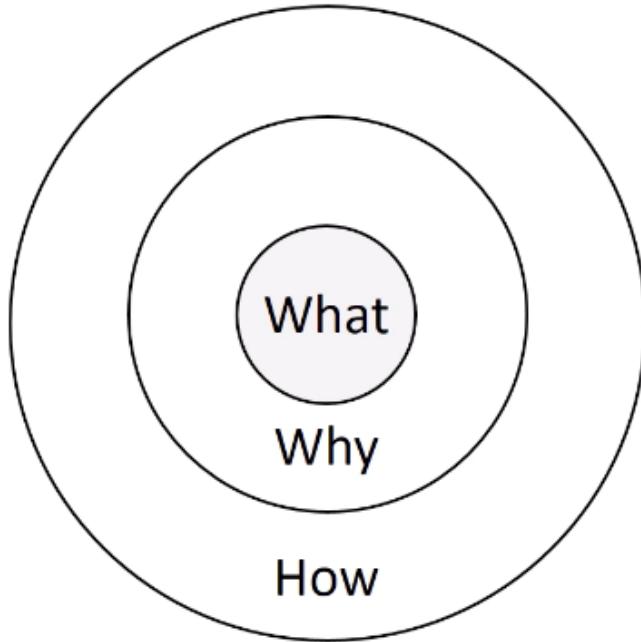


Figure 18.3: Visualizing state space modeling

of the system which remembers the past and acts based on it in the present and the future. A state-space form is expressed as:

$$\dot{x}(t) = f(x(t), u(t)) \quad (18.6)$$

$$y(t) = h(x(t), u(t)) \quad (18.7)$$

A special case of state-space form is autonomous (as described above) and linear system which is expressed as linear combinations of state vectors and input vectors as expressed below:

$$\dot{x} = Ax + Bu \quad (18.8)$$

$$y = Cx + Du \quad (18.9)$$

The why part describes why should one use state-space modeling approach:

1. They are natural for modeling physical systems
2. are easy to use due to matrix format
3. also makes it great to use for MIMO systems (which are hard to use with LT's)
4. easy to implement for ex: the control equation  $u = Lx + Kr$  is a one line code which contains simple vector or matrix algebra

## *Nomenclature*

The how part deals with how the state space model can be implemented from development phase to implementation phase, the general method is as follows:

1. Modeling
2. Analysis (For system knowledge)
3. Controller design (State Feedback controller)
4. Observer / Estimator design: To measure the required states used for state feedback controller. Generally, in a system measuring required states is not direct and easy but with state space method there are tools available that makes it possible to do so

# 19 State space representation

*This chapter deals with modeling linear systems using state space approach and linearization techniques for nonlinear systems.*

LTI systems can be expressed in state space form in the following way:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (19.1)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (19.2)$$

Here, matrix  $B$  is due to the control variables and depends on the actuators used to actuate various states of the system. Matrix  $B$  is also called control matrix or actuator matrix. Similarly, since matrix  $C$  is used to sense the required states, matrix  $C$  is called the sensor matrix and  $D$  is called direct or feedthrough matrix.

If the matrices  $A$ ,  $B$ ,  $C$  and  $D$  are constants and do not depend on time, then it is equivalent to the coefficients of linear differential equation (6.10) being constants. The system in this case is called Linear and time-invariant system.

## 19.1 Laplace transform from state-space model

Given a LTI system of the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (19.3)$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \quad (19.4)$$

Taking LT of equation (19.3)

$$\mathcal{L}\{\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}\} \implies s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s) \quad (19.5)$$

$$\mathcal{L}\{\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}\} \implies \mathbf{Y}(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s) \quad (19.6)$$

which gives,

$$(sI - A)\mathbf{X}(s) = \mathbf{B}\mathbf{U}(s) + \mathbf{x}(0) \implies \mathbf{X}(s) = (sI - A)^{-1}\mathbf{B}\mathbf{U}(s) + (sI - A)^{-1}\mathbf{x}(0) \quad (19.7)$$

$$\mathbf{Y}(s) = [C(sI - A)^{-1}B + D]\mathbf{U}(s) + C(sI - A)^{-1}\mathbf{x}(0) \quad (19.8)$$

Note that in equation (19.8) is the solution of the differential equation given by (19.3), the first term is the particular solution and second term is the null solution. The part of

## Nomenclature

the first term  $C(sI - A)^{-1}B + D = G(s)$  is also the TF of the system, given that the initial state  $x(0) = 0$  such that:

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D$$

## 19.2 Solution of State-space model

As described in section 6.3 Linear differential equation (LDE) written in scalar form

$$\dot{x} = ax + bu \quad (19.9)$$

the null solution, particular solution, and the complete solution are given as described by equations (6.31), (6.36) and (6.37) respectively also given here below for reference  
Null Solution:

$$x_n = x_0 e^{at} \quad (19.10)$$

Particular Solution using convolutional integral:

$$x_p = \int_0^t g(t - \tau) u(\tau) d\tau = \int_0^t b e^{a(t-\tau)} u(\tau) d\tau \quad (19.11)$$

Solution:

$$x(t) = x_n(t) + x_p(t) = x_0 e^{at} + \int_0^t b e^{a(t-\tau)} u(\tau) d\tau \quad (19.12)$$

Similarly, from the above results for a scalar equation, the solution is found similarly for a matrix equation like a StSp. For a matrix equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (19.13)$$

Null Solution analogous to equation 6.31

$$\mathbf{x}_n = x_0 e^{\mathbf{A}t} \quad (19.14)$$

Particular solution analogous to equation 6.36

$$\mathbf{x}_p = \int_0^t \mathbf{B} e^{\mathbf{A}(t-\tau)} u(\tau) d\tau \quad (19.15)$$

Solution of StSp (19.3) analogous to equation (6.37)

$$\mathbf{x}(t) = \mathbf{x}_n(t) + \mathbf{x}_p(t) = x_0 e^{\mathbf{A}t} + \int_0^t \mathbf{B} e^{\mathbf{A}(t-\tau)} u(\tau) d\tau \quad (19.16)$$

Also including output signal

$$\mathbf{y}(t) = \mathbf{C} e^{\mathbf{A}t} x_0 + \int_0^t \mathbf{B} e^{\mathbf{A}(t-\tau)} u(\tau) d\tau + \mathbf{D} u(t) \quad (19.17)$$

## 19.3 Transition Matrix or Matrix exponential

The special term  $e^{\mathbf{A}t}$  is an exponential matrix which can in general be calculated using infinite time series expansion as described in section 6.1 of Euler's Identity. However, this term can also be determined by comparing already developed mathematics so far. By comparing the solutions from Laplace transform (19.8) and solution from LDE (19.17), it can be seen that in both the equations the coefficients  $e^{\mathbf{A}t}$  and  $(sI - A)^{-1}$  are always same. Therefore  $e^{\mathbf{A}t}$  in time domain can be calculated using inverse Laplace

$$e^{\mathbf{A}t} = \mathcal{L}^{-1}\{(sI - A)^{-1}\} \quad (19.18)$$

## 19.4 Poles and zeros and Eigenvalues of State-space

From equation (19.8), the transfer function of the state-space model was determined to be the term due to the input signal

$$G(s) = C(sI - A)^{-1}B + D \quad (19.19)$$

So, in order to determine the TF, the inverse of  $(sI - A)$  should be calculated, this however, is rather difficult and hence an alternative proof from mathematics is used which calculates the TF in whole

$$G(s) = C(sI - A)^{-1}B + D = \frac{\det\begin{pmatrix} sI - A & -B \\ C & D \end{pmatrix}}{\det(sI - A)} \quad (19.20)$$

These determinants are actually the characteristic equations for the system, so the poles of the system are determined for the values of denominator that makes the denominator go to zero

$$\det(sI - A) = 0 \quad (19.21)$$

Similarly, zeros can be found from the solutions of numerator

$$\det\begin{pmatrix} sI - A & -B \\ C & D \end{pmatrix} = 0 \quad (19.22)$$

Furthermore, it can be seen that finding the poles using  $\det(sI - A) = 0$  is same as finding the Eigen values of  $\mathbf{A}$  which is found using  $\det(\lambda I - A) = 0$ , in which case only the  $s$  term is replaced with a  $\lambda$  term. Therefore, it can be concluded that since Eigen values determine the stability of the system, it is only sufficient to determine the Eigen values of  $\mathbf{A}$  using  $\det(\lambda I - A) = 0$  which would in turn describe the stability of the system.

Similar to stability of LDE using real values of  $\sigma$ , it can be said that the StSp model is asymptotically stable iff  $\lambda_i < 0$ . In all other cases the system is not asymptotically stable.

## 19.5 State-space transformations

A State-space representation is not unique. That is, for a given set of state variables chosen there is always a relationship with other state-variables that could have been chosen to represent the system model. In order to establish this transformation of the system model from one state to another state, consider a new state variable  $z$  such that  $x$  can be transformed to  $z$  by using a transformation matrix  $\mathbf{T}$

$$\mathbf{x} = \mathbf{T}\mathbf{z} \quad (19.23)$$

If  $\mathbf{T}$  is invertible, then the new state-space model can be expressed with the new state variable  $z$  using

$$\dot{\mathbf{z}} = \mathbf{T}^{-1}(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}) = \mathbf{T}^{-1}(\mathbf{A}\mathbf{T}\mathbf{z} + \mathbf{B}\mathbf{u}) = \mathbf{T}^{-1}\mathbf{A}\mathbf{T}\mathbf{z} + \mathbf{T}^{-1}\mathbf{B}\mathbf{u} = \tilde{\mathbf{A}}\mathbf{z} + \tilde{\mathbf{B}}\mathbf{u} \quad (19.24)$$

and new response matrix

$$\mathbf{y} = \mathbf{C}\mathbf{T}\mathbf{z} + \mathbf{D}\mathbf{u} = \tilde{\mathbf{C}}\mathbf{z} + \tilde{\mathbf{D}}\mathbf{u} \quad (19.25)$$

## 19.6 State space for a 2D particle

Consider a particle  $p$ , which can translate along two coordinates  $x$  and  $y$  and the motion is actuated due to the control inputs  $u_x$  and  $u_y$  which accelerate the particle along  $x$  and  $y$  coordinates, such that  $\ddot{p}_x = u_x$  and  $\ddot{p}_y = u_y$ . The state space model can be formulated as given below:

First states are recognized with variables:

$$\begin{aligned} x_1 &= p_X \\ x_2 &= \dot{p}_x \\ x_3 &= p_y \\ x_4 &= \dot{p}_y \end{aligned}$$

We have two control inputs, therefore, two variable can be recognized:

$$\begin{aligned} u_1 &= u_x \\ u_2 &= u_y \end{aligned}$$

There are two output variable:

$$y_1 = p_x$$

$$y_2 = p_y$$

The dynamics of the state variables is given by  $\dot{\mathbf{x}}$ :

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_2 \\ u_1 \\ x_3 \\ u_2 \end{bmatrix}$$

with states of the system  $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4]^T$ , the  $A$  and  $B$  matrices can be written as:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

Such that  $\dot{\mathbf{x}}$  can be written in the form  $Ax + Bu$  in matrix form.

Similarly, the output equation can be written in state space using sensor matrix  $C$ ,  $\mathbf{y}$  has to be written such that  $\mathbf{y} = [x_1 \ x_3]^T$ :

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

## 19.7 Dimension check for matrices [EgeND, p.3.1.6]

if  $x \in \mathbb{R}^n \implies \mathbf{A} : n \times n$

if  $u \in \mathbb{R}^m \implies \mathbf{B} : n \times m$

if  $y \in \mathbb{R}^p \implies \mathbf{C} : p \times n$

It is important to see that the size of the matrix matches the algebraic equations  $\mathbf{x} = Ax + Bu$  and  $\mathbf{y} = Cx$ .

Also,

$$x : n \times 1$$

$$u : m \times 1$$

$$y : p \times 1$$

Remembering the matrix dimensions in state space would help write the matrices more easily and also ensures the validity of the written matrices.

*Nomenclature*

# 20 Modeling Automotive Systems

With reference to three phase modeling approach described in section 7.1 consider modeling of following systems.

## 20.1 Longitudinal vehicle dynamics model

**Phase I:** The whole of the vehicle can be treated as subsystem of following with respective inputs and outputs

1. Powertrain { Pedal position – Wheel traction forces }
2. Chassis { Wheel traction forces – Vehicle velocity }



Figure 20.1: Subsystems of entire vehicle model

In order to reduce modeling complexity, the above system can be further reduced into following subsystems

1. Powertrain { Pedal position – Wheel traction forces }
  - a) Engine { Pedal position – Engine torque }
  - b) Gear box { Engine Torque – Gear box torque }
  - c) Wheel { Gear box torque – Wheel traction forces }
2. Chassis { Wheel traction forces – Vehicle velocity }

## Nomenclature

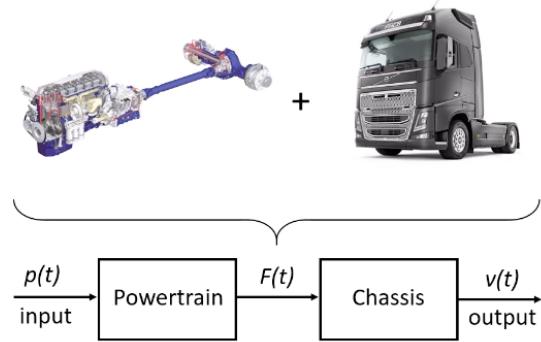


Figure 20.2: Subsystems of entire vehicle model

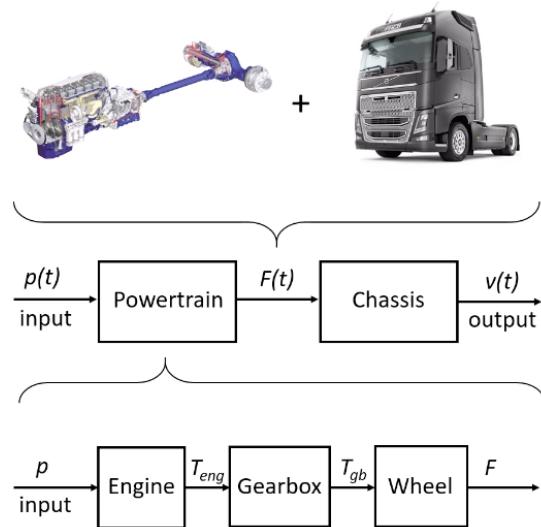


Figure 20.3: Further Subsystems of vehicle model

**Phase II:** The following mathematical models for following subsystems are required

1. Engine
2. Gearbox
3. Wheel
4. Chassis

For modeling engine, an assumption (approximation) is made such that the engine torque developed can be expressed using first order system response as

$$\dot{T}_{eng} = -\frac{1}{\tau}T_{eng} + \frac{K}{\tau}u \quad (20.1)$$

where,  $\tau$  is time-constant of first order systems,  $K$  is system gain and  $u$  is system input.

The gearbox model is assumed to be a fixed gear ratio model with gear ratio  $i$

$$T_{gb} = iT_{eng} \quad (20.2)$$

The wheel model converts the torques to ground traction forces

$$F = \frac{T_{gb}}{r_w} \quad (20.3)$$

which can be further expressed using equations (20.1) and (20.2)

$$\dot{F} = -\frac{1}{\tau}F + \frac{Ki}{\tau r_w}u \quad (20.4)$$

For modeling the chassis, conservation laws can be used from Newton's

$$\sum_i F_i = ma \quad (20.5)$$

further using free body diagram using FBD, the following model can be expressed

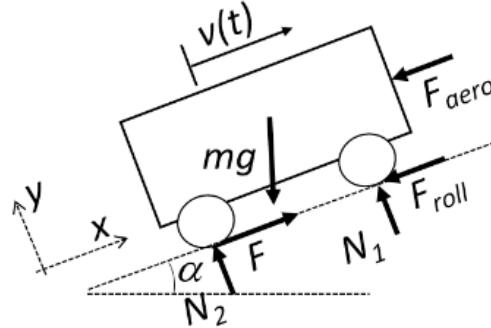


Figure 20.4: Free body diagram chassis model

$$x : ma = F - F_{roll} - F_{Aero} - F_{grad} \quad (20.6)$$

$$y : 0 = N_1 + N_2 - mg \cos \alpha \quad (20.7)$$

Using constitutive relationships

$$a = \frac{dv}{dt} \quad (20.8)$$

$$F_{roll} = mg f \cos \alpha \quad (20.9)$$

$$F_{Aero} = 0.5 \rho C_d A_f v^2 \quad (20.10)$$

$$F_{grad} = m g \sin \alpha \quad (20.11)$$

## Nomenclature

**Phase III:** From the equations developed in Phase II, it can be seen that there is a time derivative term for velocity and force has time derivatives associated with them. So choosing  $F$  and  $v$  as the state variables (There is another constitutive relationship  $v = dy/dt$ , but here the output will be written in terms of  $v$  which will be the required state space. Therefore, ignoring this state variable)

From the states chosen writing the state-space model. Start from  $F$  using equation (20.4)

$$\frac{dF}{dt} = -\frac{1}{\tau}F + \frac{Ki}{\tau r_w}u \quad (20.12)$$

this equation is already in the form  $\dot{x} = f(x, u, d)$ . Further for state  $v$ , using equation (20.6)

$$ma = F - F_{roll} - F_{Aero} - F_{grad} \quad (20.13)$$

the above equation is not in the form  $\dot{x} = f(x, u, d)$ , therefore using constitutive relations expressed in phase II

$$\frac{dv}{dt} = \frac{1}{m} \{ F - mgf \cos\alpha - 0.5\rho C_d A_f v^2 - mgsin\alpha \} \quad (20.14)$$

the above equation is now in the form  $\dot{x} = f(x, u, d)$ , this completes the state-space model. Further for the sake of simplicity new variable names can be used (entirely optional step)

$$x_1 = F \quad (20.15)$$

$$x_2 = v \quad (20.16)$$

which leads to

$$\dot{x}_1 = -\frac{1}{\tau}x_1 + \frac{Ki}{\tau r_w}u \quad (20.17)$$

$$\dot{x}_2 = \frac{1}{m} \{ x_1 - mgf \cos\alpha - 0.5\rho C_d A_f x_2^2 - mgsin\alpha \} \quad (20.18)$$

## 20.2 Modeling Drivetrain

The simplified drivetrain model as described in section 20.1 considers the entire drive train dynamics as a first order response to the step control input. This model could be used in case when simulating vehicle scenarios at higher speeds. At lower speeds when the engine torque is high, the drive shaft and gear box shafts gets twisted and due to this oscillations are generally observed in the torque transmitting driveshaft. A controller can be used to avoid these oscillations so that a smooth torque be delivered to wheel shaft. In order to do so firstly, a model describing this dynamic behavior is captured used Modeling techniques in more detail.

Following the three phase modeling method as described in section 7.1,

### 20.2.1 Phase I

: Since a more detailed dynamics of the system is now required to be captured, a subsystem model is now more detailed with most of the dynamic parts included into separate subsystems as shown in figure 20.5 This is a rotatory system, therefore, Newton's

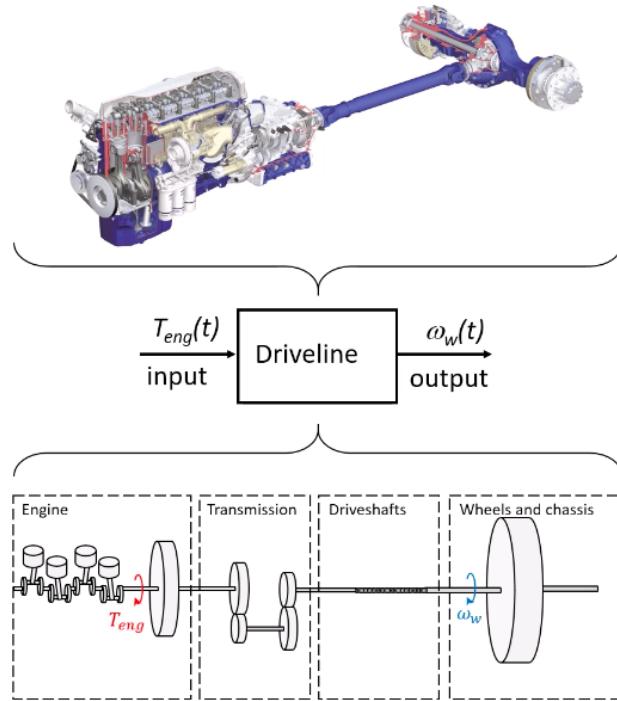


Figure 20.5: Detailed subsystems of drivetrain

law on rotating bodies is used. Using equation of angular momentum as expressed in Prelims in equation (6.65), the sum total of all the torques acting on the subsystem is expressed as the time-derivative of equation (6.65).

$$\sum_i \tau = I \dot{\omega} \quad (20.19)$$

The term from  $\mathbf{v}_{A/O} \times \mathbf{P}_{/O}$  equation (6.65) goes to zero as the distance vector  $\mathbf{v}_{A/O} = 0$  or both the vectors  $\mathbf{v}_{A/O}$  and  $\mathbf{P}_{/O}$  are parallel.

### 20.2.2 Phase II

: Drawing FBD's for the drivetrain subsystem given in figure 20.5 as shown in figure 20.6 While connecting these subsystems the following internal variables indicated in red comes to picture, also consider the following assumptions that are made for simplification

## Nomenclature

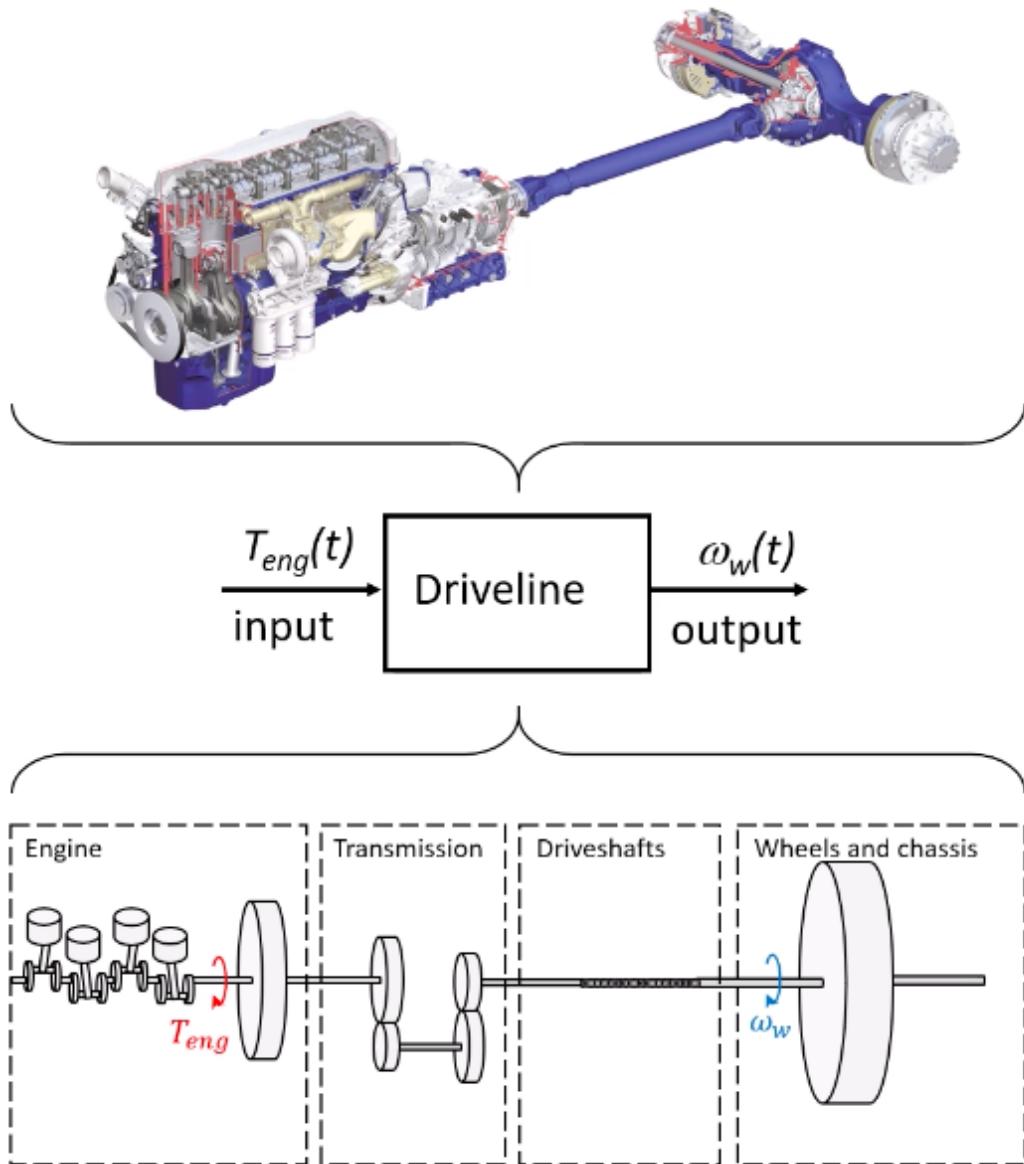


Figure 20.6: FBD of subsystems of drivetrain

1. Subsystem Engine: Inertia is modeled for the flywheel
2. Subsystem Transmission: Inertia is considered negligible and subsystem is modeled as gear ratio between engine shaft and the transmission shaft
3. Subsystem Driveshaft: Again inertia is neglected and subsystem is modeled as spring-damper system in parallel in order to include the first order dynamics of the shaft for an input torque
4. Subsystem Wheels: Inertia is modeled as a lumped parameter

5. The load at the end  $T_{load}$  can be determined using Longitudinal model (reverse dynamics)

### 20.2.2.1 Equation for engine

For the conservation law, there are two torques that are acting at the inertial flywheel, one from applied engine torque and one as resistance torque from the transmission. With no constitutive relationships (while ignoring friction)

$$\mathbf{I}\dot{\omega} = T_{eng} - T_{tr1} \quad (20.20)$$

### 20.2.2.2 Equations for transmission

The conservation law for transmission is simple as there is no inertia, the term  $I\dot{\omega} = 0$ . Therefore, the input torque should be equal to the output torque. Additionally, since the output torque gets multiplied by the transmission ratio, this should be included at the output torque

$$0 = T_{tr2} - iT_{tr1} \quad (20.21)$$

the constitutive relation is the speed reduction from the gear

$$\omega_{eng} = i\omega_{tr} \quad (20.22)$$

### 20.2.2.3 Equations for drivetrain

For the conservation law in this case, as there is no inertia element, this subsystem also does not add any additional torques at the output, so the input torque should be equal to the output torque

$$0 = T_{ds} - T_{tr2} \quad (20.23)$$

the constitutive relationship for this subsystem describes how the output torque changes dynamically due to spring-damper system like behavior of drivetrain

$$T_{ds} = c_s \left( \int (\omega_{tr} - \omega_w) dt \right) + d_s(\omega_{tr} - \omega_w) \quad (20.24)$$

### 20.2.2.4 Equations for wheel and chassis

The conservation law is again from Newtons law, here there is an inertia element, so there some dynamic loading of the output torque

$$\mathbf{I}_c \dot{\omega}_w = T_{ds} - T_{load} \quad (20.25)$$

the inertia term here has inertia due to wheels rotation and the inertia on the rotating mass due to the translating chassis, which is fixed to it

$$\mathbf{I}_c = I_{wheel} + mr_{wheel}^2 \quad (20.26)$$

where  $m$  is the mass of chassis

## Nomenclature

### 20.2.3 Phase III

In order to choose state variable, it can be seen from equations developed in phase II, there are two time-derivatives of  $\omega_{eng}$  and  $\omega_w$ , further there is also an integral of these two components  $\int(\omega_{tr} - \omega_w)dt$ . This term is actually, the torsion of the drivetrain and called here  $\varphi$ . Therefore, 3 states variables have been identified.

Also note that the driveshaft torque can now be written as a function of torsion

$$T_{ds} = c_s\varphi + d_s(\omega_{tr} - \omega_w) \quad (20.27)$$

Lets start from state  $\omega_{eng}$

$$\dot{\omega}_{eng} = \frac{1}{I}(T_{eng} - T_{tr1}) \quad (20.28)$$

there is inputs in this equation but there are no state variables, using equation (20.21)

$$\dot{\omega}_{eng} = \frac{1}{I}(T_{eng} - \frac{T_{r2}}{i}) \quad (20.29)$$

continuing with equations (20.23) and (20.24)

$$\dot{\omega}_{eng} = \frac{1}{I}(T_{eng} - \frac{1}{i}(c_s\varphi + d_s(\omega_{tr} - \omega_w))) \quad (20.30)$$

finally with equation (20.22)

$$\dot{\omega}_{eng} = \frac{1}{I}(T_{eng} - \frac{1}{i}(c_s\varphi + d_s(\frac{\omega_{eng}}{i} - \omega_w))) \quad (20.31)$$

similarly, for state  $\omega_w$

$$\begin{aligned} \dot{\omega}_w &= \frac{1}{I_c}(T_{ds} - T_{load}) \\ &= \frac{1}{I_c}(c_s\varphi + d_s(\omega_{tr} - \omega_w) - T_{load}) \\ \dot{\omega}_w &= \frac{1}{I_c}(c_s\varphi + d_s(\frac{\omega_{eng}}{i} - \omega_w) - T_{load}) \end{aligned} \quad (20.32)$$

Finally, for state  $\varphi$ , the derivative is available directly

$$\dot{\varphi} = \frac{\omega_{eng}}{i} - \omega_w \quad (20.33)$$

the output is the wheel speed state variable

$$y = \omega_w \quad (20.34)$$

Optionally, the state-space model can be written using new variables  $x_1 = \omega_{eng}$ ,  $x_2 = \omega_w$  and  $x_3 = \varphi$ , load  $T_{load}$  is considered a disturbance to the system

$$\dot{x}_1 = \frac{1}{I}(u - \frac{1}{i}(c_s\varphi + d_s(\frac{x_1}{i} - x_2))) \quad (20.35)$$

$$\dot{x}_2 = \frac{1}{I_c} (c_s \varphi + d_s (\frac{x_1}{i} - x_2) - d) \quad (20.36)$$

$$\dot{x}_3 = \frac{x_1}{i} - x_2 \quad (20.37)$$

The above state-space form is linear, therefore it can be written in matrix form

$$\dot{x} = Ax + Bu + Hd \quad (20.38)$$

$$y = Cx + Du \quad (20.39)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} \frac{-d_s}{I_c^2} & \frac{d_s}{I_c i} & \frac{-c_s}{I_c i} \\ \frac{d_s}{I_c i} & \frac{-d_s}{I_c} & \frac{c_s}{I_c} \\ \frac{1}{i} & -1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} \frac{1}{I} \\ 0 \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{-1}{I_c} \\ 0 \end{bmatrix} d \quad (20.40)$$

$$y = [0 \quad 1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [0]u \quad (20.41)$$

## 20.3 Modeling suspension system

### 20.3.0.1 Phase I

For modeling the suspension system, a quarter car model is used as shown in figure 20.7

## Nomenclature

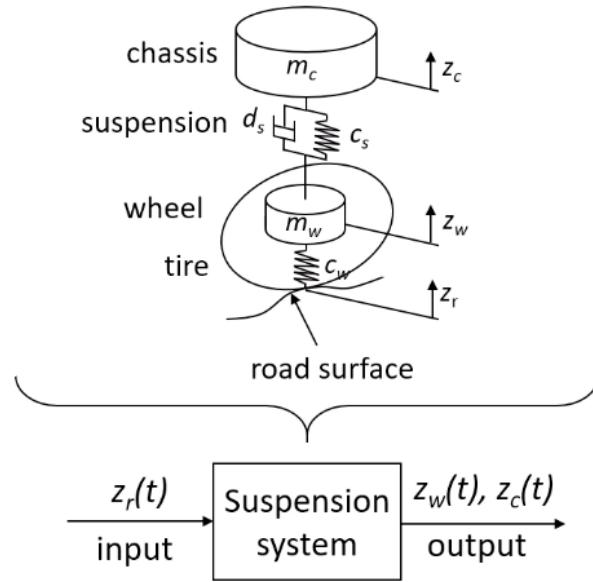


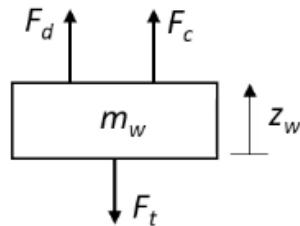
Figure 20.7: Suspension model

### 20.3.0.2 Phase II

Using FBD's as shwon in figure 20.8

#### Free body diagram:

Wheel:



Chassis:

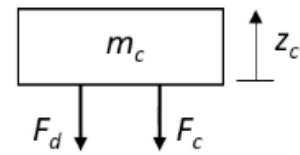


Figure 20.8: Suspension model

For modeling wheel, using conservation law's (Newtons Law), a force balance can be used

$$m_w a_w = F_c + F_d - F_t \quad (20.42)$$

For modeling chassis, again using conservation law on force balance

$$m_c a_c = -F_d - F_c \quad (20.43)$$

Constitutive relationships:

### 1. acceleration and velocity

$$a_w = \frac{dv_w}{dt} = \dot{v}_w \quad (20.44)$$

$$a_c = \frac{dv_c}{dt} = \dot{v}_c \quad (20.45)$$

$$v_w = \frac{dz_w}{dt} = \dot{z}_w \quad (20.46)$$

$$v_c = \frac{dz_c}{dt} = \dot{z}_c \quad (20.47)$$

(20.48)

### 2. spring and damper elements – using Hooke's law

$$F_t = c_w(z_w - z_r) \quad \text{Tire force} \quad (20.49)$$

$$F_c = c_s(z_c - z_w) \quad \text{Suspension spring} \quad (20.50)$$

$$F_d = d_s(v_c - v_w) \quad \text{Suspension damper} \quad (20.51)$$

(20.52)

#### 20.3.0.3 Phase III

From phase II it can be seen that there are four time derivatives, so choosing states as

$$\{z_w, v_w, z_c, v_c\}$$

the state-space model can be expressed with new variables

$$\{x_1 = z_w, x_2 = v_w, x_3 = z_c, x_4 = v_c\}$$

as (the model is linear, so it can be written in the Matrix form)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-(c_w+c_s)}{m_w} & \frac{-d_s}{m_w} & \frac{c_s}{m_w} & \frac{d_s}{m_w} \\ 0 & 0 & 0 & 1 \\ \frac{c_s}{m_c} & \frac{d_s}{m_c} & \frac{-c_s}{m_c} & \frac{-d_s}{m_c} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{c_w}{m_w} \\ 0 \\ 0 \end{bmatrix} u \quad (20.53)$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + [0 \ 0] u \quad (20.54)$$

If the dynamics of comfort is controlled by only designing the parameters of spring and damping of the suspension it is called ***Passive Suspension***, however, if an actuator is used to control the comfort dynamics against the disturbance from road then it is called ***Active Suspension***

## 20.4 Single track vehicle model

This is the model required to control the motion of the vehicle. This model defines the lateral dynamics of the vehicle when it is steered.

### 20.4.1 Modeling assumptions

1. Only planar motion is considered ie., along x, y directions and orientation  $\psi$
2. Pitch, roll and vertical motions are not included, however, the vertical motion of the vehicle can be studied using the suspension model. For including pitch and roll an additional 2 DOF (about the pitch and roll axis) model has to be constructed.

There are two kinds of planar motion models used to describe and control the vehicles motion

1. Two track model (4 Wheeled vehicle model with steering and lateral dynamics)
2. Single track model (Bicycle model)
  - a) Linear single track model
  - b) Nonlinear bicycle model (nonlinear Tyre characteristics among others)

### 20.4.2 Phase I

In single track model, the 4 wheeled vehicle model is limited to having one steerable wheel and one rear wheel. Since it is a lateral dynamic model, the model input is steering and output is the lateral motion described by  $v_y$  and  $\psi$  as shown in figure 20.9

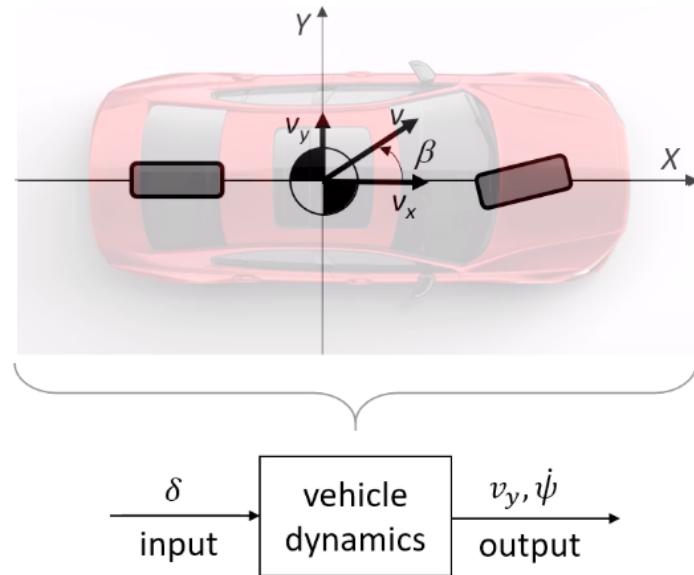


Figure 20.9: Single track model

### 20.4.3 Phase II

As this is a mechanical system, to describe the dynamics, consider the FBD as shown in figure 20.10

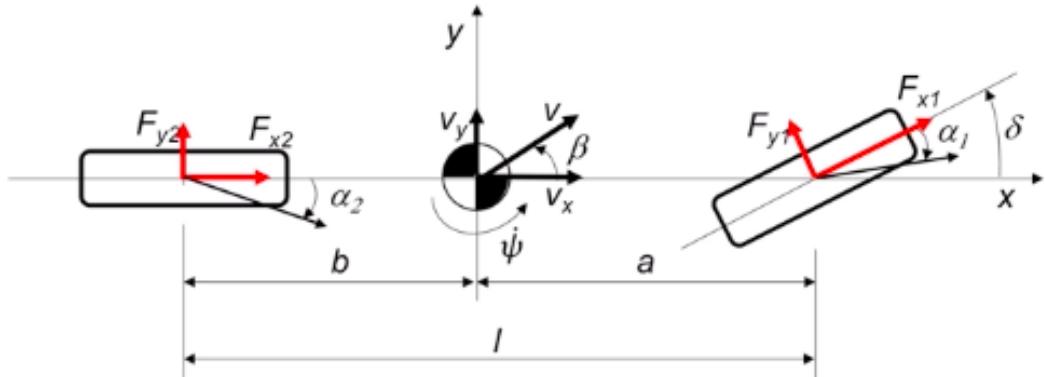


Figure 20.10: Single track model

$F_i$  are the longitudinal and lateral forces in the individual wheel frames.  $\alpha_1$  and  $\alpha_2$  are the front and rear wheel slip angles, i.e., the angles that the wheel propagation velocities make with the actual bicycles propagation (this is how the required centrifugal force develops).  $\beta$  is the angle that the propagation velocity makes with the bicycles frame. Here, the component velocities  $V_x$  and  $V_y$  are in the bicycles frame, in the sim-

## Nomenclature

plified case shown in figure 20.10, this is also the frame of the rear wheel.  $\delta$  is the steering angle with reference to bicycles frame.

Additionally, also note that the lateral forces  $F_{y1}$  and  $F_{y2}$  shown in figure 20.10 are given by lateral tire forces which is described in section 20.4.3.

There are both forces and moments in the FBD, in order to capture both of them, we have to consider both force balance and moment balance from the conservation laws.

### Force Balance

Forces are in both x and y directions, therefore, deriving forces balance equations both in x and y. Expressing forces in bicycles frame, the forces in the steering wheel are resolved along horizontal and vertical directions

$$x : \sum ma_x = F_{x1}\cos\delta + F_{x2} - F_{y1}\sin\delta \quad (20.55)$$

$$y : \sum ma_y = F_{x1}\sin\delta + F_{y2} + F_{y1}\cos\delta \quad (20.56)$$

### Moments Balance

balancing moments about z-axis:

$$\sum J\ddot{\psi} = F_{x1}\sin\delta a + F_{y1}\cos\delta a - F_{y2}b \quad (20.57)$$

### Constitutive relationships

The longitudinal and lateral forces  $\{F_x, F_y\}$  generated by the vehicle comes from the tire forces. These tire forces are nonlinear and are described popularly using Pacejka's tire models. Consider the lateral force generation given in Pacejka as described in [Ahm18, p.29] as a function of a quantity called slip angle. Lateral slip is defined as the angle between the direction of the center of the tires propagating versus the direction to which the tires common plane is pointing. It is due to this lateral or side slip a force is produced parallel to the wheel's axle. Lateral slip is expressed as [Ahm18, p.31]

$$\alpha = \tan^{-1} \left( \frac{v_y}{v_x} \right) \quad (20.58)$$

where  $v_x$  and  $v_y$  are the propagation velocity at resolved along x and y directions. Further, the tires side slip angle also depends on the steering angle  $\delta$ , so it can be re-written as

$$\alpha_1 = \delta - \tan^{-1} \left( \frac{v_y + a\dot{\psi}}{v_x} \right) \quad (20.59)$$

the lateral slip angle for the rear wheel is only due to the propagation velocity

$$\alpha_2 = -\tan^{-1} \left( \frac{v_y - b\dot{\psi}}{v_x} \right) \quad (20.60)$$

Using these lateral slip angles, the lateral tire forces generated are expressed using a relationship:

$$F_{lat} = -c_i\alpha_i \quad (20.61)$$

where  $C_\alpha$  is tire stiffness coefficient [ $N/m$ ] which is dependent of tire material properties.

Further, the external forces and moments described in section 20.4.3 need to be associated with the associated accelerations as per  $F = ma$ . Consider the following bicycle (N) and wheel frames (B) for this purpose:

$$\begin{aligned} N &: \{\hat{I}, \hat{J}, \hat{K}\} \\ B &: \{\hat{i}, \hat{j}, \hat{k}\} \end{aligned}$$

Consider, the accelerations of individual wheel contact points with reference to inertial frame N,

$$\begin{aligned} {}^B v_B &= v_x \hat{i} + v_y \hat{j} \\ {}^B \frac{d}{dt} v_B &= \dot{v}_x \hat{i} + \dot{v}_y \hat{j} \\ {}^N \frac{d}{dt} v_B &= {}^N a_B = (\dot{v}_x \hat{i} + \dot{v}_y \hat{j}) + \dot{\psi} \hat{K} \times (v_x \hat{i} + v_y \hat{j}) \\ &= (\dot{v}_x \hat{i} + \dot{v}_y \hat{j}) + v_x \dot{\psi} \hat{j} - v_y \dot{\psi} \hat{i} \\ {}^N a_B &= (\dot{v}_x - v_y \dot{\psi}) \hat{i} + (\dot{v}_y + v_x \dot{\psi}) \hat{j} \end{aligned}$$

where  ${}_{sub}^N a$  is the accelerations of each of the contact points of the front and the rear wheels. The accelerations can be finally expressed along the directions  $\hat{i}$  and  $\hat{j}$  respectively as follows:

$$a_x = \dot{v}_x - v_y \dot{\psi} \quad (20.62)$$

$$a_y = \dot{v}_y + v_x \dot{\psi} \quad (20.63)$$

#### 20.4.4 Phase III

First writing down the EOMs in the form:

$$\begin{aligned} \sum \mathbf{F}_{ext} &= m {}^N a_B \\ \sum \mathbf{M}_{ext} &= J \dot{\omega}_{/N} \end{aligned}$$

The forces due to lateral slips are externally applied at the contact points, let  $c_1$  and  $c_2$  be coefficients for each of the wheels. Additionally, the lateral forces in figure ?? acting externally to the wheel contact points show that for the rear wheel  $F_{y2} = c_2 \alpha$  and for the front wheel contact point  $F_{y1} = c_1 \alpha$ . Finally writing the equations for  $\sum \mathbf{F}_{ext} = m {}^N a_B$  along  $\hat{I}$ :

$$F_{x1} \cos \delta + F_{x2} - c_1 \left( \delta - \tan^{-1} \left( \frac{v_y + a\dot{\psi}}{v_x} \right) \right) \sin \delta = m \dot{v}_x - v_y \dot{\psi} \quad (20.64)$$

## Nomenclature

similarly, writing the equations for  $\sum \mathbf{F}_{ext} = m^N a_B$  along  $\hat{J}$ :

$$F_{x1} \sin \delta - c_2 \left( \tan^{-1} \left( \frac{v_y - b\dot{\psi}}{v_x} \right) \right) + c_1 \left( \delta - \tan^{-1} \left( \frac{v_y + a\psi}{v_x} \right) \right) \cos \delta = \dot{v}_y + v_x \dot{\psi} \quad (20.65)$$

Finally writing the equations for  $\sum \mathbf{M}_{ext} = J\ddot{\omega}_{/N}$  about  $\hat{K}$ :

$$F_{x1} \sin \delta a + ac_1 \left( \delta - \tan^{-1} \left( \frac{v_y + a\psi}{v_x} \right) \right) \cos \delta + bc_2 \left( \tan^{-1} \left( \frac{v_y - b\dot{\psi}}{v_x} \right) \right) = J\ddot{\psi} \quad (20.66)$$

### 20.4.5 Constructing State-Space equation

Choosing state variables  $\{v_x, v_y, \dot{\psi}\}$  (highest derivatives from Phsae II), the following equations can be formed for each of the state variable as shown in the follow equation:

$$\dot{v}_x = v_y \dot{\psi} + \frac{1}{m} \left\{ F_{x1} \cos \delta + F_{x2} - c_1 \left( \delta - \tan^{-1} \left( \frac{v_y + a\dot{\psi}}{v_x} \right) \right) \sin \delta \right\} \quad (20.67)$$

$$\dot{v}_y = -v_x \dot{\psi} + \frac{1}{m} \left\{ F_{x1} \sin \delta - c_2 \left( \tan^{-1} \left( \frac{v_y - b\dot{\psi}}{v_x} \right) \right) + c_1 \left( \delta - \tan^{-1} \left( \frac{v_y + a\dot{\psi}}{v_x} \right) \right) \cos \delta \right\} \quad (20.68)$$

$$\ddot{\psi} = \frac{1}{J} \left\{ aF_{x1} \sin \delta + ac_1 \left( \delta - \tan^{-1} \left( \frac{v_y + a\dot{\psi}}{v_x} \right) \right) \cos \delta + bc_2 \left( \tan^{-1} \left( \frac{v_y - b\dot{\psi}}{v_x} \right) \right) \right\} \quad (20.69)$$

As these equations are nonlinear they cannot be written directly in the matrix form. Observing equations given in the above set of equations, it can be seen that lateral dynamics of the vehicle is a function of longitudinal  $F_x$ , lateral  $F_y$  forces as well as input steering angle  $\delta$ . As the lateral dynamics is a function of longitudinal forces as well, it is this reason an ESP uses braking and acceleration to control the yaw-rate of the vehicle.

#### Assumptions for linearizing the model

1. Longitudinal velocity is constant  $\dot{v}_x = 0$ , this leads the the equation (20.67) to be eliminated. Because, if the state is going to be constant, there is no point in tracking it anymore.
2. Only highway driving, this leads to a small slip angles, in such a case the terms with  $\tan^{-1} \left( \frac{v_y - b\dot{\psi}}{v_x} \right) \approx \frac{v_y - b\dot{\psi}}{v_x}$
3. Further since there is no longitudinal accelerations, the longitudinal force  $F_{xi} = 0$

4. Small steering angles, leads to  $\cos\delta \approx 1$  and  $\sin\delta \approx \delta$

From the assumptions made above, the state equations of the single-track model will be approximated to as shown in the following equations:

$$\dot{v}_y = -v_x \dot{\psi} + \frac{1}{m} \left\{ -c_2 \left( \frac{v_y - b\dot{\psi}}{v_x} \right) + c_1 \left( \delta - \left( \frac{v_y + a\dot{\psi}}{v_x} \right) \right) \right\} \quad (20.70)$$

$$\ddot{\psi} = \frac{1}{J} \left\{ ac_1 \left( \delta - \left( \frac{v_y + a\dot{\psi}}{v_x} \right) \right) + bc_2 \left( \frac{v_y - b\dot{\psi}}{v_x} \right) \right\} \quad (20.71)$$

In summary, the single track linearized model for lateral dynamics of the vehicle can be expressed as following (substituting states  $x_1 = v_y$ ,  $x_2 = \dot{\psi}$ , steering input  $\delta = u$  and required outputs:  $y_1 = x_1$  and  $y_2 = x_2$ )

$$\dot{x}_1 = -v_x x_2 + \frac{1}{m} \left\{ -c_2 \left( \frac{x_1 - bx_2}{v_x} \right) + c_1 u - c_1 \left( \frac{x_1 + ax_2}{v_x} \right) \right\} \quad (20.72)$$

$$\dot{x}_2 = \frac{1}{J} \left\{ ac_1 u - ac_1 \left( \frac{x_1 + ax_2}{v_x} \right) + bc_2 \left( \frac{x_1 - bx_2}{v_x} \right) \right\} \quad (20.73)$$

writing the state space in the matrix form,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{(c_2+c_1)}{mv_x} & \frac{-ac_1+bc_2}{mv_x} - v_x \\ -\frac{ac_1-bc_2}{Jv_x} & \frac{-(a^2c_1+b^2c_2)}{Jv_x} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{c_1}{m} \\ \frac{ac_1}{J} \end{bmatrix} u \quad (20.74)$$

## 20.5 Differential drive robot

This section provides a kinematic modeling technique used to model a differential drive robot as well as provides a model that can be used for control system design of mechanical systems.

### 20.5.1 Part I: Deriving Equations of Motion for differential drive robot

Consider figure 20.11, this is a generic description of the kinematics of a differential drive robot.

## Nomenclature

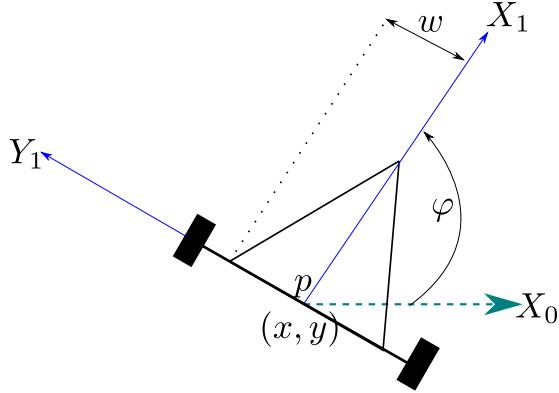


Figure 20.11: Differential drive robot

In deriving kinematic model, it is important to recognize the geometry of importance. In this case, the center of mass of the robot is considered. For the rotation of each wheels a scenario is considered and the kinematics of the robot is determined. For the first case, consider figure 20.12, where a case is considered such that  $v_r \neq 0$  and  $v_l = 0$ , where  $v_l$  and  $v_r$  are the velocities of the left and right wheels respectively. with

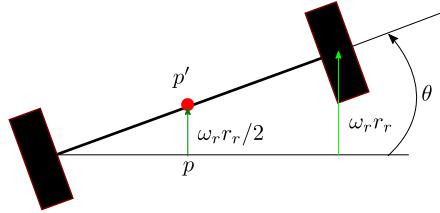


Figure 20.12: Kinematic configuration with  $v_r \neq 0$  and  $v_l = 0$

reference to figure 20.12,  $r_r$  is the radius of the right wheel and  $\omega_r$  is angular velocity of the robot when only  $v_r \neq 0$ .  $p$  is the position of mass center when at rest and  $p'$  is the position with a certain  $\omega_r$ .

The distance covered by the right wheel is expressed by,

$$d = \omega_r r_r \quad (20.75)$$

therefore, the position  $p'$ , can be expressed as,

$$\dot{p}'_x = \frac{\omega_r r_r}{2} \quad (20.76)$$

Assuming that the wheel do not slip, the velocity normal to forward direction will be zero such that  $\dot{p}'_y = 0$ .

Additionally, using the relationship  $\mathbf{v} = \dot{\theta} \hat{j} \times \mathbf{r}$ , where  $\mathbf{r}$  is the position of the displaced point from the center of rotation, the following expression for  $\dot{\theta}$  can be evaluated:

$$\omega_r r_r = 2w\dot{\theta} \implies \dot{\theta} = \frac{\omega_r r_r}{2w} \quad (20.77)$$

Performing same calculations as in the earlier case, this time using configuration  $v_r = 0$  and  $v_l \neq 0$  as shown in figure 20.13. The distance covered by the right wheel is

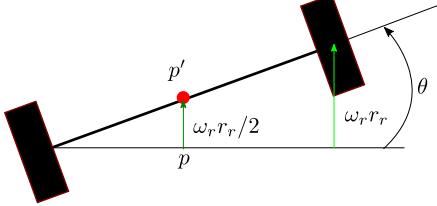


Figure 20.13: Kinematic configuration with  $v_r = 0$  and  $v_l \neq 0$

expressed by,

$$d = \omega_l r_l \quad (20.78)$$

therefore, the position  $p'$ , can be expressed as,

$$\dot{p}'_x = \frac{\omega_l r_l}{2} \quad (20.79)$$

Assuming that the wheel do not slip, the velocity normal to forward direction will be zero such that  $\dot{p}'_y = 0$ .

Additionally, using the relationship  $\mathbf{v} = \dot{\theta} \hat{j} \times \mathbf{r}$ , where  $\mathbf{r}$  is the position of the displaced point from the center of rotation, the following expression for  $\dot{\theta}$  can be evaluated:

$$\omega_l r_l = -2w\dot{\theta} \implies \dot{\theta} = -\frac{\omega_l r_l}{2w} \quad (20.80)$$

Using equations (20.76) and (20.79), since in any given case, both the velocities are non-zero, the effect of both of these can be added together. Since velocity is a vector, using vector addition, the followinf relation is determined for transnational velocity of the robots mass center,

$$\mathbf{v} = \frac{\omega_l r_l}{2} + \frac{\omega_r r_r}{2} = \frac{R}{2} (\omega_r + \omega_l) \quad (20.81)$$

where  $R$  is the radius of the wheels.

Similarly, angular velocity is also a vector quantity, performing vector addition for the case when both the wheel velocities are non-zero,

$$\dot{\varphi} = \frac{\omega_r r_r}{2w} - \frac{\omega_l r_l}{2w} = \frac{R}{2L} (\omega_r - \omega_l) \quad (20.82)$$

Where  $L$  is replaced with  $w$  and  $\varphi$  is replace with  $\theta$  which is in terms of notations in robotics.

Both the equations (20.81) and (20.82) together represent the kinematic equations for describing the motion of the differential robot.

These vector quantities can be expressed now in the global frame using the relation,

$$\mathbf{v} = {}^N\mathbf{R}_B \mathbf{b} \quad (20.83)$$

where

### Nomenclature

- $\mathbf{v}$  are the vectors expressed in frame N
- $\mathbf{b}$  are the vectors expressed in frame B
- ${}^N\mathbf{R}_B$  is the rotation matrix when B is rotated with reference to N

using equation 20.83,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{R}{2}(\omega_r + \omega_l) \\ 0 \\ \frac{R}{2L}(\omega_r - \omega_l) \end{bmatrix} \quad (20.84)$$

therefore, the vectors are expressed in frame N as,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{R}{2}(\omega_r + \omega_l) \cos\phi \\ \frac{R}{2}(\omega_r + \omega_l) \sin\phi \\ \frac{R}{2L}(\omega_r - \omega_l) \end{bmatrix} \quad (20.85)$$

# 21 Linearization

A simple nonlinear system can be linearized in case it has simple relationships with sin or cos or tan. For example the state of pendulum's angular acceleration due to an input  $u$  is expressed as:

$$\ddot{\theta} = -\frac{g}{l} \sin \theta + cu \quad (21.1)$$

where,  $c$  is a mapping coefficient that maps input torque  $u$  to output accelerations  $\ddot{\theta}$ . Linearization is performed at defined points so called operating points (OP). In general, the OP are chosen such that the changes in the states of a system is zero. OP are local minima of a system, for example in a pendulum a local minima is when the pendulum is hanging straight down. Mathematically, this is defined as the path of the pendulum where the derivatives of the path are zero. Therefore, about OP  $\theta_0$  when the pendulum is hanging straight down, the OP can be written as  $(\theta_0, u_0)$ . In the case of pendulum is hanging straight down, the input  $u_0$  at this point is zero.

After having established this operating point, the conditions for which is completely known for us, any other state relative from this OP can be found. Theoretically, if any state  $x$  is a relative distance  $\delta x$  from  $x_0$  and the inputs for which are at a relative distance  $\delta u$  from the OP  $u_0$ , then it can be said that

$$(x_0, u_0) \rightarrow (x = x_0 + \delta x, u = u_0 + \delta u) \quad (21.2)$$

For a nonlinear system such as one expressed by (21.1) written in the form:

$$\dot{x} = f(x, u) \quad y = h(x) \quad (21.3)$$

for the new deviations, the deviation dynamics can be written using equation (21.2):

$$\dot{\delta x} = \frac{d}{dt}(x + x_0) = \dot{x} \quad (21.4)$$

where  $\dot{x}_0 = 0$  (because it is a constant OP) .The derivatives (evolution's) of these deviations becomes

$$\dot{\delta x} = \dot{x} = f(x_0 + \delta x, u_0 + \delta u) \quad (21.5)$$

From equations (21.4) and (21.5), it can be seen that the evolution of the states are same as the evolution of the deviations. Further, this evolution of the deviations as a function of the deviations and inputs when having a reference OP can be found using Taylor series expansion as given below

$$\dot{\delta x} = f(x_0 + \delta x, u_0 + \delta u) \quad (21.6)$$

## Nomenclature

$$\dot{\delta}x \approx f(x_0, u_0) + \frac{\partial f}{\partial x}(x_0, u_0)\delta x + \frac{\partial f}{\partial u}(x_0, u_0)\delta u + \mathbf{H.O.T} \quad (21.7)$$

Only the first two terms of Taylor's series are considered as they are only tangents to the actual deviations and also linear to the deviations. By considering such deviations to be linear to the actual deviations, the dynamics of the system now become linear. Although this assumption has reduced the dynamics to only the linear part, in real systems it has been found to produce excellent results (provided that Linearization is possible).

Similarly, the output deviation can be linearized as:

$$y = h(x_0, u_0) \approx h(x_0) + \frac{\partial h}{\partial x}(x_0, u_0)\delta x \quad (21.8)$$

Additionally, in equations (21.6) and (21.8), the OP are chosen such a way that their derivatives are zero (local minima or maxima). Therefore, the first terms from both these equations goes to zero, the remaining terms can be group into:

$$\frac{\partial f}{\partial x}(x_0, u_0) = \mathbf{A} \quad (21.9)$$

$$\frac{\partial f}{\partial u}(x_0, u_0) = \mathbf{B} \quad (21.10)$$

$$\frac{\partial h}{\partial x}(x_0, u_0) = \mathbf{C} \quad (21.11)$$

where, matrices  $A, B, C$  are called **Jacobians**, each of these terms are the elements of the matrices A, B and C.

Now from equations (21.6), (21.8) and (21.9), the state and output equations can be written as:

$$\dot{\delta}x = \mathbf{A}\delta x + \mathbf{B}\delta u \quad (21.12)$$

$$y = \mathbf{C}\delta x \quad (21.13)$$

Therefore, the non-linear state equations are linearized using this method. Because the states in the linearized equation is  $\delta x = x - x_0$ , and the input is  $\delta u = u - u_0$  and output  $\delta y = y - y_0$ , the operating points have to be added and subtracted from the feedback system as shown in figure 21.1

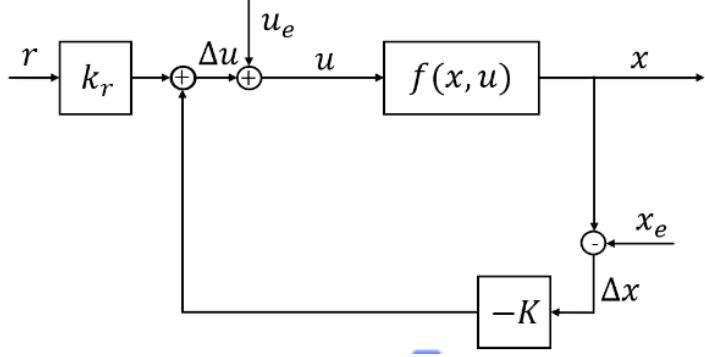


Figure 21.1: Adding and subtracting OP's in feedback loop

## 21.1 Linearizing EoM of a pendulum

Consider a pendulum with the state dynamics due to an input  $u$  expressed by:

$$\ddot{\theta} = -\frac{g}{l} \sin \theta + cu \quad (21.14)$$

where,  $c$  is the mapping factor that gives acceleration on  $\theta$  coordinate for an input torque  $u$ . The term  $\sin \theta$  makes the EoM (21.14) nonlinear. Therefore, selecting an OP about  $(\theta_0 = 0, u_0 = 0)$  from the position where the pendulum is hanging straight down. Also, consider the variable of state space as follows:

$$\begin{aligned} x_1 &= \theta \\ x_2 &= \dot{\theta} \end{aligned}$$

and output variable:

$$y = \theta$$

State dynamics can be written by the vector  $\dot{\mathbf{x}}$  as:

$$\dot{\mathbf{x}} = \begin{bmatrix} x_2 \\ \dot{\theta} \end{bmatrix} = f(x, u) = \begin{bmatrix} x_2 \\ -\frac{g}{l} \sin x_1 + cu \end{bmatrix} \quad (21.15)$$

the output equation is just the state variable  $x_1$ :

$$h(x) = x_1 \quad (21.16)$$

Jacobians are needs to determine matrices  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  about the OP  $(\theta_0, u_0)$  using (21.15):

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}_{(\theta_0, u_0)} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} \cos x_1 & -u \sin(x_1) \end{bmatrix}_{(0,0)} = \begin{bmatrix} 0 & 1 \\ \frac{g}{l} & 0 \end{bmatrix}$$

*Nomenclature*

$$\mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} \\ \frac{\partial f_2}{\partial u_2} \end{bmatrix}_{(\theta_0, u_0)} = \begin{bmatrix} 0 \\ \cos(x_1) \end{bmatrix}_{(0,0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{C} = [1 \ 0]_{(\theta_0, u_0)} = [1 \ 0]_{(0,0)} = [1 \ 0]$$

# 22 State space behavior

The state space model is generally described by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (22.1)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (22.2)$$

This chapter focuses on how the behaviors are linked to the parameters in the matrices **A**, **B**, **C** and **D**. As described in section 19.1, the LT of the state-space model is expressed by equations given below, which represent the input/output behavior from the state-space model

$$(sI - A)X(s) = BU(s) + x(0) \implies X(s) = (sI - A)^{-1}BU(s) + (sI - A)^{-1}x(0) \quad (22.3)$$

$$Y(s) = [C(sI - A)^{-1}B + D]U(s) + (sI - A)^{-1}x(0) \quad (22.4)$$

Further, the TF of the system was expressed from the output transformation (22.4) as

$$G(s) = C(sI - A)^{-1}B + D \quad (22.5)$$

It is important to note that this **TF is of the Open-Loop system**, as using the state-space models, the concept of feedback is not yet been defined. Further, it was also noted that since the term  $(sI - A)^{-1}$  in TF comes in the denominator due to the inverse, the poles of the system are defined by this characteristic equation

$$\det(sI - A) = 0 \quad (22.6)$$

Additionally, the Eigenvalues of matrix **A** are computed using the definition  $|sI - A| = 0$  which is again same as the poles of the system. Therefore, the behavior of a system is defined by the Eigenvalues (poles) of the matrix **A**.

## 22.1 State-Transition matrix $e^{\mathbf{At}}$

An introduction to state-transition matrix was defined in section 19.3 in which by comparing the coefficients from the output transform **Y** given by equation (19.8) with the solution of state-space given by equation (19.17) from an ODE, it was found that the state-transition matrix is expresses as

$$e^{\mathbf{At}} = \mathcal{L}^{-1}\{(sI - A)^{-1}\} \quad (22.7)$$

## Nomenclature

From the solution of the differential of matrix exponential  $e^{\mathbf{At}}$  as described in section 6.6, it was found that

$$\frac{d}{dt}e^{\mathbf{At}} = \mathbf{A}e^{\mathbf{At}} \quad (22.8)$$

From this solution it can be seen that the solution contains the original state  $\mathbf{x} = e^{\mathbf{At}}$  itself. That is to say, that the solution *flows* (the evolution of the states) as a function of the state  $\mathbf{x} = e^{\mathbf{At}}$  itself. Therefore, the states evolve about a certain vector field that is a function of the state  $\mathbf{x} = e^{\mathbf{At}}$ . Therefore, the vector field is defined as

$$\Phi(t) = f_D(e^{\mathbf{At}}) \quad (22.9)$$

here  $e^{\mathbf{At}}$  is an invertible function, such that there is always a non-zero value at any given time  $t$  and this solution can be assigned to the evolution of state if the initial value of the state  $x_0$  is known. So that state  $x_t$  at any time  $t$  can be expressed as a function of

$$x_t = e^{\mathbf{At}}x_0 \quad (22.10)$$

therefore, the evolution of states can be expressed using the state-transition matrix  $\Phi(t)$  as

$$x(t) = \Phi(t)x(0) \quad (22.11)$$

where

$$\Phi(t) = \mathcal{L}^{-1}\{(sI - A)^{-1}\} \quad (22.12)$$

In summary, given the state equation  $\dot{x} = \mathbf{Ax}$  can be solved using  $x(t) = \Phi(t)x(0)$ . Where  $\Phi(t) = \mathcal{L}^{-1}\{(sI - A)^{-1}\}$

# 23 State Feedback Design

## 23.1 Principles

From the state-space model as described by

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (23.1)$$

$$y(t) = Cx(t) + Du(t) \quad (23.2)$$

It was found that the behavior of the system depends on the Eigenvalues of matrix A, which are also the poles of the open-loop system. From the feedback design principles from classical control theory, it was found that the required behavior of a system can be achieved by first closing the loop (getting required output variables) and changing the poles of this closed-loop system so as to match the desired behavior (by changing the dynamics of the system). In state feedback design, a process something similar is used. The idea with control design is to modify the Eigenvalues of system matrix A by using a control input  $u(t)$ . The difference state-space based feedback design is that contrary to classical design of using system output  $y(t)$  for feedback design, here a measured state variable  $x(t)$  is used.

The state feedback control design can be expressed mathematically as

$$u(t) = -Kx(t) + k_r r(t) \quad (23.3)$$

The measured state variable  $x(t)$  is used to calculate the desired control action required with a feedback gain  $K$  to amplify / attenuate the measured state-variables. Additionally, the error concept similar to classical control design can be implemented using the reference signal  $r(t)$  with steady-state reference gain  $k_r$  so as to make the system track the desired steady-state.

Equation (23.3) can be read as one control action ( $-Kx(t)$ ) to reduce the dynamics by any input and other control action ( $k_r r(t)$ ) to amplify the system towards the steady-state. Additionally, the control action is negative towards the state gain  $-Kx(t)$ , that is because, the idea is to reduce the dynamics of the system to give input, rather than amplify them. In most of the control design approaches the idea of attenuating the dynamics is widely employed.

From the initial design of state feedback system which was essentially an open-loop as shown in figure 23.1 (here, A is not a feedback matrix, but is instead required to calculate the dynamics  $\dot{x}$ ), the state feedback design is essentially now a feedback with a measured state  $x(t)$  as shown in figure 23.2. Further, in most of the system, the feed-through matrix D is not used to influence the system output, therefore, in general, the state feedback design of the system is as shown in figure 23.3.

## Nomenclature

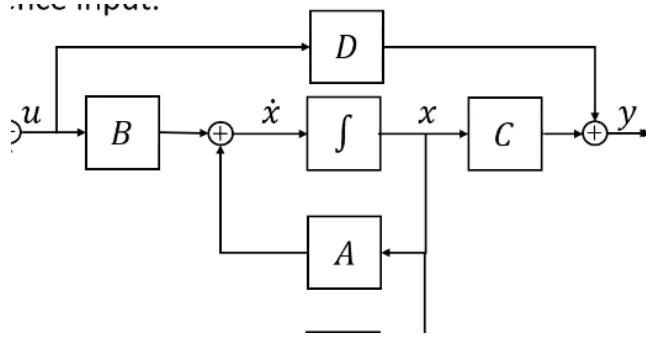


Figure 23.1: Block diagram of state-space model

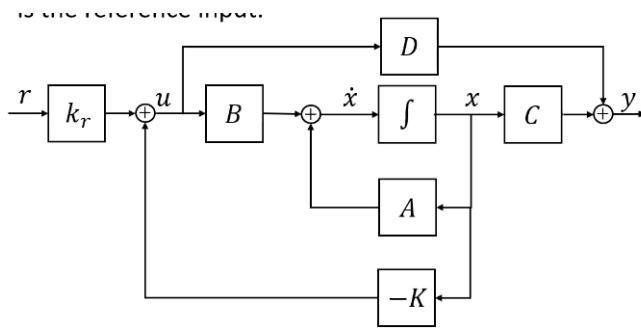


Figure 23.2: Block diagram of state feedback model

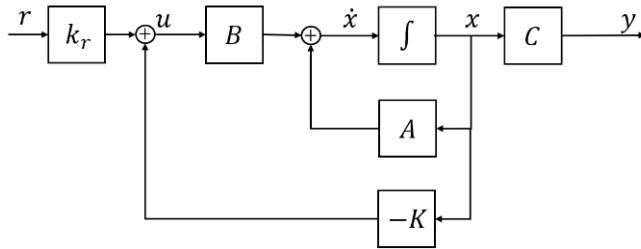


Figure 23.3: Block diagram of state feedback model without D

Using the control equation now in the state equation, the state equation becomes

$$\dot{x}(t) = Ax(t) + B(-Kx(t) + k_r r(t)) \quad (23.4)$$

$$= (A - BK)x(t) + Bk_r r(t) \quad (23.5)$$

It can be observed that now the new system matrix is given by  $(A - BK)$  matrix instead of matrix A in the previous case. With this new system matrix, the control objective can be stated: **Control objective:** Choose  $K$  such that the closed-loop dynamics  $A - BK$  get desired properties such as stability, tracking etc.,

The size of  $K$  matrix can be obtained using the relationship  $u = -Kx$ , so if the inputs are of the size  $m \times 1$  and the states  $x$  are of the size  $n \times 1$ , then the  $K$  matrix should

be equal to a size  $(m \times n)$  in order to satisfy the relation  $u = -Kx$ .

### 23.1.1 Poles of the closed loop system

Equation (23.5) gives a closed loop (state feedback) system, in order to find the poles, similar to previous open-loop models, a LT is performed on equation (23.5)

$$sX(s) = (A - BK)X(s) + Bk_r R(s) \implies (sI - (A - BK))X(s) = Bk_r R(s) \quad (23.6)$$

substituting in the LT of the output equation (lets consider matrix D to be a zero matrix)

$$Y(s) = CX(s) \implies C[(sI - (A - BK))]^{-1}Bk_r R(s) \quad (23.7)$$

the TF of the closed-loop system is then

$$G(s) = \frac{Y(s)}{R(s)} = \frac{CBk_r}{sI - (A - BK)} \quad (23.8)$$

The poles of the closed-loop system is given by the values of the denominator of the TF (23.8), which can be found by setting the denominator of TF to zero

$$\{p\} := \det\{sI - (A - BK)\} = 0 \quad (23.9)$$

In expressions containing matrices, the expression is set to zero first by taking the determinant which gives the value of the matrix and then setting the value of the determinant to zero.

### 23.1.2 Reference tracking

The steady-state reference gain  $k_r$  does not affect the stability of the system, but it does affect the steady-state solution. And choosing steady-state depends on the tracking required. A  $k_r$  is chosen such that

$$y(t) \approx r(t) \quad t \rightarrow \infty$$

And to choose the steady-state reference gain, the steady-state of the system has to be determined. This can be done by using the state and the output equations and solve them at the steady-state. For a differential  $\dot{x}$ , the steady-state is when  $\dot{x} = 0$

$$0 = (A - BK)x(t) + Bk_r r(t) \quad (23.10)$$

$$y(t) = Cx(t) + Du(t) \quad (23.11)$$

Additionally, in most of the system matrix D is zero, therefore

$$0 = (A - BK)x(t) + Bk_r r(t) \quad (23.12)$$

$$y(t) = Cx(t) \quad (23.13)$$

## Nomenclature

therefore, the steady-state can be expressed by solving the above equation for  $x(t)$  and substituting in the output equation

$$y(t) = -C(A - BK)^{-1}Bk_r r(t) \quad (23.14)$$

Finally, using the steady-state output equation, steady-state reference gain  $k_r$  can be determined using the condition  $y(t) \approx r(t) \quad t \rightarrow \infty$ . From equation (23.14),

$$k_r = (-C(A - BK)^{-1}B)^{-1} \quad (23.15)$$

or

$$k_r = \frac{1}{-C(A - BK)^{-1}B} \quad (23.16)$$

In summary, the design of state feedback is a two step process, first for  $K$  and then from  $k_r$ .

## 23.2 Integral control

Although using state-space model by designing a steady-state reference gain  $k_r$  to track the reference  $r(t)$ , in reality due to model inconsistencies (with the real model), the ss will in general not be tracked perfectly by the controller designed using equation (23.3). In such a case, similar to the classical control design, a integral control action is implemented so as to track the error and apply a control proportional to it.

In the state-feedback model in equation (23.5) there is no variable to track the error, therefore, new state has to be introduced that tracks the error being added up into the system due to ss-error. Suppose the state is introduced as

$$z(t) = \int (y(t) - r(t))dt \quad (23.17)$$

the state derivative is then

$$\dot{z}(t) = y(t) - r(t) \quad (23.18)$$

Now with this additional state along with the original state equation of the system (23.1), the two states can be tracked by the model

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ y - r \end{bmatrix} \quad (23.19)$$

From equation (23.13),  $y = Cx$  using this

$$\begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} Ax + Bu \\ Cx - r \end{bmatrix} \implies \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ -1 \end{bmatrix} r \quad (23.20)$$

where the control signal is now given by

$$u(t) = -Kx(t) - K_I z(t) + k_r r(t) \quad (23.21)$$

### 23.3 Example of State-feedback design

Consider an example dynamics given of a single track model, where the two states are lateral position and orientation angle respectively.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} av_0/b \\ v_0/b \end{bmatrix} u \quad (23.22)$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u \quad (23.23)$$

with given parameters  $v_0 = 12m/s$ ,  $a = 2m$ ,  $b = 4m$ . The input  $u$  is the steering angle  $\delta$ . The idea is to design a controller that stabilizes the dynamics and tracks a given lateral position of the vehicle. This condition is a simplified way of a lane keeping control.

Starting from the controller equation, we have two states  $x_1$  and  $x_2$

$$u(t) = -Kx(t) + k_rr(t) = u = -K_1x_1 - K_2x_2 + k_rr \quad (23.24)$$

The closed loop system dynamics are obtained by using the closed-loop control equation (23.24) to the state equation (23.22)

$$\dot{x} = (A - BK)x(t) + Bk_rr(t) \implies \dot{x} = (A - [B][K_1 \ K_2])x(t) + Bk_rr(t) \quad (23.25)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \left( \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} av_0/b \\ v_0/b \end{bmatrix} [K_1 \ K_2] \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} av_0/b \\ v_0/b \end{bmatrix} k_rr \quad (23.26)$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -K_1av_0/b & v_0 - K_2av_0/b \\ -K_1v_0/b & -K_2v_0/b \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} k_rav_0/b \\ k_rv_0/b \end{bmatrix} r \quad (23.27)$$

The closed-loop poles are determined using equation (23.9) for state feedback controller. In order to design the parameters of the controller, a pole-placement technique is used similar to classical control design. Therefore, choosing a desired polynomial to do so as

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 \quad (23.28)$$

using equation (23.9), the characteristic equation for this example problem can be found as

$$\det(\lambda I - A + BK) = \lambda^2 + \frac{v_0}{b}(aK_1 + K_2)\lambda + \frac{K_1v_0^2}{b} \quad (23.29)$$

matching coefficients with desired polynomial

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = \lambda^2 + \frac{v_0}{b}(aK_1 + K_2)\lambda + \frac{K_1v_0^2}{b} \quad (23.30)$$

the feedback gain can be determined as

$$K_1 = \frac{b\omega_n^2}{v_0^2} \quad (23.31)$$

## Nomenclature

$$K_2 = \frac{2\zeta\omega_n b}{v_0} - \frac{ab\omega_n^2}{v_0^2} \quad (23.32)$$

A feedback gain in the system stabilizes the system through pole placements. However, for tracking, the system also should now track the reference  $r(t)$  for which the reference gain  $k_r$  should be designed. This  $k_r$  value can be determined using the previously determined equation for  $k_r$  equation (23.16)

$$k_r = \frac{1}{-C(A - BK)^{-1}B} \implies k_r = \frac{b\omega_n^2}{v_0^2} \quad (23.33)$$

therefore, now the controller equation becomes

$$u = -K_1x_1 - K_2x_2 + k_r r \implies -\frac{b\omega_n^2}{v_0^2}x_1 - \frac{2\zeta\omega_n b}{v_0} - \frac{ab\omega_n^2}{v_0^2}x_2 + \frac{b\omega_n^2}{v_0^2}r \quad (23.34)$$

Further, the behavior of this closed loop system can be determined by using simulations, simulating for different values to system parameters  $\zeta$  and  $\omega_n$ . For a unit step input, the following results can be visualized in Simulink as shown in figure 23.4 Run-

**Simulations with different values of  $\zeta$  and  $\omega_n$ :**

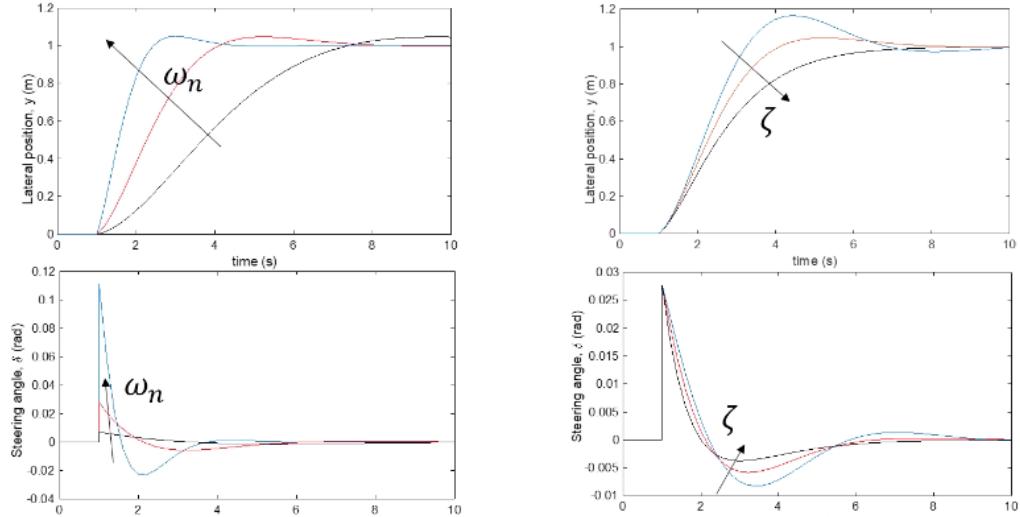


Figure 23.4: Simulation results for different design parameters

ning another simulation for a trajectory which is sinusoidal and see if the model is tracking the reference signal. For such a purpose a sinusoidal trajectory was chosen such that it produces an oscillation of  $2\pi$  time period 10s. Figure 23.5 shows the tracking of the system for the reference input.

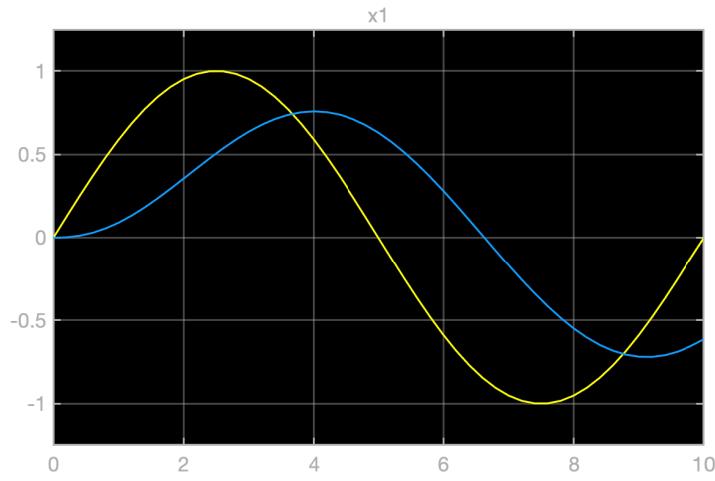


Figure 23.5: Simulation results for sinusoidal trajectory

It can be seen from figure 23.5 that the system is able to trace the trajectory by the raise time is too big. From figure 23.4 for the simulations for a step-response, it can be seen that the raise time can be reduced by increasing  $\omega_n$ , therefore setting a larger  $\omega_n$ , the resulting figure is shown in 23.6, reaching a perfect tracking.

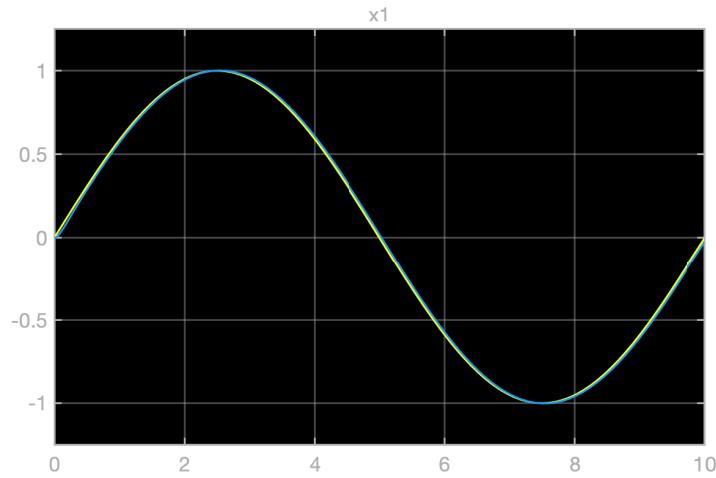


Figure 23.6: Simulation results for sinusoidal trajectory with better tracking

## 23.4 Example of reaching non-controllability

Consider another system which is similar to Example 1 but with a different matrix B as given by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \quad (23.35)$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u \quad (23.36)$$

with control signal same as the previous example

$$u(t) = -Kx(t) + k_r r(t) = u = -K_1 x_1 - K_2 x_2 + k_r r \quad (23.37)$$

Solving for state dynamics

$$\dot{x} = (A - BK)x + Bk_r r = \begin{bmatrix} -K_1 & 1 - K_2 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} k_r \\ 0 \end{bmatrix} r \quad (23.38)$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (23.39)$$

Solving for closed-loop system poles

$$\det(\lambda I - A + BK) = \det \begin{Bmatrix} \lambda + K_1 & K_2 - 1 \\ 0 & \lambda \end{Bmatrix} = \lambda^2 + K_1\lambda = \lambda(\lambda + K_1) \quad (23.40)$$

while matching the systems characteristic polynomial, the following two results are observed

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 = \lambda(\lambda + K_1) \quad (23.41)$$

1. Not all coefficients of characteristic polynomial can be matched with the desired polynomial. That means all the Eigenvalues cannot be controlled. Therefore, the dynamics of the system cannot be shaped. Such a system is not controllable or not reachable.
2. One of the Eigenvalues is zero, the system is not asymptotically stable (it may oscillate with same magnitude without amplification, such a condition is also not asymptotically stable).

# 24 Tuning of state feedback systems

There is always a way of pole-placements using the desired polynomial function. But this method is only suited for canonical systems and systems that are of low order. For complex systems there are other methods that are optimized for usage. One such method is the Ackermann's Formula.

## 24.1 Ackermann's Formula for pole placements

The approach given here relies on clear definitions of two polynomials:

1. Open-loop polynomial

- $p_0 = s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0$

2. Closed-loop polynomial

- $p_c = s^n + \alpha_{n-1}s^{n-1} + \dots + \alpha_1s + \alpha_0$

A Cayley-Hamilton theorem is used which says that if a matrix A satisfies its own characteristic equation. Such as if a characteristic equation of a matrix A is given by

$$|\lambda I - A| = 0 \implies \lambda^n + a_{n-1}\lambda^{n-1} + \dots + a_0 = 0 \quad (24.1)$$

where  $a_n$  are the coefficients of the matrix A. According to Cayley-Hamilton theorem, if the scalar value  $\lambda$  in the characteristic equation is replaced by the matrix A itself, it still leads to the characteristic equation to be satisfied (RHS is equal to zero).

$$A^n + a_{n-1}A^{n-1} + \dots + a_0A^0 = 0 \quad (24.2)$$

The above result leads to the following corollaries

- if the open-loop polynomial is given by
  - $p_0 = s^n + a_{n-1}s^{n-1} + \dots + a_1s + a_0$ , then substituting for system matrix A satisfies the equation:
    - $A^n + a_{n-1}A^{n-1} + \dots + a_0I = 0$
- if the closed-loop polynomial is given by
  - $p_c = s^n + \alpha_{n-1}s^{n-1} + \dots + \alpha_1s + \alpha_0$ , then substituting for system matrix (A - BK) satisfies the equation:

## Nomenclature

$$-(\mathbf{A} - \mathbf{BK})^n + a_{n-1}(\mathbf{A} - \mathbf{BK})^{n-1} + \dots + a_0\mathbf{I} = 0$$

The second result from the closed-loop polynomial is an important result as from this result it can be seen that there exists a values of  $K$  for which the Cayley-Hamilton theorem is satisfied. In such a case, the closed-loop gain  $K$  is determined in a mathematical way.

### 24.1.1 Ackermann's approach

The intention is to exploit the latter case of these identities and the full rank nature of the matrix. First using the later case identity, following definitions on closed-loop polynomials can be developed

$$p_c(A - BK) = (\mathbf{A} - \mathbf{BK})^n + a_{n-1}(\mathbf{A} - \mathbf{BK})^{n-1} + \dots + a_0\mathbf{I} \quad (24.3)$$

there is another definition that is possible, in which case there is  $K$  absent, such as

$$p_c(A) = \mathbf{A}^n + a_{n-1}\mathbf{A}^{n-1} + \dots + a_0\mathbf{I} \quad (24.4)$$

Now we have two polynomials  $p_c(A - BK)$  and  $p_c(A)$ , by definition if there exists a closed-loop gain  $K$  such that it satisfies the polynomial  $p_c(A - BK) = 0$ , then the absence of  $K$  cannot make the same polynomial go to zero such as  $p_c(A) \neq 0$ . Since the result of Cayley-Hamilton's proof of closed-loop polynomial is established, we can unpack the terms of this polynomial and look at what we get. In general, a polynomial expansion of matrix is defined as

$$(A - BK)^n = A^n - A^{n-1}BK + \dots + (-1)^{n-1}A(BK)^{n-1} + (-1)^n b f_n(A, BK) \quad (24.5)$$

using this definition, an example second order polynomial can be expanded as follows

$$(A - BK)^2 = A^2 - ABK + BKA + (BK)^2 \quad (24.6)$$

Infact, the polynomial  $(A - BK)^n$  can be written in terms of controllability matrix form as follows:

$$(A - BK)^n = A^n + [B, AB, \dots, A^{n-2}B, A^{n-1}B] \begin{bmatrix} (-1)^n f_n(A, BK) \\ (-1)^{n-1} K(BK)^{n-2} \\ \vdots \\ \vdots \\ -K \end{bmatrix} \quad (24.7)$$

As an example, a cubic polynomial can be written in terms of controllability matrix as follows:

$$(A - BK)^3 = A^3 - A^2BK + A(BK)^2 - Bf_n(A, BK) \quad (24.8)$$

$$(A - BK)^3 = A^3 + [B, AB, A^2B] \begin{bmatrix} (-1)^n f_n(A, BK) \\ K(BK) \\ -K \end{bmatrix} \quad (24.9)$$

In essence, the cubic polynomial is written in the form of  $A^3 + M_c v_3$ , where  $M_c$  is the controllability matrix and  $v_3$  is some vector of 3 rows that multiplies  $M_c$ . In general, the matrix polynomial expansion can be written in the form:

$$(A - BK)^n = A^n + M_c v_n \quad (24.10)$$

Following this above result, the following polynomial expansion can also be defined

$$(A - BK)^n = A^n + M_c v_n \quad (24.11)$$

$$(A - BK)^{n-1} = A^{n-1} + M_c v_{n-1} \quad (24.12)$$

$$\dots \quad (24.13)$$

$$\dots \quad (24.14)$$

$$\dots \quad (24.15)$$

$$(A - BK)^2 = A^2 + M_c v_2 \quad (24.16)$$

$$(A - BK) = A + M_c v_1 \quad (24.17)$$

Substituting these results in the closed-loop polynomial itself

$$\begin{aligned} p_c(A - BK) &= (\mathbf{A} - \mathbf{BK})^n + a_{n-1}(\mathbf{A} - \mathbf{BK})^{n-1} + \dots + a_0 \mathbf{I} \\ \implies p_c(A - BK) &= A^n + \alpha_{n-1} A^{n-1} + \dots + \alpha A + a_0 \mathbf{I} + M_c[v_n + \alpha_{n-1} v_{n-1} + \dots + \alpha v_1] \end{aligned}$$

Using the previously derived results on closed-loop polynomials also given here

$$p_c(A - BK) = 0 \quad (24.18)$$

$$p_c(A) \neq 0 \quad (24.19)$$

Also, note that there is a piece of  $p_c(A)$  in the previous equation of  $p_c(A - BK)$ , which can be now written as:

$$0 = p_c(A) + M_c[v_n + \alpha_{n-1} v_{n-1} + \dots + \alpha v_1] \quad (24.20)$$

rearranging the above equation

$$M_c^{-1} p_c(A) = -[v_n + \alpha_{n-1} v_{n-1} + \dots + \alpha v_1] \quad (24.21)$$

Note that from equation (24.7),  $K$  lies at the bottom of vector  $v_n$ , in order to extract this  $K$ , matrix  $v_n$  has to be pre-multiplied with another matrix  $[0 \ 0 \ 0 \dots 1]$  such that

$$[0 \ 0 \ 0 \dots 1] v_n = -K \quad (24.22)$$

therefore, pre-multiplying both sides of equation (24.21) with  $[0 \ 0 \ 0 \dots 1]$   $K$  can be extracted as follows

$$[0 \ 0 \ 0 \dots 1] M_c^{-1} p_c A = K \quad (24.23)$$

## Nomenclature

Therefore, the state-feedback gain  $K$  is now defined by mathematical definition and can be found in a precise way of using the definition of closed-loop polynomial  $p_c$ . In summary, if a desired closed-loop polynomial is defined as

$$p_c = s^n + \alpha_{n-1}s^{n-1} + \dots + \alpha_1s + \alpha_0$$

the required feedback is given by the formula

$$[0 \ 0 \ \dots \ 1] M_c^{-1} p_c A = K$$

where  $p_c(A)$  is defined by its definition  $p_c(A) = A^n + \alpha_{n-1}A^{n-1} + \dots + \alpha A + a_0 \mathbf{I}$

## 24.2 Example solving for $\mathbf{K}$ using Ackermann's formula

Consider a system given by matrices

$$\mathbf{A} = \begin{bmatrix} -1 & -2 \\ 1 & -0.4 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$$

$$\mathbf{C} = [3 \ 4]$$

the goal is to chose  $K$  such that the closed-loop poles are located at  $(-1, -2)$ . This leads to the desired polynomial to be

$$p_c = s^2 + 3s + 2$$

In order to find closed-loop polynomial  $p_c(A)$ , we can replace matrix  $\mathbf{A}$  with scalar  $s$  in the desired polynomial as follows

$$p_c(A) = \begin{bmatrix} -1 & -2 \\ 1 & -0.4 \end{bmatrix}^2 + 3 \begin{bmatrix} -1 & -2 \\ 1 & -0.4 \end{bmatrix} + \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} -2 & -3.2 \\ 1.6 & -1.04 \end{bmatrix} \quad (24.24)$$

further controllability matrix is

$$M_c = [B \ AB] = \begin{bmatrix} 1 & 3 \\ -2 & 1.8 \end{bmatrix} \quad (24.25)$$

finally, use Ackermann's formula to find  $K$

$$[0 \ 1] M_c^{-1} p_c(A) = K = [-0.31 \ -0.95] \quad (24.26)$$

## 24.3 Example with Single track model

Using the example model of the single track model given in section 23.3 also provided here

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} av_0/b \\ v_0/b \end{bmatrix} u \quad (24.27)$$

$$y = [1 \ 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0]u \quad (24.28)$$

With the desired closed-loop characteristic polynomial

$$p_c(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (24.29)$$

finding closed-loop polynomial  $p_c(A)$  by substituting scalar  $s$  with matrix A:

$$p_c(A) = \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix}^2 + 2\zeta\omega_n \begin{bmatrix} 0 & v_0 \\ 0 & 0 \end{bmatrix} + \omega_n^2 = \begin{bmatrix} \omega_n^2 & 2v_0\zeta \\ 0 & \omega_n^2 \end{bmatrix} \quad (24.30)$$

the controllability matrix:

$$M_c = [B \ AB] = \begin{bmatrix} av_0/b & v_0^2/b \\ v_0/b & 0 \end{bmatrix} \quad (24.31)$$

using the Ackermann's formula to find  $K$

$$K = [0 \ 1] M_c^{-1} p_c(A) = \begin{bmatrix} b\omega_n^2 & 2\zeta\omega_n b - ab\omega_n^2 \\ v_0^2 & v_0 \end{bmatrix} \quad (24.32)$$

## 24.4 Linear Quadratic Regulator

For MiMo systems in general a LQR controller design method is used. LQR is an alternative method for pole placement so that the closed-loop system optimizes a cost function given by:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (24.33)$$

equation (24.33) is called the cost function of the closed-loop system where  $Q$  is a penalty matrix for the states costs and is symmetric and semi-positive definite ( $Q \geq 0$ ) and  $R$  is also symmetric but positive definite ( $R > 0$ ). The penalties are chosen so as to minimize the area under the curve formed by the integration given by equation (24.33). Also, note that the area under the curves formed by each of the function  $x^T Q x$  and  $u^T R u$  is quadratic. This is done so as to form a definite minimum as a quadratic function is parabolic and always has a definite minima compared to a first order polynomial.

## Nomenclature

The optimization problem is defined as minimizing  $J$  using control inputs  $u \in \mathbb{R}^m$ , such that the system is subjected the dynamics (constraints) given by  $\dot{x} = Ax + Bu$ . Note that the constraint is minimized by determining some control law  $u$  such that the problem is minimized. This can be explained as follows, suppose we have a non-zero system state  $x(0) \neq 0$ , such as a tilt of a satellite angle which will remain in this state for ever. Such a state may not be desirable and it needs to be driven back to zero such that the state evolves to  $x(t) = 0$ . For a state to be driven to some other state, it needs an input, in our case the input comes from the controls such that a positive definite control action needs to be provided. Therefore, the minimization problem is to find the control law  $u(t)$  such that it drives the states to the required positions. Also note that some of the states may not be so important and some of the control actions may be very expensive. Therefore, penalty matrices  $Q$  and  $R$  are used to weigh-in their importance. As we have two knobs to turn using the penalty matrices  $Q$  and  $R$ , consider two cases

1. if  $Q$  is bigger than  $R$ , then more importance is given to the states so the cost function will penalize control action such that higher control action is provided using the control law  $u(t)$  (aggressive controller)
2. if  $R$  is bigger than  $Q$ , then more importance is given to control such that the states evolution will be penalized. The states will evolve slowly (conservative controller)

### 24.4.1 Solution to the optimization problem

To the solution to minimizing the problem  $J$  given by:

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \quad (24.34)$$

subjected to the constraints:

$$\dot{x} = Ax + Bu \quad (24.35)$$

is a full state-feedback control given in standard form:

$$u = -Kx \quad (24.36)$$

In other words, the optimal solution to minimize  $J$  is to use a full state-feedback control. where the control matrix is given by:

$$K = R^{-1} B^T S \quad (24.37)$$

where  $S$  is a positive definite, symmetric matrix given by solving algebraic-Riccati equation:

$$A^T S + S A - S B R^{-1} B^T S + Q = 0 \quad (24.38)$$

the size of  $S$  is  $(n \times n)$

In Summary, the procedure to solve the LQR problem is as follows:

1. The state and the control matrices A and B are given
2. Choose the penalty matrices Q and R
3. In order to find full state-feedback control gain  $K$ , the algebraic-Riccati equation has to be solved for  $S$
4. Compute full state-feedback control gain  $K$
5. There are multiple solutions to  $S$ ,  $K = R^{-1}B^T S$  is chosen for values of  $S$  such that it yields a stable system. Which is found using Eigenvalues.

## 24.5 Examples on LQR

### 24.5.0.1 Mass damper system

**Step1:** Consider a mass-damper system given by its state-space model:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 - c/m & \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

choosing numeric values for the system to be solved using Matlab, such that  $m = 1$  and  $c = 0.2$

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 - 1/5 & \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

**Step2:** choosing Q and R

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$R = [0.01]$$

to solve the algebraic-Riccati equation consider a symmetric solution matrix S given by

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{bmatrix}$$

Simplifying the algebraic-Riccati equation in Matlab yields:

$$A^T S + S A - S B R^{-1} B^T S + Q = \begin{pmatrix} 1 - 100 s_{12}^2 & s_{11} - \frac{s_{12}}{5} - 100 s_{12} s_{22} \\ s_{11} - \frac{s_{12}}{5} - 100 s_{12} s_{22} & -100 s_{22}^2 - \frac{2s_{22}}{5} + 2 s_{12} + 1 \end{pmatrix}$$

because the RHS of algebraic-Riccati equation is zero, that means that each of the elements of the above equation should go to zero, in Matlab solving for the above

### Nomenclature

equation with each of the element equal to zero yields  $SolS = solve(ARE(1, 1) == 0, ARE(1, 2) == 0, ARE(2, 1) == 0, ARE(2, 2) == 0, s11, s12, s22)$ :

$$s_{11} = \begin{pmatrix} \frac{\sqrt{2001}}{50} \\ -\frac{\sqrt{2001}}{50} \\ -\frac{\sqrt{3001}}{50} \\ \frac{\sqrt{3001}}{50} \end{pmatrix}$$

$$s_{12} = \begin{pmatrix} -\frac{1}{10} \\ -\frac{1}{10} \\ \frac{1}{10} \\ \frac{1}{10} \end{pmatrix}$$

$$s_{22} = \begin{pmatrix} -\frac{\sqrt{2001}}{500} - \frac{1}{500} \\ \frac{\sqrt{2001}}{500} - \frac{1}{500} \\ -\frac{\sqrt{3001}}{500} - \frac{1}{500} \\ \frac{\sqrt{3001}}{500} - \frac{1}{500} \end{pmatrix}$$

**Step4:** Compute  $K = R^{-1}B^T S$  Rearranging the solutions to be written in the form:

$$S = \begin{bmatrix} s_{11} & s_{12} \\ s_{12} & s_{22} \end{bmatrix}$$

$$S_1 = \begin{bmatrix} \frac{\sqrt{2001}}{50} & -\frac{1}{10} \\ -\frac{1}{10} & -\frac{\sqrt{2001}}{500} - \frac{1}{500} \end{bmatrix}$$

$$S_2 = \begin{bmatrix} -\frac{\sqrt{2001}}{50} & -\frac{1}{10} \\ -\frac{1}{10} & \frac{\sqrt{2001}}{500} - \frac{1}{500} \end{bmatrix}$$

$$S_3 = \begin{bmatrix} -\frac{\sqrt{3001}}{50} & \frac{1}{10} \\ \frac{1}{10} & -\frac{\sqrt{3001}}{500} - \frac{1}{500} \end{bmatrix}$$

$$S_4 = \begin{bmatrix} \frac{\sqrt{3001}}{50} & \frac{1}{10} \\ \frac{1}{10} & \frac{\sqrt{3001}}{500} - \frac{1}{500} \end{bmatrix}$$

for each of the solutions, compute  $K$

$$K_1 = R^{-1}B^T S_1 = [-10.0 \quad -9.14651]$$

$$K_2 = R^{-1}B^T S_2 = [-10.0 \quad 8.74651]$$

$$K_3 = R^{-1}B^T S_3 = [10.0 \quad -11.1563]$$

$$K_4 = R^{-1}B^T S_4 = [10.0 \quad 10.7563]$$

**Step5:** Compute Eigenvalues of the closed-loop system for each of the gains  $K_i$   
 $(\text{lambda1} = \text{eig}(A - B^*K1))$

$$\begin{aligned}\lambda_1 &= \det(\lambda I - A + BK1) = 0 \implies \lambda_1 = [9.95 \quad -1.00] \\ \lambda_2 &= \det(\lambda I - A + BK2) = 0 \implies \lambda_2 = [-9.95 \quad 1.00] \\ \lambda_3 &= \det(\lambda I - A + BK3) = 0 \implies \lambda_3 = [9.95 \quad 1.00] \\ \lambda_4 &= \det(\lambda I - A + BK4) = 0 \implies \lambda_4 = [-9.95 \quad -1.00]\end{aligned}$$

From the above results we can see that the the Eigenvalues of both negative for  $\lambda_4$  therefore, the system is stable in this case and hence, the full state-feedback control gain  $K$  is  $K_4$ . Doing this in Matlab just by using the "lqr(A,B,Q,R)" gives back the direct result of  $K_4$ . This procedure is only for demonstration purposes on how full state-feedback control gain  $K$  is selected using the regorous algebraic procedure. In practice using only the command "lqr(A,B,Q,R)" the gains can be selected. Additionally, the solution to algebraic-Riccati equation as well as the Eigenvalues of the closed-loop system can be determined directly using Matlab command "[K,S,E] = lqr(A,B,Q,R)"

### 24.5.1 Choosing penalty matrices

The tuning of LQR is to choose the weighting matrices  $Q$  and  $R$ . To guarantee that the solution exists, the system must be controllable and that  $Q \geq 0$  and  $R > 0$ .

1. The simplest choice is using Identity matrices, such that  $Q = I$  and  $R = \rho I$ . Here, the control weighting matrix is penalized based on the control expense.  $\rho$  serves as the only variable that can be varied and the cost of controller can be minimized. This kind of choosing strategy reduces the choice to be applied equally to all states and control inputs. This strategy is a trade-off between states costs and control costs in whole, as increase  $\rho$  will decrease the control action and vice-versa.
2. Output weighting: In this strategy, a state for which the output to be reduced is weighed individually such that if  $z = C_z x$  be the output that is needed to be kept small. Choose  $Q = C_z^T C_z$  and  $R = \rho I$ .
3. Diagonal weighting: In this strategy each of the individual states and control inputs can be weighted independently by diagonalizing the weighting matrices  $Q$  and  $R$  such that

$$Q = \begin{bmatrix} q_1 & 0 & 0 \\ 0 & q_2 & 0 \\ 0 & 0 & q_3 \end{bmatrix}$$

$$R = \begin{bmatrix} \rho_1 & 0 & 0 \\ 0 & \rho_2 & 0 \\ 0 & 0 & \rho_3 \end{bmatrix}$$

## Nomenclature

Choose the individual diagonal elements based on how much each state or input signal should contribute to the overall cost. There is also a way to choose these individual weightings based on a rule called **Bryson's rule** where the diagonal weights as  $q_i = \alpha_i^2/x_{i,max}^2$  and  $\rho = \beta_i^2/u_{i,max}^2$  where  $x_{i,max}$  and  $u_{i,max}$  represents the largest response.  $\alpha$  and  $\beta$  are used for additional individual weightings of the state and control cost

$$\sum_{i=1}^n \alpha_i^2 = 1 \quad \sum_{i=1}^p \beta_i^2 = 1$$

4. Finally, a trial and error method can also be used.

# 25 Controllability and Observability

## 25.1 Stability of State-space systems

In this section a formal definition of stability of systems in the context of state-space models are provided. Initially defining the equilibrium for a system, a core test to check if there exists an equilibrium point for a given system such that a state remains in this point once there and also (for stability) converges to that the point from anywhere near this neighborhood.

### 25.1.1 Equilibrium

In the most obvious definition for equilibrium is that the state dynamics are zero at the equilibrium point such that the derivative of the states are zero, which is as expressed:

$$\dot{x} = 0 = Ax + Bu \quad (25.1)$$

this follows the condition that the update of a state at the next instance is the same position of the state in the state configuration such that

$$x_{k+1} = x_k \quad (25.2)$$

which simply means that the states do not evolve. To arrive at the equilibrium point if the system also has inputs  $u$  to it, then different values of  $u$  will lead to different equilibrium points of the system. However, for the sake of simplicity, the control input  $u$  at the equilibrium point is considered to be zero. As this is a LTI system, for any other values of  $u$ , the systems response can be super-imposed later.

### 25.1.2 Stable Equilibrium

A stable equilibrium is defined using **norms** of a vector  $\|x(t)\|$  and establishes that once a state is within a certain distance from the equilibrium point, thereafter, that state will never go farther away from a certain relative distance. This definition does not allow for convergence but it does allow for the boundedness such as in the case of oscillations.

**Note:** A norm of a vector defines the radial distance from the origin, in this case, the origin being the equilibrium point. Therefore, using norm the definition of state's equilibrium can be established.

## Nomenclature

Mathematically, stable equilibrium can be expressed as:

$$\|x(t)\| \leq \delta \implies \|x(t + t_1)\| \leq r\delta, \quad t > 0 \quad (25.3)$$

where  $r$  is a scaling factor (radial position of the boundary  $\delta$ ) to be determined for a given system. In case of **linear** systems,  $r$  is independent of the magnitude  $x$ . The above definitions allows for permanent oscillations without convergence.

### 25.1.3 Asymptotic stability

This case is similar to stable equilibrium with the difference that in this case the definition insists on convergence such that

$$\lim_{t \rightarrow \infty} \|x(t)\| = 0 \quad (25.4)$$

Therefore, by the definition, pure oscillation modes or sinusoids are not allowed if the system is to be defined asymptotically stable.

### 25.1.4 BIBO

This case is defined when using control inputs such that, assuming the inputs are bounded beneath some boundary value  $\delta$ , it can be guaranteed that the states are also beneath some boundary value  $\epsilon$ . The boundary values just like in previous cases can be expressed using the norm of the vector.

$$\|u(t)\| \leq \delta, \forall t \geq 0 \implies \|x(t)\| \leq \epsilon, \forall t \geq 0 \quad (25.5)$$

This definition now allows for system with an input to be within well defined boundary of equilibrium without being blown-up. In general, the norms may depend upon the initial condition, but for linear systems, one could make these relationships more precise if desired.

**Note:** In general, for analyzing the stability of forced systems, BIBO definition is used and for analyzing stability of free systems, definition of Asymptotic stability is used.

## 25.2 Defining controllability

In this example a discrete system is considered because using a discrete system it would be fairly easier to define the idea of controllability of the system. In any case, the resulting solutions apply the same to continuous systems as well.

Consider a linear discrete system given by,

$$\dot{\mathbf{x}}_k = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k u_k \quad (25.6)$$

at an initial condition  $x_0 = 0$ . If this system is completely controllable, such that it can be driven some arbitrary point  $x^*$ , then a map of all the states and control inputs can be made for each of the discrete step as following,

$$\begin{aligned}\mathbf{x}_1 &= \mathbf{A}_0 x_0 + \mathbf{B}_0 u_0 = \mathbf{B}_0 u_0 \\ \mathbf{x}_2 &= \mathbf{A}_1 x_1 + \mathbf{B}_1 u_1 = \mathbf{A} \mathbf{B}_0 u_0 + \mathbf{B}_1 u_1 \\ \mathbf{x}_3 &= \mathbf{A}_2 x_2 + \mathbf{B}_2 u_2 = \mathbf{A}^2 \mathbf{B}_0 u_0 + \mathbf{A} \mathbf{B}_1 u_1 + \mathbf{B}_2 u_2\end{aligned}$$

Performing the similar calculations up until  $n^{th}$  iteration when  $x$  reaches the final destination  $x^*$ ,

$$\mathbf{x}_n = \mathbf{A}^{n-1} \mathbf{B} u_0 + \dots + \mathbf{B} u_{n-1} \quad (25.7)$$

therefore, in general it can be said that in order to reach a final point  $x^*$ , the linear controllable model can be expressed in matrix form as

$$x^* = [\mathbf{B} \quad \mathbf{AB} \quad \dots \quad \mathbf{A}^{n-1} \mathbf{B}] \begin{bmatrix} u_{n-1} \\ \vdots \\ u_1 \\ u_0 \end{bmatrix} \quad (25.8)$$

Lets consider the matrix  $[\mathbf{B} \quad \mathbf{AB} \quad \dots \quad \mathbf{A}^{n-1} \mathbf{B}] = \Gamma$ , controllability is possible, iff  $\text{rank}(\Gamma) = n$ . That is to say that, controllability is only possible iff there are  $n$  independent columns in matrix  $\Gamma$ . In other words, controllability is only possible iff each of the state can be placed using control inputs  $(u_1, u_2, \dots, u_n)$  individually without being affected by any other placement of poles (states) by other control inputs.

Following this theory, the following theorems can be laid down concerning controllability as follows, ***Completely controllable (CC) Theorem 1:*** The system is CC if and only if  $\text{rank}(\Gamma) = n$ . ***Completely controllable (CC) Theorem 2:*** Pole placement to arbitrary eigenvalues is possible if and only if the system is CC.

## 25.3 Controllability vs Trajectories

Consider an arbitrary system,

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad (25.9)$$

Using Matlab, it can be found that the rank of the controllability matrix  $G = \text{rank}(\text{ctrb}(A, B)) = 2$ , which is also equal to  $n$ . Therefore, this system is CC. However, note that a CC system does not follow any arbitrary trajectory to achieve the goal from  $x$  to  $x^*$ . Consider

### Nomenclature

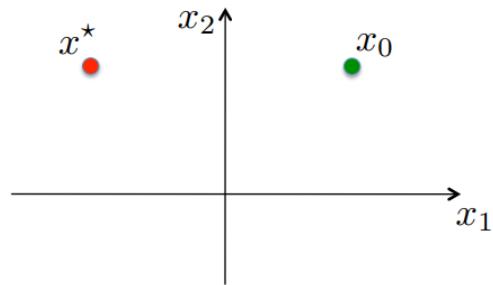


Figure 25.1: A model for trajectory vs Controllability

figure 25.1 where  $x_1$  and  $x_2$  are the states for positions and velocities respectively. Notice from figure 25.2, that is not possible to simply go from position  $x$  to  $x^*$  in this fashion. As by doing so, the state  $x_2$  ie., the velocity is either a constant or positive, in both of such cases, it is not possible to move to a negative position by having a positive velocity. Therefore, this trajectory is not possible.

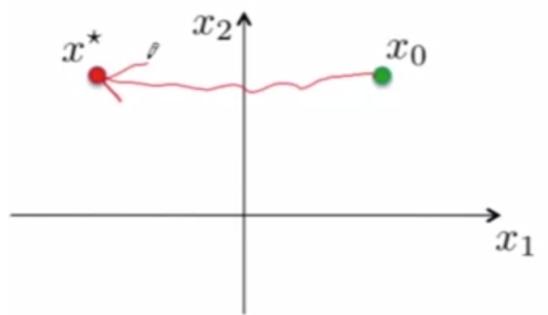


Figure 25.2: Non possible trajectory

Further consider, figure 25.3, this would one of the possible trajectories that this system can follow.

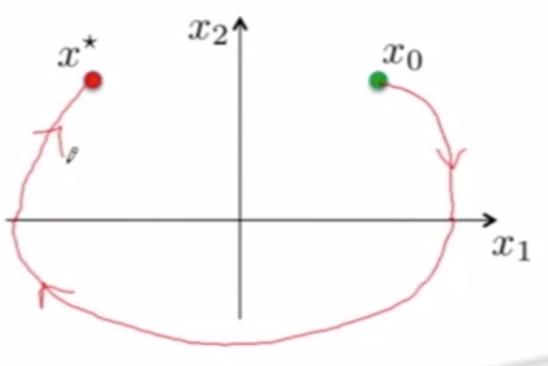


Figure 25.3: Possible trajectory

*Nomenclature*

# 26 State Estimation

## 26.1 Principle of State estimation

The design of a State-feedback controller assumes that all the states required for calculating the controller gains are readily available. However, this is rarely the case, as it would either be physically impossible to fit a sensor or there may be no sensor available at all to measure a quantity. State estimation is a way of getting access to all such states that are not feasible for sensing. In automotive industry this concept is sometimes referred to as *Soft Sensors*, when software is used to measure a state instead of a hardware.

Recall, from state-feedback design, all the states of the system are measured and controller gains are designed so as to modify each of system's Eigenvalues which in-turn modifies the systems dynamics to required behavior as shown in figure 26.1. The design principle can be visualized as shown in figure 26.1

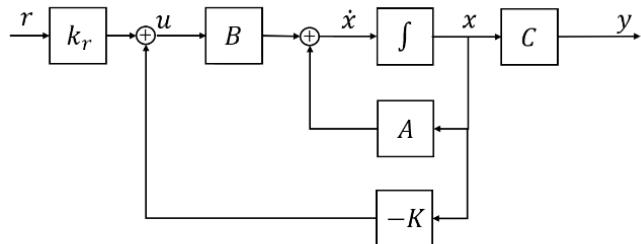


Figure 26.1: State-feedback control design principle

In order to estimate each of the states required by the system given in figure 26.1, the idea of state estimation is to develop an another system called an observer that provides an estimate of dynamical system's states as shown in figure 26.2. Further to improve the estimations, a feedback of the original dynamical system's output can be fed into the observer also shown in figure 26.2. By using a feedback to the observer, the error dynamics between the observer and dynamical system can be mapped and driven to zero so as to make the estimation more accurate.

## Nomenclature

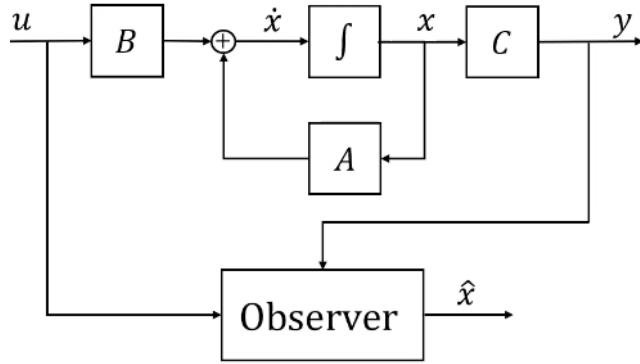


Figure 26.2: State estimation principle with observer design

## 26.2 Goal of State estimation

Let  $\hat{x}$  is the estimated states and  $x$  is the actual states, such that the error between them is expressed as,

$$e = x - \hat{x} \quad (26.1)$$

the error dynamics can be expressed as,

$$\dot{e} = \dot{x} - \dot{\hat{x}} \quad (26.2)$$

The goal of state estimation can then be formulated as

$$\dot{e} = \dot{x} - \dot{\hat{x}} = 0$$

the idea is to drive the error between the actual states and the derived states to zero such that we get a replica of the actual states. Such that as  $\dot{e} = \dot{x} - \dot{\hat{x}} = 0$ ,  $\hat{x} \rightarrow x$ .

In general, the following approach is adopted in order to design a observer and derive required states from it. Consider figure 26.3, Only the outputs from the system can

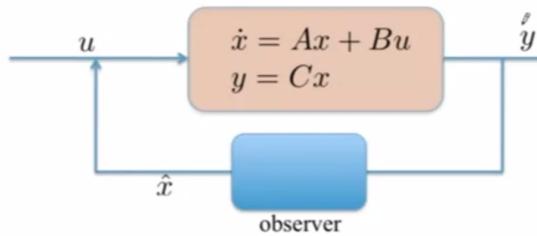


Figure 26.3: Generic Observer Design

be measured using sensors, the idea is to figure out the actual system states using the

sensed system outputs. As shown in figure 26.3, the outputs of the system is fed into a device, here so called an observer, based on the sensed outputs, system states are derived.

**Note:** If the observer design does not estimate all the required system states, then a new set of sensors has to be added to the system. New sensed outputs will then drive the system to be completely observable as described in 26.4.

## 26.3 Observers design

### 26.3.1 Theory

Once observability has been established, the idea of state-estimation using an observer design is to use the system inputs and the outputs of the original system to estimate the states of the original system. In essence, the idea is to use the copy of the original linear dynamical system that uses  $u$  as well and  $y$  from the original system and has its states  $\hat{x}$  which estimates the states of the original system as follows

$$\frac{d\hat{x}}{dt} = F\hat{x} + Gu + Hy \quad (26.3)$$

Lets begin with a simple observer design, starting for only input  $u$  as shown in figure 26.4, therefore the observer model is expressed as,

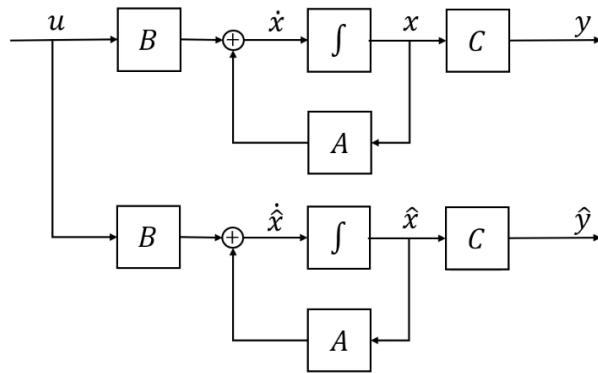


Figure 26.4: Open-loop observer design

$$\frac{d\hat{x}}{dt} = A\hat{x} + Bu \quad (26.4)$$

As described in section 26.2, the goal of an observer is to eliminate the error in the prediction, such that

$$e = x - \hat{x} = 0 \quad (26.5)$$

## Nomenclature

(this error dynamics is also called as ***Estimator Dynamics***)analyzing error dynamics of this open-loop observer design,

$$\dot{\tilde{x}} = \dot{x} - \hat{x} = Ax + Bu - A\hat{x} - Bu = Ax - A\hat{x} = A(x - \hat{x}) = A\tilde{x} \quad (26.6)$$

from the above equation, it can be seen that the error dynamics of the state estimation is completely dependent on the actual error itself and independent of inputs to the system. In order to determine the behavior of the error dynamics given by the system above, the solution to  $\dot{\tilde{x}} = A\tilde{x}$  has to be found. The system  $\dot{\tilde{x}} = A\tilde{x}$  is in standard canonical form, the solution to which is expressed as  $\tilde{x} = e^{At}\tilde{x}(0) = e^{At}(x(0) - \hat{x}(0))$ . This says that the error response will be dependent on the eigenvalues of  $A$  as well as the initial conditions. Therefore, if  $A$  has eigenvalues in positive half-plane then this open-loop observer design system will be un-stable. Which means that the estimation of states would not be appropriate to be used in most / any of the systems at all.

Now, consider another system with feedback. Because feedback has earlier proven to be effective by shifting the eigenvalues of the system towards the negative half-plane. A closed-loop observer design is employed using output feedback to the observer design as shown in figure 26.5 In case of closed-loop observer design, the estimated

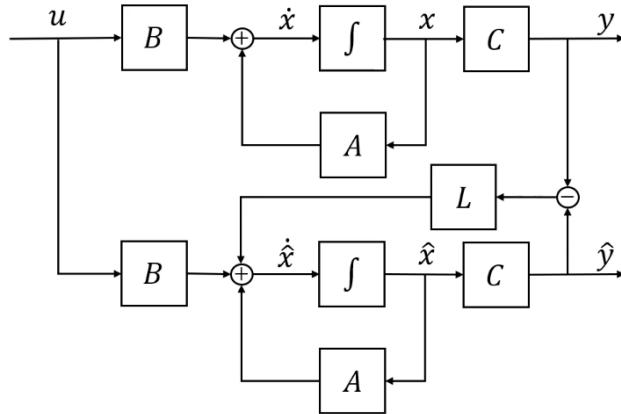


Figure 26.5: Closed-loop observer design

output and the measured output are first compared  $y - \hat{y}$  and then fed back into the observer design using a feedback gain  $L$ , so that  $L$  can be adjusted to shift the eigenvalues towards the left half-plane. In this case now, the new observer can be modeled using figure 26.5 as following

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (26.7)$$

now the system (26.7) is finally appearing to be modeled according to (26.3). Further, analyzing error dynamics similar to the open-loop observer design,

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}}$$

$$\begin{aligned}
&= Ax + Bu - A\hat{x} - Bu - L\tilde{y} \\
&= A(x - \hat{x}) - L(y - \hat{y}) \\
&= A(x - \hat{x}) - L(Cx - C\hat{x}) \\
&= A\tilde{x} - LC\tilde{x} \\
&= (A - LC)\tilde{x}
\end{aligned}$$

from the above result, it can be seen that the closed-loop observer design give a possibility to tune the eigenvalues of the system which would otherwise be unstable because of the eigenvalues of  $A$ . Further, the response can be determined using the solution of the equation. The system  $\dot{\tilde{x}} = (A - LC)\tilde{x}$  is in standard canonical form, the solution to which is expressed as  $\tilde{x} = e^{(A-LC)t}(x(0) - \hat{x}(0))$ . With the system response given by

$$y = \mathbf{C}\hat{x} \quad (26.8)$$

### 26.3.2 State estimation gain

The way to choose state estimator gain is similar to that used fo control design. Using the closed loop estimator, the error dynamics becomes

$$\dot{\tilde{x}} = (A - LC)\tilde{x} \quad (26.9)$$

Therefore, now using system (26.7) it is now possible to choose feedback gain  $L$  such that the error dynamics converges to  $\tilde{x} \rightarrow 0$  as  $t \rightarrow \infty$ , (if observable).

In order to make the error dynamics go to zero, poles of the observer has to be selected such that the error states  $\tilde{x}$  go to zero. The closed-loop poles for this observer design can be found using,

$$\det(s - A + LC) = 0 \quad (26.10)$$

### 26.3.3 Predictor-Corrector

Predictor-Corrector is essentially the same closed-loop observer design only modeled in a different way.

This is one of the simplest observer designs. Consider a system by ignoring the system inputs  $Bu$  such that the state dynamics and output be expressed by,

$$\dot{x} = Ax \quad (26.11)$$

$$y = Cx \quad (26.12)$$

The observer design is as follows,

- Predictor: Make a copy of the system such that the new system has the state dynamics  $\dot{\hat{x}} = A\hat{x}$

## Nomenclature

- Corrector: Next add the error between the copy systems output and the output of the actual system. The output of the actual system is given by  $y = Cx$  and that of copy system is  $\hat{y} = C\hat{x}$ . Since this is a linear system, the effects of error signal can be directly superimposed into the system as follows,

$$\dot{\hat{x}} = A\hat{x} + (y - C\hat{x}) \quad (26.13)$$

- Additionally, also add a scaling factor, a gain  $L$  so that the error signal  $e = (y - C\hat{x})$  can be adjusted, such that it can be driven to zero. Therefore,

$$\dot{\hat{x}} = A\hat{x} + L(y - C\hat{x}) \quad (26.14)$$

- The term  $e = (y - C\hat{x})$  from (26.14) is the estimation error, with the goal of driving this error to zero, such that  $e = x - \hat{x}$ , the idea is to drive the dynamics of error to zero. In this case,

$$\dot{e} = \dot{x} - \dot{\hat{x}} = Ax - A\hat{x} - L(y - C\hat{x}) \quad (26.15)$$

if  $\dot{e}$  needs to be driven to zero, the equation (26.15) can be stabilized asymptotically such that, the dynamics are driven to zero asymptotically. The only difference here would be that, in this case, the asymptotic stability will be when the dynamics is towards zero.

Further, simplify (26.15),

$$\dot{e} = A(x - \hat{x}) - LC(x - \hat{x}) = (A - LC)e \quad (26.16)$$

- It can be seen that using the error dynamics equation (26.15), the error of the observer can be directly diminished to zero without even considering the determined states  $\hat{x}$ .
- In the final step, consider the matrix  $(A - LC)$  there is only one variable in this matrix, that is  $L$ ,  $L$  can be adjusted appropriately by choosing appropriate eigenvalues such that the error dynamics diminish to zero. For the simplest case, a pole placement method can be used.

### 26.3.4 Example of observer design

Consider a system,

$$\dot{x} = \begin{bmatrix} -1 & 2 \\ 0 & -2 \end{bmatrix} x, \quad y = [1 \quad -1]x \quad (26.17)$$

The error dynamics can be determined using the expression (26.15)

$$\dot{e} = (A - LC)e$$

here  $A$  is of size  $(2 \times 2)$ ,  $C$  is of the size  $(1 \times 2)$ , therefore,  $L$  should be sized to  $(2 \times 1)$  as  $[L_1 \ L_2]^T$

$$\begin{aligned}\dot{e} &= \left( \begin{bmatrix} -1 & 2 \\ 0 & -2 \end{bmatrix} - \begin{bmatrix} L_1 \\ L_2 \end{bmatrix} [1 \quad -1] \right) e \\ &= \begin{bmatrix} -1 - L_1 & 2 + L_1 \\ -L_2 & -2 + L_2 \end{bmatrix} e\end{aligned}$$

for pole placements, now the characteristic equation (23.9)  $sI - (A - LC)$  in this case,

$$\begin{aligned}\det(sI - (A - LC)) &= \det \left( \begin{bmatrix} \lambda + 1 + L_1 & -2 - L_1 \\ L_2 & \lambda + 2 - L_2 \end{bmatrix} \right) \\ &= \lambda^2 + \lambda(3 + L_1 - L_2) + 2 + 2L_1 + L_2\end{aligned}$$

choosing a characteristic equation  $p_{des} = \lambda^2 + 2\lambda + 1$  for desired pole placements and matching the coefficients,

$$\lambda(3 + L_1 - L_2) = 2 \quad (26.18)$$

$$2 + 2L_1 + L_2 = 1 \quad (26.19)$$

$L_1$  and  $L_2$  can be determined as,

$$L_1 = -2/3 \quad (26.20)$$

$$L_2 = 1/3 \quad (26.21)$$

Once  $L_1$  and  $L_2$  have been determined, the determined states dynamics can be determined using

$$\dot{\hat{x}} = A\hat{x} + \begin{bmatrix} -2/3 \\ 1/3 \end{bmatrix} (y - C\hat{x}) \quad (26.22)$$

### 26.3.5 Example observer design fo Single Track Model

Consider a single track model shown below which is modeled as per in section 20.4,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 12 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 6 \\ 3 \end{bmatrix} u \quad (26.23)$$

$$y = [1 \quad 0] \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \quad (26.24)$$

where  $x_1$  is the lateral position  $Y$ ,  $x_2$  is the heading orientation  $\theta$  and  $u$  is the steering angle  $\delta$ . The estimator design requirement is to design an observer such that it uses the measurement of vehicles lateral position to estimate vehicle's states.

The requirement that the observer should stabilize the estimated states earlier than the controller can use them, the poles of the observer are placed at the negative LHP as

## Nomenclature

$\{s_1, s_2\} = \{-6, -4\}$ . Therefore, a desired characteristic polynomial can be expressed as

$$p_{des}\lambda = (\lambda + 6)(\lambda + 4) = \lambda^2 + 10\lambda + 24 \quad (26.25)$$

The closed-loop observer design is chosen as given in (26.7) where the estimated states are determined by

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (26.26)$$

and the estimator dynamics (error dynamics) are given by the characteristic equation

$$\tilde{\dot{x}} = (A - LC)\hat{x} = \left( \begin{bmatrix} 0 & 12 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} l_1 \\ l_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix} \right) \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} \quad (26.27)$$

here,  $C$  is of size  $1 \times 2$ , the required size of  $LC$  is  $2 \times 2$  therefore,  $L$  should be chosen in size  $2 \times 1$

Simplifying, the above equation

$$\tilde{\dot{x}} = \left( \begin{bmatrix} 0 & 12 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} l_1 & 0 \\ l_2 & 0 \end{bmatrix} \right) \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \left( \begin{bmatrix} -l_1 & 12 \\ -l_2 & 0 \end{bmatrix} \right) \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} \quad (26.28)$$

the poles of this estimator is given by

$$\det(sI - A + LC) = \lambda^2 + l_1\lambda + 12l_2 \quad (26.29)$$

for the observer design, these poles should match the required polynomial in (26.25)

$$\lambda^2 + 10\lambda + 24 = \lambda^2 + l_1\lambda + 12l_2 \implies l_1 = 10 \quad l_2 = 2 \quad (26.30)$$

Finally, plugging in the values of observer gains into the closed-loop observer,

$$\dot{\hat{x}} = \begin{bmatrix} 0 & 12 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 6 \\ 3 \end{bmatrix} u + \begin{bmatrix} 10 \\ 2 \end{bmatrix} (y - \hat{x}_1) \quad (26.31)$$

### 26.3.6 Control using the estimated states

From the observer gains now determined, estimated states are derived using (26.26) and used for the feedback control design  $u = -K\hat{x} + k_r r$ . The control algorithm using the controller itself and the observer as a whole now forms the whole of the controller (developed as software) of the system as shown by the highlighted figure 26.6 At this point of system design, now we have a given system  $\dot{x} = Ax + Bu$  with output  $y = Cx$  and the feedback controller  $u = -K\hat{x} + k_r r$ . Additionally with a state estimator  $\hat{x} = A\hat{x} + Bu + L(y - C\hat{x})$ , for which the eigenvalues are adjusted such that the error dynamics  $\tilde{\dot{x}} = (A - LC)\tilde{x}$ . The complete closed-loop system can be written as

$$\begin{bmatrix} \dot{x} \\ \tilde{\dot{x}} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - CL \end{bmatrix} \begin{bmatrix} x \\ \tilde{x} \end{bmatrix} \begin{bmatrix} Bk_r \\ 0 \end{bmatrix} \quad (26.32)$$

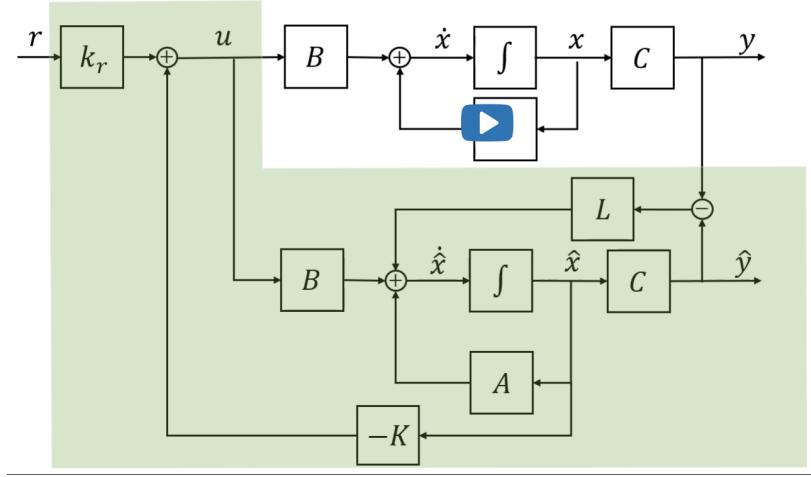


Figure 26.6: Control algorithm using estimated states

in (26.32), the states  $x$  are replaced with the estimated states  $\hat{x}$ . There are two closed-loops in this system with the closed loop poles given by  $\lambda(s)$ . Each of these closed-loop poles are determined separately and as a system as a whole can be expressed as

$$\lambda(s) = \det(sI - A + BK) \quad \det(sI - A + LC) \quad (26.33)$$

provided that this system is both controllable and observable, the polynomial (26.33) can be assigned roots such that the eigenvalues for each of the closed-loop can be adjusted to stabilize the system. In general, the poles of observer are chosen such that they are 4-5 times faster than the poles of feedback controller. As an observer is only a piece of software code and no actuator, the observer cannot be simply saturated with high observer gains. Therefore, this system is practical.

### 26.3.7 Full System with Observer and Feedback controller

Consider the observer design as describe in section 26.3.5 for a single track vehicle model. For the observer poles placed at  $\{s_1, s_2\} = \{-4, -6\}$ , the observer gains  $\{l_1, l_2\}$  were found to be  $\{10, 2\}$ .

Consider a feedback controller with a double pole placed at  $\{s_1, s_2\} = \{-1, -1\}$  for which the controller gains  $\{k_1, k_2\} = \{0.0278, 0.6111\}$ , additionally with the steady-state reference tracking gain  $k_r = 0.0278$  therefore the control law can be written as,

$$u = -Kx + k_r r = -0.0278x_1 - 0.6111x_2 + 0.0278r \quad (26.34)$$

the full system with both the closed loop controller and observer are expressed as,

$$\dot{\hat{x}} = Ax + Bu + L(y - C\hat{x}) = \begin{bmatrix} 0 & 12 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 6 \\ 3 \end{bmatrix} u + \begin{bmatrix} 10 \\ 2 \end{bmatrix} (y - \hat{x}_1) \quad (26.35)$$

### 26.3.8 Implementation - Digital Computer Language

The system developed so far needs to be transformed into a discrete system before being implemented as a software for the computer. From the system developed in the previous section, the state-estimator can be developed in discrete form using a first order Euler's approximation in this case as

$$\dot{\hat{x}} \approx \frac{\hat{x}(t_{k+1}) - \hat{x}(t_k)}{t_{k+1} - t_k} = A\hat{x}(t_k) + Bu(t_k) + L(y(t_k) - C\hat{x}(t_k)) \quad (26.36)$$

the next state can be derived by rearranging the above equation as

$$\hat{x}(t_{k+1}) = \hat{x}(t_k) + (t_{k+1} - t_k)(A\hat{x}(t_k) + Bu(t_k) + L(y(t_k) - C\hat{x}(t_k))) \quad (26.37)$$

where  $(t_{k+1} - t_k) = h$  is the sampling time. In pseudo-code, the control algorithm can be expressed as,

```
// control algorithm - main loop
r = digital_read(ch1) // read reference
y = diital_read(ch2) // read process outputs
// from previous computed state and r
u = -K*xhat + k_r*r // compute control variable
analog_out(ch1, u) // set the analog output
/* (ch1 is the summation point both for reference
and control variable) */
xhat = xhat + h*(A*x + B*u + L*(y-C*x))
// update next state
```

## 26.4 Observability

**Definition:** A linear system is observable if for every  $T > 0$ , it is possible to determine the state of the system  $x(t)$  through the measurements of  $y(t)$  and  $u(t)$  on the interval of  $[0, T]$ .

Similar to Controllability not being able to control any arbitrary system, not all systems can also be completely observable. In most of such cases, a rich set of sensor skirt has to be added additionally, such that it improves the output  $y$  and from there  $\hat{x}$  can be determined.

Consider a modest example of a discrete system with no input as given below,

$$x_{k+1} = Ax_k \quad (26.38)$$

$$y_k = Cx_k \quad (26.39)$$

The question is, can the initial state be recovered given  $n$  number of outputs,

$$y_0 = Cx_0 \quad (26.40)$$

$$y_1 = Cx_1 = CAx_0 \quad (26.41)$$

iterating this upto  $n$  number of outputs,

$$y_{n-1} = CA^{n-1}x_0$$

writing the above iteration in matrix form,

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \ddots \\ \ddots \\ CA^{n-1} \end{bmatrix} x_0 \quad (26.42)$$

here the matrix on the RHS is called the Observability matrix  $\Omega$ , with the system being observable only when  $\Omega$  has a full rank of  $n$ . If it is the case that with enough number of outputs, the initial states can be tracked that means, that each of the intermediate states from the current outputs upto the initial states can also be tracked.

#### 26.4.1 Observability Theorem 1

The system is completely observable (CO), if it is possible to recover the initial state from the output. Therefore, the system is CO iff  $\text{rank}(\Omega) = n$

#### 26.4.2 Observability Theorem 2

Pole-placement to arbitrary eigenvalues is possible iff the system is CO.

\*\*\*Note:\*\*\* With observer design, a close approximation of the states  $\hat{x}$  can be made, all by having only the outputs  $y$ .

### 26.5 Separation Principle

So far, in this book most of the fundamental tools have been developed from Controllability, Observability, State feedback, Observers and Pole-Placement technique. In practice for example, pole-placement is a technique used to stabilize the dynamics of both the controller and the observer. However, each of them is dependent on another, a controller needs the observer to estimate the states of the system such that  $\hat{x} \approx x$ . Only a fairly estimated states by the observer can lead to the controller to work properly as expected. On the other hand, the observer also needs the inputs from the controller for the states estimation. So it appears that there are two dependent systems the inputs of each are dependent on the outputs from the another.

This complication is solved using the Separation principle.

### 26.5.1 Method of separation principle

Assume a system that is completely controllable and completely observable, given by

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx\end{aligned}$$

the separation principle works in the following steps,

1. Design the state feedback controller as if full set of  $x$  states were already known (currently which are unknown) Using this assumption, the controller can be designed such that

$$u = -Kx \implies \dot{x} = (A - BK)x$$

As  $\hat{x}$  instead of  $x$  is known in reality, it is better to use the controller design,

$$u = -K\hat{x} \implies \dot{x} = (A - BK)\hat{x}$$

This completes the required control design, now observer design can be proceeded with the observer design

2. Estimate  $x$  using an observer (that now also contains  $u$ ) Previously, in the Predictor-Corrector observer design,  $u$  was ignored as in a linear system such external signals can be added at anytime. Consider a system with  $u$  inputs,

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

where  $A\hat{x} + Bu$  is the predictor part and  $L(y - C\hat{x})$  is the corrector part of the observer. The error dynamics remains the same as,

$$\dot{e} = (A - LC)e$$

3. The dynamics of both  $x$  and  $e$  needs to be stabilized, therefore in the separation principle both  $\dot{x}$  and  $\dot{e}$  are considered together for analyzing the dynamics as,

$$\dot{x} = Ax - BK\hat{x} = Ax - BK(x - e) = (A - BK)x + BKe \quad (26.43)$$

here, as the controller is always designed for the estimated states  $\hat{x}$ , therefore in (26.43)  $Bu = -K\hat{x}$ . Also, using the relationship  $e = (x - \hat{x})$ , (26.43) is replaced with  $e$  so as to write the following equations.

$$\begin{bmatrix} \dot{x} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A - BK & BK \\ 0 & A - LC \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} \quad (26.44)$$

The strategy of separation principle works if and only if, (26.44) is an asymptotically stable system. Observe, that the matrix in the RHS of (26.44) is a upper triangular block matrix, with the blocks of matrices  $A - BK$  and  $A - LC$  along the diagonals. In such a triangular block matrix, the eigenvalues of the entire matrix are simply the eigenvalues of the diagonal blocks. Such that, if this whole triangular matrix is represented by  $m$ , then

$$\det(m) = \det(A - BK)\det(A - LC) \quad (26.45)$$

### 26.5.2 Choosing eigenvalues for controller and observer

In general, as the controller is dependent on the correct prediction of the observer to correctly control the system as desired, it is always should be the case that the observer poles are placed such that the slowest pole of the observer is always much faster than the controller poles. For example, the poles are arbitrarily placed as shown in figure 26.7

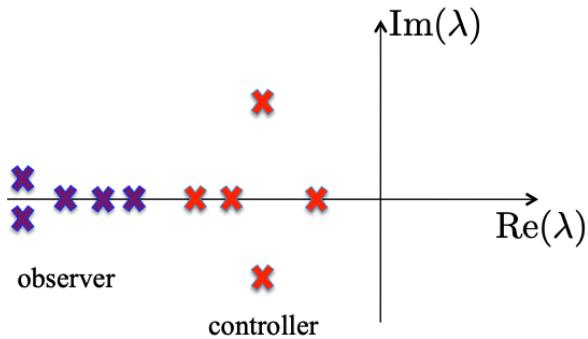


Figure 26.7: Poles of controller and observer

## 26.6 Kalman Filter

Consider again the linear time invariant system which is added with process disturbances and measurement noise given by:

$$\dot{x} = Ax + Bu + v \quad (26.46)$$

$$y = Cx + w \quad (26.47)$$

where  $v$  and  $w$  are stochastically added process disturbances and measurement noise, included into the system so as to define the use of a filter. Earlier, a feedback observer design was used which had the state estimator dynamics as (26.7)

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x}) \quad (26.48)$$

as now the process disturbances and measurement noise are included into the system, the error  $\tilde{x} = x - \hat{x}$  will be effected by these parameters and the error dynamics can be determined using the the equation

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}} \quad (26.49)$$

substituting for  $\dot{x}$  and  $y$  from the equations stated above, the following equation is derived

$$\dot{\tilde{x}} = Ax + Bu + v - A\hat{x} - Bu - L(Cx + w - C\hat{x})$$

## Nomenclature

$$\begin{aligned}
&= Ax + v - A\hat{x} - L(Cx + w - C\hat{x}) \\
&= Ax - A\hat{x} - LC(x - \hat{x}) + v - Lw \\
&= A\tilde{x} - LC\tilde{x} + v - Lw \\
&= (A - LC)\tilde{x} + v - Lw
\end{aligned}$$

From the above expression it can be seen that the error dynamics is characterized by the system matrix  $(A - LC)$ , therefore, if  $(A - LC)$  is stable, then the estimation error becomes stable. In this case, since a stochastic process is included into the system, the error here becomes **stationary stochastic process**. Using covariance, the error can be linearly transformed using  $P_{\tilde{x}} = \mathbb{E}(\tilde{x}(t)\tilde{x}^T(t))$ , using this formulation, the covariance of the estimation error is given by the following equation

$$0 = (A - LC)P_{\tilde{x}} + P_{\tilde{x}}(A - LC)^T + R_v + LR_wL^T \quad (26.50)$$

the idea here is to minimize the covariance matrix ( $P_{\tilde{x}}$ ) using matrix  $L$ . The optimal observer therefore, minimizes  $P_{\tilde{x}}$ .

\*\*\*Note\*\*\* If the system is time-varying, the system will also be time dependent.

### 26.6.0.1 Kalman - Bucy Filter

Kalman - Bucy discovered that when the system is observable, the optimal observer gain matrix  $L$  is given by

$$L = P_{\tilde{x}}C^T R_w^{-1} \quad (26.51)$$

where  $P_{\tilde{x}} = P_{\tilde{x}}^T \geq 0$  is the solution to the Riccati equation

$$0 = AP_{\tilde{x}} + P_{\tilde{x}}A^T + R_v - P_{\tilde{x}}C^T R_w L^{-1} C P_{\tilde{x}} \quad (26.52)$$

This observer is called the **Kalman - Bucy Filter** which has the following characteristics

- always stable
- the optimal linear filter for state estimation
- $R_v$  and  $R_w$  are regarded as the design parameters.  $R_v$  can be decided using the sensor characteristics.

There are similarities between Kalman Filter and LQR, in-fact, using them both in the controller design **linear quadratic Gaussian controller**.

### 26.6.1 Vehicle Steering Example

Consider the steering model which is modeled as per in section 20.4, added with stochastic processes  $v$  and  $w$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 12 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 6 \\ 3 \end{bmatrix} u + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \quad (26.53)$$

$$y = [1 \ 0] \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + w \quad (26.54)$$

The process disturbance and the measurement noise are zero mean with covariance (linear transformation of disturbances)

$$R_v = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (26.55)$$

$$R_w = \rho \quad (26.56)$$

The idea is to design a Kalman filter to estimate the vehicles states, from the measurement of the lateral position. The standard state estimator equation is used ( $\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$ ) with  $L$  as the optimal observer gain given by the Kalman filter equation  $L = P_{\tilde{x}}C^T R_w^{-1}$ . Where  $P_{\tilde{x}} = P_{\tilde{x}}^T \geq 0$  is the solution to the Riccati equation

$$0 = AP_{\tilde{x}} + P_{\tilde{x}}A^T + R_v - P_{\tilde{x}}C^T R_w L^{-1} C P_{\tilde{x}} \quad (26.57)$$

and, the solution to the Riccati equation is given by (for example)

$$\begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \quad (26.58)$$

inserting the solution  $P_{\tilde{x}}$  into the Riccati equation gives a non-linear matrix equality

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 12 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} + \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 12 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \rho^{-1} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} p_1 & p_2 \\ p_2 & p_3 \end{bmatrix} \quad (26.59)$$

this Riccati equation can be solved using Matlab **[P,E,L] = care(A',C',[1 0]; [0 1], 1)**, choosing noise covariance of  $\rho = 1$ , gives the following result

$$P_{\tilde{x}} = \begin{bmatrix} 5.0 & 1.0 \\ 1.0 & 0.4167 \end{bmatrix}$$

and

$$L = \begin{bmatrix} 5.0 \\ 1.0 \end{bmatrix}$$

## 26.6.2 Matlab's Kalman Filter

The whole idea of Kalman Filter is to used a measurement output to determine the internal states of the system. This is an idea of observing the external behavior to predict the internal states. The prediction in the control system happens using a combination of both a mathematical model which pre-predicts the next state as all the possible sates are mapped out mathematically in prior along with measurements (in most cases some in-direct variable is measured).

The output from the mathematical model is compared with the output from the measurements, if both the outputs match, then the estimated states using the mathematical

## Nomenclature

model is exactly same as the original states. If not there is certainly a discrepancy. This discrepancy is reduced using a feedback loop similar to a feedback control system, where the error is driven to zero using a feedback gain  $K$ . In this case, a feedback gain  $L$  is used to drive the estimation error to zero. In both these cases, the eigenvalues of the estimator system are pushed the negative half plane, so as to drive the error exponentially to zero.

### 26.6.2.1 Kalman filter for a single state system

A system is considered where the position of a vehicle is measured in 1D, which is expressed in discrete state space as

$$x_k = Ax_{k-1} + Bu_k + w_k \quad (26.60)$$

$$y_k = Cx_k + v_k \quad (26.61)$$

where  $w_k$  and  $v_k$  are introduced as process and sensor disturbances. These process disturbances are defined using a Gaussian distribution and represent as  $v \approx N(0, R)$ . Which means that all the sensor reading are distributed about the point 0 with a covariance of  $R$ .  $R$  is the area under which 68.3% of all the sensor readings are distributed about the point 0. The graph pf which looks as follows 26.8. Similarly, the process disturbance is modeled as a Gaussian distribution with a covariance of  $Q$ .

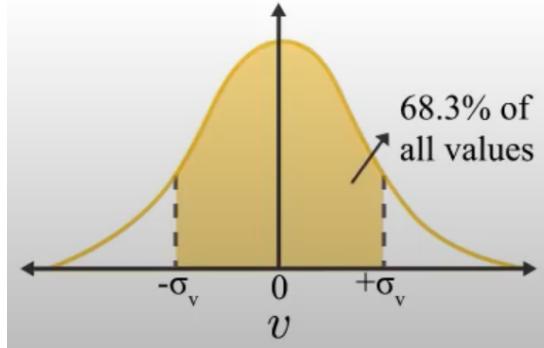


Figure 26.8: Gaussian distribution of sensor disturbances  $v$

A mathematical estimator model would predict the current states based on the same inputs to estimator model as given to the real model. This estimator model can be modeled as follows:

$$x_k = A\hat{x}_{k-1} + Bu_k + w_k \quad (26.62)$$

$$y_k = C\hat{x}_k + v_k \quad (26.63)$$

Theoretically, the estimator model cannot accurately estimate the exact current states of the system as the real system would be affected by the process disturbances which are not included in the estimator model as can be seen comparing 26.60 and (26.62).

This is where a Kalman filter comes into play, where it combines the results of both the real system output  $y_k$  and the estimated states  $\hat{x}_k$  to find the optimal estimate of the vehicle's position.

In order to understand how this optimization works, consider figure 26.9

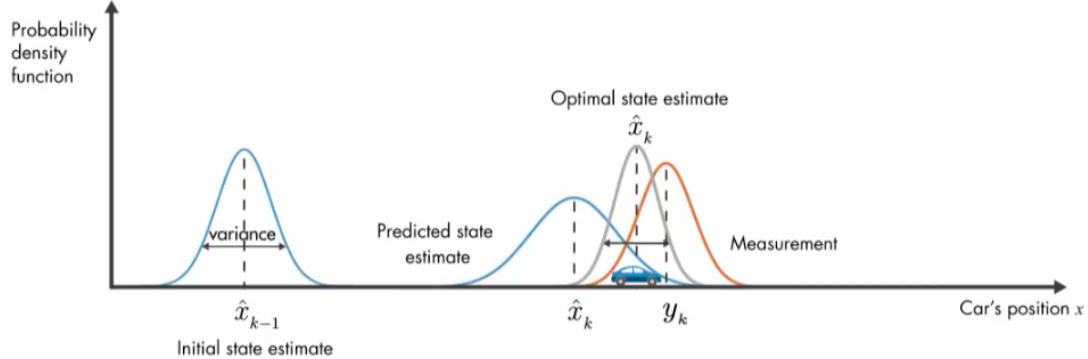


Figure 26.9: Understanding Kalman filter optimization using graph

the initial estimate from the predictor model (mathematical model) is shown by the left blue graph, this graph says that the predicted position of the vehicle would be anywhere among that variance, by most likely at the center with zero variance (shown by the dotted line). From this initial state estimate, the predicted state estimate is shown by the right blue graph. Additionally, there is also the system output from the real system, which is also graphed, shown here by the orange graph to the right. Again, in this case, the predicted state (position) of the vehicle is anywhere among that variance, by most likely at the center with zero variance (shown by the dotted line). According to the theory of Kalman filter, the best estimate comes from combining the results of both the predictor model (mathematical model) as well as the system output from the real system.

Note\*\*\* Also note that the area under this graph is minimized mathematically while solving for  $P_{\hat{x}}$  tuning with gain  $L$ .

By combining, it means that both these graphs are multiplied, the result of which is that the resultant variance (shown by the gray graph) is smaller than either of the graphs it was generated from. Combining these two results is an optimal way of finding a solution between these two available results. The optimal state estimated is shown by the gray dotted line (most likely position) with variance as shown in the gray graph. (Note\*\*\* The results are combined by multiplying, then scaling (estimator gain) and finally computing the mean (dotted line) to obtain the optimal estimated state).

The multiplication of these two probability density function relates to the discrete Kalman filter equation given by

$$\hat{x}_k = A\hat{x}_{k-1} + Bu_k + K_k(y_k - C(A\hat{x}_{k-1} + Bu_k)) \quad (26.64)$$

The left half of (26.64)  $A\hat{x}_{k-1} + Bu_k$  is the predictor model and the right half of (26.64) is the scaled (estimator gain) error feedback. Therefore, the summation of these two

## Nomenclature

leads towards an optimal state estimate. (26.64) is same as the state observer equation given below

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K(y_k - C\hat{x}_k) \quad (26.65)$$

infact Kalman filter is a type of state observer which is designed for stochastic systems with variance as shown in figure 26.9 where as the state observer given by (26.65) is designed for the deterministic systems.

In (26.64), the left half term  $A\hat{x}_{k-1} + Bu_k$  is a prior estimate based on the previous state and current inputs, this is denoted as  $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$ , therefore, (26.64) can now be re-written as

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-) \quad (26.66)$$

in (26.66), the part  $\hat{x}_k^-$  is called the predict and the part  $K_k(y_k - C\hat{x}_k^-)$  is called the update as it incorporates the measured values which are updated with the prior estimate  $\hat{x}_k^-$ , the sum of which is called a ***A Posteriori Estimate***:

$$PosterioriEstimate = Predict + Update \quad (26.67)$$

actually, there are sets of equations that describe the prediction and update as shown below: **Prediction**

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \quad (26.68)$$

$$P_k^- = AP_{k-1}A^T + Q \quad (26.69)$$

## Update

$$K_k = \frac{P_k^- C^T}{C P_k^- C^T + R} \quad (26.70)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-) \quad (26.71)$$

$$P_k = (I - K_k C)P_k^- \quad (26.72)$$

where  $P_k$  is the error covariance matrix and  $P_k^-$  is the variance of the prior estimate as shown in 26.10. It can be considered as uncertainty in the measurement of the estimated states. the Kalman filter gain  $K_k$  is calculated such that it minimizes this error covariance matrix  $P_k$ . Infact, the gain  $K_k$  can be adjusted to either trust the measurements more or the predict model more, based on how these results deviate from the actual states. If the measured states have very little variance with the actual states due to low sensor disturbances, then the predict model is trusted more and gain tuning  $K_k$  depends more on the measurement than the prediction model. This can be formulated mathematically by making the measurement disturbance go to zero and formulating the gain  $K_k$

$$\lim_{R \rightarrow \infty} K_k = \lim_{R \rightarrow \infty} \frac{P_k^- C^T}{C P_k^- C^T + R}$$

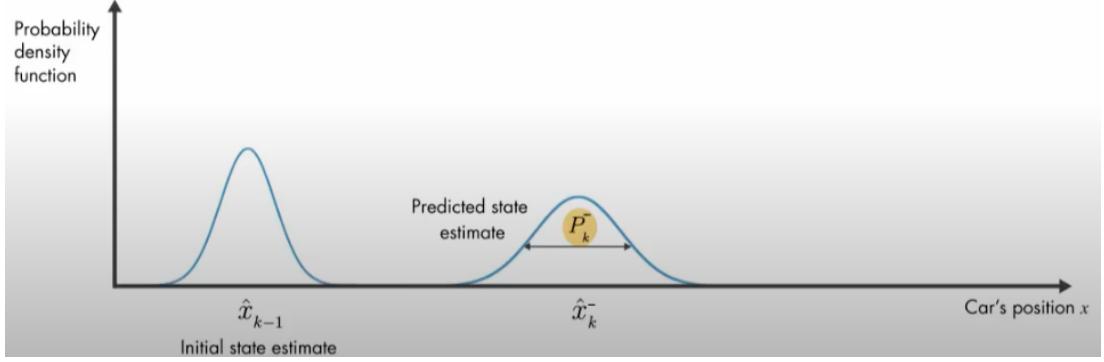


Figure 26.10: Definition of  $P_k^-$

$$= \lim_{R \rightarrow \infty} \frac{P_k^- C^T}{C P_k^- C^T + 0} = C^{-1}$$

for this particular system, the output matrix  $C^{-1} = 1$ . Further the update state equation becomes

$$\begin{aligned} \hat{x}_k &= \hat{x}_k^- + C^{-1}(y_k - C\hat{x}_k^-) \\ &= \hat{x}_k^- + C^{-1}y_k - C^{-1}C\hat{x}_k^- \\ \hat{x}_k &= C^{-1}y_k = y_k \end{aligned} \quad (26.73)$$

therefore, as described in (26.73), the estimated states are only described by the measurements. The plot of such a estimated state on the density function can be seen as 26.11, where the estimated states are reduced to a single line, as the measurement disturbance  $R = 0$  or the measurement is most trusted. on the other hand, if the pre-

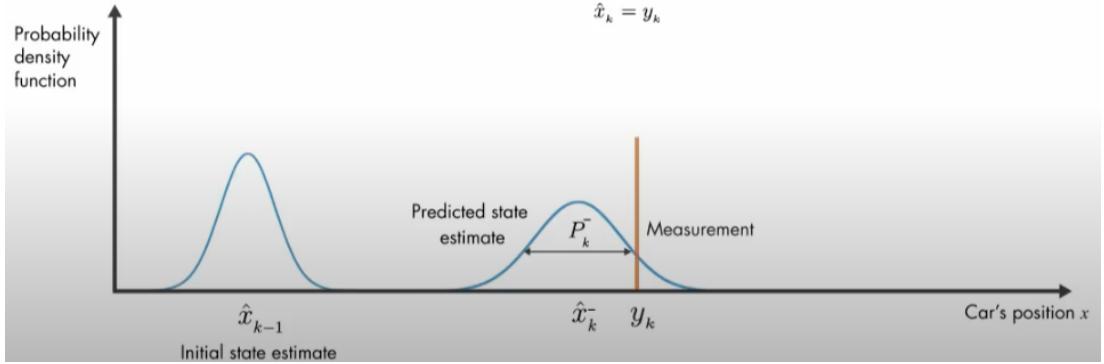


Figure 26.11: Definition of  $P_k^-$

diction model is trusted more, it means that the prior state estimate covariance  $P_k^- = 0$ , therefore,

$$\lim_{P_k^- \rightarrow \infty} K_k = \lim_{P_k^- \rightarrow \infty} \frac{P_k^- C^T}{C P_k^- C^T + R}$$

## Nomenclature

$$= \lim_{P_k^- \rightarrow \infty} \frac{0}{0 + R} = 0$$

Further the update state equation becomes

$$\begin{aligned}\hat{x}_k &= \hat{x}_k^- + 0(y_k - C\hat{x}_k^-) \\ \hat{x}_k &= \hat{x}_k^-\end{aligned}\tag{26.74}$$

from (26.74), the estimated states are described only by the prior state predictions and the contribution of the measurements are ignored completely. On the graph, this is represented by 26.12

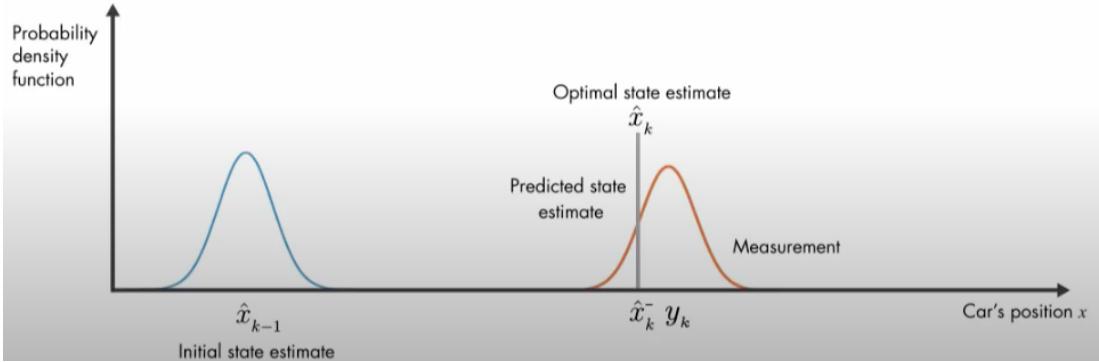


Figure 26.12: Definition of  $P_k^-$

### 26.6.2.2 Kalman filter for Sensor Fusion

If more than one sensor is used in the system, the Kalman filter equation does not change much. The size of the matrices of  $K$ ,  $y$  and  $C$  will be modified and the rest of the solution remains exactly the same. The only difference is that now more than two graphs in the probability density function are merged to optimize the estimation of the states as shown in figure 26.13

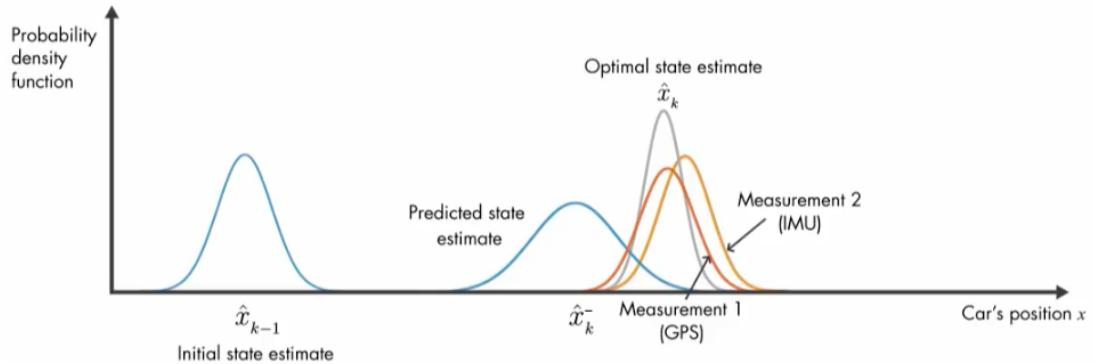


Figure 26.13: Definition of  $P_k^-$

### 26.6.3 Kalman Filter Matrix Equations

The design of Kalman filter has five matrices as described below:

1.  $\mathbf{x}$  the state matrix
2.  $\mathbf{P}$  the uncertainty matrix. This is of size  $(n \times n)$  where  $n$  is the number of states. This matrix says about how confident we are when the states prediction is concerned by the filter. The higher the number the higher uncertainty is when states prediction is concerned. In case of Kalman filter, this is the only tuner that will be available to tune the filter to make better predictions. Only elements along the diagonals of this matrix are tuned.
3.  $\mathbf{F}$  the state transition matrix. This is of size  $(n \times n)$  where  $n$  is the number of states. This matrix is same as the  $A$  matrix in linear state-space equation
4.  $\mathbf{H}$  the measurement matrix. This is of size  $(z \times n)$  where  $z$  is the number of sensors. This matrix gives a relationship that is required to extract measured states such as velocity from the speed-o-meter. In most of the simple cases the  $\mathbf{H}$  has 1's along its diagonal indicating a one-to-one relationship between measurement and the states. This matrix is similar to  $C$  matrix in the linear state space equation.
5.  $\mathbf{R}$  the measurement uncertainty matrix. This is of size  $(n \times n)$ . This matrix has to say about the certainty of the measurements recorded. If the measurements recorded can be claimed to be 100% accurate, then the corresponding terms in this matrix can be made zero. The values are plugged only along the diagonal and the weighted values are determined through trial and error until a suitable solution is converged.

Once these matrices have been established, then a computation method is employed in order to perform the ***Predictions*** and ***Measurements and Update***. Both predictions

## *Nomenclature*

and measurements and update computations are central to all of the filters. The part in the measurements and update (state update for the next time-integration) will be employed specifically for specific type of a filter such as a Kalman Filter.

Using the five matrices generated, the following computations are performed:

### **1. *Predictions***

- a)  $x = \mathbf{F}x$
- b)  $\mathbf{P} = \mathbf{F}\mathbf{P}\mathbf{F}^T$

### **2. *Measurements and Update***

- a)  $\mathbf{Z}$  =measurements
- b)  $y = \mathbf{Z}^T - \mathbf{H}x$
- c)  $\mathbf{S} = \mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R}$
- d)  $\mathbf{K} = \mathbf{P}\mathbf{H}^T\mathbf{S}^{-1}$
- e)  $x = x + \mathbf{K}y$
- f)  $\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}$

# 27 Example Systems with control applications

## 27.1 Rotor of a Humanoid Robotic arm

Consider a system of an electric motor given by,

$$\ddot{\theta} = \frac{1}{J}(Ki - b\dot{\theta}) \quad (27.1)$$

there are two highest derivatives in system (27.1), considering the both  $\ddot{\theta}$  and  $\dot{\theta}$  are variables. The control input  $u = i$  and the required output of the system is the angular displacement  $\theta$ . Therefore, the state-space equation can be expressed as,

$$\dot{x} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -b/J \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ K/J \end{bmatrix} u \quad (27.2)$$

$$y = [1 \ 0] \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} \quad (27.3)$$

Assume that this system is both completely controllable and observable. For reference tracking, there are two methods, one is the use of reference signal to drive the system to track  $r(t)$  by including  $-k_r r(t)$  for in the control equation itself, as described in section 23.3. Another method is using error dynamics as described here, for tracking the reference, lets describe error for each of the states as,

$$\begin{bmatrix} \theta - \theta_d \\ \dot{\theta} \end{bmatrix} \quad (27.4)$$

such that for velocities the reference  $\dot{\theta}_d = 0$ , the reference velocity is when the arm is stabilized to a certain position.

## 27.2 Full system control design for a single track model

Consider the already developed single track model linearized as described in 20.4 also given here,

$$\begin{bmatrix} \dot{v}_y \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} -(c_2+c_1) & -ac_1+bc_2 \\ \frac{mv_x}{m} & \frac{-mv_x^2 - v_x}{(a^2c_1+b^2c_2)} \end{bmatrix} \begin{bmatrix} v_y \\ \psi \end{bmatrix} + \begin{bmatrix} c_1 \\ \frac{m}{ac_1} \end{bmatrix} \delta \quad (27.5)$$

## Nomenclature

where  $x_1$  is the vehicle's speed,  $x_2$  is the yaw-rate and  $u$  is the steering angle  $\delta$  of the vehicle. Here the state  $x_1$  which is  $v_y$  and vehicles steering angle  $\delta$  ie.,  $x_2$  are given in the body coordinates. To get the vehicles real position and heading angle in the global reference frame, consider figure 27.1 using angle cosines, the projections of  $v_x$  and  $v_y$

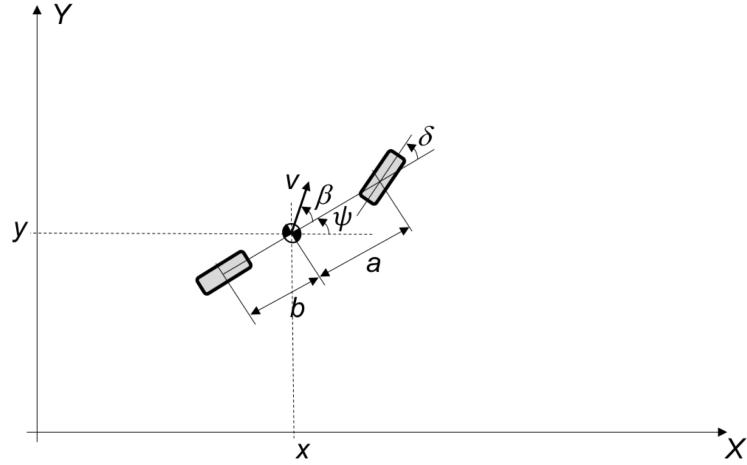


Figure 27.1: Global coordinate system for single track model

on global coordinate system can be determined as,

$$\dot{x} = v_x \cos \psi - v_y \sin \psi \quad (27.6)$$

$$\dot{y} = v_x \sin \psi + v_y \cos \psi \quad (27.7)$$

for this autonomous system, there is an actuator for the steering which is modeled as a non-linear system

$$\dot{\delta} = -\frac{1}{\tau} \delta + \frac{k_1}{\tau} u + \frac{k_2}{\tau} u^3 \quad (27.8)$$

where  $\delta$  is the steering angle and  $u$  is the input voltage to the actuator. Consider the following model parameters given in 27.1

Now with the inclusion of actuator steering dynamics, a new state space model needs to be created. In this single track model, the longitudinal velocity  $v_x$  is assumed / maintained constant. Further only lateral dynamics are of concern as a lateral control is developed here. Therefore, the variables that are to be included in order to properly capture all of the lateral dynamics states are as following  $[y \ v_y \ \psi \ \dot{\psi}]$  additionally, there is a actuator dynamics variable  $\dot{\delta}$  which is also important state to be recorded for a proper lateral control. Therefore, the complete set of states that needs to be considered are as follows

$$[y \ v_y \ \psi \ \dot{\psi} \ \dot{\delta}]^T = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T \quad (27.9)$$

Description	Parameters	Value
vehicle mass	m	1700 [kg]
vehicle inertia (around z-axis)	J	2000 [ $kg \cdot m^2$ ]
front wheel cornering stiffness	$c_1$	80000 [ $kg \cdot m/s^2$ ]
rear wheel cornering stiffness	$c_1$	100000 [ $kg \cdot m/s^2$ ]
front wheel to center of mass	a	1.3 m
rear wheel to center of mass	b	1.5 m
actuator time constant	$\tau$	0.1 s
actuator input gain 1	$k_1$	0.01 rad/V
actuator input gain 2	$k_2$	0.0056 rad/ $V^3$

Table 27.1: Single track model parameters

re-writing the new set of equations for the above mentioned new states in addition to (27.5),

$$\dot{y} = v_x \sin\psi + v_y \cos\psi \quad (27.10)$$

$$\dot{v}_y = -\frac{(c_2 + c_1)}{mv_x} v_y + \left( \frac{-ac_1 + bc_2}{mv_x} - v_x \right) \dot{\psi} + \frac{c_1}{m} \delta \quad (27.11)$$

$$\dot{\psi} = \dot{\psi} \quad (27.12)$$

$$\ddot{\psi} = \left( -\frac{ac_1 - bc_2}{Jv_x} \right) v_y - \left( \frac{a^2 c_1 + b^2 c_2}{Jv_x} \right) \dot{\psi} + \frac{ac_1}{J} \delta \quad (27.13)$$

$$\dot{\delta} = -\frac{1}{\tau} \delta + \frac{k_1}{\tau} u + \frac{k_2}{\tau} u^3 \quad (27.14)$$

The above set of equations form a set of nonlinear equations describing the dynamics of single track vehicle model concerned only with lateral vehicle dynamics.

### 27.2.1 Linearizing single track model

Linearizing the set of nonlinear lateral dynamics single track model as described in the section above such that it can be written in Matrix form  $\dot{x} = Ax + Bu$ . In order to do so, an operating point has been chosen around  $x_e = (0, 0, 0, 0, 0)$  and  $u_e = 0$ . This operating point is chosen such that all lateral motions are zero along with zero inputs for lateral steering, therefore, this operating point  $x_e$  denotes a straight forward driving. Using a small step approximation from the operating point of the system as described in chapter Linearization in this book, a set of matrices  $A$ ,  $B$  and  $C$  are to be developed which denote the Jacobian's of functions with each of the states. These matrices are developed as follows

## Nomenclature

$$A = \begin{bmatrix} 0 & \cos\psi & v_x \cos\psi - v_y \sin\psi & 0 & 0 \\ 0 & -(c_1 + c_2)/mv_x & 0 & -v_x - (ac_1 - bc_2)/mv_x & c_1/m \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -(ac_1 - bc_2)/Jv_x & 0 & -(c_1 a^2 + c_2 b^2)/Jv_x & ac_1/J \\ 0 & 0 & 0 & 0 & -1/\tau \end{bmatrix}_{(x_e, u_e)} \quad (27.15)$$

applying the conditions of operating point,

$$A = \begin{bmatrix} 0 & 1 & v_x & 0 & 0 \\ 0 & -(c_1 + c_2)/mv_x & 0 & -v_x - (ac_1 - bc_2)/mv_x & c_1/m \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -(ac_1 - bc_2)/Jv_x & 0 & -(c_1 a^2 + c_2 b^2)/Jv_x & ac_1/J \\ 0 & 0 & 0 & 0 & -1/\tau \end{bmatrix} \quad (27.16)$$

similarly, finding the matrix B

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 3k_2 u^2 / \tau + k_1 / \tau \end{bmatrix}_{(x_e, u_e)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ k_1 / \tau \end{bmatrix} \quad (27.17)$$

the next step is to design a state feedback controller for the system. Before designing the controller, we need to check if the system is reachable or not.

### 27.2.2 Controllability

Controllability or more formally reach-ability is the condition from an initial state  $x_0$  of a system to reach a controlled state  $x_n$  given control input  $u$ . As described in section of controllability in 25.2, for the case that each of the discrete state ( $x_0$  through  $x_n$ ) can be mapped to each of the control inputs ( $u_0$  through  $u_{n-1}$ ), the controllability matrix  $\Gamma$  should be a full row-rank matrix (In linear matrices having a row-rank is same as having a column rank). In other words, having such one-to-one mapping ensures that each of the discrete states are possible by applying required control inputs.

The same of  $\Gamma$  depends on the order of the system states, therefore for a five order state system of the linear single track model, the controllability matrix is expressed as,

$$\Gamma = [B \ AB \ A^2B \ A^3B \ A^4B] \quad (27.18)$$

If  $\Gamma$  is a full rank matrix, a test to check weather determinant of  $\Gamma \leq 0$  will tell if the rank of  $\Gamma$  is not a full rank. In our case,  $\text{rank}(\Gamma) = 9.5247 \times 10^5$  which is  $\Gamma > 0$ , which is a full rank matrix. Therefore, our system with the set of control inputs on steering satisfies that the system is completely controllable.

### 27.2.3 Feedback Control

Once the controllability is determined, now a controller itself can be designed in order to achieve our target of implementing a lateral control of the vehicle using inputs to the steering. The form of the state-feedback controller should be of,

$$\Delta u = -K \Delta x + k_r \Delta r \quad (27.19)$$

Before deciding the tuning parameters for the controller, consider the following requirements for the controller.

- the lane change maneuver should take not more than 2 seconds
- the steady-state vehicle's position should be within the boundary of 5%.

The control design is proceeded as follows, the system is of fifth order, therefore, in theory there are five independent poles of the system that can be tuned to bring the system to perform this behavior. For the sake of simplicity, choosing repeated poles such that two repeated poles determine the dynamics of the system. Therefore, being the slowest of the five poles, the other three repeated poles can be placed to be faster than these two first slower repeated poles. Therefore, the required control system should look something like this,

$$p(s) = (s + p_1)^2(s + p_2)^3 \quad (27.20)$$

the faster poles  $p_2$  are chosen at least 3 times faster than  $p_1$ . Here a simple pole-placement technique is used and Ackermann's formula is used to determine gain  $K$  which is detailed in section 24.1 of this book. Now since it is already established that poles  $p_2$  should be atleast 3 times bigger than  $p_1$ , in order to find  $p_{des}$  required by Ackermann's formula for controller tuning, there are many possibilities such as the poles being placed at locations as follows  $\{p_1, p_2\} = \{-1, -3\}$ ,  $\{p_1, p_2\} = \{-2, -6\}$ ,  $\{p_1, p_2\} = \{-3, -9\}$  or at  $\{p_1, p_2\} = \{-4, -12\}$ .

Poles  $p_1$  being the dominating poles, are selected based on the fact that the setting time as per the requirement should take 2 sec (settling time, here defined as from start of maneuver until the vehicle's position is within 5% of its steady state level). The reference gain should be chosen such that the steady state gain is 1. From the previously listed possibilities of poles  $\{p_1, p_2\}$ , running simulations can determine the required placement of the poles. (Model: <https://github.com/riz-ahmed/Autonomous-Vehicle-Controls.git>)

Running simulations shows that the chosen poles at  $\{p_1, p_2\} = \{-3, -9\}$  satisfies all the required control objectives. A square wave function is used as a reference input signal as which simulates a periodic lane-changing of the vehicle. The controller satisfies the objectives when the simulation results are consistent with the settling time and the required steady-state performance.

## Nomenclature

### 27.2.4 Estimator design using Kalman Filter

An estimator design is employed using Kalman-Filter optimization for covariance  $P_{\tilde{x}}$  using gain  $L$ , the resulting Riccati's equation for which  $P_{\tilde{x}}$  is the solution is define as follows, considering that there are no process disturbances considered into the system:

$$AP + PA^T - PC^T R_w^{-1} CP = 0 \quad (27.21)$$

The estimator gain that minimizes the state error covariance  $P = E(\tilde{x}\tilde{x}^T)$  is given by Kalman filter:

$$L = PC^T R_w^{-1} \quad (27.22)$$

A sensor disturbance with a covariance of  $R_w = 0.01$  is considered. In Matlab, this Riccati's equation can be solved to find gain  $L$  using the care function as  $[P, E, L] = \text{care}(A', B', R_w)$ . Simulations seem to be stable even for the conditions when the model parameters are uncertain, such as varying mass and vehicle velocity does not modify the behavior of the controller.

# 28 Robust Control

In general, a robust system is a system that is capable of meeting the requirements (which is stability or other performances) even in the presence of model uncertainties and/or disturbances. Robust control is an approach to designing a system in a way that can handle uncertainties.

## 28.1 Reason for a Robust control

In general, for any system based on a mathematical model, an input / output system model is created such that, the outputs of the system measurements can be used directly / indirectly in some form by a controller. The controller then develops a control law which is then fed back into the input/output model as shown in figure 28.1 In

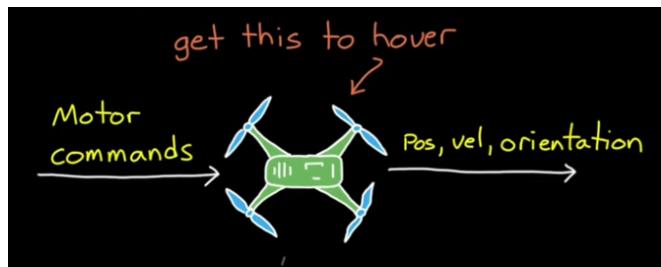


Figure 28.1: Input/Output model of a plant

this system shown in figure 28.1, the inputs and the outputs are labeled. Using these outputs (or measurements) from this input/output plant model, a following feedback control system can be developed as shown in 28.2 the part where the controller goes

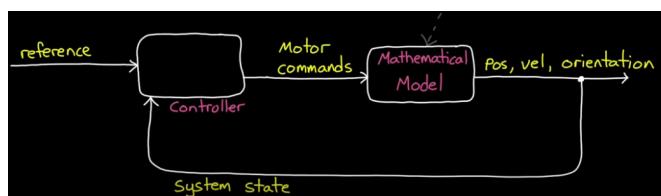


Figure 28.2: Control system developed using the measurements from the input/output model

could be anything from a classical designed PID to a full state feedback controllers

## Nomenclature

tuned using LQR etc., as well as a neural network can be used as a control algorithm. Once the controller meets the requirements, then the control algorithm can be used to develop a C-Code and flashed into an embedded system. This controller can then be tested finally with a original system so as to verify the requirements with actual model as shown in figure 28.3 The validity of this controller is based on the accuracy of the

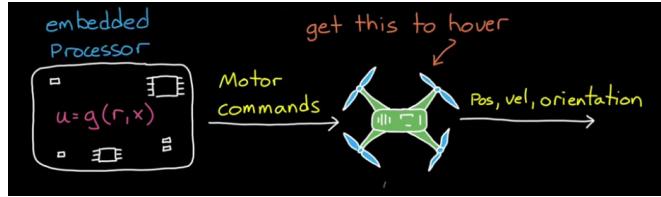


Figure 28.3: Control system developed using the measurements from the input/output model

model. Obviously, as the mathematical model deviates from the original model, the performance deviations also deviate accordingly or may fail entirely. This could happen in most of the real world applications as most of the models are prone to one or more of the following reasons as to why mathematical models deviate / fail to represent the required plant model.

1. ***Complex Dynamics***: It may be very complex to understand and capture complex dynamics of a system such as high frequency vibrations in a given plant system. In most of such cases, the mathematical model could not have included such complexities in order to simplify things. Even in case of black-box modeling, there would not have been relevant sensors to actually measure such complexities directly.
2. ***Uncertainties in the model***: this could raise due to uncertain disturbances, actuator uncertainties as well as uncertainty on how the system would respond to such uncertain actuations among others.
3. ***Intentional Simplicity***: In most of the cases, model are linearized so as to simplify whole of the development process. Realism is sacrificed for simplicity.
4. ***Stochastic events***: this is one of the reasons where the model would naturally vary over time and the changes are driven by unknown or stochastic events. Short term of such events could be treated as noise and the long term events could be of the performance deterioration as it ages.
5. ***Process variations***: after producing series of products it is often noticed that there are slight variations in each of the model produced due to slight process variations which now have been taken into consideration while developing a mathematical model.

Due to these above mentioned model uncertainties, the controller's performance can be expected to drift as the model uncertainties begin to drift. In order to combat this model uncertainty, one of the simplest ways is to increase the band of stability. Such that the system being designed to just be stable is not enough in this case, the system is in fact designed to have some margin of stability. In such a case, as the model uncertainty pushes the stability of the system in reality, it will be only eating up this band of stability that was previously included in the design. Therefore, maintaining controller stability even in the cases of some model uncertainty variations.

The gain and phase margins discussed earlier in classical control theory does exactly this purpose. For example, if a control system is specified by a case that, gain margin > 6dB and phase margin > 45 deg. These margins say on the limits the system can deviate before it becomes unstable. Such that, this system has 6dB of additional gain that can be added before it goes unstable, similarly, phase can be lagged upto 45 deg before the system goes unstable. Now the decision on choosing these margins depends entirely on the uncertainties. However, it should also be noted that as these margins are designed to have higher limits, it also increases the cost of the project. On the other hand, too little of these margins will also result in failed requirement testing.

It should also be noted that though gain and phase margins are very useful in determining the stability of the systems they are not a complete picture of robustness of a system. For an example consider a open-loop transfer function of a system:

$$G = \frac{0.38s^2 + 0.038s + 0.209}{s^4 + 1.06s^3 + 0.56s^2 + 0.5s} \quad (28.1)$$

the closed-loop stability can be determined by open-loop gain and phase margins, using Matlab function *margin(G)*, the following figure 28.4 was generated

## Nomenclature

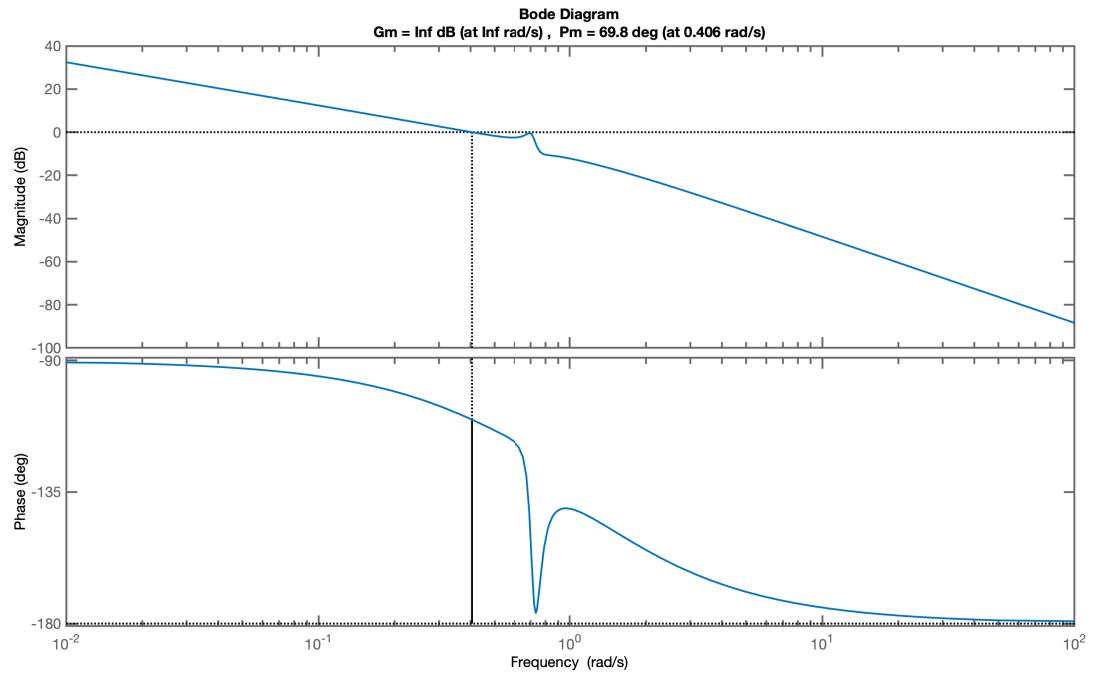


Figure 28.4: open-loop margins of an example system

It can be seen in 28.4, that the gain margin for this system is literally  $\infty$ , this system appears to be fairly stable under large gain additions as it can be seen in 28.4, that the phase never crosses the  $180^\circ$  line. This can also be seen by adding a closed loop gain using Matlab's *feedback*( $G * addGain, 1$ ) which generates 28.5

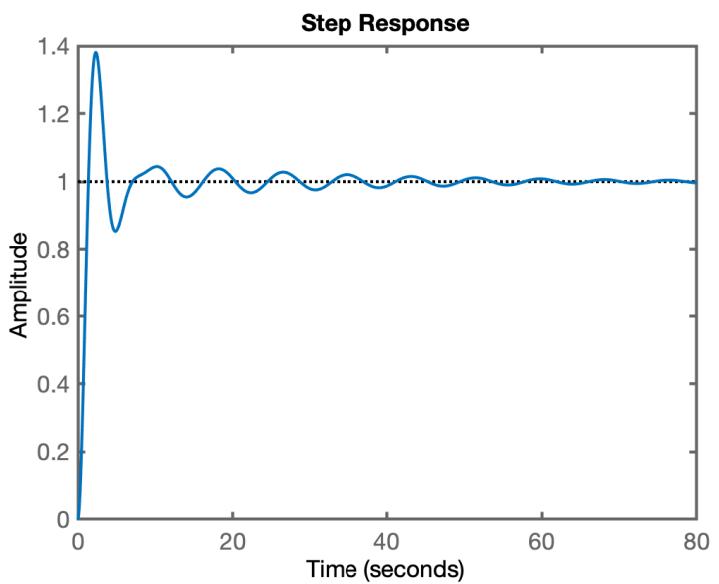


Figure 28.5: adding gain to the example system

where  $\text{addGain} = 10^{\text{amplitude}/20}$  in dB and amplitude = 15 in metric scale. further, adding phase delay of 2 sec using Matlab's  $\text{feedback}(G * \exp(-s * 2), 1)$ , generates 28.6

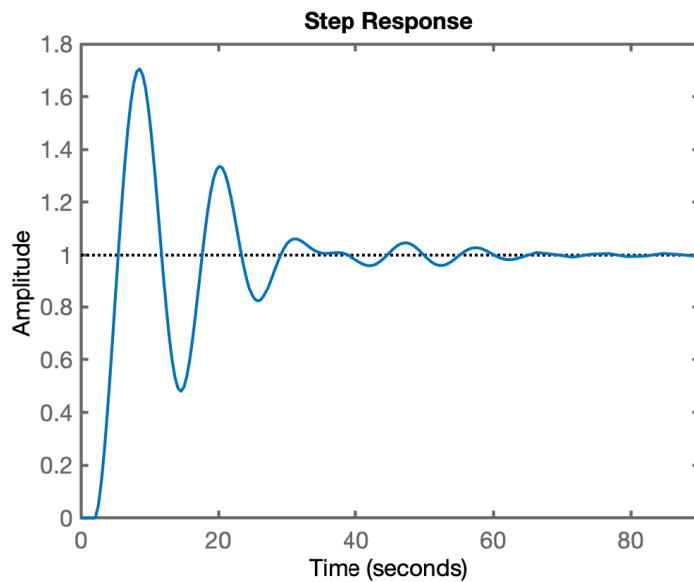


Figure 28.6: adding phase to the example system

## Nomenclature

But when the system is added with both gain and phase margins, the system would go completely unstable as shown in 28.7 this is due to the combined effect of adding both

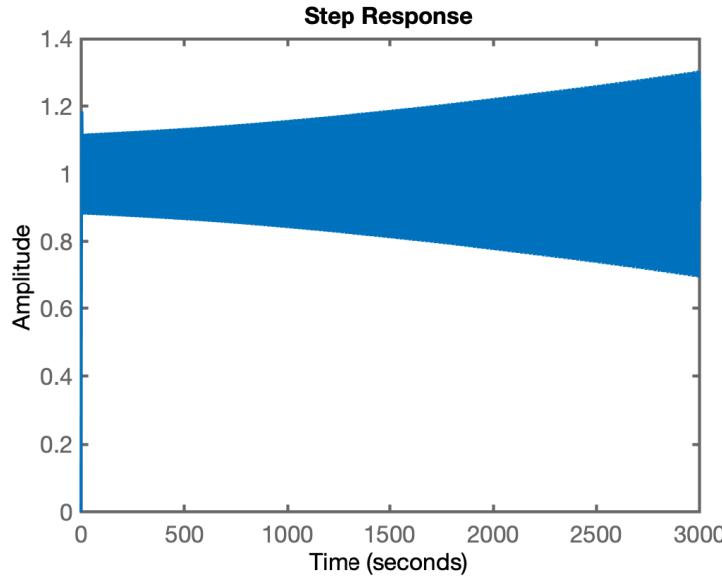


Figure 28.7: adding phase to the example system

the margins together which cannot be determined only using the bode plots. Therefore, in robust control theory, another concept called disk margin is used which accounts for both gain and phase margins simultaneously. Disk margins further also have an advantage of being able to be implemented to MIMO systems which unlike in classical margins (from bode plot) is not in general easy to be implemented. By using Matlab's *diskmargin(G)*, the following structure is generated

```
struct with fields :
```

```

GainMargin: [0.7379 1.3552]
PhaseMargin: [-17.1526 17.1526]
DiskMargin: 0.3016
LowerBound: 0.3016
UpperBound: 0.3016
Frequency: 0.7137

```

from this struct it can be seen that the actual gain and phase margins are actually 1.3552 and 17.1526 which is much lower than what was originally determined using classical gain and phase margins. Now, lets plot a step response again using these new results generated using disk margins,

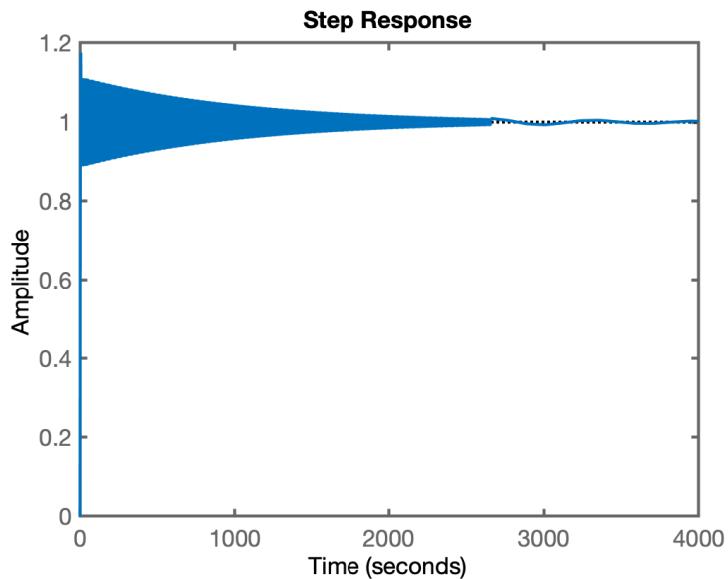


Figure 28.8: adding phase to the example system

To summarize this introduction, for robust control analysis, it is important to assess and scale the model variations and other disturbances into the system before designing the controller. In most of the cases, it is not possible to know in prior how exactly the model would vary and how other parameters such as sensor noise or high frequency dynamics would make the model uncertain. However, it is possible to predict how these uncertainties would possibly develop during real time operations of the system. These uncertainties can be modeled as some stochastic disturbance or disturbances that goes into the system and therefore, a valid range of band limits can be implemented directly into the system while designing the system. Other than stochastic modeling there are other simple ways to model these uncertainties such as in case of classical modeling approach, one can consider that only the gain margin or only the phase margin to be designed with. In some of the real time systems for example, one can have more certainty of system performance at lower operating frequencies than at higher operating frequencies. In such cases, a lower band limit at lower frequencies and higher band limit at higher operating frequencies can be implemented into the design.

So far in this introduction only the analysis part of robust control methodology has been discussed. For the synthesis part (design the controller), it should be noted that in case of a robust controller, it is not the case that there is a specific controller such as a PID or a full state feedback controller. Rather, a robust control discusses a methodology to tune controller parameters such as PID to make the system robust.

In classical control synthesis, in book a brief introduction is already provided earlier for a robust control design using loop shaping (root locus and Bode plots) by adding poles or zeros, to make the system more robust in case of uncertainties. However, this

## *Nomenclature*

classical method is not suitable for MIMO system and we have other methods such as  $H_\infty$  or  $\mu$  synthesis.

In all cases, it is important to note that the basis of robust control analysis and synthesis is pure basd on adressing uncertainties. For this the following points have to be noted

1. understand uncertainties and represent them in the model
2. analyse the system to see if the system is still stable with these added uncertainties
3. if not, then tweak the design based in robust controll analysis and synthesis such that the system is stable again

# Bibliography

- [Ahm18] AHMED, Rizwan: Model Validation of Two-wheeled Autonomous Vehicle. (2018)
- [EgeND] EGERSTEDT, Magnus: *Control of Mobile Robots: Linear Systems*. N.D. – Lecture slides: Control of Mobile Robots
- [Hil13a] HILL, Rick: *Video Lec: System Dynamics and Control: Module 11 - Stability and Second-Order Systems*. <https://youtu.be/h73CudC0FQo>. Version: 2013
- [Hil13b] HILL, Rick: *Video Lec: System Dynamics and Control: Module 13 - Introduction to Control, Block Diagrams*. <https://youtu.be/p4KEKCenc0E>. Version: 2013
- [Hil13c] HILL, Rick: *Video Lec: System Dynamics and Control: Module 14 - Control Specifications and PID control*. <https://www.youtube.com/watch?v=RYWPqb2hAEM>. Version: 2013
- [MT19a] MESSNER, Bill ; TILBURY, Dawn: *Control Tutorials with Matlab and Simulink*. <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SystemModeling>. Version: 2019
- [MT19b] MESSNER, Bill ; TILBURY, Dawn: *Control Tutorials with Matlab and Simulink*. <http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SystemAnalysis>. Version: 2019

*Nomenclature*