# INTERPRETING SPECIFICATIONS FOR CONTROL SYSTEM DESIGN

**Kevin Edward Harris**

A research report submitted to the Faculty of Engineering and the Built Environment, of the University of the Witwatersrand, in partial fulfillment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, 2007

# Declaration

I declare that this thesis is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination in any other university.

Signed this ____________ day of ____________ 20___

_______________________

Kevin Edward Harris

## Abstract

Current control theories require control specifications to be posed in a very specific manner, which is often not easily obtained from the specifications posed on the system.

The goal of this project report is to develop a method that is capable of evaluating the set of specifications placed on a control system for consistency, and generate the specifications that are missing, but required by the control system design algorithm which is to be used. Further, these specifications must be consistent with the other specifications placed on the system as well the plant dynamics.

A method using the Bode ideal characteristic as a nominal loop transmission was developed in order to generate prototype models (dynamics) for use in specifying control system bounds was discussed. The goals were:

1. To generate the prototype models which meet the specifications that were provided.
2. To provide the designer with a means of trading off the control loop characteristics which are not covered by the specifications.
3. To bring the plant characteristics into the generation and evaluation of the prototype models.

The prototype models generated are then used to generate the bounds required the the control system design process to be used.

## Acknowledgments

I would like to thank:

- Dr. B. Wigdorowitz for his advice, guidance and time throughout the duration of research. I have learnt a great deal.

- My parents for their encouragement and support which was greatly appreciated.

# Contents


1Introduction........................................................ 17

2The control design process........................................................20

2.1Functional requirements........................................................20

2.1.1Physical plant........................................................21

2.1.2Plant model........................................................ 21

2.1.3Performance requirements........................................................24

2.1.4Controller algorithm........................................................24

2.1.4.1Quantitative feedback theory........................................................ 25

2.1.4.2Optimal control theory........................................................25

2.1.5Theoretical simulation........................................................27

2.1.6Physical controller........................................................ 27

2.1.7Hardware-in-loop simulation........................................................27

2.1.8System test........................................................ 28

2.2The problem under consideration........................................................28

2.3The architecture of the control system that will be considered........................................................ 30

2.3.1Other control system architectures........................................................31

2.3.2Dealing with sensor and actuator characteristics........................................................31

3Operational requirements........................................................33

3.1Time response of the control system to the plant input........................................................33

3.1.1Step Response........................................................ 33

3.1.1.1Rise time........................................................33

3.1.1.2Overshoot........................................................ 35

3.1.1.3Settling time........................................................35

3.1.2Responses to other control inputs........................................................35

3.1.3Tracking command inputs........................................................37

3.1.4The steady state response of the control system to the plant input........................................................38

3.2Time response of the control system to disturbances entering the plant........................................................39

3.2.1Step response........................................................39

3.2.1.1Fall time........................................................ 39

3.2.1.2Undershoot........................................................ 40

3.2.1.3Settling time........................................................40

3.2.2The response to other disturbance inputs........................................................42

3.2.3Steady state response to a disturbance input........................................................ 42

3.3Response of the control system to sensor noise........................................................42

3.4Translating the time domain specifications into the frequency domain........................................................ 43

3.4.1Response of the control system to the plant input........................................................43

3.4.1.1Rise time........................................................45

3.4.1.2Overshoot........................................................ 46

3.4.1.3Settling time........................................................48

3.4.1.4Steady state error........................................................49

3.4.1.5Tracking the command input........................................................51

3.4.2Response of the control system to disturbances........................................................ 51

3.4.2.1Fall time........................................................ 52

3.4.2.2Undershoot........................................................ 52

3.4.2.3Settling time........................................................52

3.4.2.4Steady state response........................................................53

3.4.3Response of the control system to sensor noise........................................................55

4Response prototype models........................................................ 56

4.1Types of response prototype models........................................................56

4.1.1Command input to the controlled variable prototype model set........................................................ 58

4.1.2Disturbance input to the controlled variable prototype model set.............................. 58
4.1.3The sensor noise to the controlled variable prototype model set............................... 59
4.1.4The set of plant input prototype models for the command input ...............................59
4.1.5The set of plant input prototype models for the disturbance inputs............................60
4.1.6The set of plant input prototype models for the sensor noise.....................................60
4.2Determining the set of prototype models that meet all the specifications...........................60
4.2.1Independent prototype models.................................................................................... 61
4.2.2Using one of the sets of prototype models to define all the prototype model sets....... 61
4.2.3Using the loop transmission to define all the prototype models................................ 65
5Translating the frequency domain specifications to bounds on the loop transmission................70
5.1Response of the control system to the plant input................................................................71
5.1.1Bandwidth..................................................................................................................... 71
5.1.2Maximum resonant peak............................................................................................. 72
5.1.3Steady state error..........................................................................................................74
5.1.4Tracking the command input...................................................................................... 76
5.1.4.1One degree of freedom systems...........................................................................76
5.1.4.2Two degree of freedom systems.......................................................................... 78
5.2Response of the control system to disturbances................................................................... 79
5.2.1Bandwidth..................................................................................................................... 82
5.2.2Resonant peak............................................................................................................... 83
5.2.3Steady state response to disturbances.......................................................................... 84
5.3Response of the control system to sensor noise.....................................................................84
6Designing for a stable minimum phase plant............................................................................... 86
6.1Desirable properties of the loop transmission.......................................................................86
6.2The Bode ideal characteristic.............................................................................................. 88
6.2.1The Bode cut-off characteristic....................................................................................88
6.2.2The Bode semi-infinite characteristic..........................................................................90
6.2.3The Bode step characteristic........................................................................................92
6.2.4Putting all the characteristics together........................................................................ 93
6.3Relating the parameters of the Bode Ideal characteristic to the specifications................... 97
6.3.1Low frequency parameters...........................................................................................97
6.3.1.1Steady state error..................................................................................................97
6.3.1.2Tracking specifications..................................................................................... 103
6.3.2Relative stability.........................................................................................................104
6.3.2.1Phase margin......................................................................................................104
6.3.2.2Gain margin....................................................................................................... 105
6.3.2.3Resonant peak.................................................................................................... 106
6.3.3Speed of response.......................................................................................................110
6.3.4High frequency properties of the closed loop response.............................................110
6.3.5The trade offs between the parameter values which should be considered.............. 111
6.4Example............................................................................................................................... 112
6.4.1The plant model..........................................................................................................113
6.4.2The operational requirements.....................................................................................113
7Designing for unstable plants..................................................................................................... 139
7.1The suitability of the Bode ideal characteristic for the loop transmission for an unstable plant..................................................................................................................................... 139
7.2The properties of unstable poles.........................................................................................140
7.2.1Unstable real poles......................................................................................................140
7.2.2Unstable complex conjugate pole pairs..................................................................... 141
7.2.3General properties of unstable poles..........................................................................142
7.3Modifying the Bode ideal characteristic............................................................................. 143
7.3.1The all-pass function.................................................................................................. 143
7.3.2Modifying the nominal loop transmission to take the unstable poles into account...145
7.4Relating the parameters of the modified Bode ideal characteristic to the specifications ..146

7.4.1Steady state error....146
7.4.1.1The steady state error due to the command input for a system with a Bode ideal characteristic loop transmission....146
7.4.1.2The steady state error due to a disturbance input for a system with a Bode ideal characteristic loop transmission....149
7.4.2The trade offs between the parameter values which should be considered....150
7.5Example....150
7.5.1The plant model....150
7.5.2The operational requirements....151
8Designing for non-minimum phase plants....174
8.1The suitability of the Bode ideal characteristic for the loop transmission for a non-minimum phase plant....174
8.2The properties of non-minimum phase zeros....175
8.2.1Non-minimum phase real zeros....175
8.2.2Non-minimum phase complex conjugate zero pairs....176
8.2.3General properties of non-minimum phase zeros....178
8.3Modifying the Bode ideal characteristic....178
8.3.1The all-pass function....178
8.3.2Modifying the nominal loop transmission to take the non-minimum phase zeros into account....180
8.4Relating the parameters of the modified Bode ideal characteristic to the specifications ..181
8.4.1Steady state error....182
8.4.1.1The steady state error due to the command input for a system with a Bode ideal characteristic loop transmission....182
8.4.1.2The steady state error due to a disturbance input for a system with a Bode ideal characteristic loop transmission....183
8.4.2The trade offs between the parameter values which should be considered....184
8.5The design procedure....184
9Conclusion....186
10Appendix A – Relating the open loop and closed loop frequency responses....188
10.1Relating the closed loop magnitude to the open loop magnitude and phase....188
10.2Relating the open loop magnitude to the closed loop magnitude and the open loop phase....189
10.3Relating the open loop phase to the open loop magnitude and closed loop magnitude...190
10.4Finding the minimum and maximum open loop magnitudes for a specific closed loop magnitude....190
10.5Finding the minimum and maximum open loop phases for a specific closed loop magnitude....193
10.6Relating the gain and phase margins specifications to the resonant peak specification..193
10.6.1Relating the gain margin specifications to the resonant peak specification....194
10.6.2Relating the phase margin specifications to the resonant peak specification....194
11 Appendix B – Useful equations....195
11.1Calculating the range of a point on a line given the angle....195
12Appendix C – Matlab Code....197
12.1Architecture....197
12.2M-circle class....198
12.2.1Overview....198
12.2.2Attributes....198
12.2.3Methods....199
12.2.3.1 calcMag....199
12.2.3.2 calcPhase....200
12.2.3.3getClosedLoopMag....201
12.2.3.4 getMinMag....202

12.2.3.5 getMaxMag........................................................................................................203
12.2.3.6 getMinPhase.....................................................................................................204
12.2.3.7 getMaxPhase....................................................................................................205
12.2.3.8 getGainMargin.................................................................................................206
12.2.3.9 getPhaseMargin...............................................................................................207
12.2.3.10 nichols............................................................................................................208
12.3The rotated M-circle class.............................................................................................208
12.3.1Overview.....................................................................................................................208
12.3.2Attributes....................................................................................................................210
12.3.3Methods.......................................................................................................................210
12.3.3.1calcMag.............................................................................................................210
12.3.3.2calcPhase..........................................................................................................211
12.3.3.3getMinMag.......................................................................................................212
12.3.3.4getMaxMag.......................................................................................................212
12.3.3.5getMinPhase.....................................................................................................213
12.3.3.6 getMaxPhase....................................................................................................214
12.3.3.7getGainMargin..................................................................................................215
12.3.3.8getPhaseMargin................................................................................................216
12.3.3.9nichols...............................................................................................................216
12.4QFT template class.........................................................................................................217
12.4.1Overview.....................................................................................................................217
12.4.2Attributes....................................................................................................................219
12.4.3Methods.......................................................................................................................219
12.4.3.1 getMinMag.......................................................................................................219
12.4.3.2 getMaxMag......................................................................................................220
12.4.3.3 getMinPhase.....................................................................................................221
12.4.3.4 getMaxPhase....................................................................................................222
12.4.3.5 getBound..........................................................................................................223
12.4.3.6 getNom.............................................................................................................224
12.4.3.7 getFreq.............................................................................................................224
12.4.3.8 calcPhase..........................................................................................................225
12.4.3.9 calcMag.............................................................................................................228
12.4.3.10 sampleClosedLoopMag..................................................................................231
12.4.3.11 nichols.............................................................................................................232
12.4.3.12 init...................................................................................................................233
12.4.3.13 calcBoundary..................................................................................................234
12.5Stability bound class.......................................................................................................238
12.5.1Overview.....................................................................................................................238
12.5.2Attributes....................................................................................................................239
12.5.3Methods.......................................................................................................................239
12.5.3.1 calcPhase..........................................................................................................239
12.5.3.2 getMinMag.......................................................................................................243
12.5.3.3 getMaxMag......................................................................................................245
12.5.3.4getMinPhase.....................................................................................................247
12.5.3.5getMaxPhase....................................................................................................249
12.5.3.6getFreq.............................................................................................................251
12.5.3.7nichols..............................................................................................................252
12.6Uncertainty bound class..................................................................................................252
12.6.1Overview.....................................................................................................................252
12.6.2Attributes....................................................................................................................253
12.6.3Methods.......................................................................................................................253
12.6.3.1calcMag............................................................................................................253
12.6.3.2nichols..............................................................................................................254
12.7Bode semi-Infinite characteristic class...........................................................................255
12.7.1Overview.....................................................................................................................255
12.7.2Attributes....................................................................................................................256

12.7.3Methods....256
12.7.3.1calcResponse....256
12.7.3.2set....257
12.7.3.3get....258
12.7.3.4fit....259
12.7.3.5bode....260
12.7.3.6nichols....261
12.8Bode step class....262
12.8.1Overview....262
12.8.2Attributes....262
12.8.3Methods....263
12.8.3.1calcResponse....263
12.8.3.2set....264
12.8.3.3get....265
12.8.3.4fit....266
12.8.3.5bode....268
12.8.3.6nichols....268
12.8.3.7calcFreqRange....269
12.8.3.8calcSecondCornerFreq....270
12.9Bode cut-off class....271
12.9.1Overview....271
12.9.2Attributes....272
12.9.3Methods....272
12.9.3.1set....272
12.9.3.2get....273
12.9.3.3calcBandwidth....274
12.9.3.4calcCrossoverFreq....275
12.9.3.5calcFreqAtMag....276
12.9.3.6calcFreqAtPhase....277
12.9.3.7calcPhaseMargin....278
12.9.3.8calcResponse....280
12.9.3.9fit....281
12.9.3.10bode....282
12.9.3.11nichols....283
12.9.3.12calcFreqRange....284
12.9.3.13calcPhaseError....285
12.10 Bode ideal characteristic class....286
12.10.1Overview....286
12.10.2Attributes....286
12.10.3Methods....288
12.10.3.1calcResponse....288
12.10.3.2fit....289
12.10.3.3bode....290
12.10.3.4nichols....291
12.10.3.5calcFreqRange....292
12.11The second order model class....293
12.11.1Overview....293
12.11.2Attributes....294
12.11.3Methods....294
12.11.3.1calcResponse....294
12.11.3.2getOvershoot....295
12.11.3.3getResonantPeak....296
12.11.3.4getRiseTime....297
12.11.3.5getSettlingTime....298
12.11.3.6getTurningPoints....299
12.11.3.7set....300

12.11.3.8get....................................................301
12.12 Utility functions....................................................302
12.12.1 Fit a line to two sets of cartesian co-ordinates....................................................302
12.12.2 Calculate a point on a line using polar co-ordinates....................................................303
12.12.3 Calculating the frequency response of a rational transfer function....................................................305
12.12.4Calculating the frequency response of a root....................................................307
13References....................................................309
14Bibliography....................................................310

# List of Figures

Figure 2.1: The control system design process....22
Figure 2.2: A second degree of freedom control system architecture.... 30
Figure 2.3: An alternative second degree of freedom control system architecture....32
Figure 3.1: The rise time bound on the step response specification.... 34
Figure 3.2: The overshoot bound on the step response specification....34
Figure 3.3: The settling time bound on the step response specification.... 36
Figure 3.4: An example of bounds on a systems response to a ramp input....36
Figure 3.5: Bounds on the system for the tracking of the command input....38
Figure 3.6: The fall time specification on the step response specification to a disturbance input. 40
Figure 3.7: The undershoot specification on the step response specification to a disturbance input. .... 41
Figure 3.8: The settling time specification on the step response specification to a disturbance input....41
Figure 3.9: Command response prototypes that meet the step response specifications....44
Figure 3.10: Disturbance response prototypes that meet the disturbance specifications....44
Figure 3.11: The normalised rise time versus the damping ratio for a second order response.... 46
Figure 3.12: The relationship between the percentage overshoot and the resonant peak of a second order model.... 48
Figure 3.13: The normalised settling time versus the damping ratio of a second order step response....49
Figure 4.1: Using the set of response prototype models for a command input to derive the other sets of prototype models.... 63
Figure 4.2: Using the loop transmission to define all the sets of prototype models.... 69
Figure 1-5.1: Example of a plant template.... 70
Figure 5.2: The -3dB m-circle plotted on the Nichols chart.... 72
Figure 5.3: The bandwidth bound for an uncertain plant.... 73
Figure 5.4: M-circles plotted on the Nichols chart....74
Figure 5.5: Calculating the stability bound.... 75
Figure 5.6: Bounds on the loop transmission set due to the closed loop tracking bounds for a single degree of freedom system.... 77
Figure 5.7: Tracking bounds for an uncertain plant of a one degree of freedom system.... 78
Figure 5.8: Calculating an uncertainty bound.... 79
Figure 5.9: Examples of uncertainty bounds on the nominal loop transmission....80
Figure 5.10: The m-circles plotted on the rotated Nichols chart.... 82
Figure 6.1: The Bode cut-off characteristic.... 89

Figure 6.2: The Bode semi-infinite characteristic........................................................................91
Figure 6.3: The Bode step characteristic with,and........................................................................94
Figure 6.4: The Bode plot of the Bode ideal characteristic............................................................95
Figure 6.5: The Nichols chart of the Bode ideal characteristic......................................................96
Figure 6.6: The Bode plot of the Bode ideal characteristic with a pole at the origin. .................100
Figure 6.7: The Nichols chart of the Bode ideal characteristic with a pole at the origin. ...........100
Figure 6.8: Bounds due to the gain margin for a single frequency...............................................105
Figure 6.9: The relationship between the gain margin, phase margin and resonant peak specifications..............................................................................................................107
Figure 6.10: Relating the gain margin property of the Bode ideal characteristic to the resonant peak specification...................................................................................................108
Figure 6.11: Setting the phase margin property of the Bode ideal characteristic to the phase margin of the m-circle related to the 0 dB point of the resonant peak specification...................109
Figure 6.12: Setting the phase margin property of the Bode ideal characteristic to the maximum phase of the m-circle related to the resonant peak specification.................................................109
Figure 6.13: A bode plot of the pitch loop....................................................................................116
Figure 6.14: The Bode plot of the magnitude of the initial nominal plant selected for the pitch control loop...............................................................................................................116
Figure 6.15: The Bode plot of the new nominal plant..................................................................117
Figure 6.16: The Bode plot of the new nominal plant and the plant set.......................................118
Figure 6.17: The plant templates for the pitch control loop.........................................................118
Figure 6.18: The stability bounds due to the 1 dB resonant peak at a number of frequencies.....119
Figure 6.19: The Bode cut-off part of the Bode ideal characteristic plotted with the stability bounds it must meet....................................................................................................122
Figure 6.20: The desired Bode Ideal characteristic......................................................................123
Figure 6.21: The Bode plot of the rational transfer function fitted to the desired Bode Ideal Characteristic..............................................................................................................124
Figure 6.22: The Nichols chart of the rational transfer function fitted to the desired Bode Ideal Characteristic..............................................................................................................125
Figure 6.23: The Bode plot of the required controller..................................................................126
Figure 6.24: The Nichols chart of the loop transmission.............................................................127
Figure 6.25: The Bode plot of the loop transmission set..............................................................128
Figure 6.26: The closed loop frequency response.........................................................................129
Figure 6.27: The response to the command input........................................................................130
Figure 6.28: The transfer function relating the plant input to the command input.....................131
Figure 6.29: The time response of the plant input........................................................................132
Figure 6.30: The final response to the command input................................................................134

Figure 6.31: The final time response of the plant input. 134
Figure 6.32: Upper and lower bounds on the systems' closed loop response for single degree of freedom QFT designs. 136
Figure 6.33: Uncertainty bounds for two degree of freedom QFT designs. 136
Figure 6.34: Upper bound on the sensitivity function for use in H-infinity design. 137
Figure 7.1: The Bode plot of an unstable pole. 141
Figure 7.2: The Bode plot of an unstable complex pole pair. 142
Figure 7.3: The all-pass function for an unstable real pole. 144
Figure 7.4: The all-pass function for an unstable complex pole pair. 145
Figure 7.5: A bode plot of the bank loop. 153
Figure 7.6: Phase versus frequency of the all-pass function. 155
Figure 7.7: The Bode plot of the new nominal plant and the plant set. 155
Figure 7.8: The plant templates for the bank control loop. 156
Figure 7.9: The stability bounds due to the 1 dB resonant peak at a number of frequencies. 157
Figure 7.10: The desired Bode Ideal characteristic. 160
Figure 7.11: The Bode plot of the rational transfer function fitted to the desired Bode Ideal Characteristic. 162
Figure 7.12: The Nichols chart of the rational transfer function fitted to the desired Bode Ideal Characteristic. 162
Figure 7.13: The Bode plot of the required controller. 163
Figure 7.14: The Nichols chart of the loop transmission. 164
Figure 7.15: The Bode plot of the loop transmission set. 165
Figure 7.16: The closed loop frequency response. 165
Figure 7.17: The response to the command input. 167
Figure 7.18: The transfer function relating the plant input to the command input. 168
Figure 7.19: The time response of the plant input. 169
Figure 7.20: The loop transmission set for the fitted controller. 170
Figure 7.21: The closed loop response set for the fitted controller. 171
Figure 7.22: The closed loop step response set. 171
Figure 7.23: The closed loop step response set with the pre-filter added and the steady state error removed. 173
Figure 8.1: The Bode plot of a non-minimum phase zero. 176
Figure 8.2: The Bode plot of a non-minimum phase complex zero pair. 177
Figure 8.3: The all-pass function for a non-minimum phase zero. 179
Figure 8.4: The all-pass function for a non-minimum phase complex zero pair. 180
Figure 12.1: The QFT plant template. 218

Figure 12.2: Calculating the minimum and maximum phase of a QFT template at a specified magnitude....226
Figure 12.3: Calculating the minimum and maximum magnitude of a QFT template at a specified phase....228
Figure 12.4: Calculating a reference point for the plant template....235
Figure 12.5: Converting the position of a plant in the plant set to polar co-ordinates....236
Figure 12.6: Fitting a line between two plants in the QFT templates plant set....238
Figure 12.7: The stability bound and its properties....239
Figure 12.8: Sampling of the m-circles phase....241
Figure 12.9: Finding the nominal phase of the QFT template which places its boundary on one of the m-circle samples....242
Figure 12.10: Finding the nominal phase of the QFT template that places one of its boundary points on the m-circles boundary....243
Figure 12.11: Calculating the minimum magnitude of the stability bound....245
Figure 12.12: Calculating the maximum magnitude of the stability bound....247
Figure 12.13: Calculating the minimum phase of the stability bound....249
Figure 12.14: Calculating the maximum phase of the stability bound....251

# List of Tables

Table 1: The steady state error due to power of time command inputs........................................ 51
Table 2: The steady state error due to power of time disturbance inputs.......................................54
Table 3: The steady state error due to a power of time command input for a system with the Bode ideal characteristic as the loop transmission................................................................................ 102
Table 4: The steady state error due to a power of disturbance input for a system with the Bode ideal characteristic as the loop transmission................................................................................ 103
Table 5: The steady state error due to a power of time command input for a system with the Bode ideal characteristic as the loop transmission and unstable poles................................................. 148
Table 6: The steady state error due to a power of time disturbance input for a system with the Bode ideal characteristic as the loop transmission and unstable poles........................................ 150
Table 7: The steady state error due to a power of time command input for a system with the Bode ideal characteristic as the loop transmission and non-minimum phase zeros............................. 183
Table 8: The steady state error due to a power of time disturbance input for a system with the Bode ideal characteristic as the loop transmission and non-minimum phase zeros.....................184

## List of Symbols

| | |
|---|---|
| $D_n(s)$ | The transfer function of the $n^{th}$ disturbance entering the plant. |
| $E(s)$ | The transfer function of the error in the closed loop system response. |
| $F(s)$ | The transfer function of the pre-filter. |
| $G(s)$ | The transfer function of the loop compensator. |
| $H(s)$ | The transfer function of the sensor dynamics. |
| $L(s)$ | The transfer function of the loop transmission. |
| $L_0(s)$ | The transfer function of the nominal loop transmission. |
| $N(s)$ | The power spectrum of the sensor noise. |
| $P(s)$ | The transfer function of the plant. |
| $P_0(s)$ | The transfer function of the nominal plant. |
| $P_{Dn}(s)$ | The transfer function of the plant's response to the $n^{th}$ disturbance entering the plant. |
| $S(s)$ | The sensitivity function (in the frequency domain). |
| $T(s)$ | The complementary sensitivity function (in the frequency domain). |
| $U(s)$ | The transfer function of the plant input. |
| $W_1$ | The performance weighting function. |
| $W_2$ | The uncertainty weighting function. |
| $Y(s)$ | The transfer function of the plant output. |
| $t_f$ | The maximum time the closed loop system may take to fall to the percentage fall time specification, $x_f$ , for a step response to the command input. |
| $t_r$ | The maximum time the closed loop system may take to reach the percentage rise time specification, $x_r$ , for a step response to the command input. |
| $t_s$ | The maximum time the closed loop system may take to reach the percentage settling time specification, $x_s$ , for a step response to the command input or a disturbance input. |
| $x_f$ | Percentage fall time specification for a step response to the disturbance input. (%/100). |
| $x_{os}$ | Percentage overshoot specification for a step response to the command input. (%/100). |
| $x_r$ | Percentage rise time specification for a step response to the command input. (%/100). |
| $x_s$ | Percentage settling time specification for a step response to the command input or a disturbance input.(%/100). |
| $x_{us}$ | Percentage undershoot specification for a step response to the disturbance input. (%/100). |

# 1 Introduction

There are a number of theories available to design compensators which are used to control plants of many types. These theories vary from simple algorithms which tune predefined compensator structures in order to get the best possible performance from them, to theories that build up the required compensator structure based on the plant and specifications on the desired performance.

The algorithms which tune a predefined compensator structure are common in process control problem's where PID controllers are tuned in an attempt to get an optimum response for the predefined structure. There are algorithms which can be used in to identify the parameters of the compensator such as the Ziegler-Nichols tuning for PID compensators (Franklin et al, 1986). There has also been research into using expert systems to tune lead-lag compensators (James et al, 1987).

For control systems which do not have a predefined compensator structure there are a number of theories which may be used for designing the compensator. This extends from classical control methods, which do not take plant uncertainty into account, to more modern methods such as quantitative feedback theory and H-infinity optimal control theory which do.

These theories however require the specifications to be posed in a very specific manner, which is often not easily obtained from the specifications posed on the system performance as defined in the plant domain. The system performance specifications are usually in a form which suites the domain of knowledge of the plant, whereas the control algorithm design procedure specifications are more general in nature.

Although it is common practice to fit a set of dynamics to the control system specifications in order to translate the control system specifications into the form desired by the control theory used, this is not sufficient. This is due to the fact that the control system specifications are often not complete from the control theory perspective and it is often required that the control system provide the best possible performance from the plant for the specifications that are missing. The best possible performance is viewed from the domain of knowledge in which the plant resides, rather than from a general dynamics perspective.

The common method of deriving prototype models, as discussed in the previous section, often makes use of a predefined structure, such as second order rational transfer function or a third order rational transfer function with a zero (Horowitz, 1993). This structure does not directly take into account the dynamics of the plant which is desirable. An experienced designer, in the field, can from previous experience select prototype models which give performance requirements which are achievable by the plant.

For example, in the case of aircraft, research has been done into the desirable dynamic characteristics of an aircraft from a pilots perspective (Stevens, 2003). These dynamic characteristics can be used to define the acceptable prototype models used in the design of an autopilot. The aircraft should have been designed to have the performance which will meet these requirements on its dynamics. Therefore, the aircraft should be capable of meeting the prototype models derived using these dynamic characteristics.

Therefore, a method is needed which allows the designer to generate the specifications in a form required by the control theory to be used, while allowing the designer to:

1. Evaluate the set of specifications placed on a control system for consistency.
2. Generate the specifications that are missing, but required by the control system design algorithm which is to be used. Further, these specifications must be consistent with the other specifications placed on the system as well as the plant dynamics.

This report proposes a method for generating the specifications required by modern control theories such as quantitative feedback theory and H-infinity optimal control theory, which allow the designer to evaluate them from the perspective of the plant performance. The method developed will not cater for multi-input single output (MISO) plants. Plants with unstable poles or non-minimum phase zeros will be catered for.

The method developed uses the Bode ideal characteristic as a nominal loop transmission in order to generate a set of dynamics, which will be referred to as prototype models, for use in specifying control system bounds. This method's goals are:

1. To generate the prototype models which meet the specifications that were provided.
2. To provide the designer with a means of trading off the control loop characteristics which are not covered by the specifications and so generate any specifications which are missing.
3. To bring the plant characteristics into the generation and evaluation of the prototype models.

Lurie and Enright (2000) in their book on classical feedback control, use the Bode ideal characteristic for loop shaping in classical frequency control designs. The Bode ideal characteristic is used for the loop transmission, but no plant uncertainty is catered for. As far as is known, there is no literature on using the Bode ideal characteristic to generate and evaluate prototype models in order to generate a consistent set of specifications for use in control system design.

The project report is structured as follows:

Chapter 2 discusses the control system design process. In order to fully understand the problem discussed in this report, the high level control system design process and the possible architectures of a control system must be considered.

Chapter 3 discusses the common operational requirements placed on the control system. These requirements place bounds on the response of the control system to the command input, disturbance inputs and sensor noise. These bounds are generally defined in the time domain. However, as the method discussed in this report is in the frequency domain, the common method of using a second order dynamic model to estimate the frequency response version of these bounds is discussed (Horowitz, 1993).

Chapter 4 discusses different strategies for determining a set of prototype models which meet all the bounds placed on the systems response to the command input, disturbance inputs and sensor noise. The strategy which makes use of the loop transmission to find the set of prototype models is found to be the most useful.

Chapter 5 discusses a method of translating the bounds on the closed loop control systems response in the frequency domain to bounds on the loop transmission. This is further extended to placing bounds on a nominal loop transmission, selected by the designer, despite the plant uncertainty.

Chapter 6 discusses a method for determining the set of prototype models that meet the specifications while taking the plants dynamics into account. For a stable minimum phase.

Chapter 7 discusses a method for determining the set of prototype models that meet the specifications while taking the plants dynamics into account. For plants with unstable poles.

Chapter 8 discusses a method for determining the set of prototype models that meet the specifications while taking the plants dynamics into account. For plants with non- minimum phase zeros.

# 2 The control design process

In this chapter the complete control design process (Houpis et al, 1966, 1999) will be discussed. The process discussed here can vary slightly depending on the type of plant for which the control system is being designed.

For a simple system which has a plant which is linear or very close to linear may only require linear simulation before the control algorithm is tested in the system. An example of such a system is an oven with temperature control. These systems are often modelled using a first order linear differential equation, so a full non-linear simulation will not be necessary. Saturation of the heating element will however need to be looked at, but in this case the control algorithm and the effects of saturation can be safely tested using the actual oven itself. This is due to the fact that the system is slow and human intervention can easily shut down the system before any damage can occur.

A more complex system such as an aircraft, is non-linear and has coupling between the individual control loops. Some loops are also multi-input multi-output in nature. In this case the whole process discussed here will probably be used, as the control algorithm cannot be tested using the aircraft without risking life or serious damage to the aircraft. So the control algorithm will be tested as much as possible using linear and non-linear simulation techniques, in order to reduce the risk to life and the aircraft.

The design process is represented in Figure 2.1. The white blocks represent the tasks that the system design team follow and the gray blocks represent the output of these tasks. The arrows show the order in which the steps are performed. It should be noted that at any time in the design process, should it become apparent that the functional requirements should not be met, then the design team would need to re-iterate some of the steps in order to insure that the requirements are met. This has not been represented on the diagram as it will unnecessarily complicate it.

It is also possible that it may be found that the functional requirements are impossible to meet and will therefore have to change in order for them to be attainable.

## *2.1 Functional requirements*

The functional requirements are the high level requirements on the system and are usually defined by the client who requires the system. These requirements generally consist of the following two types of requirements:

1. Operational requirements
2. Environmental requirements

The environmental requirements define the type of environment in which the system is intended to function.

The operational requirements define the tasks that the system is required to perform, as well as placing constraints on the systems' performance. From a control perspective these requirements can be expressed as the following bounds placed on the response of the control system:

1. The time response of the control system to a reference command signal. This response is often broken up into the steady state and transient response of the control system.
2. The control systems sensitivity to plant variations and measurement noise.
3. The time response of the system due to process disturbances entering it.

Not all of the above mentioned specifications may be defined. It is even possible that no specifications are defined at all and it is the design teams responsibility to simply get the best possible performance from the system.

The operational requirements that may be specified are discussed in more detail in section 3.

### 2.1.1 Physical plant

If the plant does not already exist, experts in the field of the plant type, design the physical plant based on the functional requirements. Ideally the control engineers will have an input to the design, as they need to ensure that the plant inputs are effective enough to allow the operational requirements to be met. However, this is not always the case, and the control engineers sometimes have to make the best of the plant that has been designed.

### 2.1.2 Plant model

The plant model is a theoretical dynamic model of the plant. This plant model is often non-linear and/or time varying in nature and is constructed using inputs from the design of the physical plant and measurements from the actual physical plant (for modeling and verification).

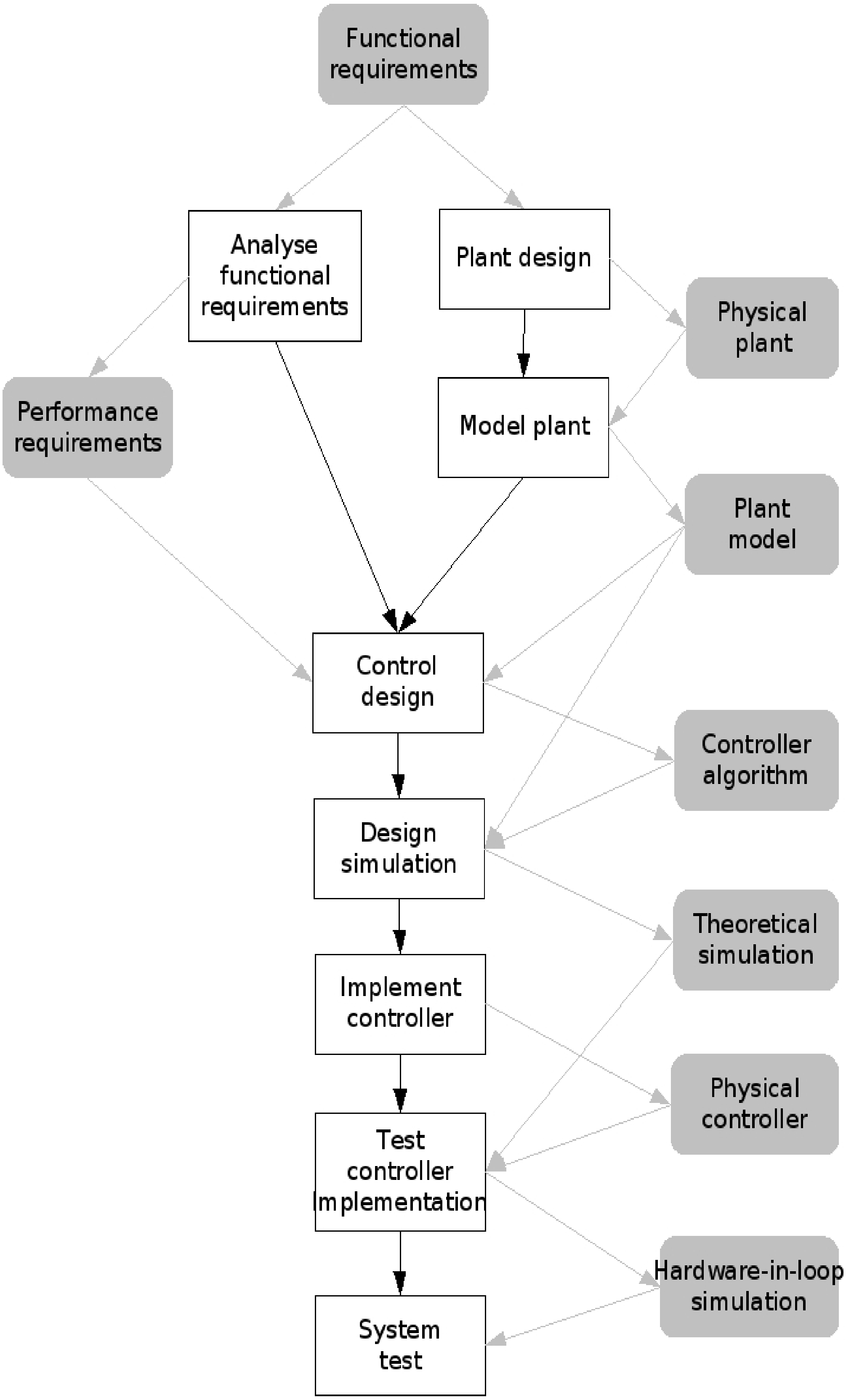


**Figure 2.1: The control system design process.**

The team or person responsible for the model should make every effort to ensure that the model should contain all the important dynamics. However, there will always be uncertainty in the plant model. These uncertainties can be divided into two types:

1. Structured uncertainties, and
2. unstructured uncertainties.

Structured uncertainties are those uncertainties which give rise to uncertainty in the parameters used in the model. This can be due to variations due to tolerances in the manufacture of components that make up the plant or changes in the environment, which are modelled but are not measured, which change the dynamics of the system. The unstructured uncertainties are the dynamics which are part of the physical plant but not part of the model and therefore can not simply be incorporated into the plant dynamics.

The model is normally linearised for use in the design of the controller. This linearisation is done to create a set of models which represent the non-linear plant at a number of points over its range of operation. These models can be generated using a number of methods such as

- modeling small perturbations around the non-linear models equilibrium points or
- generating linear time invariant equivalent (LTIE) models

The linear time invariant equivalent models are linear models which give the same output as the non-linear system that is being linearised for a given input (Horowitz, 1993).

Steps may also be taken to reduce the nonlinearity of the plant before the set of linear models are generated. A common method is by non-linear cancellation (Horowitz, 1993). This is achieved by adding non-linearities to the output of the compensator in an attempt to minimize the plants non-linearity. An example of this is adding the inverse of the throttle to engine speed characteristic of an internal combustion engine to the speed control compensator. This will not perfectly cancel the non-linearity, but it will help make the plants performance more linear in nature. This cancellation is then often added to the plant model for design purposes.

The variation in the plant due to external disturbance inputs or the plants non-linearity can also be reduced before generating the set of linear models. The most common method of achieving this is by gain scheduling. Gain scheduling makes use of different compensators for different operating conditions of the plant. Normally the structure of the compensator remains fixed but parameters used by the compensator vary with the operating conditions and disturbance inputs. This however requires that the external disturbance inputs and or operating conditions can be measured.

The plant under consideration will therefore be a set of models having the following form in the time domain,

$$a_n \frac{d^n y}{d t^n} + a_{n-1} \frac{d^{n-1} y}{d t^{n-1}} + \ldots + a_1 \frac{dy}{dt} + a_0 = b_m \frac{d^m u}{d t^m} + b_{m-1} \frac{d^{m-1} y}{d t^{m-1}} + \ldots + b_1 \frac{dy}{dt} + b_0$$

which simply gives the following transfer function in the frequency domain,

$$P(s) = \frac{Y(s)}{U(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \ldots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \ldots + a_1 s + a_0}$$

Plants containing pure time delays will not be considered, as the the time delay can be approximated by the Pade approximation (Stevens,2003)(Franklin,1986) which will put the plant in the form above.

### 2.1.3 Performance requirements

As no actual dynamic response is necessarily specified in any of the functional requirements, the desired dynamics of the control system cannot necessarily be directly obtained from the operational requirements. This leaves the control system designer with the task of selecting responses (dynamics) that meet the specifications. These responses will be referred to as the response prototype models.

The word prototype, in this context, is used to highlight the fact that these responses are not in fact the requirements that must be met, but an attempt to specify dynamics consistent with the physical plant and the functional requirements. It is possible that a control algorithm exists that meets the functional requirements but not the prototype models. This is due to the difficulty of finding dynamics that are consistent with all the functional requirements and the dynamics of the physical plant. This issue is discussed in more depth in section **4**.

### 2.1.4 Controller algorithm

The next task is to determine dynamics which will be used in the control algorithm to modify the performance of the closed loop system so that it meets the functional requirements. The inputs to this stage are the prototype models, which define the desired closed loop dynamics, and the plant models, which define the dynamics of the current open loop system.

The most common robust control design processes used are:

1. Quantitative feedback theory (QFT)
2. Optimal control theory (such as Linear Quadratic Gaussian (LQG) and H-infinity control theory)

A very brief overview of both these theories will be discussed in the following sections.

### 2.1.4.1 Quantitative feedback theory

The quantitative feedback theory (QFT) design method (Horowitz,1993)(Houpis,1999) requires the performance requirements to be bounds on the dynamics of the closed loop responses. These bounds are determined from the prototype models (see section 2.1.3) which were calculated from the operational requirements (see section 2.1).

The plant model (discussed in section 2.1.2) is converted to the frequency domain, if it is not already in that form. A nominal plant is chosen, which will be used in the rest of the design process. The nominal plant chosen has no effect on the outcome of the design, but it has an effect on the difficulty in implementing the design process (Horowitz,1993).

The plant set and the nominal plant are then used to generate plant templates. A plant template is a region in the Nichols chart defining all the magnitudes and phases that an uncertain plant may have at a particular frequency relative to the magnitude and phase of the nominal plant.

The plant templates and the closed loop bounds are used to generate bounds on the nominal loop transmission, which is made up of the nominal plant and the loop compensator. The designer then adds dynamic elements (such as poles, zeros, lead-lag terms) to the nominal plant such that the nominal loop transmission generated meets the bounds generated.

### 2.1.4.2 Optimal control theory

The goal of optimal control (Toivonen,1998)(Doyle et al,1990) is to design a controller which gives a stable closed loop system which minimizes a cost function used to measure the performance of the system.

The optimal control problems of interest here, are the ones which guarantee robust stability. Robust stability is achieved if the following h-infinity norm is met,

$$\left\| W_2 \cdot T \right\|_\infty < 1$$

Where $W_2$ is the uncertainty weighting function which relates the plant set to the nominal plant, $P$, in the following manner,

$$\tilde{P}=(1+\Delta\cdot W_2)\cdot P$$

$\Delta$ is a variable stable transfer function which satisfies $\|\Delta\|_\infty<1$. The uncertainty weighting function describes the upper bound on the magnitude of the plant uncertainty and is calculated from the plant model which was discussed in section 2.1.2. Methods for calculating the weighting function will not be discussed in detail here. Doyle, Francis and Tannenbaum (1990) give a detailed discussion on the subject.

The cost function used to measure the performance of the closed loop system may take many forms. The cost function may be placed on the performance of the nominal plant or the whole plant set. If the H-infinity bound is used, an upper bound can be placed on the cost function which defines the minimum performance that is considered acceptable.

The H-infinity bound placed on the nominal performance normally takes the following form.

$$\|W_1\cdot S\|_\infty<1$$

And the bound on robust performance (a performance specification on the entire plant set) often has the following form,

$$\||W_1\cdot S|+|W_2\cdot T|\|_\infty<1$$

The transfer function $W_1$ is a weighting function which is used to define the desired performance. Note that the performance bounds are placed on the sensitivity function rather than the complementary sensitivity function as in the case of the robust stability bound. The robust performance bound contains both the uncertainty weighting function and the complementary sensitivity function in order to insure that the performance weighting bound is met for the whole plant set.

The performance weighting function must be determined from the prototype models (see section 2.1.3) which were calculated from the operational requirements (see section 2.1).

### 2.1.5 Theoretical simulation

Once a control strategy has been designed, the strategy is tested by simulation. The simulation is normally done in two stages

1. Linear simulation
2. Non-linear simulation

Linear simulation of the system is normally done before non-linear simulation as it is simpler and results are achieved quicker. This is done in order to add extra confidence in the design before the time and expense is used in generating a non-linear simulation.

The linear simulation is used to verify the control algorithm making use of the linear models used in the control algorithm design process. The goal is to identify any errors made when implementing the control design process.

The non-linear simulation is used to verify that the control algorithm will meet the performance specifications when implemented on the full non-linear plant. Note that this does not guarantee that the control algorithm will work in practice as it relies on the accuracy of the non-linear model. It does however, check that the method used and the assumptions made when generating the simplified linear models are correct.

### 2.1.6 Physical controller

Now that the control algorithm has been designed and verified by simulation, the control algorithm has to be manifested in physical form.

The physical controller may be implemented in a number of ways, whether it be discrete electronics which implement the desired dynamics or code which runs on a digital system.

The interface between the the physical controller and the sensors and actuators  also needs to be implemented.

### 2.1.7 Hardware-in-loop simulation

The hardware-in-loop simulation is a simulation where the control hardware is linked to a non-linear simulation of the plant. This will allow for the verification of the physical implementation of the control algorithm.

This step is normally performed if serious consequences will result if the system for which the control algorithm has been designed does not function as specified. For example, a flight control system for an aircraft can cause loss of life and/or damage to an expensive aircraft if it malfunctions during testing. In this case it would be prudent to check the physical implementation in a simulated environment before testing the system as a whole. However, a control temperature control system for an incubator can be tested using the actual incubator, provided it contains no infant, as the response is slow enough to prevent damage before it can occur.

### 2.1.8 System test

The final step is to integrate the physical controller and its sensors and actuators into the physical plant. Tests are then performed to verify that the system's performance meets the functional specifications.

If these requirements are not met, it will necessitate a redesign of the controller and/or physical plant. This will be a very costly exercise, which highlights the need for the testing via simulation at intermediate stages in the design process.

## *2.2 The problem under consideration*

From the previous section it can be seen that there is a large body of theory which is used when designing a control system. This body of theory contains all kinds of engineering knowledge, from the specialized knowledge required to design and model the plant, to the more generalized knowledge used to simulate the plant and the control algorithm obtained.

There is also a selection of theories that can be used to design the necessary control algorithm. These theories however require the specifications to be posed in a very specific manner, which is often not easily obtained from the specifications posed on the system performance as a whole. The system performance specifications are usually in a form which suites the domain of knowledge of the plant, whereas the control algorithm design procedure specifications are more general in nature.

Although it is common practice to fit a set of dynamics to the control system specifications in order to translate the control system specifications into the form desired by the control theory used, this is not sufficient. This is due to the fact that the control system specifications are often not complete from the control theory perspective. It is often required that the control system provide the best possible performance from the plant for the specifications that are missing. The best possible performance is also viewed from the domain of knowledge in which the plant resides, rather than from a general dynamics perspective.

Therefore, a method is needed which allows the designer to generate the specifications in a form required by the control theory to be used, while guaranteeing that the control system specifications that exist are met while getting the best possible performance from the plant as seen from the plant domain.

A method which allows the designer to translate the control system specifications to a form useful for the control theory used to design the control algorithm is proposed in this report.

## 2.3 The architecture of the control system that will be considered

The architecture of the control system considered is a second degree of freedom architecture with a pre-filter, $F(s)$, and the loop compensator, $G(s)$, in the forward loop. A block diagram of this architecture is shown in Figure 2.2.

The plant considered is a multi-input, single output (MISO) plant. The plant has one desirable input, $U(s)$, and undesirable disturbance inputs, $D_1(s),\ldots,D_n(s)$. The response of the plants output, $Y(s)$, to the desirable input, $U(s)$, is given by a linear dynamic model $P(s)$, and the response to the disturbance inputs, $D_1(s),\ldots,D_n(s)$, is given by the linear dynamic models $P_{DI}(s),\ldots,P_{Dn}(s)$ respectively.

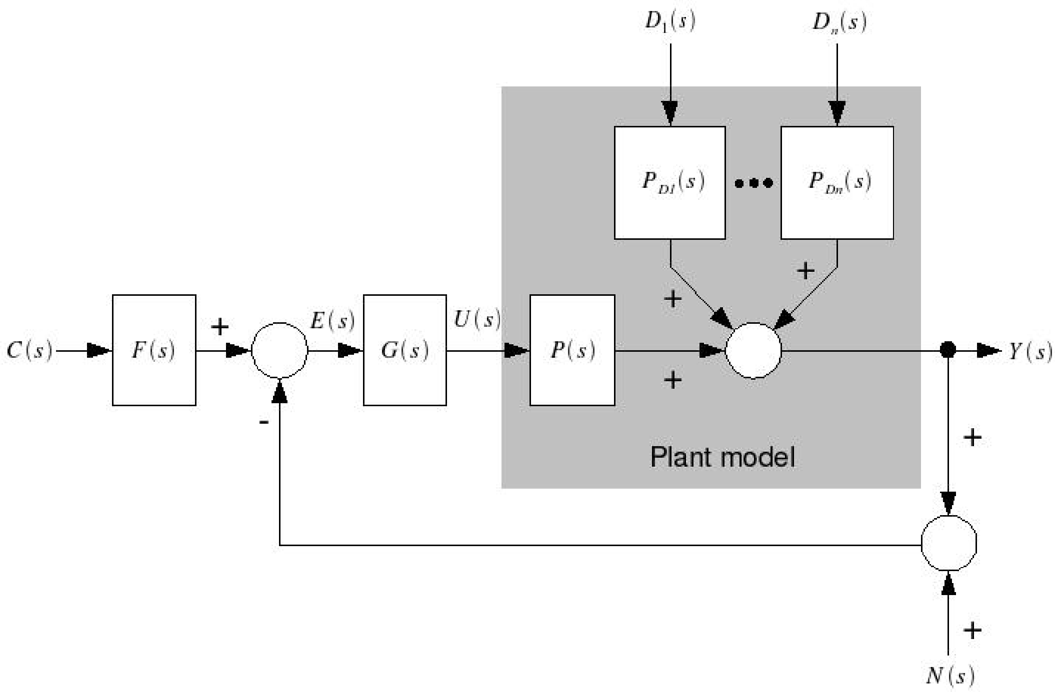


**Figure 2.2: A second degree of freedom control system architecture.**

The equation giving the output of the closed loop system output given all the system inputs is given by:

$$Y(s)=F(s)\cdot T(s)\cdot C(s)-T(s)\cdot N(s)+P_{DI}(s)\cdot S(S)\cdot D_1(s)+\ldots$$
$$+P_{Dn}(s)\cdot S(S)\cdot D_n(s)$$

where:

$$S(s)=\frac{1}{1+L(s)}$$ is the sensitivity function.

$$T(s)=\frac{L(s)}{1+L(s)}$$ is the complementary sensitivity function.

$$L(s)=P(s)\cdot G(s)$$ is the loop transmission.

The plant inputs response to the system inputs is given by:

$$U(s)=S(s)\cdot G(s)\cdot[F(s)\cdot C(s)-N(s)-P_{DI}(s)\cdot D_1(s)-\ldots-P_{Dn}(s)\cdot D_n(s)]$$

The error response (the difference between the command input and the output of the closed loop system) to the system inputs is given by:

$$E(s)=S(s)\cdot[F(s)\cdot C(s)-N(s)-P_{DI}(s)\cdot D_1(s)-\ldots-P_{Dn}(s)\cdot D_n(s)]$$

### 2.3.1 Other control system architectures

The are other possible control system architectures that can be considered. Figure 2.3 Shows an example of another two degree of freedom structure. Horowitz has shown that any two degree of freedom control system architecture can be transformed into any other two degree of freedom architecture (Horowitz, 1993). So the architecture discussed in the previous section will be used.

### 2.3.2 Dealing with sensor and actuator characteristics

The system has a sensor noise input, $N(s)$ and the dynamic characteristic is assumed to be unity, i.e. $H(s)=1$ . The actuator characteristic is assumed to modelled as part of the plants response to the control input, $P(s)$ . If this is not the case Horowitz has shown that the loop compensator and pre-filter can easily be modified to take into account the sensor and actuator characteristics (Horowitz, 1993), so the system need not be redesigned.

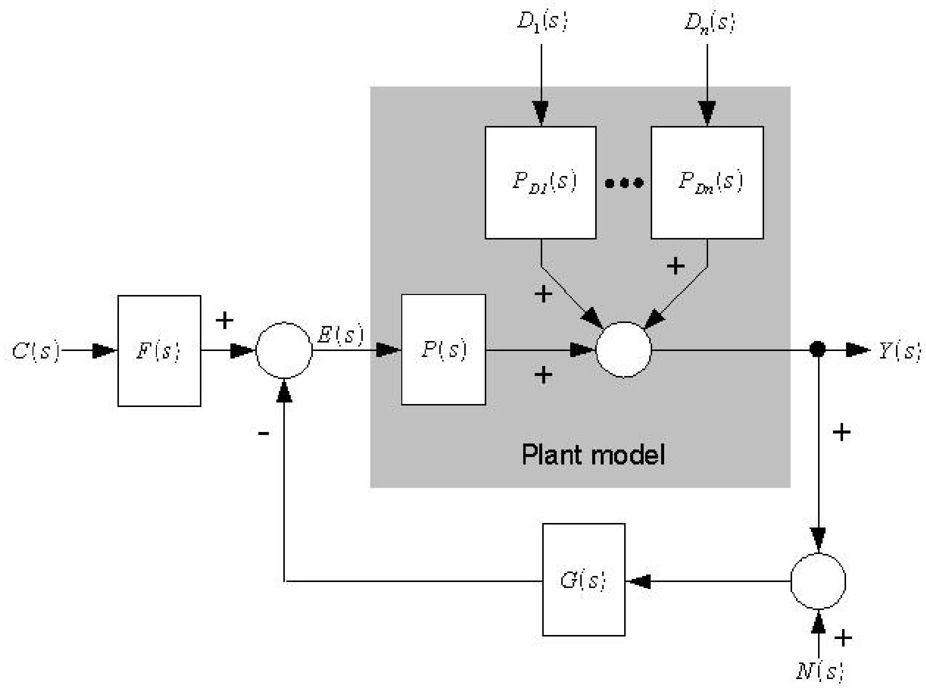


**Figure 2.3: An alternative second degree of freedom control system architecture.**

# 3 Operational requirements

As discussed in section 2.1, the operational requirements define the tasks that the system is required to perform, as well as places constraints on the systems' performance. This chapter will focus on the most likely form that the constraints placed on the systems' performance will have, as well as a simple method to translate these constraints into the frequency domain. The translation to the frequency domain is required as the method discussed later for generating and evaluating the prototype models from these constraints makes use of the frequency domain.

## *3.1 Time response of the control system to the plant input*

The time response specifications are usually expressed as bounds on the control system's response to a specified command input. The command input used is normally a step input as there are well defined and understood bounds on this type of response. The step response also effectively evaluates the dynamic and steady state behaviors of the control system as it commands an instantaneous change in the controlled variable, followed by commanding that it be held constant as time tends to infinity. Therefore, the step response will be mainly discussed.

A tracking system however, can have any type of command input. The designer of such a system will use a single command or set of commands that will be expected to be presented to the control system. This is discussed in more detail in section 3.1.3.

### 3.1.1 Step Response

The most common type of command input used to specify the performance of a control system is the step response. The step response is a systems response to a step input. The following sections describe some of the bounds that are often placed on the step response.

The bounds discussed are:

1. Rise time
2. Overshoot
3. Settling time

#### 3.1.1.1 Rise time

The rise time specification places bounds on speed of response of the control system. This specification defines the maximum amount of time, $t_r$, the control system may take to reach a certain percentage, $x_r$, of the step inputs' magnitude. The value of $x_r$ is normally 66.7%. However, the value of $x_r$ can be anything between 0 and 100%.

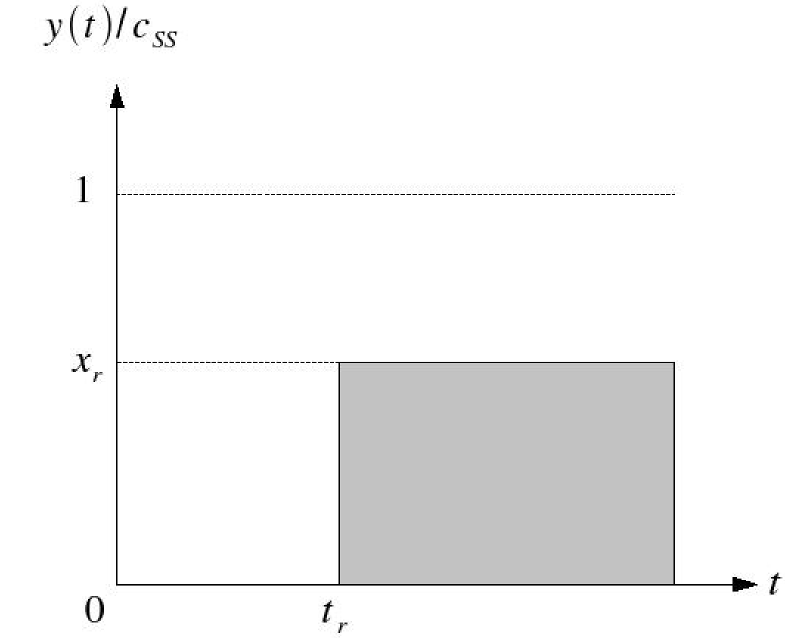


**Figure 3.1: The rise time bound on the step response specification.**

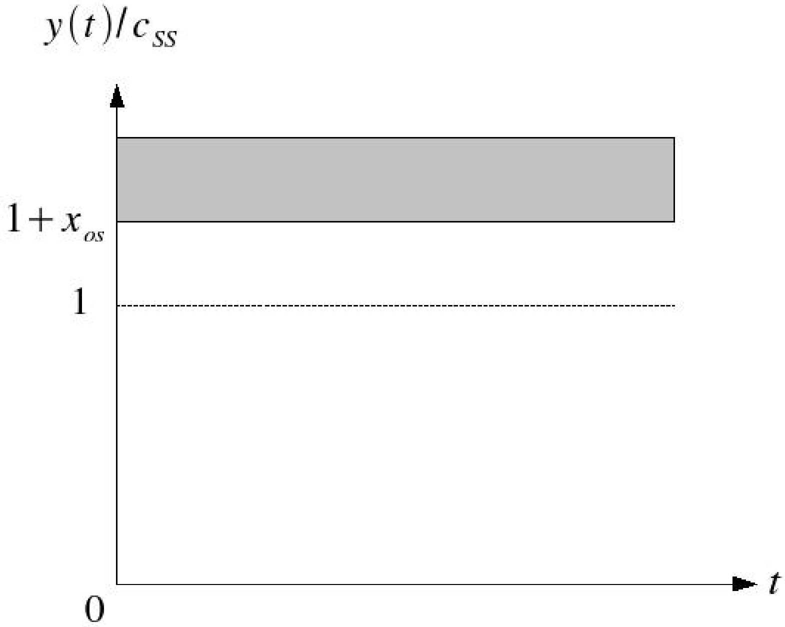


**Figure 3.2: The overshoot bound on the step response specification.**

Figure 3.1 shows a graphical representation of the rise time bound. The shaded block represents the region into which the step response may not go if it is to meet the specification.

#### 3.1.1.2 Overshoot

The overshoot specification expresses the maximum value that the step response may reach. This value is equal to or greater than the final value that step response reaches. The overshoot specification, $x_{os}$, is normally expressed as a percentage of the final value of the systems' response to the step input.

Figure 3.2 shows a graphical representation of the overshoot bound. The shaded block represents the region into which the step response may not go if it is to meet the specification.

#### 3.1.1.3 Settling time

The settling time bound places a bound on the maximum time, $t_s$, the system may take before it falls between $\pm x_s$% of the step inputs' magnitude (final value).

This bound is very useful when describing a regulator. As a regulator tries to keep a system output at a defined value. This bound then describes the maximum time the regulator may take before it starts to regulate effectively.

Figure 3.3 shows a graphical representation of the settling time bound. The shaded blocks represent the regions into which the step response may not go if it is to meet the specification.

### 3.1.2 Responses to other control inputs

The step response is not always the best means to describe the time domain response required. In a tracking system, for example, the input signal is seldom a step.

The designer normally knows the form of the input signal or signals. These signals will be a better way of describing the required time response of the control system.

Figure 3.4 shows the possible bounds a designer may place on a control system that will be used to track a truncated ramp input.

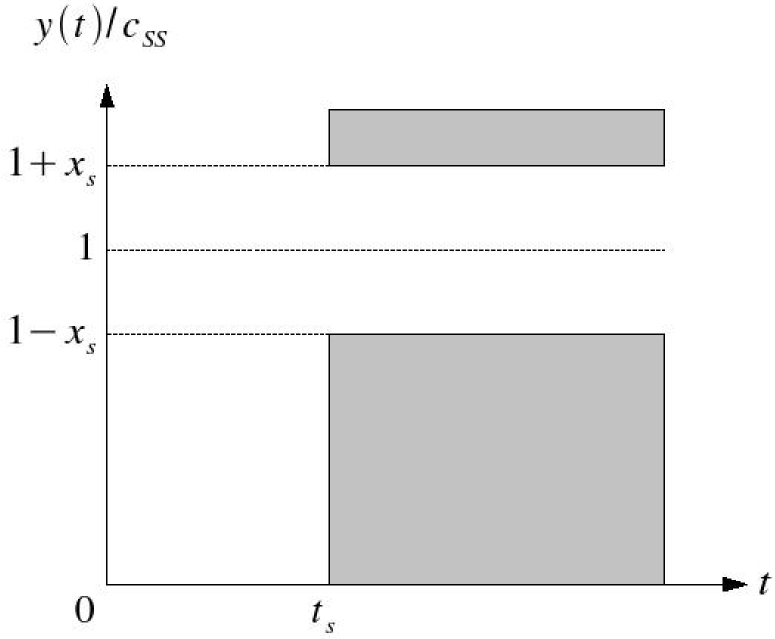


**Figure 3.3: The settling time bound on the step response specification.**

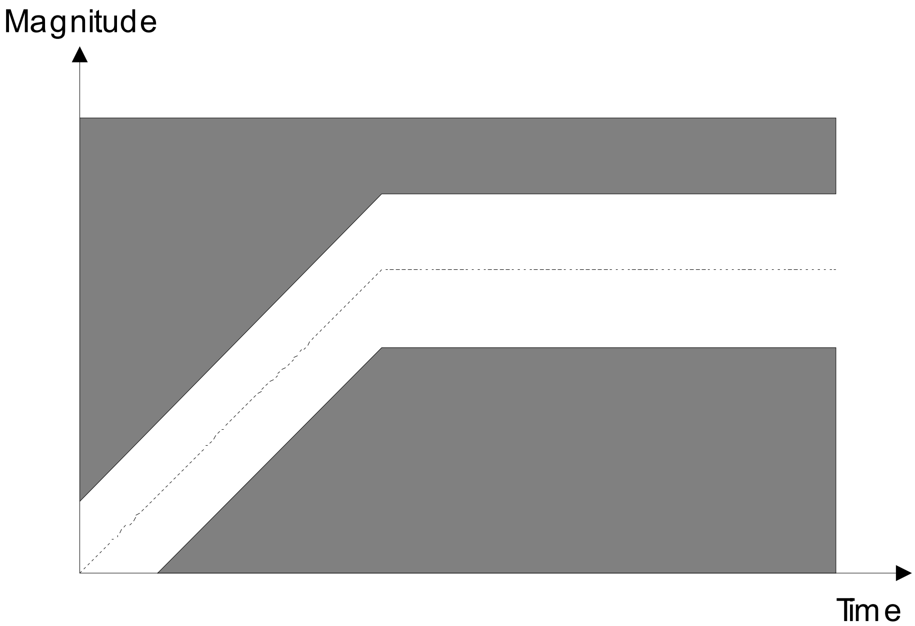


**Figure 3.4: An example of bounds on a systems response to a ramp input.**

There are no standard bounds for a ramp response as in the case of the step response. The designer is therefore left with the task of deciding on what will be reasonable and useful bounds to place on the control system. These bounds will be influenced by the problem domain in which the designer is working.

### 3.1.3 Tracking command inputs

The bounds on the tracking of the command input may be expressed as bounds on a single or set of command inputs that the control system is expected to encounter. These bounds then fall into the same category as the other command inputs which are not a step, discussed in the previous section.

Another method is to consider the control system as a low pass filter. This allows the designer to use the standard low pass filter bounds. The designer then only needs to ensure that these bounds do not “distort” the command input so as the system output has an unacceptable difference (error) between itself and the command input.

Figure 3.5 shows the bounds normally placed on a low pass filter (Stanley et al,1984). The passband defines the range of frequencies that should ideally have a magnitude of 0 dB, as this range of frequencies contains the significant frequency components of the command input set. For a low pass filter the phase in this band should be linear with frequency which will ensure that the command input is not distorted due to the phase characteristic of the system. This is not however stringent enough for a control system as this will allow the system to have a significant delay between the command input and the system output. For a control system it is also required that the phase in this band should remain close to $0^o$, as this will ensure that the time delay remains small.

In practice these ideal bounds cannot be met, so rather than requiring that the magnitude of the control system is 0 dB in this band, the designer should place bounds on how far the magnitude may deviate from 0 dB. Bounds on the phase may also be specified, but they are often unnecessary as the magnitude bounds will require a high loop gain which should ensure that the phase stays close to $0^o$.

The lack of bounds in the transition band, allows the system to meet the relative stability specification (such as the maximum overshoot specification). This band therefore does not need to be specified as these frequencies are covered by other specifications.

The stop band in a low pass filter defines the maximum magnitude that the filter may have in order to attenuate the noise in the input signal to an acceptable level. In a control system this range of frequencies normally contains the sensor noise as the designer would have selected a sensor that does not have any significant noise in the passband frequency range. Noise may also be present in the command input. The noise in the command input is due to the means by which it is calculated, which may contain measurements from other sensors. This noise can also be attenuated by the pre-filter. So the most important source of the noise would be the sensor used in the feedback loop. The stopband would therefore have similar bounds to the low pass filter, except that the noise would be more likely to originate from the sensor rather than the command input. The form of the noise specifications are discussed in more detail in section 3.3.

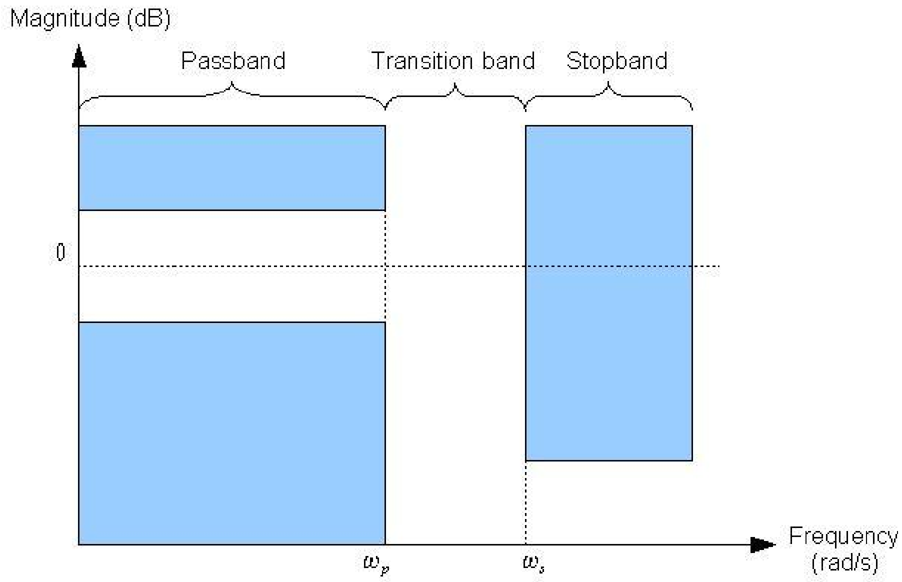


**Figure 3.5: Bounds on the system for the tracking of the command input.**

### 3.1.4 The steady state response of the control system to the plant input

The steady state error specification defines the maximum error that the control system may have when tracking a given command input as time tends to infinity.

The most common type of command input used when defining this specification is a power of time input of the following form

$$c(t)=A\cdot t^{i}$$

where

$A$ is the gain of the input.
$i$ is the power of time.

## 3.2 *Time response of the control system to disturbances entering the plant*

As disturbances are unwanted system inputs, it is desired that their effect on the controlled variable be negligible. This is not however always possible, so the best that can be asked of the control system is that it squash the effect on the output in a reasonable time and keeps the effect small enough not to adversely effect the systems performance.

A step response is the worst possible type of disturbance that a control system must contend with (Horowitz, 1993) as it has infinitely fast onset and then persists for infinite time. So the control system needs to react rapidly to squash the disturbance before it has a significant effect on the controlled variable, and then needs to continue rejecting it. A step response will therefore be used to define the desired disturbance rejection.

### 3.2.1 Step response

#### 3.2.1.1 Fall time

The fall time specification places bounds on speed of response of the control system to a disturbance entering the plant. This specification defines the maximum amount of time, $t_f$, the control system may take to fall to a certain percentage, $x_f$, of the step responses initial value for a disturbance entering at the plant output. In this case the initial value of the step disturbance response will be equal to the magnitude of the step disturbance. For a disturbance entering the plant at another point, where $P_D(s) \neq 1$, the initial value of the response to the disturbance is not the magnitude of the step input. In this case the initial value is normally zero as the transfer function $P_D(s)$ is often low pass in nature. The fall time will therefore define the maximum time that the systems response to a step disturbance may take to fall back below the value of $x_f$ percent of the step disturbance inputs' magnitude. The value of $x_f$ can be anything between 0 and 100%.

Figure 3.6 shows a graphical representation of the fall time bound. The shaded block represents the region under which the step response must pass if it is to meet the specification.

#### 3.2.1.2 Undershoot

The undershoot specification, $x_{us}$, expresses the minimum value that the disturbance step response may reach. The undershoot specification is normally expressed as a percentage of the step disturbances' magnitude and specifies the maximum magnitude that the disturbance step response may go below zero.

Figure 3.7 shows a graphical representation of the undershoot bound. The shaded block represents the region into which the step response may not go if it is to meet the specification.

#### 3.2.1.3 Settling time

The settling time bound places a bound on the maximum time, $t_s$, the system may take before it falls between $\pm x_s$% of zero.

Figure 3.8 shows a graphical representation of the settling time bound . The shaded blocks represent the regions into which the disturbance step response may not go if it is to meet the specification.

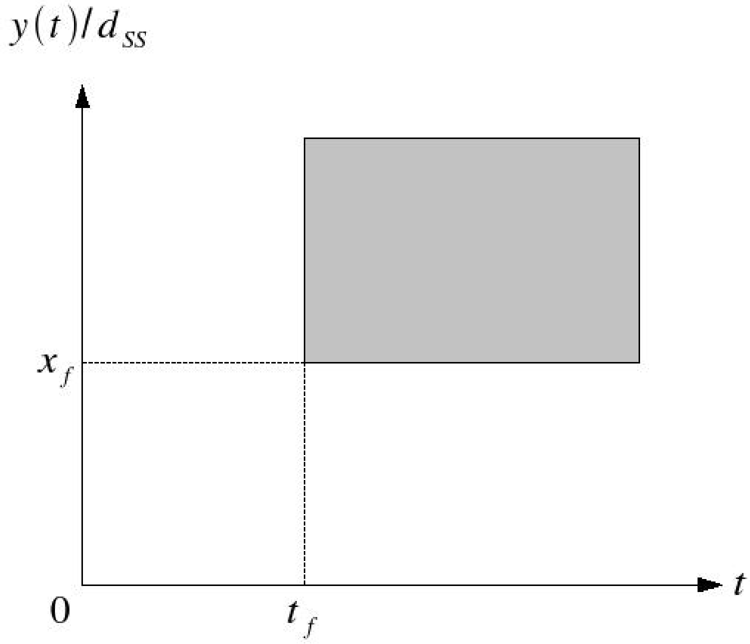


**Figure 3.6: The fall time specification on the step response specification to a disturbance input.**

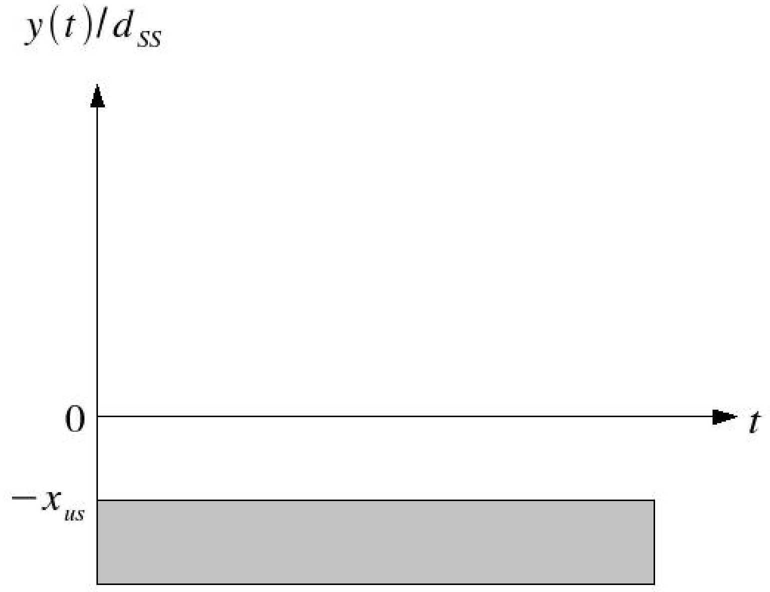


**Figure 3.7: The undershoot specification on the step response specification to a disturbance input.**

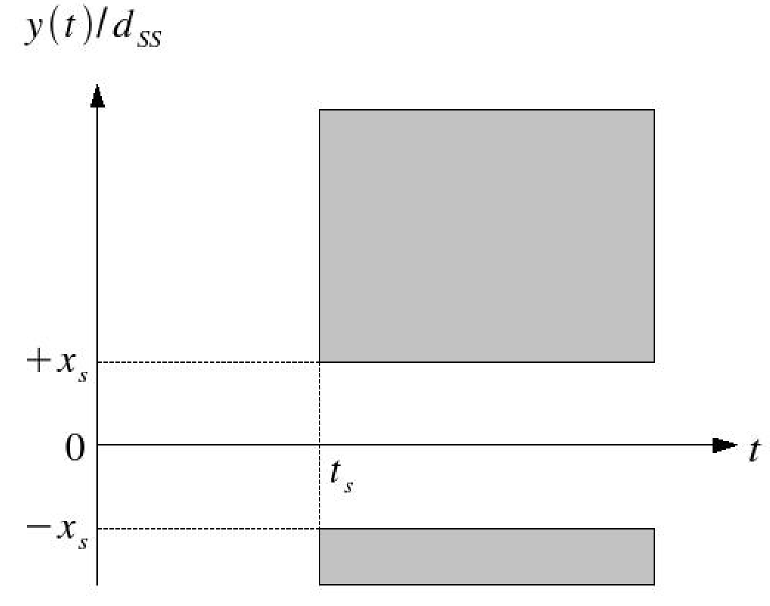


**Figure 3.8: The settling time specification on the step response specification to a disturbance input.**

### 3.2.2 The response to other disturbance inputs

As in the case of the command input, the disturbance need not be a step. The type of disturbance signals expected will differ from one problem domain to the next, so the type of bounds that will be useful will also vary. However, as a disturbance is an unwanted input to the system, the settling time specification discussed in section 3.2.1.3, should always be useful, as it defines the maximum time that the system may take before the disturbance falls below an acceptable level.

### 3.2.3 Steady state response to a disturbance input

The steady state response specification defines the maximum amplitude that the control system's response may have to a given disturbance input as time tends to infinity.

The most common type of disturbance input used when defining this specification is a power of time input of the following form (as in Section 3.1.4)

$$c(t) = A \cdot t^i$$

## *3.3 Response of the control system to sensor noise*

As the sensor noise is a random signal, the power spectrum of the noise signal is used rather than the Laplace transform.

The main concern of the control system designer should be to reduce the noise level at the output of the control system as much as possible. The usual measure of the size of a noise signal is the RMS value of the signal, which is given by:

$$N_{RMS} = \sqrt{\int_{-\infty}^{+\infty} |Y_N(j\omega)|^2 \, d\omega}$$

where:

$Y_N(j\omega)$ is the power spectrum of the noise at the control system output.

So the specification on the maximum allowed sensor noise is an upper bound on the RMS value of the sensor noise level at the system output.

A similar specification can be placed on the sensor noise at the plant input. This specification will have the following form

$$N_{U_{RMS}} = \sqrt{\int_{-\infty}^{+\infty} |U_N(j\omega)|^2 d\omega}$$

where:

$U_N(j\omega)$ is the power spectrum of the noise at the plant input.

## 3.4 Translating the time domain specifications into the frequency domain

As discussed in the previous section, the specifications on a control system performance are normally limits on the system response in the time domain to specific inputs to the system. These inputs could be the command input, one of the disturbance inputs or sensor noise. As the limits do not have dynamics they cannot be directly converted to the frequency domain. The common procedure is to find dynamics which meet the specifications and use them as bounds on the closed loop response in the frequency domain (Horowitz,1993).

Figure 3.9 Shows an example of bounds on the step response of the control system to the command input. The rise time $(t_r, x_r)$ , the settling time $(t_s, x_s)$ and the overshoot $(x_{os})$ specifications are shown, which where discussed in sections 3.1.1.1, 3.1.1.3 and 3.1.1.2 respectively. Two time responses that meet these specifications are also shown.

Figure 3.10 shows an example of bounds on the step response of the control system to a disturbance input. The settling time $(t_s, x_s)$ and the undershoot $(x_{us})$ specifications are shown, which where discussed in sections 3.2.1.3 and 3.2.1.2 respectively. Two time responses that meet these specifications are also shown.

### 3.4.1 Response of the control system to the plant input

If it is assumed that the closed loop system will have a response similar to a second order model, a second order response prototype model can be used to estimate the frequency characteristics required of the loop transmission in order to meet the specifications placed on the response to the plant input. Higher order models may also be used, such as a third order model with a zero (Horowitz,1993), but as the frequency response characteristics are only being used as an estimate, therefore this simple model will suffice.

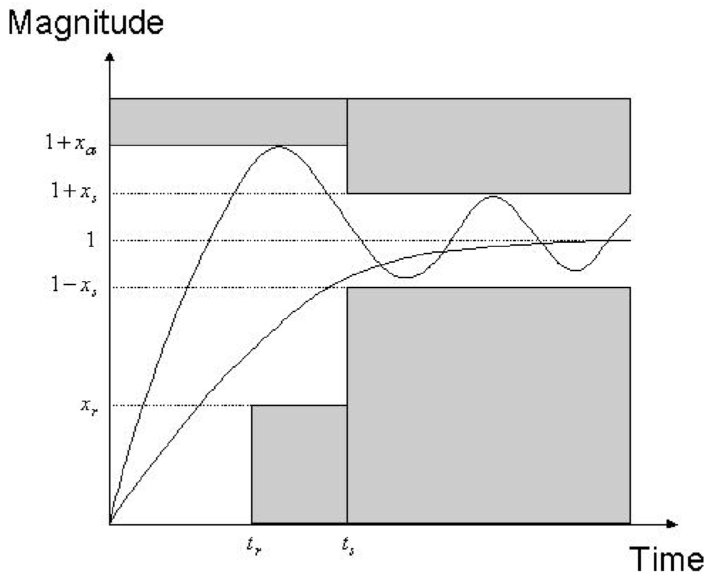


**Figure 3.9: Command response prototypes that meet the step response specifications.**

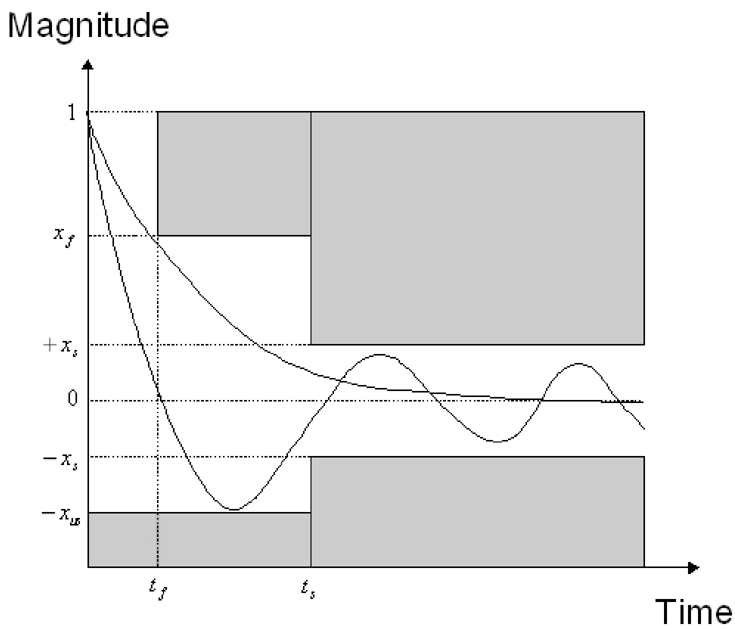


**Figure 3.10: Disturbance response prototypes that meet the disturbance specifications.**

The model used is described by the following differential equation,

$$\frac{1}{\omega_n^2}\cdot\frac{d^2y(t)}{dt^2}+\frac{2\cdot\zeta}{\omega_n}\cdot\frac{dy(t)}{dt}+y(t)=c(t)$$

where:

$y(t)$ is the closed loop system output with respect to time.

$c(t)$ is the closed loop system command input with respect to time.

$\zeta$ is the damping ratio.

$\omega_n$ is the natural frequency.

which gives the following response to a step input command with amplitude *A*. The step command was used to place bounds on the systems response to a plant input, see Section 3.1.

$$y(t)=\begin{cases} A-A\cdot e^{-\zeta\cdot\omega_n\cdot t}\cdot\left(\cos(\omega_0\cdot t)+\dfrac{\zeta\cdot\omega_n}{\omega_0}\cdot\sin(\omega_0\cdot t)\right) & for \quad \zeta<1 \\ A-A\cdot e^{-\zeta\cdot\omega_n\cdot t}\cdot(1+\omega_n\cdot t) & for \quad \zeta=1 \\ A-A\cdot e^{-\zeta\cdot\omega_n\cdot t}\cdot\left(\cosh(\omega_x\cdot t)+\dfrac{\zeta\cdot\omega_n}{\omega_x}\cdot\sinh(\omega_x\cdot t)\right) & for \quad \zeta>1 \end{cases}$$

where:

$$\omega_0=\omega_n\cdot\sqrt{1-\zeta^2}$$
$$\omega_x=\omega_n\cdot\sqrt{\zeta^2-1}$$

The above equations will now be solved for the following specifications placed on the step input:

1. Rise time
2. Overshoot
3. Settling time

Section 12.11 describes the code used to calculate the second order response that meets the specifications discussed below.

#### 3.4.1.1 Rise time

As discussed in section 3.1.1.1, the rise time defines the maximum amount of time, $t_r$, the control system may take to reach a certain percentage, $x_r$, of the step responses final value. This is shown in Figure 3.9. Due to the complex nature of the systems response to a step, this time must be solved for numerically.

Figure 3.11 shows the normalised rise time versus the damping ratio of a second order response for rise times of 10%, 20% 50% and 70%. The normalised rise time is simply the required rise time multiplied by the the natural frequency of the second order response. The designer can use a figure such as this to estimate the minimum natural frequency (closed loop bandwidth) required to meet the rise time specification given the damping ratio. The damping ratio is estimated from the overshoot specification, which is discussed in the following section.

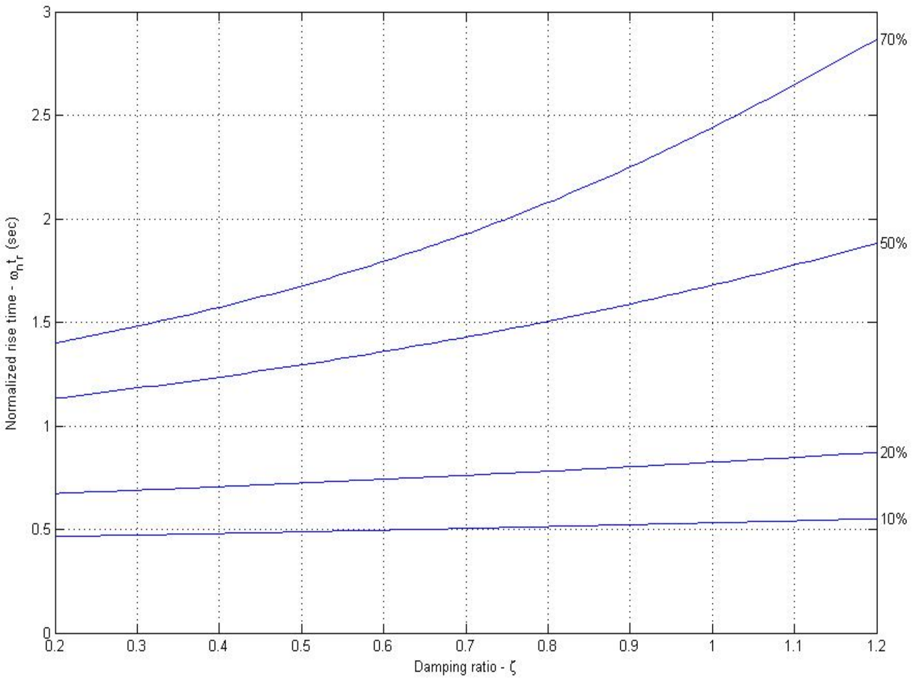


**Figure 3.11: The normalised rise time versus the damping ratio for a second order response.**

If no overshoot specification is given the figure can be used to estimate the trade off between the closed loop bandwidth of the system and its resonant peak, which is estimated from the damping ratio as discussed in the following section.

### 3.4.1.2 Overshoot

The overshoot specification is used to estimate the bound on the closed loop systems resonant peak. The first step is to convert the overshoot specification to a frequency domain specification through the use of the second order prototype model. This is achieved by calculating the damping ratio of the second order prototype model which has the overshoot specified. The damping ratio is given by:

$$\zeta=\frac{k_p}{\sqrt{\pi^2+k_p^2}}$$

where:

$k_p=\ln(M_p)$

$M_p$ is the specified percentage overshoot. Shown as $x_{os}$ in Figure 3.9.

The damping ratio is then used to calculate the resonant peak of the second order prototype which is given by

$$M_m=\frac{1}{2\cdot\zeta\cdot\sqrt{1-\zeta^2}}$$

Figure 3.12 shows the relationship between the percentage overshoot and the resonant peak of a second order model as discussed above.

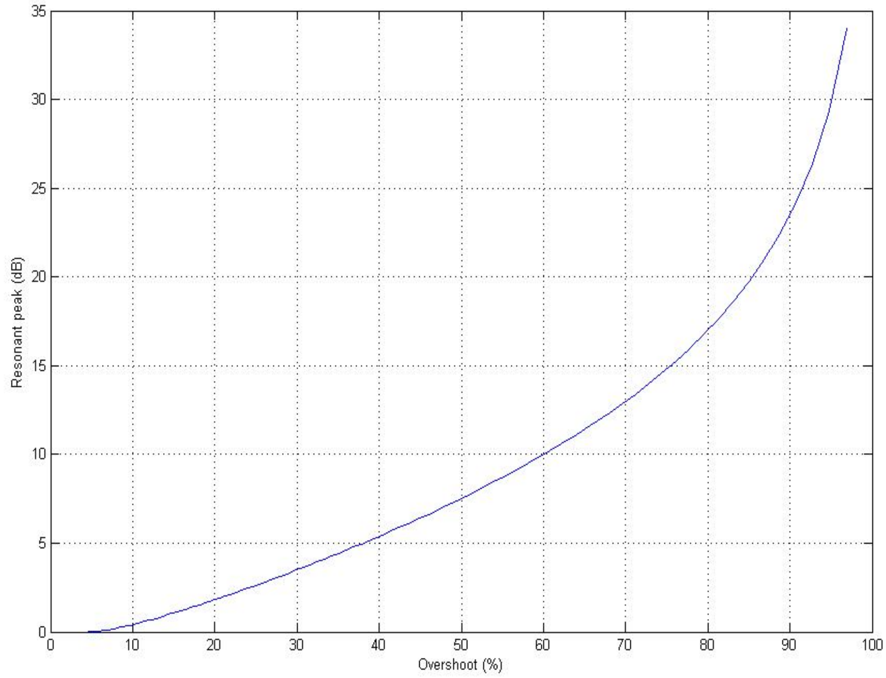


**Figure 3.12: The relationship between the percentage overshoot and the resonant peak of a second order model.**

### 3.4.1.3 Settling time

As discussed in section 3.1.1.3, the settling time defines the maximum amount of time, $t_s$, the control system may take before it falls between $\pm x_s\%$ of the step response's final value. This is shown in Figure 3.9. As in the case of the rise time, due to the complex nature of the systems response to a step, this time must be solved for numerically.

Figure 3.13 shows the normalised settling time versus the damping ratio of a second response for a $x_s$ of 1%, 2% 5% and 10%. The designer can use a figure such as this to estimate the minimum natural frequency (closed loop bandwidth) required to meet the settling time specification given the damping ratio. The damping ratio is estimated from the overshoot specification, which was discussed in the section above.

If no overshoot specification is given, the figure can be used to estimate the trade off between the closed loop bandwidth of the system and its resonant peak, which is estimated from the damping ratio as discussed in section 3.4.1.2.

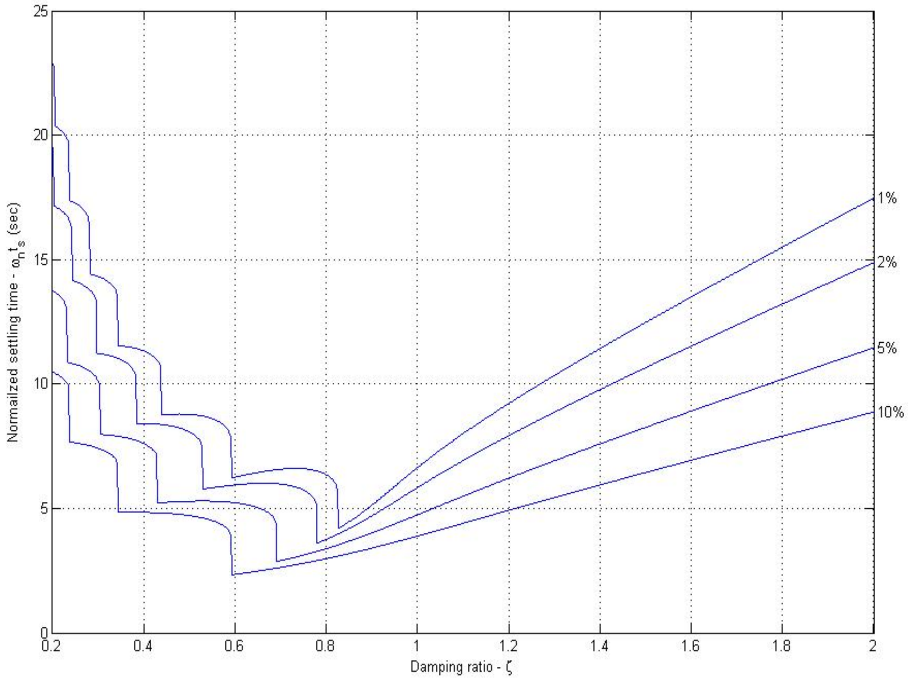


**Figure 3.13: The normalised settling time versus the damping ratio of a second order step response.**

### 3.4.1.4 Steady state error

As discussed in section 3.1.4, the steady state error specification defines the maximum error that the control system may have when tracking a given command input as time tends to infinity.

In the frequency domain, the error in the closed loop output is given by

$$E(s)=\frac{1}{1+L(s)}\cdot C(s)$$

Using the final value theorem to calculate the error as time tends to infinity gives

$$e(\infty)=\lim_{s\to 0} s\cdot E(s)=\lim_{s\to 0}\frac{s}{1+L(s)}\cdot C(s)$$

The above equation can be used to calculate the steady state error given any command input. However, it is common practice to specify the steady state error to a power of time input of the following form

$$c(t)=A\cdot t^i$$

which has the following Laplace transform

$$C(s)=\frac{A}{s^{i+1}}$$

Then the steady state error becomes

$$e(\infty)=\lim_{s\to 0}\frac{1}{1+L(s)}\cdot\frac{A}{s^i}=\begin{cases}\dfrac{A}{1+K_0} & if \quad i=0\\ \dfrac{A}{K_i} & if \quad i>0\end{cases}$$

where the error coefficient is given by

$$K_i=\lim_{s\to 0} s^i L(s)$$

The steady state error due to the power of time inputs can be summarised in Table 1.

| ***Type of L(s)*** | ***Steady state error to a step input*** $c(t)=A\cdot u(t)$ | ***Steady state error to a ramp input*** $c(t)=A\cdot t$ | ***Steady state error to a parabolic input*** $c(t)=A\cdot t^2$ |
|---|---|---|---|
| 0 | $\frac{A}{1+K_0}$ | $\infty$ | $\infty$ |
| 1 | 0 | $\frac{A}{K_1}$ | $\infty$ |
| 2 | 0 | 0 | $\frac{A}{K_2}$ |
| 3 | 0 | 0 | 0 |

**Table 1: The steady state error due to power of time command inputs.**

#### 3.4.1.5 Tracking the command input

As these specifications are already in the frequency domain, see section 3.1.3, they do not need to be translated to the frequency domain.

### 3.4.2 Response of the control system to disturbances

The response of the closed loop system to a disturbance is given by;

$$T_D(s)=P_D(s)\cdot S(s)$$

For a disturbance entering at the output (.i.e. $P_D(s)=1$ ) the second order model used to estimate the loop transmission characteristics required to meet the plant input can be modified as follows:

$$T_D(s)=S(s)=1-T(s)=1-\frac{\omega_n^2}{s^2+2\cdot\zeta\cdot\omega_n\cdot s+\omega_n^2}=\frac{s\cdot(s+2\cdot\zeta\cdot\omega_n)}{s^2+2\cdot\zeta\cdot\omega_n\cdot s+\omega_n^2}$$

which has the following response to a step input with amplitude *A* in the time domain:

$$y(t)=\begin{cases} A\cdot e^{-\zeta\cdot\omega_n\cdot t}\cdot\left(\cos(\omega_0\cdot t)+\dfrac{\zeta\cdot\omega_n}{\omega_0}\cdot\sin(\omega_0\cdot t)\right) & for \quad \zeta<1 \\ A\cdot e^{-\zeta\cdot\omega_n\cdot t}\cdot\left(1+\omega_n\cdot t\right) & for \quad \zeta=1 \\ A\cdot e^{-\zeta\cdot\omega_n\cdot t}\cdot\left(\cosh(\omega_x\cdot t)+\dfrac{\zeta\cdot\omega_n}{\omega_x}\cdot\sinh(\omega_x\cdot t)\right) & for \quad \zeta>1 \end{cases}$$

When the disturbance does not enter at the output, construction of a consistent prototype model is more complicated. The transfer function describing the response to the disturbance input, $P_D(s)$, usually acts as a low pass filter on the step input which depends on its dynamics. Therefore, the specifications on these type of disturbances are dependent on the dynamics of the plants response to a disturbance and will not be discussed further. The specifications discussed can be used as an initial estimate all be it an over-design.

### 3.4.2.1 Fall time

For disturbances entering at the plant output (.i.e. $P_D(s)=1$ ) the response to the disturbance has the same dynamics as the response to the command input. The only difference in the response is an offset and the sign of the response. Therefore the fall time for a step disturbance is equal to the rise time of a step command input, therefore Figure 3.11 can be used to estimate the bandwidth required of the closed loop system given the damping ratio as required by the resonant peak specification or trade-off the bandwidth versus the resonant peak of the closed loop response.

### 3.4.2.2 Undershoot

As in the case of the fall time specification the undershoot specification can be read off the graph used to estimate the parameters of the second order model used by the overshoot specification for the command input as shown in Figure 3.12.

### 3.4.2.3 Settling time

As in the case of the fall time and undershoot specifications the settling time specification can be read off the graph used to estimate the parameters of the second order model used by the settling specification for the command input as shown in Figure 3.13.

#### 3.4.2.4 Steady state response

The closed loop systems response to a disturbance entering the plant is given by

$$Y_D(s)=\frac{P_D(s)}{1+L(s)}\cdot D(S)$$

Using the final value theorem the steady state output due to a disturbance is given by

$$y_D(\infty)=\lim_{s\to 0} s\cdot Y_D(s)=\lim_{s\to 0}\left[s\cdot\frac{P_D(s)}{1+L(s)}\cdot D(s)\right]$$

Let $D(s)$ be a power of time input of the form

$$d(t)=A\cdot t^i$$

which has the following Laplace transforms

$$D(s)=\frac{A}{s^{i+1}}$$

Then the steady state output due to a disturbance becomes

$$y_D(\infty)=\lim_{s\to 0}\left[A\cdot\frac{P_D(s)}{s^i+s^i\cdot L(s)}\right]$$

Using the error coefficients defined for the steady state error due to plant input

$$K_i=\lim_{s\to 0} s^i\cdot L(s)$$

Then

$$y(\infty)=\lim_{s\to 0}\left[A\cdot\frac{P_D(s)}{1+K_i}\right]=\begin{cases}\dfrac{A}{1+K_i}\cdot\left(\lim_{s\to 0}P_D(s)\right) & if \quad i=0\\ \dfrac{A}{K_i}\cdot\left(\lim_{s\to 0}P_D(s)\right) & if \quad i>0\end{cases}$$

Assuming that $P_D(0)=\lim_{s\to 0}P_D(s)$ exists and is finite, the steady state response due to a disturbance input can be simplified to

$$y(\infty)=\begin{cases}\dfrac{A}{1+K_i}\cdot P_D(0) & if \quad i=0\\ \dfrac{A}{K_i}\cdot P_D(0) & if \quad i>0\end{cases}$$

Which can be summarised in Table 2.

| ***Type of L(s)*** | ***Steady state error to a step disturbance*** $d(t)=A\cdot u(t)$ | ***Steady state error to a ramp disturbance*** $d(t)=A\cdot t$ | ***Steady state error to a parabolic disturbance*** $d(t)=A\cdot t^2$ |
|---|---|---|---|
| 0 | $\frac{A}{1+K_0}\cdot P_D(0)$ | $\infty$ | $\infty$ |
| 1 | 0 | $\frac{A}{K_1}\cdot P_D(0)$ | $\infty$ |
| 2 | 0 | 0 | $\frac{A}{K_2}\cdot P_D(0)$ |
| 3 | 0 | 0 | 0 |

**Table 2: The steady state error due to power of time disturbance inputs.**

### 3.4.3 Response of the control system to sensor noise

The sensor noise at the output of the control system is given by:

$$Y_N(j\omega)=N(j\omega)|T(j\omega)|^2$$

where:

$N(j\omega)$ is the power spectrum of the sensor noise.
$Y_N(j\omega)$ is the power spectrum of the noise of the control system output.
$T(j\omega)$ is the frequency response of the control system to the command signal.

The specification on the allowable sensor noise level at the control system output is a maximum bound on the RMS value of the noise signal (See section 3.3), which is given by:

$$N_{RMS}=\sqrt{\int_{-\infty}^{+\infty} N(j\omega)|T(j\omega)|^2 d\omega}$$

Therefore, the frequency domain specification is an upper bound on the area under the magnitude of the complementary sensitivity function, $|T(j\omega)|$ .

# 4 Response prototype models

In this chapter, the selection of prototype models will be focused on, which are used as specifications for the control algorithm design as discussed in section 2.1.3.

The other parts of the design process have a large body of theory backing them, but the process of deriving the prototype models (and optimal control's performance weighting functions) seem to rely on engineering experience and trial and error.

In the case of aircraft, research has been done into the desirable dynamic characteristics of an aircraft from a pilots perspective (Stevens et al,2003). These dynamic characteristics can be used to define the acceptable prototype models used in the design of an autopilot. The aircraft should have been designed to have the performance which will meet these requirements on its dynamics. Therefore, the aircraft should be capable of meeting the prototype models derived using these dynamic characteristics. However in the case of an unmanned air vehicle (UAV) a pilot is not present. The dynamics of the aircraft need not meet the expectations of a pilot or be limited in a manner that will ensure the pilots safety (such as limiting the g forces that the pilot will experience). The desirable dynamics from the pilots perspective are no longer necessarily a good estimate of the desired aircraft performance. The desired performance in this case is highly likely to be limited only by the performance of the aircraft for which the autopilot is to be designed. The prototype models in this case should be selected in such a manner that they define the best performance of the aircraft while meeting any specifications that have been supplied (such as accuracy of airspeed or altitude control of the aircraft for example).

The common method of deriving prototype models, as discussed in the previous section, often makes use of a predefined structure, such as second order rational transfer function or a third order rational transfer function with a zero (Horowitz,1993). This structure does not directly take into account the dynamics of the plant which is desirable as in the case of the UAV being discussed. An experienced designer, in the field, can from previous experience select prototype models which give performance requirements which are achievable by the plant.

Different types of prototype models and different methods of selecting and generating them will now be discussed.

## 4.1 Types of response prototype models

The prototype models are dynamics relating to groups of specifications. For example the specifications on the systems output response to a step input command. Not all the specifications in the group are always specified. And some groups are not specified at all. For example the systems output due to sensor noise is not always specified but the designer is expected to minimise its effect.

Ideally the set of all the prototype models of any order that meet a group of specifications which have been supplied should be considered. This is not however practical, so the designer normally settles for a subset of the prototype models, often of a single order, which meet the specifications.

For a single output system there are normally two sets of signals of interest to the designer. The obvious signal is the system output itself. The other is the plant input, as the designer needs to ensure that the actuators are not saturated too much.

The equation giving the output of the closed loop system output given all the system inputs is given by:

$$Y(s)=H_{YC}(s)\cdot C(s)+H_{YN}(s)\cdot N(s)+H_{YDI}(S)\cdot D_1(s)+\ldots+H_{YDn}\cdot D_n(s)$$

which has the following plant response prototype model sets:

- The system output prototype model set for the command input ( $H_{YC}(s)$ ), which gives the required control systems response to the control input.
- The system output prototype model sets for the disturbance inputs ( $H_{YDI}(s),\ldots H_{YDn}(s)$ ), which gives the required control systems responses for each of the disturbance inputs.
- The system output prototype model set for the sensor noise ( $H_{YN}(s)$ ), which gives the required control systems response to the sensor noise.

The equation giving the plant inputs response to the system inputs is given by:

$$U(s)=H_{UC}(s)\cdot C(s)+H_{UN}(s)\cdot N(s)+H_{UDI}(s)\cdot D_1(s)+\ldots+H_{Dn}(s)\cdot D_n(s)$$

The common plant response prototype model sets are:

- The plant input prototype model set for the command input ( $H_{UC}(s)$ ), which gives the acceptable plant input signal to the command input.

- The plant input prototype model set for the sensor noise( $H_{UN}(s)$ ), which gives the acceptable plant input signal to sensor noise.
- The plant input prototype model sets for the disturbance inputs ( $H_{UD1}(s),\ldots H_{UDn}(s)$ ) , which gives the acceptable plant input signal to the disturbance inputs.

Each prototype model set will now be discussed.

### 4.1.1 Command input to the controlled variable prototype model set

The response prototype model set is the set of desired dynamics of the control system in response to the command input. The set of desired control system dynamics required (in the frequency domain) can be calculated from:

$$H_{YC}(s)=\frac{Y(s)}{C(s)}=F(s)\cdot T(s)$$

This set of prototype models should meet the following specifications that are specified for a step input

1. overshoot
2. rise time
3. settling time

The tracking specifications and the steady state error specifications to the command input must also be met.

For a one degree of freedom system all the specifications specified in the set above must be met by the set of complementary sensitivity functions, $T(s)$ . For a two degree of freedom system some of the specifications may be slackened for $T(s)$ as they can be modified to meet the specifications by the pre-filter, $F(s)$ . The steady state error specification must be met by $T(s)$ .

### 4.1.2 Disturbance input to the controlled variable prototype model set

This set of response prototype models is the set of desired dynamics of the control system in response to the disturbance inputs. The set of desired control system dynamics required (in the frequency domain) can be calculated from:

$$H_{YDn}(s)=\frac{Y(s)}{D_n(s)}=P_{Dn}(s)\cdot S(s)$$

If the disturbance is specified as a step input the following specifications must be met:

1. undershoot
2. fall time
3. settling time

The steady state response to a disturbance specification must also be met by this set of prototype models.

### 4.1.3 The sensor noise to the controlled variable prototype model set

The sensor noise specifications must be met by the set of system output prototype models for the sensor noise which is given by the following set of transfer functions:

$$H_{YN}(s)=\frac{Y(s)}{N(s)}=-T(s)$$

As discussed in section 3.4.3 the sensor noise is a random process so the designer is often concerned with reducing the sensor noise power spectrum at the systems output. In this case the designer would be more concerned with reducing the area under the magnitude of the set of complementary sensitivity functions. Particularly at high frequencies where the major senor noise frequency components are expected.

### 4.1.4 The set of plant input prototype models for the command input

The set of dynamics of the plant input prototype models for the command input in the frequency domain are given by:

$$H_{UC}(s)=\frac{U(s)}{C(s)}=S(s)\cdot G(s)\cdot F(s)$$

The designer in this case will normally have no specifications, but would want to limit the time domain amplitude so that the actuator is not saturated too much, as this will have an adverse effect on the control systems relative stability.

### 4.1.5 The set of plant input prototype models for the disturbance inputs

The set of dynamics of the plant input prototype models for the disturbance inputs in the frequency domain are given by:

$$H_{UDn}(s)=\frac{U(s)}{D_n(s)}=-S(s)\cdot G(s)\cdot P_{Dn}(s)$$

The designer in this case will also normally have no specifications, but would want to limit the time domain amplitude so that the actuator is not saturated.

### 4.1.6 The set of plant input prototype models for the sensor noise

The set of dynamics of the plant input prototype models for the sensor noise in the frequency domain is given by:

$$H_{UN}(s)=\frac{U(s)}{N(s)}=-S(s)\cdot G(s)$$

The designer as in the case of the set of system output prototype models for the sensor noise would want to reduce the power spectrum of the sensor noise at the plant input. This is highly desirable, as the sensor noise may swamp the plant input, in which case the desired plant input will not be able to achieve the specifications on the systems response to the command and disturbance inputs.

## *4.2 Determining the set of prototype models that meet all the specifications*

At first, selecting suitable set of response prototype models may not seem a difficult task. In fact there is an infinite set of responses that meet each set of specifications. However the loop compensator and pre-filter (if a two degree of freedom system is being designed) must generate a control system which lies within the sets of prototype models.

### 4.2.1 Independent prototype models

The simplest method of determining the prototype models is to consider each of the set of prototype models separately and try to find the largest set of possible models which meet the specifications on the particular prototype model under consideration.

These sets of prototype models are then used by control design processes to generate the controller algorithm. In quantitative feedback theory these sets of prototype models are used to generate bounds on the Nichols chart. A loop transmission which meets these bounds is then formulated. In optimal control the sets of prototype models are used to generate the performance weighting function. In H-infinity theory, the performance weighting function places an upper bound on the sensitivity function. So the prototype model set which gives the lowest upper bound on the sensitivity function will need to be used to ensure that all the specifications are met.

The plant model has not been considered when generating the prototype model sets. The designer hopes that the plant performance was considered when generating the specifications which make up the operational requirements.

There is no guarantee that the sets of prototype models place requirements on the plant which it could practically meet. Even if these requirements can be met, it is highly likely that they would cause a conservative compensator to be designed. This is due to the fact that no effort was made to ensure that the requirements placed on the compensator by the prototype model sets are consistent. To overcome this problem two methods of generating prototype models are proposed.

### 4.2.2 Using one of the sets of prototype models to define all the prototype model sets

A better option is to use one of the prototype model sets to derive the other prototype model sets. This will ensure that all the prototype model sets are consistent, and therefore there will little or no over-design of the compensator due to inconsistent prototype model sets. The challenge however is to be able to adjust the selected prototype model set so that all the specification sets are met for all the prototype model sets.
But which set of prototype model sets should be used? A good candidate is the command input prototype model set as it often has the most complete set of specifications. This idea is shown in Figure 4.1.

If the systems' response to a reference command input, $C(s)$, is selected to generate all the prototype models in the command input to the controlled variable prototype model set, $H_{YC}(s)$, the following procedure will need to be followed.

A predefined structured for the response prototype models will be selected and the parameters for each model will be selected in order to meet the performance specification for the reference input. If a number of reference command inputs are specified or expected, the parameters of the response prototype models will have to be selected such that the response prototype models will have all its responses to the reference command inputs meet the relevant performance specifications.

The set of acceptable complementary sensitivity functions is then determined from the set of response prototype models for the command input by simply finding its impulse response that gives each prototype model. In the frequency domain the complementary sensitivity function is simply given by

$$T(s)=\frac{Y(s)}{C(s)}$$

where:

$Y(s)$ is one of the response prototype models for the systems' response to the command input.

$C(s)$ is the command input which generates this response.

This formula will be used on all the system response prototype models for a command input to generate the set of acceptable complementary sensitivity functions. Note that the pre-filter has not been used in the calculation. The system being discussed first is a one degree of freedom system, the set of complementary sensitivity functions will therefore be correct as they are.

The measurement noise specifications may now be checked by generating the systems response to sensor noise using the complementary sensitivity function set. The power spectrum of the measurement noise is given by,

$$R_N(j\omega)=N(j\omega)|T(j\omega)|^2$$

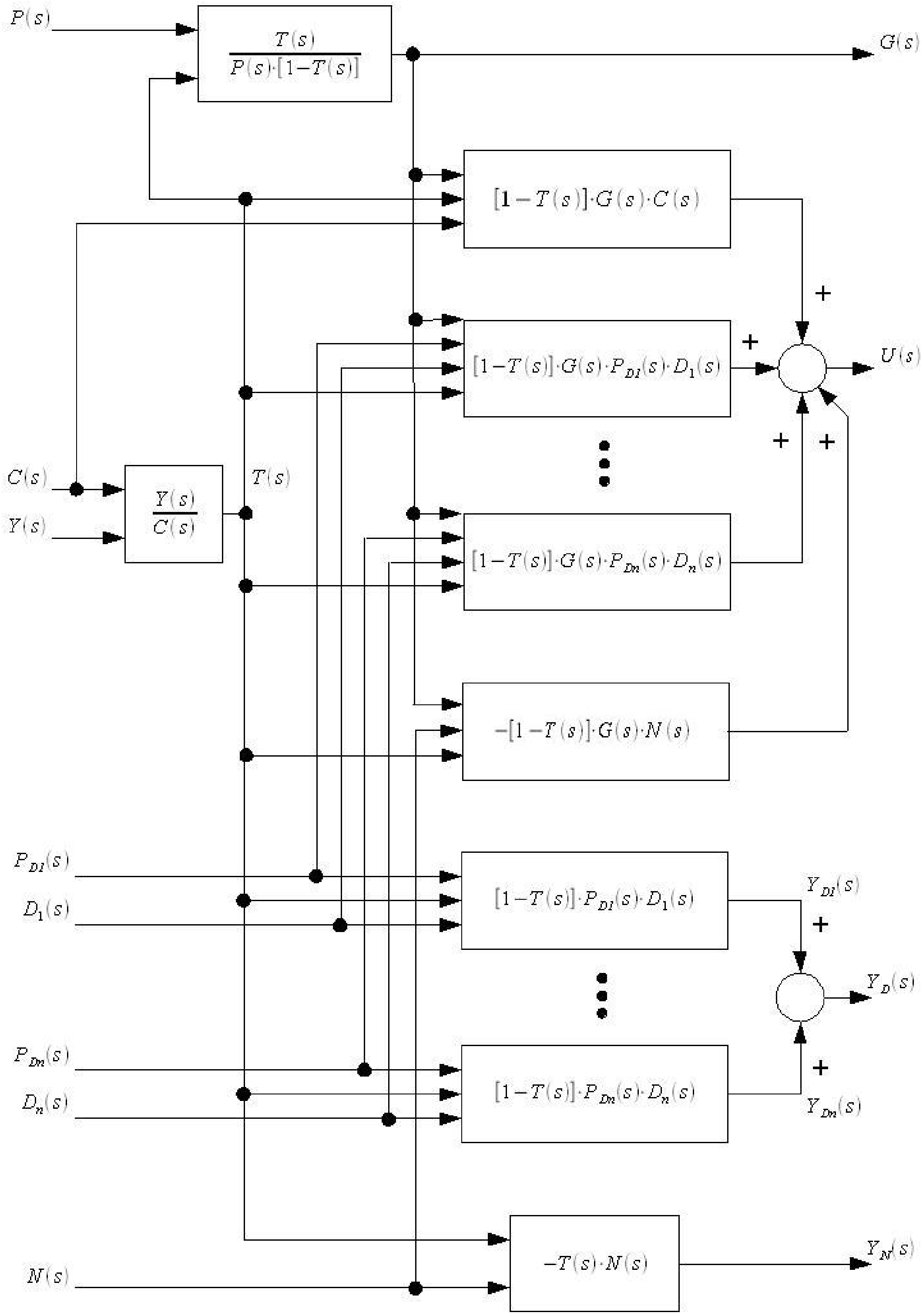


**Figure 4.1: Using the set of response prototype models for a command input to derive the other sets of prototype models**

where:

$N(j\omega)$ is the power spectrum of the sensor noise.
$R_N(j\omega)$ is the power spectrum of the noise of the control system output.
$T(j\omega)$ is the set of frequency responses of the control system to the command signal.

and the set RMS values of the noise is given by

$$N_{RMS}=\sqrt{\int_{-\infty}^{+\infty} N(j\omega)|T(j\omega)|^2 d\omega}$$

The set of sensitivity functions related to the complementary sensitivity function set may now be calculated as follows:

$$S(s)=1-T(s)$$

The set of systems' response to each set of disturbances entering the plant can now be checked against the specifications by calculating the response as follows:

$$Y_{Dn}=S(s)\cdot P_{Dn}(s)\cdot D_n(s)$$

The calculation of the signal at the plant input requires the compensator transfer function. A set of compensator transfer functions related to the plant set and complementary sensitivity function set is given by

$$G(s)=\frac{T(s)}{P(s)\cdot[1-T(s)]}$$

This step can cause a problem for plants which contain non-minimum phase zeros and/or unstable poles. If the plant contains non-minimum phase zeros, the formula above will give them as unstable poles in the compensator. In this case the problem is avoided by requiring that the non-minimum phase zeros be part of the response prototype model set. This is a practical solution, as the closed loop transfer function of a control system always contains the non-minimum phase zeros contained within the plant. In the case where the plant contains unstable poles, the compensator will end up with non-minimum phase zeros which cancel out the unstable poles. This however is highly impractical, as plant uncertainty will ensure that the unstable poles do not cancel and the closed loop response of the actual system will be unstable. So this method will not be very effective for unstable plants.

If it is possible to calculate a set of compensators, the set of system responses at the plant input due to the command input is given by

$$U_C(s)=S(s)\cdot G(s)\cdot C(s)$$

And the sets of system responses for the disturbances can be calculated similarly as follows

$$U_{Dn}(s)=S(s)\cdot G(s)\cdot P_{Dn}(s)\cdot D_n(s)$$

Now that systems response set to all the inputs and hopefully the plant inputs response set to all the inputs has been calculated, they can be evaluated against their relevant specifications. The complementary sensitivity function set must then be adjusted by the designer until all the specifications are met.

If a two degree of freedom system is being designed the complementary sensitivity function will not need to meet all the specifications on the systems' response to the command input. Specifications such as the steady state error will need to be met, but other specifications such as overshoot, rise time or settling time can be modified by the pre-filter.

For a two degree of freedom system, the complementary sensitivity function is a means of generating the other prototype models so that they can be evaluated in terms of their specifications. The complementary sensitivity function can then be modified so that it will allow the other prototype models to meet their specifications, and the pre-filter can be used to ensure that the modifications to the complementary sensitivity function do not cause it not to meet the specifications on the systems response to the command input.

This system is very complex and still does not take the plants' characteristics into account directly. The plant response is only used to generate the set of possible loop compensators which will give the prototype models specified. If the plant is unstable this cannot be done directly.

Another set of prototype models can be used but the same problems will be encountered. This procedure therefore is not very practical.

### 4.2.3 Using the loop transmission to define all the prototype models

Another option is to use the loop transmission to generate the prototype models. The loop transmission is used extensively in classical control theory frequency domain methods. From the loop transmission set the systems responses can be calculated as follows (See Figure 4.2):

The set of system responses to the command input is given by

$$Y_C(s)=F(s)\cdot\frac{L(s)}{1+L(s)}\cdot C(s)$$

The set of system responses to the set of disturbance is given by

$$Y_{Dn}(s)=\frac{P_{Dn}(s)}{1+L(s)}\cdot D_n(s)$$

The systems response to sensor noise is given by

$$R_N(s)=N\left|\frac{L(s)}{1+L(s)}\right|^2$$

and the set RMS values of the noise is given by

$$N_{RMS}=\sqrt{\int_{-\infty}^{+\infty} R_N(j\cdot\omega)\,d\omega}$$

As before, the calculation of the signal at the plant input requires the compensator transfer function. In this case the compensator transfer functions can be estimated by

$$G(s)=\frac{L(s)}{P(s)}$$

Now unlike the previous section, non-minimum phase zeros and unstable poles will not cause a problem if the designer makes them part of the loop transmission. The values of the nominal plants' non-minimum phase zeros and unstable poles should be added to the loop transmission. Then they will not form part of the compensator.

The set of system responses at the plant input due to the command input is given by

$$U_C(s)=F(s)\cdot\frac{G(s)}{1+L(s)}\cdot C(s)$$

And the sets of system responses for the disturbances can be calculated similarly as follows

$$U_{Dn}(s)=\frac{G(s)\cdot P_{Dn}(s)}{1+L(s)}\cdot D_n(s)$$

However, the above calculations do not take plant uncertainty into account. This is easily rectified by introducing a nominal plant and defining the other plants in the plant sets variation from the nominal plant. For a nominal plant, $P_0(s)$, the variation in plant performance is given by

$$\Delta P(s)=\frac{P(s)}{P_0(s)}$$

Then if the designer chooses a nominal loop transmission, the complete set of loop transmissions is given by

$$L(s)=L_0(s)\cdot\Delta P(s)$$

And the compensator transfer function can be estimated from

$$G(s)=\frac{L_0(s)}{P_0(s)}$$

The prototype models sets can be calculated as before, using the set of loop transmissions and the compensator calculated.

Once they have been calculated, the prototype model sets can be evaluated against the specifications and the necessary adjustments to the nominal loop transmission can be made.

Unlike the previous procedure where a set of prototype models were used to generate all the prototype models, this procedure does not run into problems with plants with non-minimum phase zeros and unstable poles. It also takes into account the plants characteristics early on, by working with plants uncertainty to generate the prototype model sets.

As an estimate of the compensator can be calculated, the signals at the plant input can be calculated. These signals may be used to determine if the prototype models being produced are asking for a practical performance of the plant.

This procedure will now be developed further, by first evaluating the requirements placed on the nominal loop transmission by the specifications (which is discussed in the next section) and then by finding a good candidate for the nominal loop transmission.

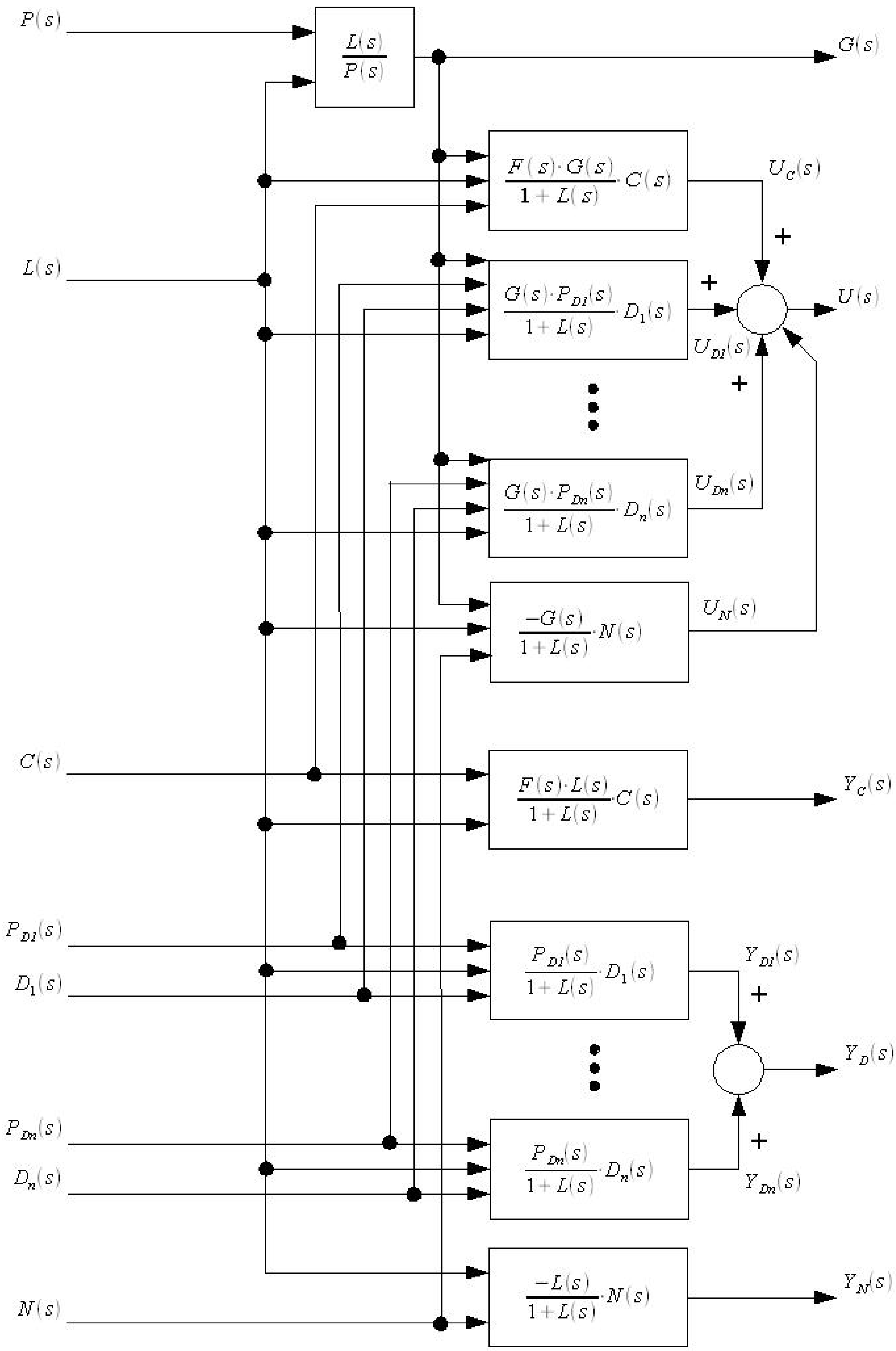


**Figure 4.2: Using the loop transmission to define all the sets of prototype models.**

# 5 Translating the frequency domain specifications to bounds on the loop transmission

This chapter discusses the calculation of bounds on the loop transmission that will ensure that the closed loop frequency domain specifications are met. These bounds are required to ensure that the specifications that have been supplied are met, and taken into account when the trade-offs between the specifications that have not been supplied are being considered.

As the plants considered in this report may have significant plant uncertainty, the bounds on the loop transmission must take the uncertainty of the plant into account. The method that shall be used to account for the uncertainty will make use of the plant templates as used in Quantitative Feedback Theory (Horowitz,1993)(Houpis et al,1999). A plant template is a region plotted on a Nichols chart that represents all the possible magnitudes and phases that the plant can have at a specific frequency. An example of a plant template is shown in Figure 1-5.1.

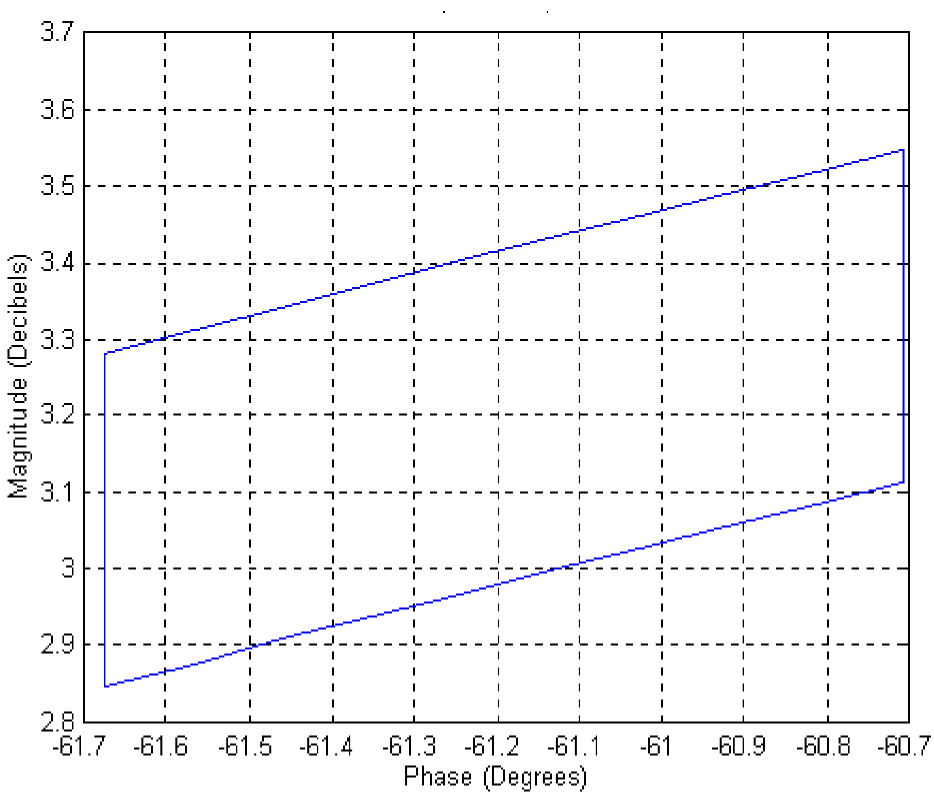


**Figure 1-5.1: Example of a plant template**

The plant template may defined as continuous dynamic plant model with parameters which can vary within bounds or a set of magnitude-phase points which have been determined experimentally.

A nominal plant is also selected for each of the plant templates. The nominal plant is usually a dynamic plant model rather than a set magnitude-phase points which are defined for each template, although the magnitude-phase points may be used. The nominal plant is used in place of the plant set by the design algorithm. This is possible as the plant templates are used to calculate the bounds and they therefore already take the plant uncertainty into account. The calculation of the bounds will now be discussed.

## *5.1 Response of the control system to the plant input*

This section will discuss the frequency domain specifications placed on the systems response to the command input. These specifications are:

1. Bandwidth
2. Resonant peak
3. Steady state error
4. Tracking specifications

### 5.1.1 Bandwidth

The specification for bandwidth is normally specified as the minimum bandwidth that the control system may have in its response to the command input. This will ensure that the system is fast enough to meet the performance requirements.

The closed loop bandwidth is defined as the frequency at which the magnitude of the closed loop system is -3 dB and the magnitude of the higher frequencies remain below this magnitude. The -3 dB m-circle then gives the minimum open loop magnitude versus open loop phase that the loop transmission may have at the desired frequency of the bandwidth in order to meet the specification. An m-circle is simply a plot of the open loop magnitude versus open loop phase for a constant closed loop magnitude. The -3 dB m-circle is shown in Figure 5.2.

As the plant is allowed to have significant plant uncertainty, which is defined by a set of plants, the bandwidth that is required is defined by the minimum bandwidth of all the plants in the plant set. It is tedious to have to calculate the bandwidth for each plant in the plant set in order to check if the bandwidth specification is met, at each step in the design algorithm. The -3 dB m-circle will therefore be modified in a manner which takes into account the plant uncertainty at the specified bandwidth frequency.

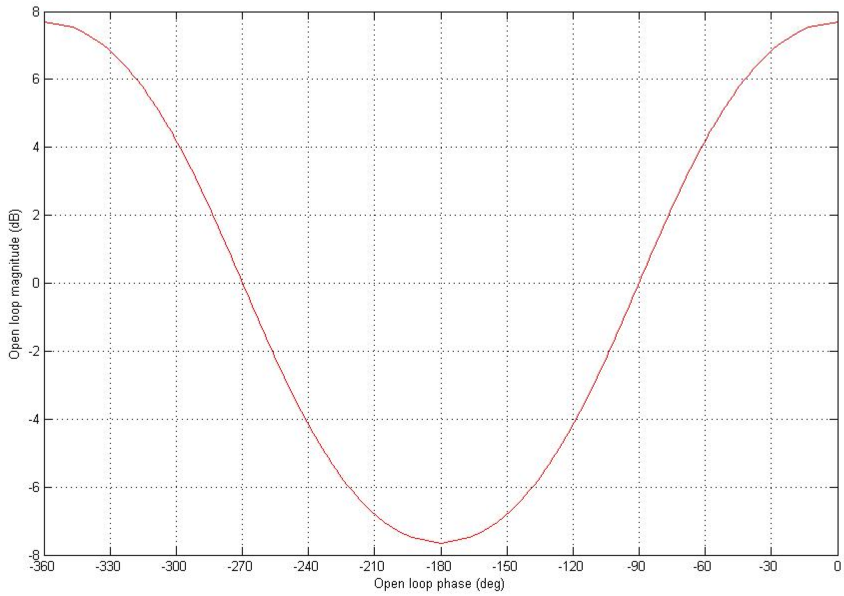


**Figure 5.2: The -3dB m-circle plotted on the Nichols chart.**

This is achieved using the plant template at the desired bandwidth frequency. As shown in Figure 5.3, the magnitude of the bound due to the bandwidth specification at a given open loop phase is the minimum magnitude that the nominal plant may have and still have the plant template lie above the -3 dB m-circle. The bandwidth bound is therefore the set of minimum magnitudes of the nominal plant versus open loop phase.

The bandwidth specification is therefore met if the nominal loop transmission's magnitude lies above the bandwidth bound at the bandwidth frequency specified.

### 5.1.2 Maximum resonant peak

The resonant peak specification defines the maximum magnitude that the closed loop transmission set may have i.e. all the magnitudes for all the plants in the plant set must be smaller than or equal to the resonant peak specification. The m-circles as plotted on the Nichols chart can be used to define a region of magnitudes and phases that the loop transmission set may have and still meet the resonant peak specification.

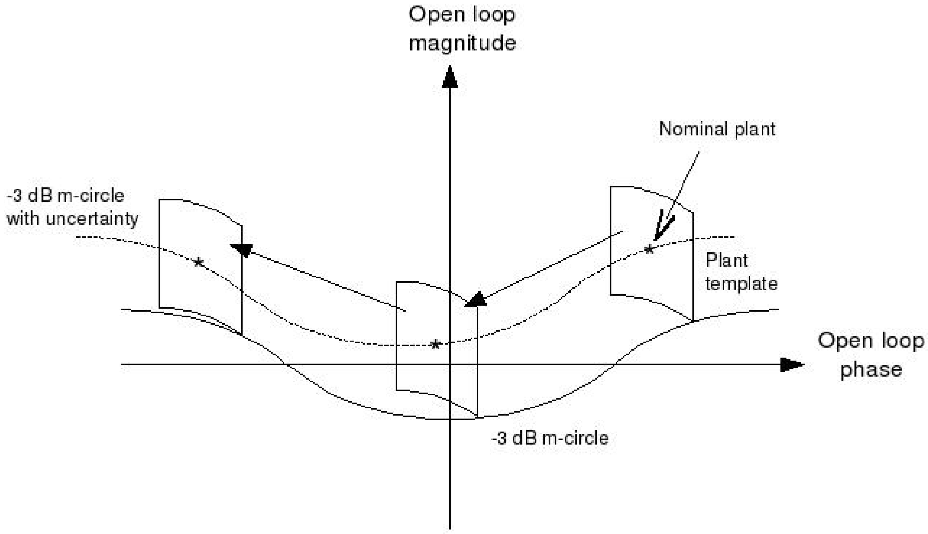


**Figure 5.3: The bandwidth bound for an uncertain plant.**

Figure 5.4 shows an example of m-circles plotted on the Nichols chart. As the resonant peak specification will always have a magnitude greater than or equal to 0 dB, only the m-circles which relate to a closed loop magnitude greater than 0 dB may need to be considered. From Figure 5.4 it can be seen that all the m-circles greater than 0 dB, are closed oval regions in shape. It can also be seen that if a specific m-circle is considered, the 1 dB m-circle for example, all the m-circles with a greater closed loop magnitude lie within the closed oval shape that defines the m-circle under consideration. Therefore, if the loop transmission set lies outside or on the m-circle contour, the closed loop magnitudes for all the loop transmissions will have a lower or equal magnitude than the closed loop magnitude defining the m-circle.

Therefore, in order for the resonant peak specification to be met, the loop transmission set must lie outside the m-circle with the same closed loop magnitude as the resonant peak specification. This m-circle can be modified so that the plant uncertainty can be taken into account, as in the bandwidth specification, so that the nominal plant may only need to be considered.

As the plant templates vary with frequency, the modified m-circle bound will vary with frequency as well. It is common practice to sample the plant frequencies, as is done with the plant templates. The modified m-circles can then be generated at these frequencies. These bounds obtained are known as stability bounds in Quantitative Feedback Theory (Horowitz,1993)(Houpis et al,1999).

The stability bound is calculated by moving the plant template so that it touches but does not cross into the m-circle. See Figure 5.5. The plant template is then moved around the m-circle, still touching it but not crossing into it. The path that the nominal plant traces on the Nichols chart gives the stability bound contour at the frequency of the plant template.

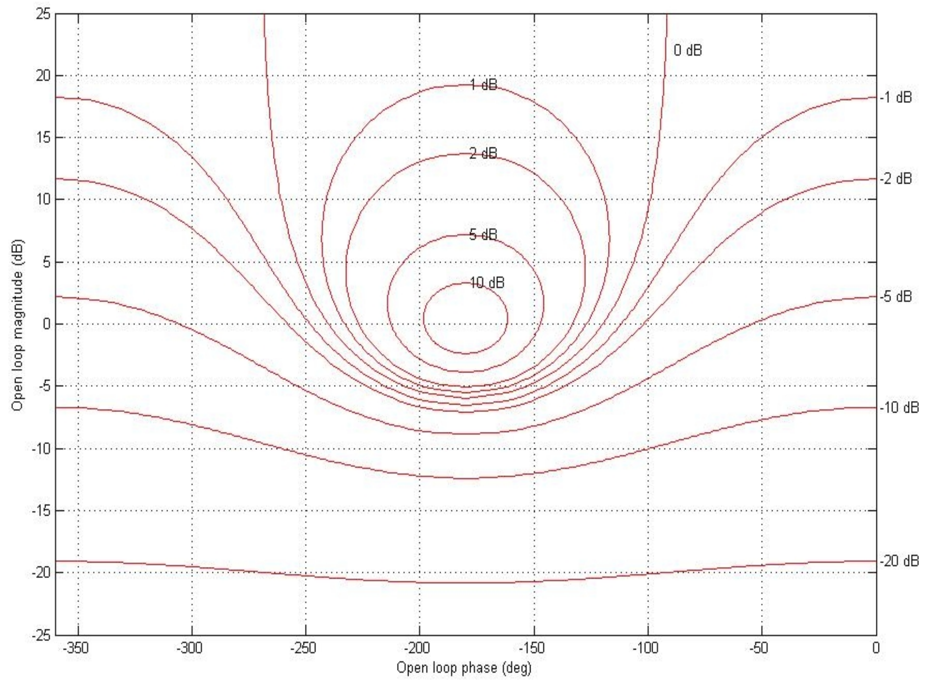


**Figure 5.4: M-circles plotted on the Nichols chart.**

As in the case of the m-circle defining the region into which the loop transmission set may not cross in order to meet the resonant peak specification, the stability bound defines the region into which the nominal loop transmission may not cross at the frequency for which the bound is defined.

### 5.1.3 Steady state error

The steady state error, as discussed in section 3.4.1.4, places bounds on the number of pure integrators contained in the loop transmission as well as the minimum DC gain that the loop transmission must have with these integrators removed.

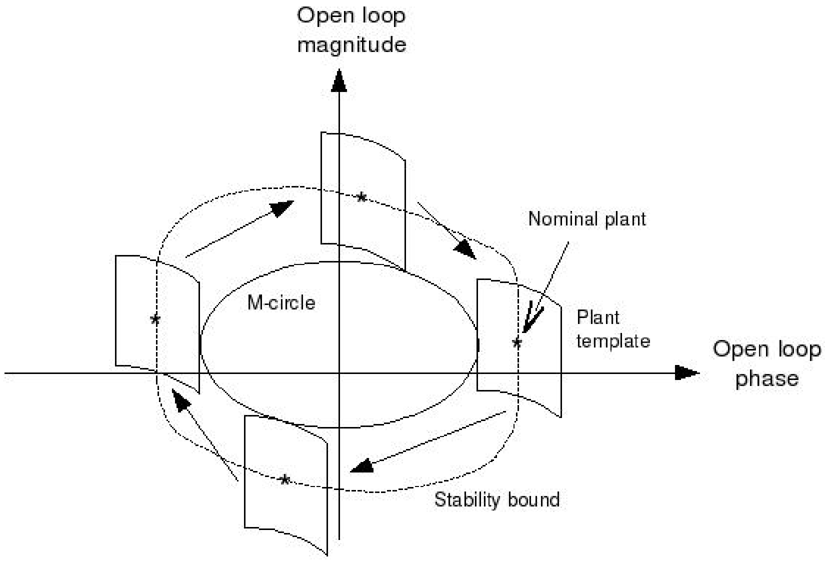


**Figure 5.5: Calculating the stability bound.**

This minimum DC gain must be met for all plants in the plant set. The DC plant template usually only has uncertainty in gain and not in phase (Horowitz,1993). So if the DC template is modelled by the following equation

$$|P_{MIN}(0)|=|P_0(0)|\cdot|\Delta P(0)|$$

where

| | |
|---|---|
| $\lvert P_{MIN}(0)\rvert$ | the minimum magnitude of the DC plant template. |
| $\lvert P_0(0)\rvert$ | the magnitude of the nominal plant at DC. |
| $\lvert\Delta P(0)\rvert$ | the ratio relating the minimum magnitude of the plant template to the nominal magnitude at DC. |

Then the factor by which the DC gain required by the steady state specification, see section 3.4.1.4, must be increased so that the steady state specification is met for all plants in the plant set is given by

$$|\Delta P(0)|=\frac{|P_{MIN}(0)|}{|P_0(0)|}$$

### 5.1.4 Tracking the command input

The calculation of bounds on the loop transmission that will ensure that the tracking bounds are met (See section 3.1.3) depend on whether the control system designed is a one degree or two degree of freedom system. The calculations for the one degree of freedom system will be discussed before the two degree of freedom system.

#### 5.1.4.1 One degree of freedom systems

In a one degree of freedom system, as there is no pre-filter, the loop transmission must ensure that the upper and lower bounds placed on the closed loop magnitude for the passband (See section 3.1.3) are met.

In the simplest case, these upper and lower bounds are constant magnitudes. Then the bounds on the loop transmission set are simply the m-circles at these magnitudes. Figure 5.6 shows an example of the bounds on the loop transmission set where the lower and upper bounds on the closed loop magnitude are -*E* dB and +*E* dB respectively. For the tracking bounds to be met the loop transmission set must lie in between these bounds for the frequencies in the passband.

If however, the lower and/or upper bounds have magnitudes which vary with frequency, the loop transmission bounds at each frequency will relate to different m-circles. So in a similar manner to the stability bound, in order to practically display these bounds, a sample of frequencies would need to be used over the frequency range of the passband.

As for the other loop transmission bounds, the m-circles which bound the whole loop transmission set can be converted to bounds on the nominal loop transmission. In this case, these conversions have already been discussed in previous sections. The upper bound for a plant with significant plant uncertainty is equivalent to the stability bound, except that the closed loop magnitude is set to the upper bounds magnitude rather than the resonant peak specification. See section 5.1.2. The lower bound is similar to the bandwidth bound except that the m-circle used has the magnitude of the lower bound rather than -3 dB. See section 5.1.1.

An example of these bounds at a frequency, ω, are shown in Figure 5.7. The nominal loop transmission must not lie within the upper bound, $E(\omega)$, and must be greater than the lower bound, $-E(\omega)$, for the frequency under consideration, ω. This applies for all the frequency samples of the passband frequency range.

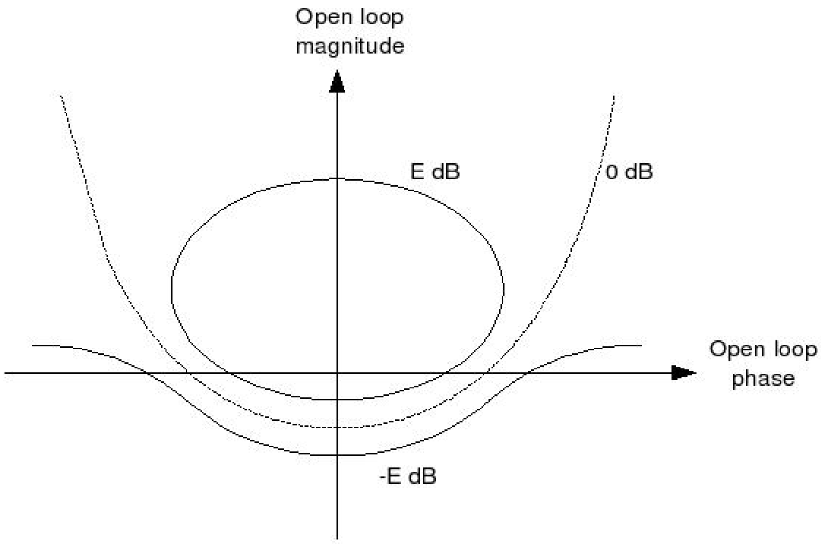


**Figure 5.6: Bounds on the loop transmission set due to the closed loop tracking bounds for a single degree of freedom system.**

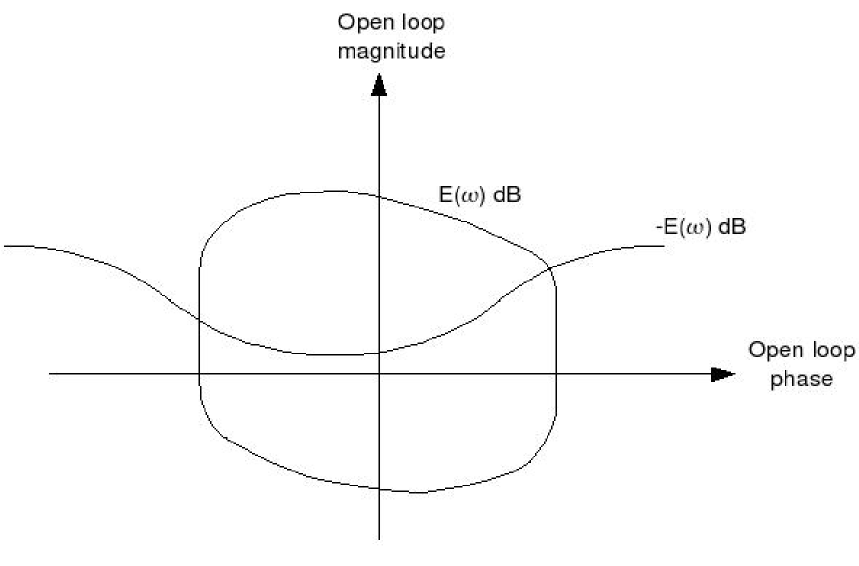


**Figure 5.7: Tracking bounds for an uncertain plant of a one degree of freedom system.**

### 5.1.4.2 Two degree of freedom systems

In a two degree of freedom system, as there is a pre-filter, the loop transmission must ensure that the difference in magnitude between the upper and lower bounds placed on the closed loop magnitude for the passband (See section 3.1.3) is met. The actual values of the magnitude are not important as the pre-filter can reshape the closed loop response so that the upper and lower bounds are met. The pre-filter however cannot change the uncertainty of the closed loop system, this must be achieved by the loop transmission.

The closed loop uncertainty bounds on the nominal loop transmission are given by the QFT uncertainty bounds. The uncertainty bound is found at each open loop phase for a given frequency, by finding the minimum magnitude that the nominal loop transmission may have at the open loop phase.

Figure 5.8 shows an example of a plant template plotted on a Nichols chart with a selection of m-circles. The m-circles all differ in magnitude from one to the other by 1 dB. Note that the plant template is plotted in two different positions but at the same phase for the nominal plant. If the maximum uncertainty allowed at the frequency of the plant template is 1 dB, then it can be seen that position A of the plant template would not meet the specification as it cuts through a couple of the m-circles. Therefore, at this position the uncertainty in closed loop magnitude will be greater than 1 dB. However position B has the plant template just touch two m-circles. The maximum uncertainty in closed loop magnitude will then be 1 dB which just meets the uncertainty specification, therefore the magnitude and phase of the nominal plant would lie on the uncertainty specification and give the minimum magnitude of the nominal loop transmission. This procedure would need to be performed over all the phases of interest to give the complete uncertainty bound at the frequency of the plant template.

The uncertainty bounds at other frequencies would need to be calculated over the passband frequency range in order to bound the loop transmission over this range. Figure 5.9 shows examples of uncertainty bounds at a number of different frequencies.

## *5.2 Response of the control system to disturbances*

This section will discuss the frequency domain specifications placed on the systems response to a disturbance input. These specifications are:

1. Bandwidth
2. Resonant peak
3. Steady state error

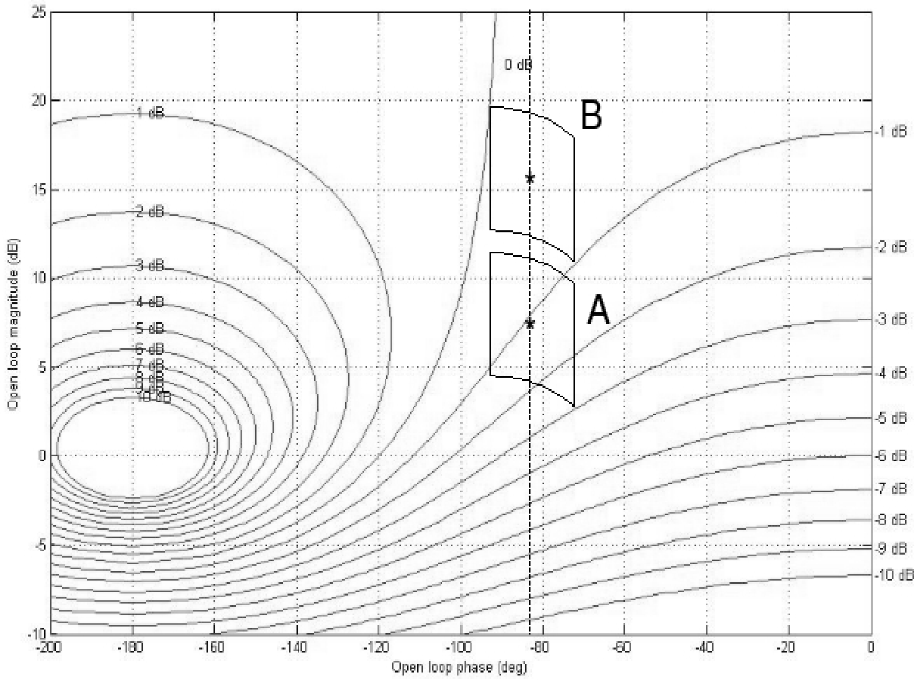


**Figure 5.8: Calculating an uncertainty bound.**

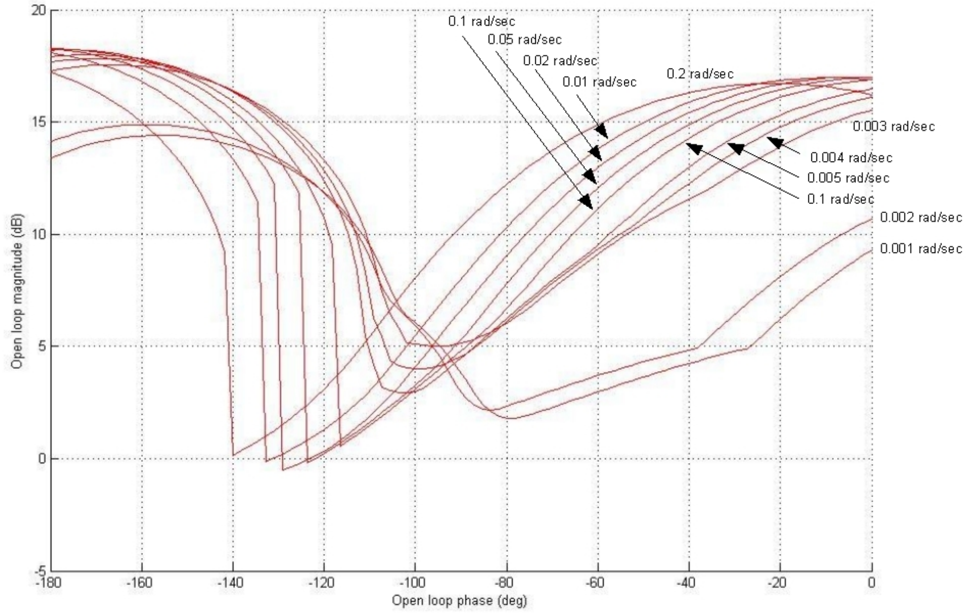


**Figure 5.9: Examples of uncertainty bounds on the nominal loop transmission.**

As discussed in section 2.3 the response of the closed loop system to a disturbance is given by;

$$T_D(s)=P_D(s)\cdot S(s)$$

Where $S(s)$ is the sensitivity function, which has the following relationship to the loop transmission

$$S(s)=\frac{1}{1+L(s)}$$

As the plants response to a disturbance, $P_D(s)$, is known (as it is part of the plant model) and features of the closed loop systems' response to a disturbance (such as the maximum resonant peak, bandwidth etc.) can be calculated from specifications related to the response to disturbances, the desired sensitivity function can be expressed as follows

$$S(s)=\frac{T_D(s)}{P_D(s)}$$

which can be used to calculate the desired features of the sensitivity function.

The bounds on the loop transmission due to the specifications on the systems response to the command input, as discussed in the previous section, made extensive use of m-circles plotted on the Nichols chart. It would be advantageous to be able to use the Nichols chart and a similar concept to the m-circles to calculate the bounds on the loop transmission due the specifications on the systems response to the disturbances entering the plant. The problem is that the m-circles are contours of open loop magnitude versus open loop phase which would give a constant magnitude for the complementary sensitivity function, $T(s)$ , which has the following relationship to the loop transmission

$$T(s)=\frac{L(s)}{1+L(s)}$$

What is desired are contours of open loop magnitude versus open loop phase which would give a constant magnitude for the sensitivity function. A simple method for converting the m-circles to the desired contours is to first calculate the m-circle for the complementary sensitivity function, $T(s)$ , using the desired magnitude of the sensitivity function. The desired contour is obtained from this m-circle by converting its magnitude and phase points using the following relationship (Houpis et al,1999)

$$l(s)=\frac{1}{L(s)}$$

which gives the following magnitude and phase relationships for the sensitivity m-circles to the complementary sensitivity m-circle

$$|l(s)|=\frac{1}{|L(s)|}$$

$$\measuredangle l(s)=-\measuredangle L(s)$$

Figure 5.10 shows a selection of sensitivity m-circles at various magnitudes plotted on the Nichols chart. The chart containing these m-circles is referred to as the rotated Nichols chart.

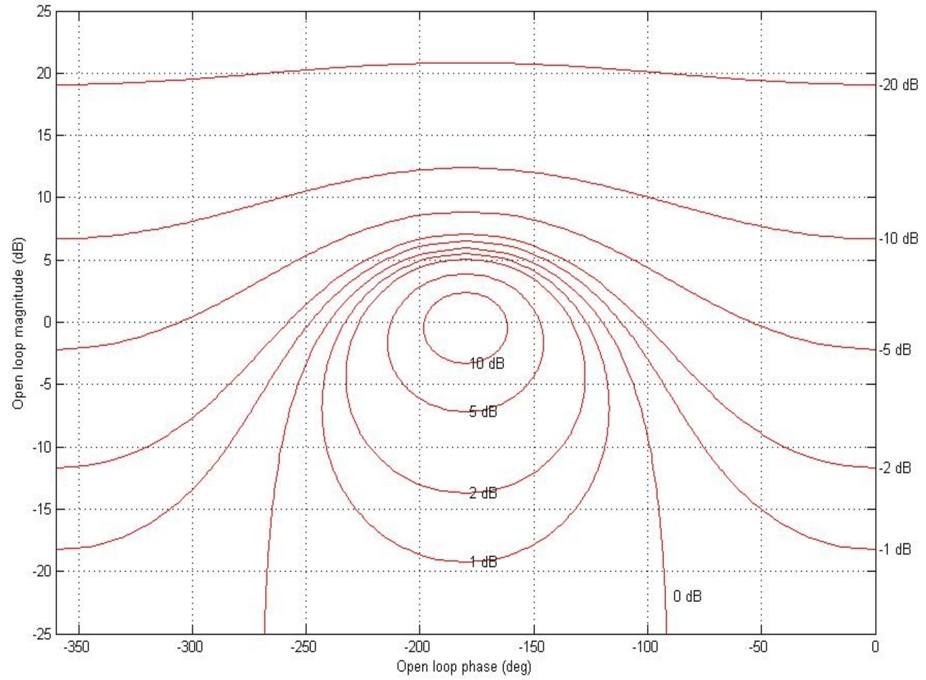


**Figure 5.10: The m-circles plotted on the rotated Nichols chart.**

### 5.2.1 Bandwidth

The closed loop bandwidth specification places a bound on the maximum magnitude of the systems response to a disturbance for the frequencies from 0 rad/s to the desired bandwidth frequency, $\omega_B$. This can be expressed mathematically as follows

$$|T_D(j\omega)| \leqslant -3\,dB \quad \text{for all} \quad \omega \in [0, \omega_B]$$

This bound can be converted to a bound on the sensitivity function as follows

$$|S(j\omega)| \leqslant \frac{-3\,dB}{|P_D(j\omega)|}$$

The sensitivity m-circles can now be calculated for a sample of the frequencies in the range from 0 rad/s to $\omega_B$, which place bounds on the minimum magnitude that the loop transmission set may have at the sampled frequencies. As in the bandwidth bounds due to the systems response to the command input, it is desirable to convert the sensitivity m-circle bounds to bounds on the nominal loop transmission so that the entire loop transmission set need not be considered.

This is achieved in a similar manner to the bandwidth bounds due to the systems response to the command input. For each sensitivity m-circle the plant template with the same frequency as the sensitivity m-circle is moved so that the entire plant template touches or lies above the sensitivity m-circle at a given open loop phase. The magnitude of the nominal plant at the current position of the plant template defines the minimum magnitude that the nominal loop transmission may have and still meet the bandwidth specification at the open loop phase under consideration. The minimum magnitudes of the nominal loop transmission is calculated over the range of open loop phases of interest which defines the bandwidth bound at the frequency of the sensitivity m-circle.

### 5.2.2 Resonant peak

The resonant peak specification places a bound on the maximum magnitude of the systems response to a disturbance for all frequencies, which can be expressed mathematically as follows

$$|T_D(j\omega)| \leqslant M_{RP} \quad \text{for all} \quad \omega$$

This bound can be converted to a bound on the sensitivity function as follows

$$|S(j\omega)| \leqslant \frac{M_{RP}}{|P_D(j\omega)|}$$

The bound above defines the maximum magnitude that the systems' sensitivity function may have versus frequency in order to meet the resonant peak specification on the systems' response to a disturbance. If the disturbance enters the plant at the plant output, $|P_D(j\omega)| = 1$, the bound is a constant magnitude equal to $M_{RP}$.

The maximum magnitude bound on the sensitivity function defines the sensitivity m-circle into which the loop transmission set may not cross. This is similar to the m-circle defining the region into which the plant set may not cross for the resonant peak specification on the systems' response to the command input.

In section 5.1.2 the calculation of the stability bound for ensuring that the resonant peak specification on the systems' response to the command input will be met was discussed. The calculation of the stability bound made use of the plant template and an m-circle with a closed loop magnitude equal to the resonant peak specification. The desired bound for ensuring the resonant peak specification for a disturbance input is met can be calculated in exactly the same manner as the stability bound. The sensitivity m-circle is substituted for the m-circle and the procedure remains the same.

### 5.2.3 Steady state response to disturbances

The steady state response to a disturbance, as discussed in section 3.4.2.4, places bounds on the number of pure integrators contained in the loop transmission as well as the minimum DC gain that the loop transmission must have with these integrators removed.

As in the case of the steady state error to the command input, see section 5.1.3, the steady state response to a disturbance must be met for all the plants in the plant set. Therefore, as before the factor by which the minimum DC gain required by the steady state specification must be increased is given by

$$|\Delta P(0)| = \frac{|P_{MIN}(0)|}{|P_0(0)|}$$

## *5.3 Response of the control system to sensor noise*

As discussed in section 3.4.3 the frequency domain specification on the allowable sensor noise level at the output of the control system is an upper bound on its RMS value, which is given by:

$$N_{RMS} = \sqrt{\int_{-\infty}^{+\infty} N(j\omega)|T(j\omega)|^2 d\omega}$$

Therefore, in order to reduce the effect of the sensor noise at the output of the control system, the designer needs to minimise the area under the magnitude of the control systems closed loop frequency response to a command input (complementary sensitivity function), $|T(j\omega)|$ .

As the major frequency components of the sensor noise lie at frequencies above the frequency range in which the performance specifications are specified, the only practical way of reducing the noise's RMS value at the system output is:

- To keep the $T(j\omega)$ 's bandwidth as low as possible without violating the performance specifications, and

- to roll-off the magnitude of $T(j\omega)$ as fast as possible at high frequencies.

If actual bounds are required for the sensor noise they will be in the form of the maximum allowable closed loop magnitude at high frequencies. These bounds are easily converted to bounds on the loop transmission. They are simply the m-circles at the maximum magnitude allowed. The plant uncertainty can be taken into account in exactly the same manner as the bandwidth bounds (See section 5.1.1).

# 6 Designing for a stable minimum phase plant

From the discussion in section 4, it was decided that the best way to evaluate the requirements that should be placed on the system is through the use of the loop transmission. What must be considered now is what this loop transmission should look like. The desirable properties that the loop transmission should have will now be discussed.

## 6.1 Desirable properties of the loop transmission

From the bounds calculated on the loop transmission (see section 5) it can be seen that it should have the following properties:

- The loop transmission should have a high gain at low frequencies. This high gain is required to meet the tracking, steady state error to a command input and steady state performance to a disturbance input specifications. All these specifications give a lower bound on the loop transmissions' magnitude at low frequencies.
- At high frequencies the loop transmission should have a low gain so that any high frequency sensor noise is attenuated and the high frequency plant uncertainty, which often contains unmodelled dynamics, does not effect the stability or performance of the system.
- The loop transmission should ensure that the system is acceptably stable by meeting the gain margin, phase margin and/or resonant peak specifications. This limits the minimum phase that the loop transmission may have when it rolls-off from high gain at low frequency to the cross-over frequency.

Also the following trade-offs should be considered when considering the performance that will be demanded of the control system. Different trade-offs will be made depending on the plant which is being considered. This means that the loop transmission defined as a starting point for considering the system requirements should also make provision for these trade-offs to be evaluated.

An increase in the low frequency magnitude improves the systems' tracking performance, disturbance rejection, steady state error due to the command input and the steady state performance due to a disturbance input. However if the relative stability of the system is not allowed to change, the bandwidth of the closed loop system increases. This is due to the fact that in order to maintain the relative stability, the roll-off in magnitude from the high to low frequency magnitude must be maintained, which means the increase in the low frequency magnitude will increase the frequency at which the loop transmission crosses the -3 dB m-circle (which means an increase in the closed loop bandwidth).

Although this will mean that the system will respond quicker to the command input, it will also mean that the magnitude of the closed loop system will be higher at high frequencies for the same high frequency roll-off in magnitude. This is undesirable as it will decrease the attenuation of the sensor noise and make it more likely that the high frequency plant uncertainty may effect the closed loop response.

The stability margins of the loop transmission can be reduced in order to allow the roll-off in magnitude of the loop transmission to be increased. This in turn will allow the bandwidth to be reduced with out having to reduce the low frequency magnitude. The decrease in the stability margins will however make the closed loop response more oscillatory which in most cases is undesirable.

So there is a trade-off between the low frequency magnitude, the bandwidth and the relative stability of the system.

Adding poles at high frequency (at frequencies greater than the cross-over frequency) can eleviate the need to limit the closed loop bandwidth as they will increase the high frequency roll-off in magnitude. This will allow the closed loop response to meet the low magnitude needed to reject the sensor noise and high frequency plant uncertainty effects.

The addition of these poles, however increases the phase lag of the loop transmission at the cross-over frequency. This in turn reduces the systems relative stability by reducing the phase margin. The higher in frequency that these poles are placed, the lower the effect on the relative stability, but the less the decrease in the closed loop magnitude at high frequency.

It is possible to increase the relative stability so that the poles may be added in closer to the cross-over frequency, but the increase in relative stability increases the cross-over frequency. This will in effect cancel any gain in getting closer to the cross-over frequency and will in effect have almost the same effect as when the stability margins were not increased.

Therefore the loop transmission should also allow the designer to evaluate the trade-offs in performance, particularly for any specifications that have not been specified.

## *6.2 The Bode ideal characteristic*

The Bode ideal characteristic is a good candidate for the loop transmission of a control system for a stable minimum phase plant (Horowitz,1993)(Lurie et al,2000), as it has the following properties:

- It has a high gain at low frequencies.
- It has a low gain at high frequencies.
- It has a definable relative stability through the gain and phase margin parameters.

The low frequency magnitude and the range of frequencies over which it is maintained is definable. The high frequency roll-off in magnitude is also definable. The relative stability is definable through the gain and phase margin parameters. Therefore these parameters can be adjusted by the designer in order to investigate the trade-offs discussed in the previous section.

The Bode ideal characteristic will now be discussed. The actual derivation of the characteristic will not be discussed as there are a number of good references containing the derivation, such as Horowitz's Book on QFT (Horowitz,1993) and Pritchard's thesis (1995).

The Bode ideal characteristic is made up of a number of parts which are:

- The Bode cut-off characteristic
- The Bode semi-infinite characteristic
- The Bode step characteristic

These parts will first be discussed and the logic behind their structure, the complete Bode ideal characteristic will then be constructed from them.

### 6.2.1 The Bode cut-off characteristic

The Bode cut-off characteristic allows the designer to specify the magnitude and phase in different frequency ranges. For a defined cut-off frequency, $\omega_0$ , the magnitude is specified for frequencies lower than $\omega_0$ and the phase for frequencies greater than $\omega_0$ . So to complete the characteristic the phase in the low frequency range needs to be calculated and the magnitude in the high frequency range must be calculated from the defined magnitudes and phases in the low and high frequency ranges respectively.

This characteristic is useful for loop shaping as high gain is normally required for the low frequency range due to the performance specifications such as tracking, steady state error and disturbance rejection specifications for the loop transmission. But the loop transmission must roll-off after $\omega_0$ in a manner which gives the system an acceptable relative stability. The roll-off must be limited so that it does not produce enough phase lag near cross-over so that the system is highly oscillatory or even unstable.

The roll-off is determined by specifying the desired phase in the frequency range above $\omega_0$. It is actually specified as a phase margin, which defines the amount of extra phase that the system must have greater than -180$^o$ at the cross-over frequency.

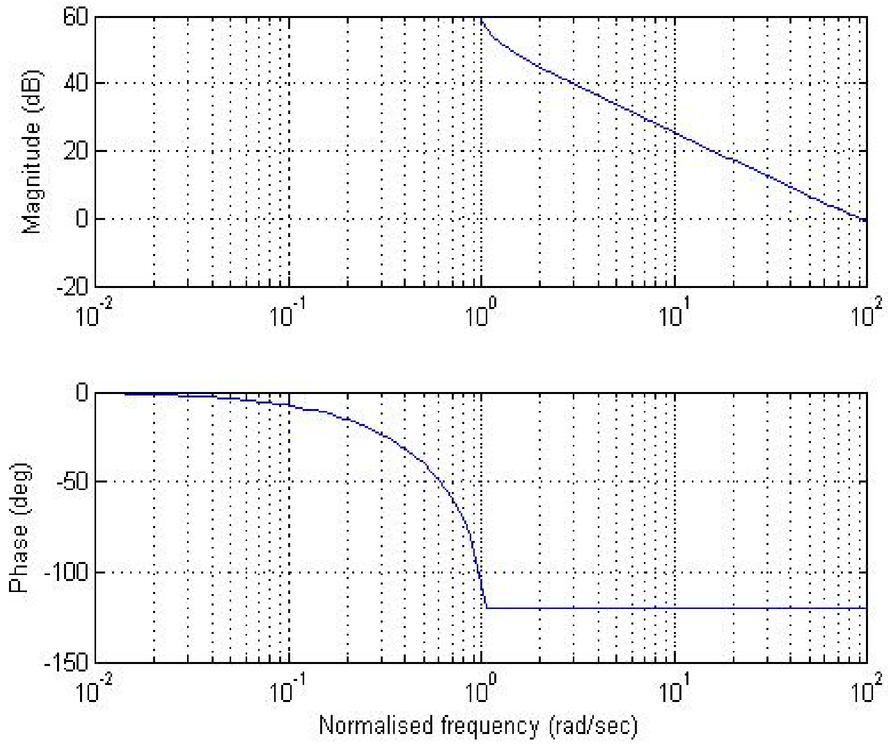


**Figure 6.1: The Bode cut-off characteristic.**

The magnitude of the bode ideal Bode loop transmission characteristic is given by:

$$|H_{BC}(\omega)| = \begin{cases} M_0 & \text{for} \quad \omega \leqslant \omega_0 \\ \dfrac{M_0}{\left(\dfrac{\omega}{\omega_0} + \sqrt{\left(\dfrac{\omega}{\omega_0}\right)^2 - 1}\right)^{2\alpha}} & \text{for} \quad \omega > \omega_0 \end{cases}$$

and the phase is given by:

$$\measuredangle H_{BC}(\omega) = \begin{Bmatrix} -2\alpha \tan^{-1}\left(\dfrac{\omega}{\sqrt{\omega_0^2 - \omega^2}}\right) & \text{for} \quad \omega \leqslant \omega_0 \\ -\alpha\pi & \text{for} \quad \omega > \omega_0 \end{Bmatrix} \text{ radians}$$

where:

$$\alpha = 1 - \frac{\phi_{PM}}{180^o}$$

| | |
|---|---|
| $M_0$ | is the low frequency gain (dB). |
| $\omega_0$ | is the maximum frequency of the low frequency gain in rad/s. |
| $\phi_{PM}$ | is the phase margin in degrees. |

The Bode cut-off characteristic meets the requirements for the loop transmission in that

1. it allows for a high gain at low frequencies through the parameter $M_0$
2. and it allows for this magnitude to roll-off at higher frequencies while specifying the desired relative stability through the phase margin parameter, $\phi_{PM}$ .

The Bode cut-off characteristic however does not allow the high frequency roll-off required to reduce the loop transmissions magnitude to an acceptable level at high frequencies.

What is required is a means to increase the roll-off after cross-over so that the high frequency specifications are met.

### 6.2.2 The Bode semi-infinite characteristic

The Bode semi-infinite characteristic defines the desired magnitude in a piece wise linear fashion. This will allow for a function to be defined which can increase the roll-off of the magnitude of the loop transmission after cut-off.

The magnitude of the Bode semi-infinite characteristic is 0 dB from 0 rad/s up until $\omega_0$ rad/s. For the frequencies greater than $\omega_0$ rad/s the slope of the magnitude is defined as $20 \cdot k$ dB/decade. The phase is then calculated from the magnitude characteristic defined.

Figure 6.2 shows an example of a semi-infinite characteristic with $\omega_0 = 1$ rad/s and $k = 1$ .

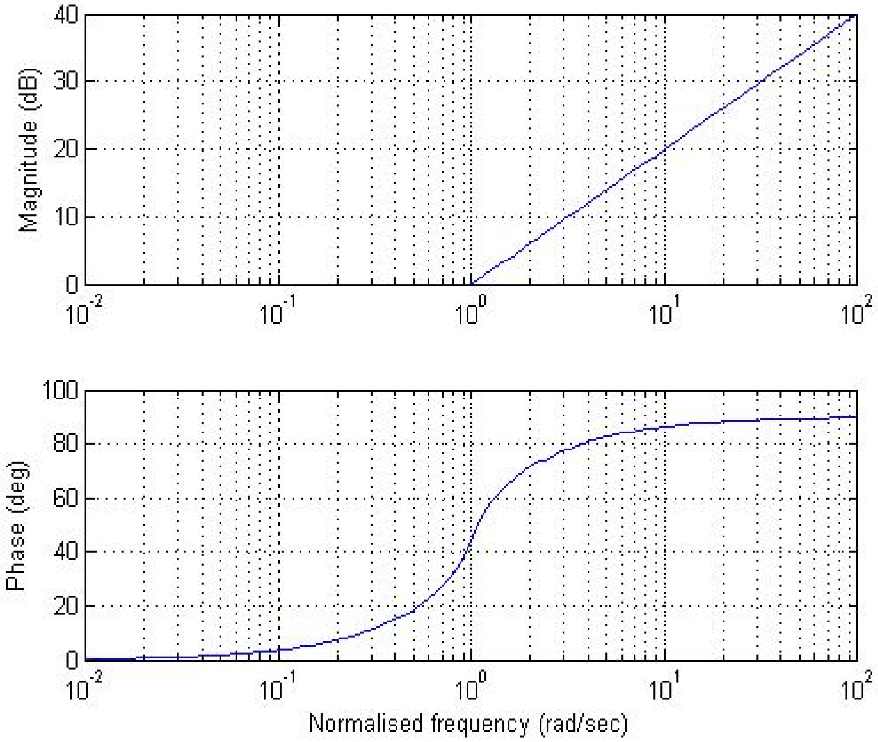


**Figure 6.2: The Bode semi-infinite characteristic.**

The magnitude of the Bode semi-infinite characteristic is given by:

$$|H_{SI}(\omega)| = \begin{cases} M_0 & for \quad x_0^C \leq 1 \\ M_0 \left(x_0^C\right)^k & for \quad x_0^C > 1 \end{cases}$$

And the phase (in radians) is given by:

$$
\measuredangle H_{SI}(\omega)=\begin{cases} k\dfrac{2}{\pi}\left[x_0^C+\dfrac{1}{9}\left(x_0^C\right)^3+\ldots\right] & \text{for} \quad 0<x_0^C\leqslant 0.414 \\ k\left[\dfrac{\pi}{4}-\dfrac{1}{\pi}\ln\left(x_0^C\right)\ln\left(y_0^C\right)-\dfrac{2}{\pi}\left(y_0^C+\dfrac{1}{9}\left(y_0^C\right)^3+\ldots\right)\right] & \text{for} \quad 0.414<x_0^C\leqslant 1.0 \\ k\dfrac{\pi}{2}-\measuredangle H_{SI}\left(\dfrac{1}{x_0^C}\right) & \text{for} \quad x_0^C>1.0 \end{cases}
$$

where :

$$x_0^C=\frac{\omega}{\omega_0}$$

$$y_0^C=\frac{1-x_0^C}{1+x_0^C}=\frac{\omega_0-\omega}{\omega_0+\omega}$$

From the equation describing the phase of the Bode semi-infinite characteristic it can be seen that the phase starts at 0° at 0 rad/s and goes to $k\cdot 90^o$ as the frequency tends to infinity. The phase at the corner frequency $\omega_0$ is $k\cdot 45^o$ .

If the semi-infinite characteristic is added to the Bode ideal characteristic to get the loop transmission with the desired high frequency roll-off, it can be seen from the semi-infinite characteristic phase, that the new loop transmission will have more phase lag at the cross-over frequency then the Bode cut-off characteristic alone.

This will jeopardize the relative stability (phase margin) of the loop transmission. It will therefore be necessary to add phase lead in order to maintain the phase margin of the loop transmission. This is achieved by the Bode step which is described in the following section.

### 6.2.3 The Bode step characteristic

The Bode step combines two semi-infinite characteristics with the goal of obtaining the desired high frequency roll-off without adding extra phase lag at frequencies lower than the lowest frequency corner frequency. This is achieved by introducing a semi-infinite characteristic with a positive slope before the semi-infinite characteristic which will achieve the high frequency roll-off, using a negative slope. The ratio between the first and second corner frequencies is selected such that no phase lag is introduced below the first corner frequency while keeping the corner frequencies as close as possible.

The magnitude of the Bode semi-infinite characteristic is given by:

$$|H_{BS}(\omega)|=|H_{SI1}(\omega)|\cdot|H_{SI2}(\omega)|$$

and the phase (in radians) is given by:

$$\sphericalangle H_{BS}(\omega)=\sphericalangle H_{SI1}(\omega)+\sphericalangle H_{SI2}(\omega)$$

The Bode step characteristic requires the following parameters:

| | |
|---|---|
| $\omega_1$ | is the first corner frequency in rad/s. |
| $k_1$ | is the desired slope of the magnitude between the corner frequencies $\omega_1$ and $\omega_2$. The slope is defined as $20\cdot k_1$ dB/decade. |
| $k_2$ | is the desired slope above the corner frequency $\omega_2$. The slope is defined as $20\cdot k_2$ dB/decade. |

The second corner frequency (in rad/s) is given by:

$$\omega_2=\frac{k_1-k_2}{k_1}\cdot\omega_1$$

The first semi-infinite characteristic, $H_{SI1}(\omega)$, has the slope of $k_1$ after its cut-off frequency $\omega_1$. In order for the Bode step to have a roll-off slope of $k_2$, the second semi-infinite characteristic, $H_{SI2}(\omega)$, must have the slope of $k_2-k_1$ after its cut-off frequency of $\omega_2$.

### 6.2.4 Putting all the characteristics together

The Bode ideal characteristic is constructed from the Bode cut-off characteristic and the Bode step characteristic.

The Bode cut-off characteristic is used to obtain the necessary low frequency gain. This gain, $M_0$, is defined for the frequency range from 0 rad/s up until a user defined frequency of $\omega_0$ rad/s. Above this frequency the phase is defined by the user so that an acceptable phase margin, $\phi_{PM}$, is achieved. The roll-off in magnitude from the low frequency magnitude, $M_0$, is therefore calculated in order to get the desired phase margin.

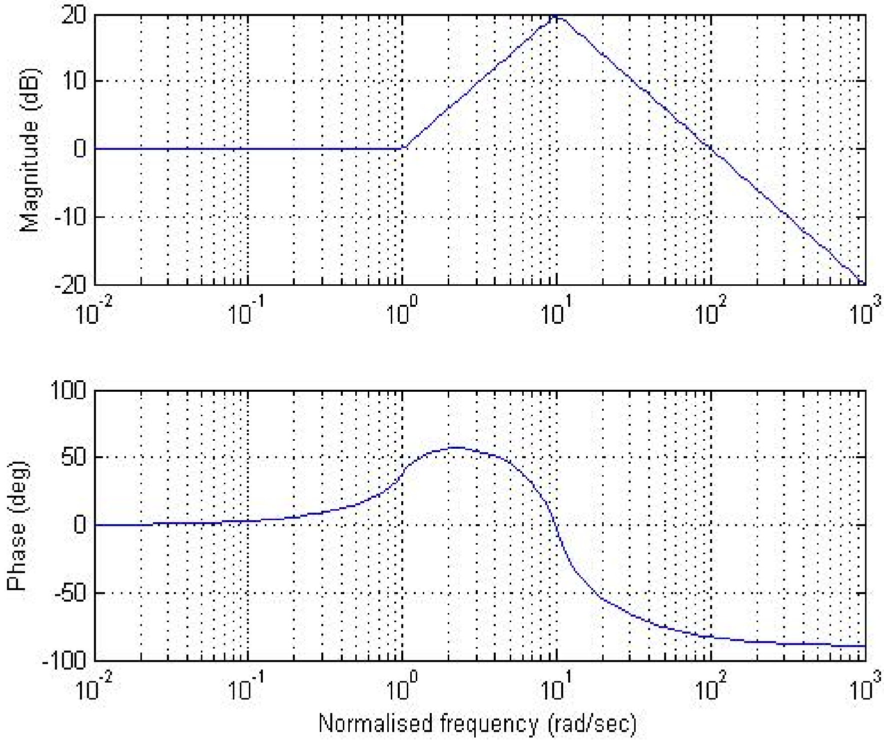


**Figure 6.3: The Bode step characteristic with $\omega = 1$, $k_1 = 1$ and $k_2 = -1$.**

The Bode step part of the Bode ideal characteristic allows the designer to specify the high frequency roll-off in magnitude without compromising the relative stability of the Bode cut-off part. This is achieved by allowing the designer to specify the gain margin and the desired pole-zero excess. The gain margin specifies at what magnitude below 0 dB the system may start to increase the roll-off in magnitude. This is necessary as the increase in magnitude roll-off has an accompanying increase in phase lag which may compromise the relative stability (The relationship between the closed loop resonant peak, gain and phase margins is discussed in section 6.3.2). Once the magnitude defined by the gain margin is reached by the Bode cut-off part of the Bode ideal characteristic, the Bode step part modifies the slope of the magnitude so that it is equal to 0 dB/decade. This is done in order to add sufficient phase lead so that the addition of the extra phase lag due to the desired high frequency magnitude roll-off does not modify the phase margin obtained by the Bode cut-off part.

The magnitude and phase of the Bode ideal characteristic is then given by:

$$|H_{BI}(\omega)| = |H_{BC}(\omega)| \cdot |H_{BS}(\omega)|$$

$$\measuredangle H_{BI}(\omega) = \measuredangle H_{BC}(\omega) + \measuredangle H_{BS}(\omega)$$

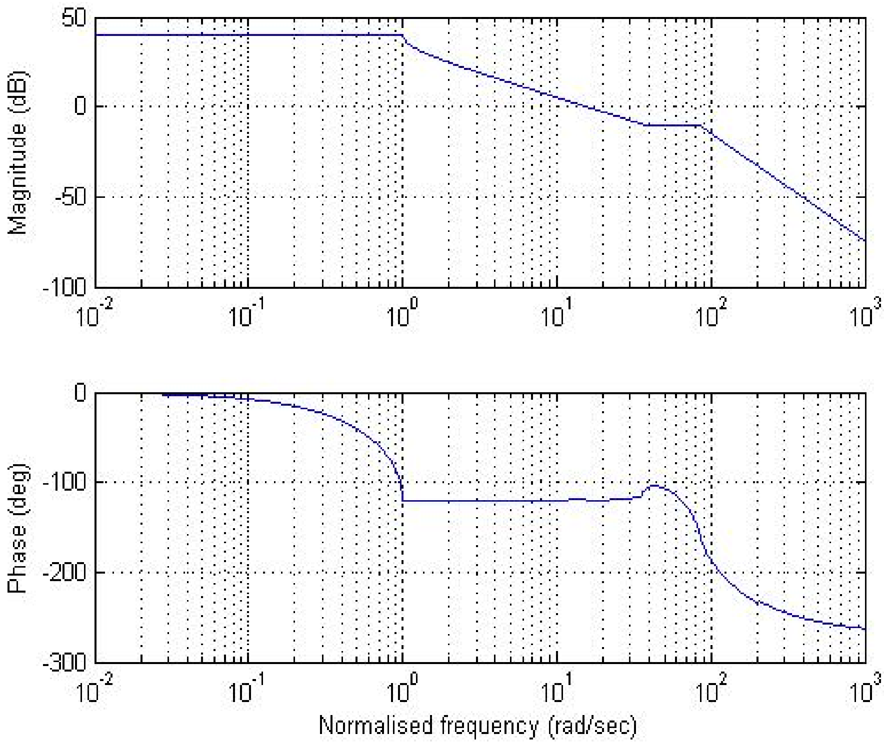


**Figure 6.4: The Bode plot of the Bode ideal characteristic.**

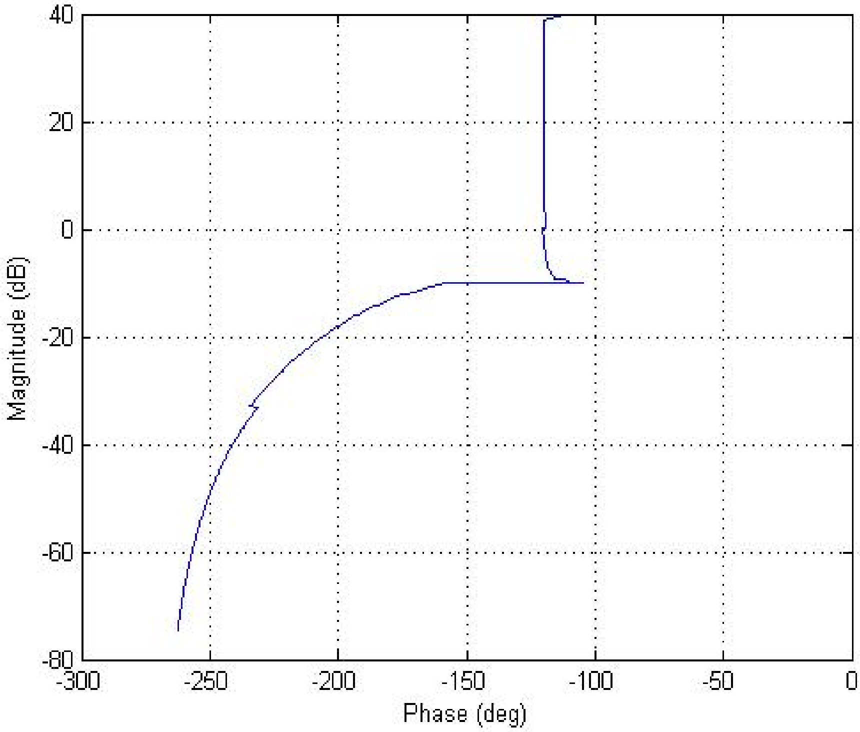


**Figure 6.5: The Nichols chart of the Bode ideal characteristic.**

The Bode ideal characteristic requires the following parameters:

| | |
|---|---|
| $M_0$ | The low frequency gain (dB). |
| $\omega_0$ | The maximum frequency of the low frequency gain in rad/s. |
| $\phi_{PM}$ | The phase margin in radians. |
| $M_{GM}$ | The gain margin (dB). |
| $e$ | The pole-zero excess. |

The corner frequency which levels the slope of the Bode ideal characteristic off to 0 dB/decade, $\omega_1$ , is given by:

$$\omega_1 = \omega_0 \cdot 10^{\frac{M_0 + M_{GM} - 12\alpha}{40\alpha}}$$

The parameters required by the Bode cut-off part of the Bode ideal characteristic are the low frequency gain, $M_0$ , the maximum frequency of the low frequency gain, $\omega_0$ , and the phase margin, $\phi_{PM}$ . As these parameters are required by the Bode ideal characteristic in the same form as required by the Bode cut-off characteristic, the values supplied can by used by the Bode cut-off characteristic directly.

The parameters required by the Bode step part of the Bode ideal characteristic are given by:

$\omega_1$ is the first corner frequency of the Bode step characteristic. This value is the same as the value calculated for $\omega_1$ above.

$k_1$ is the desired slope of the magnitude between the Bode steps first and second corner frequencies ( $\omega_1$ and $\omega_2$ respectively). This is set to a value of $2 \cdot \alpha$ which is the inverse of the slope of the Bode cut-off characteristic.

$k_2$ is the desired slope above the second corner frequency. This is set to the roll-off specified for the Bode ideal characteristic by the pole-zero excess parameter, which is given by $-e$ .

## 6.3 Relating the parameters of the Bode Ideal characteristic to the specifications

### 6.3.1 Low frequency parameters

The low frequency parameters consist of the low frequency gain, $M_0$ , the maximum frequency of the low frequency gain, $\omega_0$ , and the system type required, *k* (which will be discussed in the following section). These parameters are specified by the following specifications:

1. The steady state error specifications for both the steady state response to the command and disturbance inputs.
2. The specifications on tracking the command input.

#### 6.3.1.1 Steady state error

The Bode ideal characteristic is type 0, which severely limits the steady state specifications that can be met using it. Therefore, the first step required is to extend the Bode ideal characteristic to contain pure integrators which will allow it to have a type greater than or equal to zero.

After the extension of the Bode ideal characteristic is discussed, the calculation of the values of the Bode ideal characteristic's parameters required to meet the steady state specifications will be discussed.

#### 6.3.1.1.1 Extending the Bode ideal characteristic to be able to specify a system type parameter

The Bode ideal characteristic can easily be extended to allow for the system type, $k$, to be specified. This is achieved by adding pure integrators to the Bode cut-off characteristic part of the Bode ideal characteristic. The integrators have the following transfer function

$$H_T(s)=\left(\frac{\omega_0}{s}\right)^k$$

which has the following magnitude and phase

$$|H_T(\omega)|=\left(\frac{\omega_0}{\omega}\right)^k$$

$$\sphericalangle H_T(\omega)=\frac{-k\cdot\pi}{2}\,radians=-90\cdot k\,degrees$$

Note that the maximum frequency of the low frequency gain parameter, $\omega_0$, is included. This ensures that the modified Bode cut-off characteristic has the same magnitude at $\omega=\omega_0$ as the unmodified version.

The modified Bode cut-off characteristic will then have the following magnitude and phase

$$|H_{TBC}(s)|=|H_{BC}(s)|\cdot|H_T(s)|$$

$$|H_{TBC}(s)|=\left(\frac{\omega_0}{s}\right)^k\cdot|H_{BC}(s)|$$

$$\sphericalangle H_{TBC}(s)=\sphericalangle H_{BC}(s)+\sphericalangle H_T(s)$$

$$\measuredangle H_{TBC}(s) = \measuredangle H_{BC}(s) - \frac{\pi}{2} \cdot k$$

From the phase equation above it can be seen that the phase of the modified Bode cut-off characteristic has an extra phase lag of $\pi/2 \cdot k$ radians. This is undesirable, but can easily be remedied by adjusting the phase margin specification of the modified Bode cut-off characteristic, $\phi'_{PM}$, as follows

$$\phi'_{PM} = \phi_{PM} + \frac{\pi}{2} \cdot k$$

The modified Bode cut-off characteristic with the same phase margin as the original Bode cut-off characteristic is given by the above equations except that the phase margin used by the Bode cut-off characteristic $\phi'_{PM}$.

The complete Bode ideal characteristic modified to allow for a system type specification is obtained by replacing the Bode cut-off characteristic with the modified Bode cut-off characteristic in section 6.2.4 to get

$$|H_{TBI}(s)| = |H_{TBC}(s)| \cdot |H_{BS}(s)|$$

$$\measuredangle H_{TBI}(s) = \measuredangle H_{TBC}(s) + \measuredangle H_{BS}(s)$$

Figure 6.6 and Figure 6.7 show an example of a type 1 Bode ideal characteristic plotted on the Bode plot and Nichols chart respectively.

##### 6.3.1.1.2 The steady state error due to the command input for a system with a Bode ideal characteristic loop transmission

As discussed in section 3.4.1.4, the steady state error can be calculated using the final value theorem. For a unity feedback system with power of time inputs, an error coefficient can be defined which in turn can be used to calculate the steady state error.

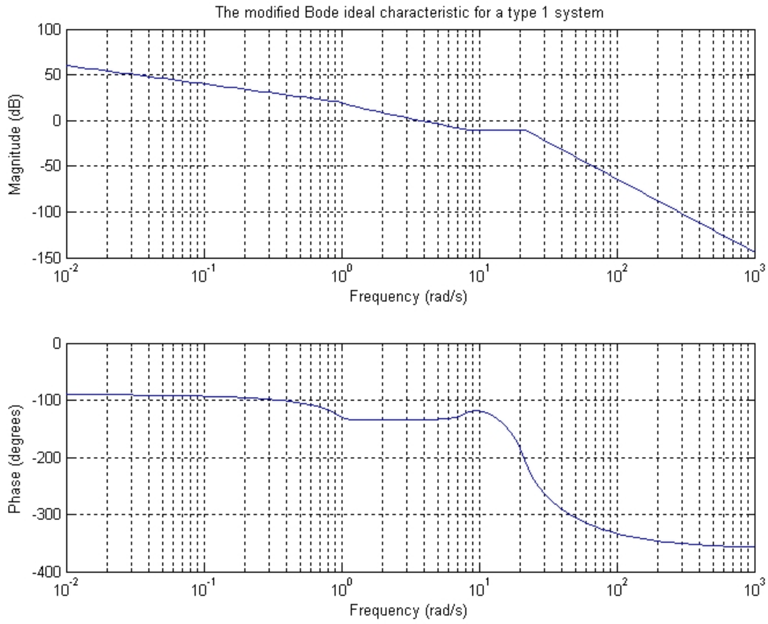


**Figure 6.6: The Bode plot of the Bode ideal characteristic with a pole at the origin.**

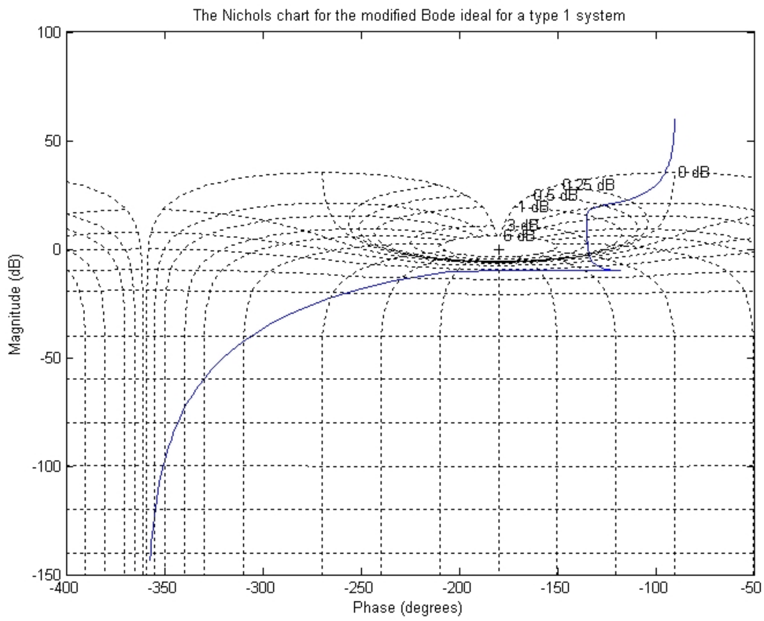


**Figure 6.7: The Nichols chart of the Bode ideal characteristic with a pole at the origin.**

Using the error coefficients defined for the steady state error due to a power of time input

$$K_i = \lim_{s \to 0} s^i \cdot L(s)$$

where the loop transmission is given by

$$L(s) = H_{TBI}(s) = \frac{\omega_0^k}{s^k} \cdot H_{BI}(s)$$

where:

k is the desired system type

Then the error coefficient becomes

$$K_i = \lim_{s \to 0} \left[ s^{i-k} \cdot \omega_0^k \cdot H_{BI}(s) \right]$$

As $s \to 0$ the Bode ideal characteristic, $H_{BI}(0) = M_0$ . So the error coefficient can be simplified to

$$K_i = M_0 \cdot \omega_0^k \cdot \left( \lim_{s \to 0} s^{i-k} \right) = \begin{cases} 0 & if \quad k < i \\ M_0 \cdot \omega_0^k & if \quad k = i \\ \infty & if \quad k > i \end{cases}$$

The steady state error can now be calculated using the error coefficients.

The steady state error to a step input (*i=0*) is given by

$$e(\infty) = \frac{A}{1 + K_0} = \begin{cases} \dfrac{A}{1 + M_0} & if \quad k = 0 \\ 0 & if \quad k > 0 \end{cases}$$

The steady state error to the other power of time inputs (*i>0*) is given by

$$e(\infty)=\frac{A}{K_i}=\begin{cases}0 & if \quad k<i\\ \dfrac{A}{M_0\cdot\omega_0^k} & if \quad k=i\\ \infty & if \quad k>i\end{cases}$$

The steady state errors can be summarised in Table 3:

| ***Type of L(s)*** | ***Steady state error to a step input*** $r(t)=A\cdot u(t)$ | ***Steady state error to a ramp input*** $r(t)=A\cdot t$ | ***Steady state error to a parabolic input*** $r(t)=A\cdot t^2$ |
|---|---|---|---|
| 0 | $\frac{A}{1+M_0}$ | $\infty$ | $\infty$ |
| 1 | 0 | $\frac{A}{M_0\cdot\omega_0^k}$ | $\infty$ |
| 2 | 0 | 0 | $\frac{A}{M_0\cdot\omega_0^k}$ |
| 3 | 0 | 0 | 0 |

**Table 3: The steady state error due to a power of time command input for a system with the Bode ideal characteristic as the loop transmission.**

#### 6.3.1.1.3 The steady state error due to a disturbance input for a system with a Bode ideal characteristic loop transmission

As discussed in the previous section the error coefficient for the Bode ideal characteristic is given by:

$$K_i=\begin{cases}0 & if \quad k<i\\ M_0\cdot\omega_0^k & if \quad k=i\\ \infty & if \quad k>i\end{cases}$$

So for a step disturbance (*i=0*) the steady state error is given by

$$e(\infty)=\begin{cases}\frac{A}{1+M_0}\cdot P_D(0) & if \quad k=0\\ 0 & if \quad k>0\end{cases}$$

The steady state error to the other power of time disturbance inputs (*i>0*) is given by

$$e(\infty)=\frac{A}{K_i}=\begin{cases}0 & if \quad k<i\\ \frac{A}{M_0\cdot\omega_0^k}\cdot P_D(0) & if \quad k=i\\ \infty & if \quad k>i\end{cases}$$

The steady state responses to disturbance inputs are summarised Table 4.

| ***Type of L(s)*** | ***Steady state error to a step input*** $r(t)=A\cdot u(t)$ | ***Steady state error to a ramp input*** $r(t)=A\cdot t$ | ***Steady state error to a parabolic input*** $r(t)=A\cdot t^2$ |
|---|---|---|---|
| 0 | $\frac{A}{1+M_0}\cdot P_D(0)$ | $\infty$ | $\infty$ |
| 1 | 0 | $\frac{A}{M_0\cdot\omega_0^k}\cdot P_D(0)$ | $\infty$ |
| 2 | 0 | 0 | $\frac{A}{M_0\cdot\omega_0^k}\cdot P_D(0)$ |
| 3 | 0 | 0 | 0 |

**Table 4: The steady state error due to a power of disturbance input for a system with the Bode ideal characteristic as the loop transmission.**

### 6.3.1.2 Tracking specifications

The tracking specifications placed bounds on the allowable magnitude and phase of the loop transmission at each frequency, see section 5.1.4. The values of $M_0$ , $\omega_0$ and $k$ must be selected such that these bounds are met.

If a steady state error specification has been specified, lower limits have been placed on these values (as discussed in section 6.3.1.1.2). It is undesirable to increase $k$ as this will make it more difficult to stabilise the system. Therefore, $M_0$ and $\omega_0$ must be adjusted to meet the tracking specifications, if necessary.

## 6.3.2 Relative stability

The frequency domain relative stability specifications are:

- Gain margin
- Phase margin
- Resonant peak

These specifications will now be discussed in terms of the values that the properties of the bode ideal characteristic must be set to in order to meet them for plants with and without plant uncertainty. The properties effected by the relative stability specifications are the phase margin, $\phi_{PM}$ , and the gain margin, $M_{GM}$ .

The relationship between these specifications will also be discussed.

### 6.3.2.1 Phase margin

The phase margin specification relates directly to the Bode ideal characteristics' phase margin property. For a plant with no uncertainty the phase margin specification can be used directly. However, for an uncertain plant the phase margin specification must be met for all the plants in the plant set. So the phase margin will need to be increased to ensure this.

This is achieved in a similar manner to the QFT stability bounds, as discussed in section 5.1.2. The plant template, at a particular frequency of interest, is placed so that the phase of the nominal loop transmission is -360$^{o}$ (the nominal plant associated with the plant template lies at -360$^{o}$). The magnitude of the nominal loop transmission is chosen so that the plant template lies above the 0 dB on the Nichols chart and touches but does not cross it. The plant template is then moved such that the phase of the nominal loop transmission increases, while the plant template still touches but does not cross over the 0 dB line. The movement of the plant template is stopped once the minimum phase of the plant template is equal to the phase margin minus 180°. See Figure 6.8. The template is now moved so that it lies below the 0 dB line (still touching but not crossing it) and is moved back to -360°. The path traversed by the nominal plant defines the boundary that insures that all the plants in the plant set at the plant templates frequency meet the phase margin specification. This is performed for a number of frequencies lying in the range of frequencies of interest. If the nominal plant does not cross into all of these bounds at their specific frequencies, then the gain margin specification is met for all the plants in the plant set.

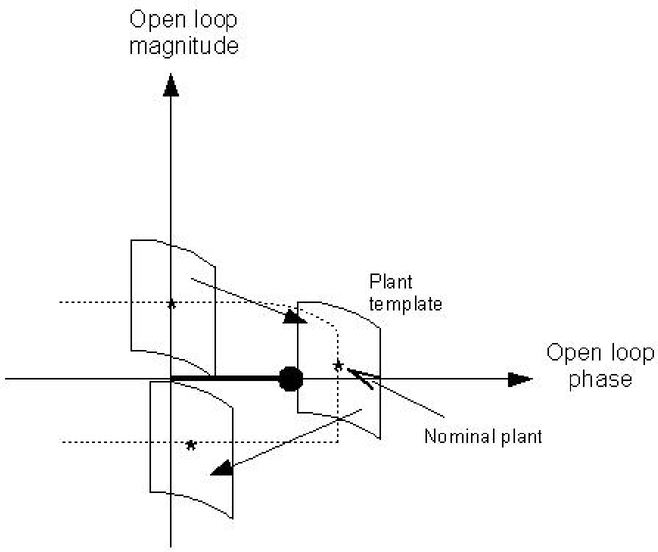


**Figure 6.8: Bounds due to the gain margin for a single frequency.**

Therefore, the phase margin property of the Bode ideal characteristic must be selected such that none of its gain and magnitude points lie within the bound at the bounds' frequency.

### 6.3.2.2 Gain margin

As in the case of the phase margin (discussed in the previous section), the gain margin specification relates directly to the Bode ideal characteristics' gain margin property. For a plant with no uncertainty the gain margin specification can be used directly.

For an uncertain plant the gain margin specification must be used to generate bounds at a selection of frequencies within the frequency range of interest. These bounds are similar to the phase margin bounds except that the plant template is moved around the open loop phase line of -180° from the open loop magnitude of 0 dB to the negative value in decibels of the gain margin specification.

As before the Bode ideal characteristic must be selected so that none of its open loop phase-magnitude points lie within the bounds at the frequency of the bounds.

### 6.3.2.3 Resonant peak

#### 6.3.2.3.1 The relationship between the gain and phase margins specifications and the resonant specification

The gain and phase margins of the loop transmission of a control system are related to the resonant peak specification. From Figure 6.9 it can be seen that the m-circle for the resonant peak specification places limits on the minimum gain and phase margins. As a loop transmission with a smaller gain or phase margin will cross into the m-circle and therefore violate the resonant peak specification. Therefore, if the control system has a resonant peak specification and/or gain and phase margin specifications, it is useful to check that they are consistent.

The gain margin related to the m-circle is given by (see section 10.6 for the derivation of the following equations):

$$M_{GM} = \frac{M_{RP}}{1 + M_{RP}}$$

where:

$M_{RP}$ is the maximum magnitude of the resonant peak allowed.

The phase margin related to the m-circle is given by:

$$\phi_{PM} = \cos^{-1}\left(\frac{1}{2\mathrm{M}_{RP}^2 - 1}\right)$$

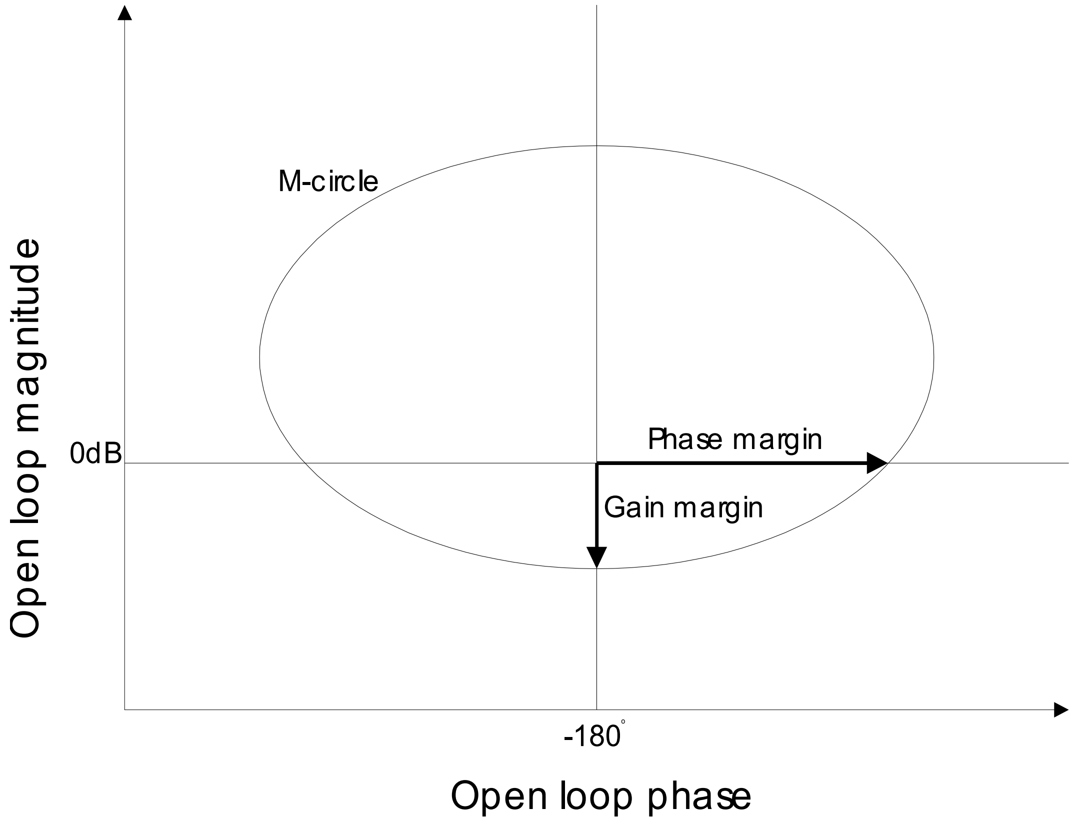


**Figure 6.9: The relationship between the gain margin, phase margin and resonant peak specifications**

6.3.2.3.2 Selecting the Bode ideal characteristics' properties so that the resonant peak specification is met for plants with no uncertainty

The gain margin specification on the Bode ideal characteristic should be equal to or greater than the minimum magnitude of the m-circle (the minimum magnitude of the m-circle at an open loop phase of -180°) in order to ensure that it does not cross into the m-circle, as shown in Figure 6.10.

The phase margin property of the Bode ideal characteristic cannot be set to the phase related to the m-circle (as discussed in the previous section), from Figure 6.11 it can be seen that the Bode ideal characteristic will cross into the m-circle. This is due to the fact that the Bode ideal characteristic has constant phase from the frequency, $\omega_0$ , until the magnitude drops to the open loop magnitude equal to the negative value of the phase margin property in decibels. While the m-circle however has a phase greater than its phase at the open loop magnitude of 0 dB.

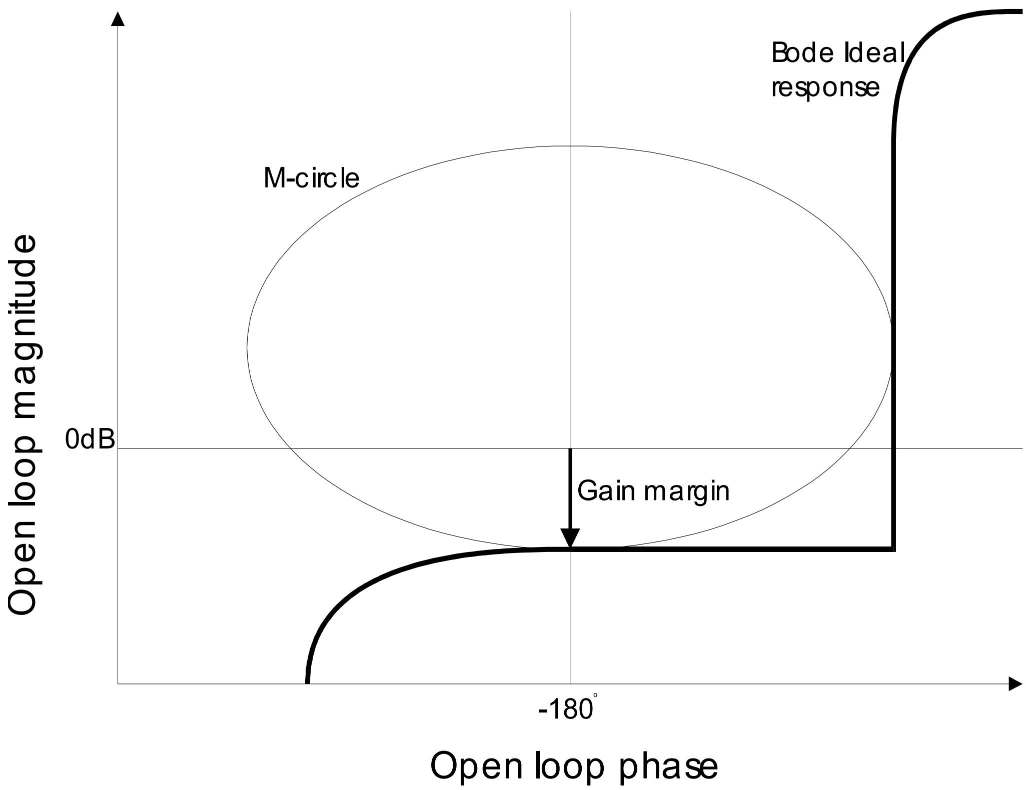


**Figure 6.10: Relating the gain margin property of the Bode ideal characteristic to the resonant peak specification.**

In order to insure that it does not cross into the m-circle, the phase margin specification should be set to the maximum phase of the m-circle. The maximum phase of the m-circle (see section 10.6 for the derivation of the following equation):

$$\phi_{MAX} = \cos^{-1}\left(\frac{-\sqrt{M_{RP}^{2}-1}}{M_{RP}}\right)$$

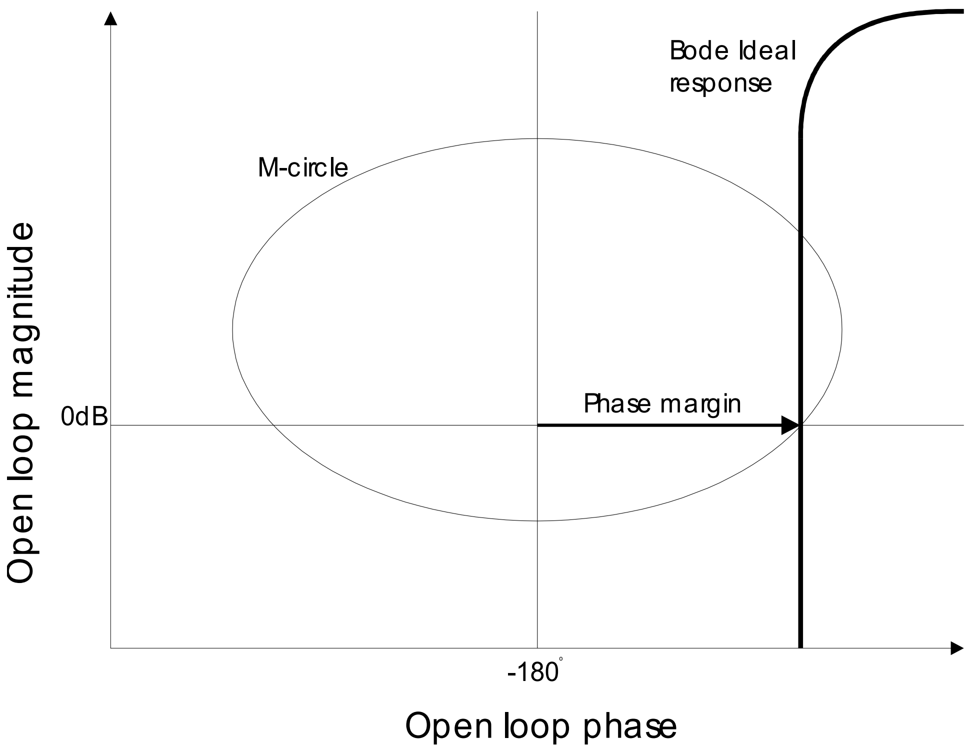


**Figure 6.11: Setting the phase margin property of the Bode ideal characteristic to the phase margin of the m-circle related to the 0 dB point of the resonant peak specification.**

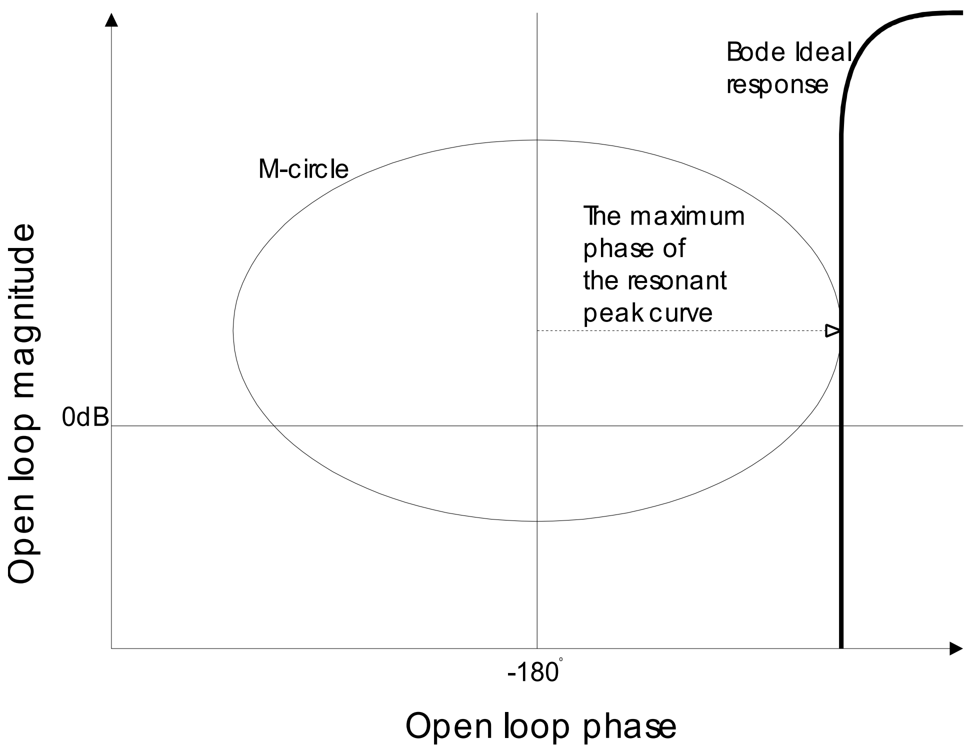


**Figure 6.12: Setting the phase margin property of the Bode ideal characteristic to the maximum phase of the m-circle related to the resonant peak specification.**

#### 6.3.2.3.3 Selecting the Bode ideal characteristics' properties so that the resonant peak specification is met for plants with uncertainty

When a plant has significant plant uncertainty, the m-circle related to the resonant peak specification (as discussed in the previous section) will not guarantee that all the plants in the plant set meet the specification if it used to calculate the Bode ideal characteristics' properties . This is due to the fact that the Bode ideal characteristic does not contain the plants' uncertainty.

In this case the m-circle needs to be replaced with the QFT stability bounds, as discussed in section 5.1.2. The Bode ideal characteristics' properties must be selected so that it meets all the stability bounds.

### 6.3.3 Speed of response

The two parameters of the Bode ideal characteristic that effect the speed of response of the closed loop system, normally measured as the closed loop bandwidth, are :

- the maximum frequency of the low frequency gain, $\omega_0$
- the phase margin , $\phi_{PM}$

The maximum frequency of the low frequency gain, $\omega_0$ , shifts the whole Bode ideal characteristic in frequency, so its value has a direct effect on the frequency at which the characteristic cuts the -3 dB m-circle (.i.e the closed loop bandwidth). Increasing $\omega_0$ will increase the bandwidth of the system.

The phase margin , $\phi_{PM}$ , effects the closed loop bandwidth by indirectly specifying the roll-off in magnitude from the low frequency magnitude, $M_0$ , to the -3 dB m-circle. Increasing the phase margin will increase the closed loop bandwidth as this will cause the roll-off in magnitude from $M_0$ to the -3 dB m-circle.

### 6.3.4 High frequency properties of the closed loop response

The high frequency properties of the closed loop response that will considered is the magnitude and roll-off of the closed loop magnitude. The main reason for considering the high frequency properties is sensor noise, as it is undesirable that the system should allow this noise through to the system output. It is therefore necessary for the magnitude of the closed loop response to be low at the frequencies of the sensor noise.

The high frequency properties of the closed loop response are effected by the pole-zero excess, $e$, and the speed of response parameters discussed in the previous section. It is obvious that the bandwidth of the closed loop system will effect the high frequency magnitude of the closed loop system. The higher the bandwidth, the higher the closed loop magnitude will be at high frequencies for the same pole-zero excess. The closed loop bandwidth should therefore be kept as small as possible, so that the closed loop magnitude at the sensor noise frequencies can be low enough to reject the noise.

The high frequency roll-off of the closed loop magnitude is determined by the pole-zero excess parameter of the Bode ideal characteristic. This parameter must be set to the desired pole-zero excess of the closed loop system. The high frequency roll-off obtained is simply $20 \cdot e$ dB. The pole-zero excess of the Bode ideal characteristic must also be greater than the pole-zero excess of the plant. If this is not the case, the controller will not be allowed to roll-off.

### 6.3.5 The trade offs between the parameter values which should be considered

Now that the parameters have been linked to the specifications, the trade-offs that should be considered by the designer will be linked to the parameters.

If the low frequency magnitude, $M_0$, is increased, the systems' tracking performance, disturbance rejection, steady state error due to the command input and the steady state performance due to a disturbance input will be improved. But if the phase margin, $\phi_{PM}$, is not allowed to change, the bandwidth of the closed loop system will increase.

The maximum frequency of the low frequency magnitude, $\omega_0$, can also be increased in order to increase the frequency range over which the systems' tracking performance and disturbance rejection is improved. This will also increase the systems bandwidth.

Although this will mean that the system will respond quicker to the command input, it will also mean that the magnitude of the closed loop system will be higher at high frequencies for the same high frequency roll-off in magnitude. This is undesirable as it will decrease the attenuation of the sensor noise and make it more likely that the high frequency plant uncertainty may effect the closed loop response.

The phase margin can be reduced in order to allow the roll-off in magnitude of the Bode ideal characteristic to be increased which will reduce the bandwidth. The decrease in the phase margin will however make the closed loop response more oscillatory.

So there is a trade-off between low frequency performance of the closed loop system as defined by the parameters $M_0$ and $\omega_0$, the bandwidth of the closed loop system and the relative stability as defined by the parameter $\phi_{PM}$.

The magnitude roll-off at high frequency as defined by the pole-zero excess, $e$, can be increased so that the high frequency magnitude of the closed loop response meets the low magnitude needed to reject the sensor noise and high frequency plant uncertainty effects. However the increase in $e$ will also increase the frequency of the corner frequency at which the high frequency magnitude roll-off starts, $\omega_2$. So the increase in $e$ may not decrease the high frequency magnitude as desired, which means that the low frequency magnitude may need to be decreased as discussed before.

The phase margin, $\phi_{PM}$, may be decreased so that the magnitude roll-off at high frequency can begin at a lower frequency. This however reduces the relative stability of the system which is often undesirable.

So there is also a trade-off between high frequency performance of the closed loop system as defined by the parameter $e$ and the relative stability as defined by the parameter $\phi_{PM}$.

So ultimately, the designer needs to trade off the low frequency performance of the closed loop system, the relative stability of the system and the high frequency performance of the system.

## 6.4 Example

The example describes the design of the pitch control loop for an Aerosonde unmanned air vehicle (UAV).

### 6.4.1 The plant model

The plant set for the Aerosonde UAV was obtained by linearising the non-linear model of the UAV, implemented using the Aerosim block set for Simulink®, at a number of points in the flight envelope. The set of linear models obtained are of the following form

$$\frac{\theta(s)}{\delta_e(s)}=\frac{K\cdot(s/\omega_1+1)\cdot(s/\omega_2+1)\cdot(s/\omega_3+1)}{(s^2/\omega_p^2+2\cdot\zeta_p/\omega_p\cdot s+1)\cdot(s^2/\omega_{sp}^2+2\cdot\zeta_{sp}/\omega_{sp}\cdot s+1)\cdot(s/\omega_{sd}+1)}$$

where:

| | |
|---|---|
| $\theta$ | is the pitch of the UAV. |
| $\delta_e$ | is the elevator deflection. |
| $K$ | is the DC gain of the pitch response to the elevator deflection. |
| $\omega_1$ , $\omega_2$ , $\omega_3$ | are the zeros of the linear model. |
| $\zeta_p$ | is the damping ratio of the phugoid poles. |
| $\omega_p$ | is the natural frequency of phugoid poles. |
| $\zeta_{sp}$ | is the damping ratio of the short period poles. |
| $\omega_{sp}$ | is the natural frequency of the short period poles. |
| $\omega_{sd}$ | is the natural frequency of the spiral divergence poles. |

### 6.4.2 The operational requirements

There are no formal specifications for this example, so the design goal will be to find an optimal pitch control loop by evaluating the trade offs between the different performance characteristics.

For practical reasons the following informal specifications will be imposed:

1. The allowable range for the pitch command will be $-15^o\leqslant\theta_{c_{MAX}}\leqslant+15^o$ .
2. The maximum overshoot will be limited to approximately $\theta_{OV}=1.5^o$ .

**Step 1: Translate the time domain specifications into the frequency domain**

*Inputs:* All the available time domain specifications on the closed loop control system.

*Outputs:* The frequency domain versions of the time domain specifications.

*Procedure*:

Using the calculations discussed in section 3.4, the frequency domain specifications may be estimated.

The only specification is the maximum overshoot that the closed loop pitch control loop may have. The maximum overshoot in pitch allowed is $\theta_{OV}=1.5^o$ . The maximum pitch command allowed is $\theta_{c_{MAX}}=15^o$ . So the maximum percentage overshoot allowed is

$$Percentage\,overshoot=\frac{\theta_{OV}}{\theta_{c_{MAX}}}\cdot 100=\frac{1.5^o}{15^o}\cdot 100 = 10\%$$

From Figure 3.12 the resonant peak of a second order model that corresponds to a 10% overshoot is approximately

$$M_{RP}=1\,dB$$

**Step 2: Translate the frequency domain bounds into bounds on the nominal loop transmission**

*Inputs:* All the available frequency domain specifications on the closed loop control system.

*Outputs:* The bounds on the loop transmission.

*Procedure*:

Using the calculations discussed in section 5, the bounds on the loop transmission obtained from the closed loop frequency domain specifications.

As the plant in this example has a significant amount of plant uncertainty due to the flight envelope, the first step is to generate the plant template and select the nominal plant.

The nominal plant selected is chosen to lie in the middle of the plant template (as far as practically possible). In a standard QFT design the nominal plant chosen has no effect on the design of the compensator (Horowitz,1993).

A QFT design starts with the selection of the nominal plant and the generation of the plant templates from the plant model. The plant templates and the nominal plant are then used to generate bounds on the loop transmission. The selection on the nominal plant will change the bounds obtained, as these bounds depend on the plant templates boundary and the difference magnitude and phase between the plant templates boundary and the nominal plant. The designer then starts with the nominal plant onto which poles and zeros of the compensator are added so that the nominal loop transmission meets the QFT bounds generated. The designer is therefore working relative to the nominal plant. As the bounds are also relative to the nominal plant, the selection of the nominal plant will not effect the the design of the compensator. Therefore, in a QFT design the nominal plant is normally selected so that the bounds can be calculated as simply as possible.

In this case the structure of the loop transmission is assumed to be the Bode ideal characteristic and the compensator can be derived from this structure through the use of the nominal plant. The nominal plant selected will therefore influence the compensator attained as the Bode ideal characteristic is not relative to the nominal plant.

As the Bode ideal characteristic is being used to evaluate the design trade-offs and not give the final design, the nominal plant selected will lie in the middle of the plant templates. This should give a good estimate of the desired loop transmission, as the selection of the middle of the plant template as the nominal plant means that average plant uncertainty will be used in the design.

Figure 6.13 shows a bode plot of the plant set (plotted in blue) and the nominal plant selected (plotted in red).

Note that the phase at DC is -180°. This is due to the fact that the plants all have a negative gain, $K$ .Matlab however plots the phase at DC at +180°. Therefore, the plots of the plant set start at +180°. This is not mathematically incorrect, but it does not follow convention.

The Bode plot of the magnitude nominal plant selected is shown in Figure 6.14. From the Bode plot it can be seen that the resonant peak of the nominal plant is highly underdamped. It should be noted that the compensator obtained would be calculated as follows:

$$G(s)=\frac{L_0(s)}{P_0(s)}$$

where:

$L_0(s)$ is the nominal loop transmission as defined by the Bode ideal characteristic.
$P_0(s)$ is the nominal plant selected.

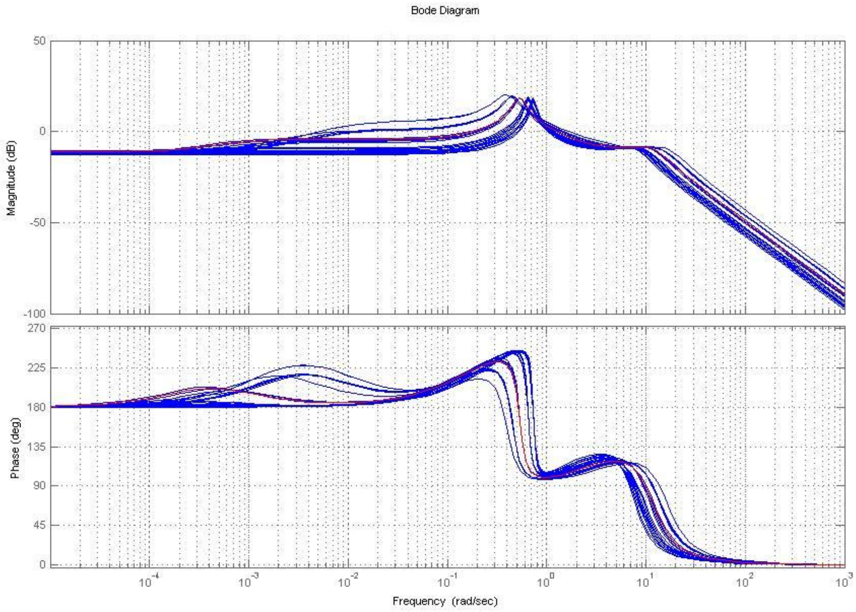


**Figure 6.13: A bode plot of the pitch loop.**

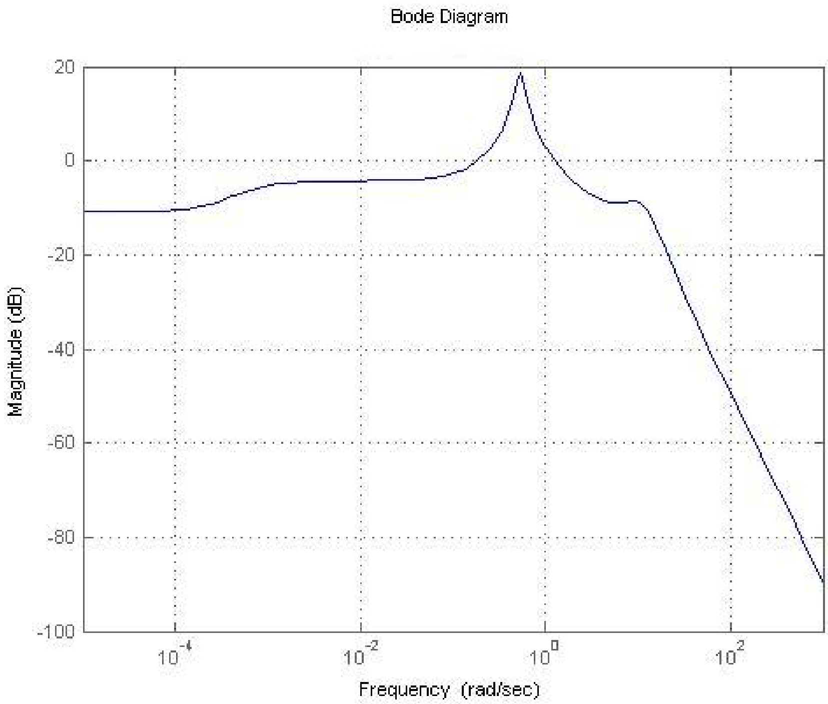


**Figure 6.14: The Bode plot of the magnitude of the initial nominal plant selected for the pitch control loop.**

This would mean that the compensator would contain highly underdamped zeros which cancel the underdamped poles. This a problematic as these poles will only be canceled for the nominal loop transmission, all the other loop transmissions due to the non-nominal plants in the plant set will contain both highly underdamped poles and zeros.

The nominal plant will have to be modified by removing the highly underdamped poles and replacing them with poles with a higher damping ratio but with the same natural frequency. The damping ratio chosen for the nominal plant is 0.7. Figure 6.15 and Figure 6.16 show Bode plots of the new nominal plant and the new nominal plant (in red) with the plant set (in blue).

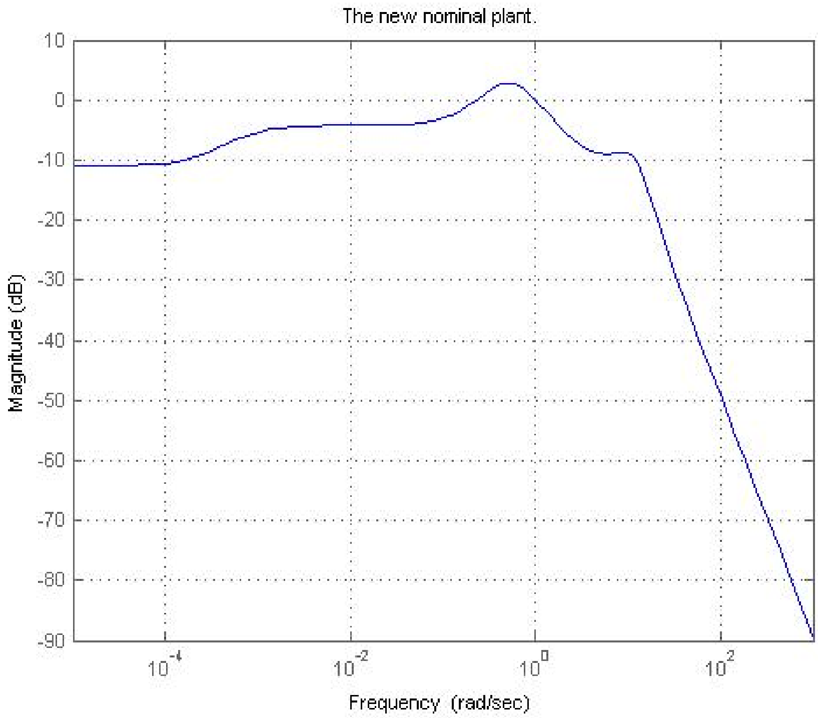


**Figure 6.15: The Bode plot of the new nominal plant.**

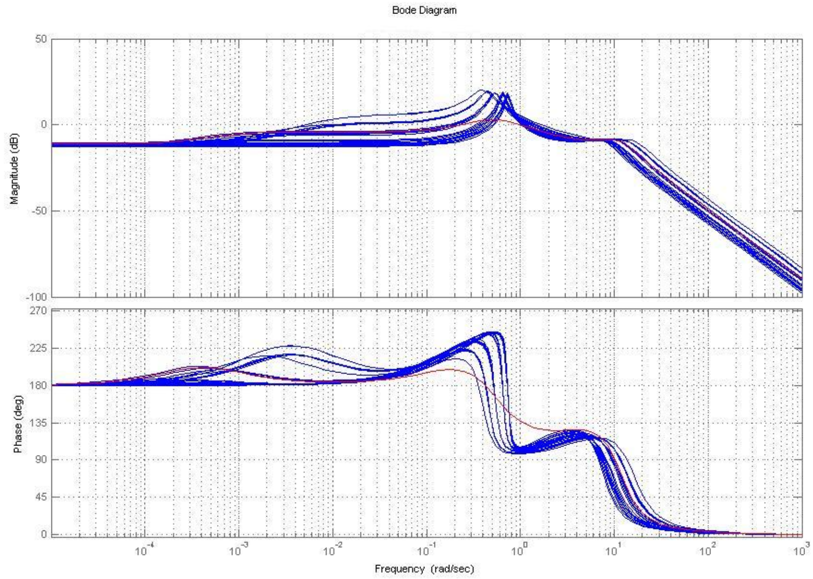


**Figure 6.16: The Bode plot of the new nominal plant and the plant set.**

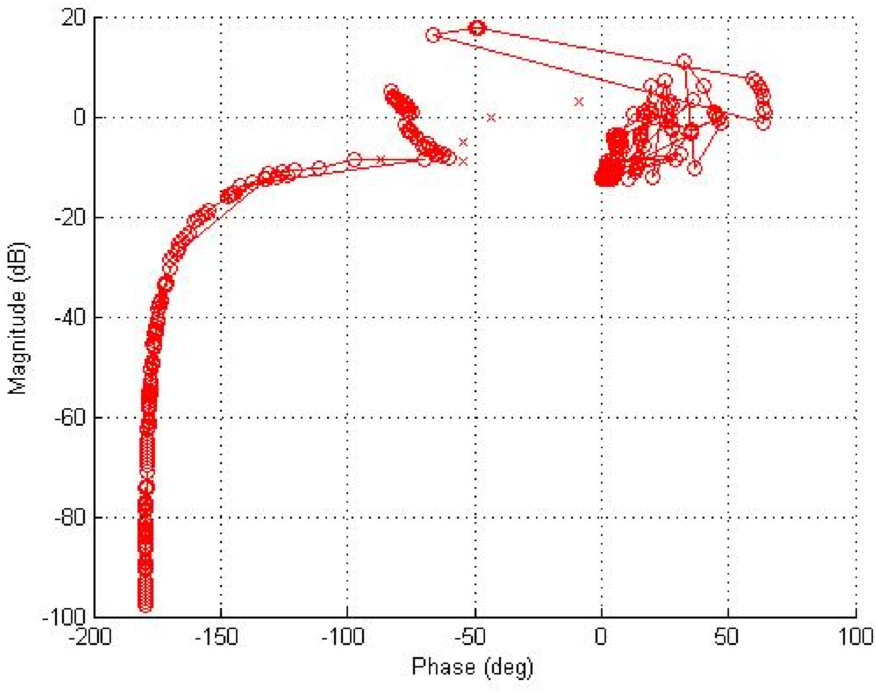


**Figure 6.17: The plant templates for the pitch control loop.**

Figure 6.17 shows a Nichols plot of the plant templates at a number of frequencies. Note that the for the frequencies around the frequency of the highly underdamped poles the nominal plant is not contained within the plant templates. The nominal plant is represented by an 'x'. The 'o' represent the non-nominal plants in the plant set that lie on the boundary.

Now that the plant templates have been generated, the frequency domain closed loop specifications can be translated into the bounds on the loop transmission. Figure 6.18 shows a Nichols plot of the stability bounds calculated from the 1 dB resonant peak specification calculated in the previous step. These stability bounds were calculated using the method discussed in section 5.1.2. The code used to generate the plant templates and the stability bounds are discussed in section 12.4 and section 12.5 respectively.

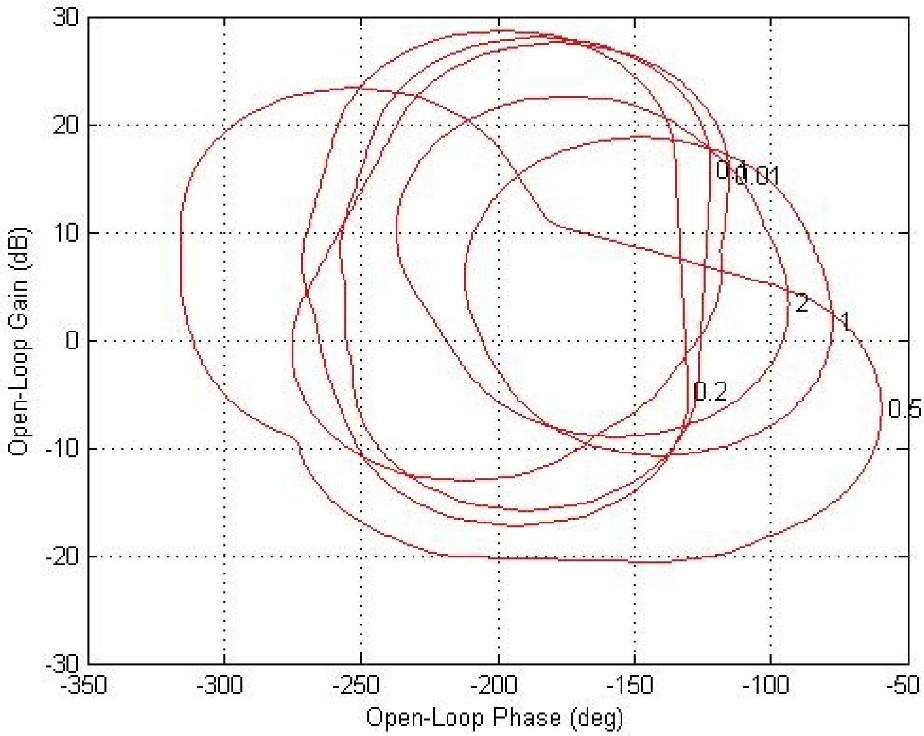


**Figure 6.18: The stability bounds due to the 1 dB resonant peak at a number of frequencies.**

**Step 3: Estimate the loop transmission required to meet the specifications**

*Inputs:* All the bounds placed on the nominal loop transmission.

*Outputs:* The typed Bode ideal characteristic, $H_{TBI}(s)$ .

*Procedure*:

Using the considerations discussed in section 6.3, the parameters required may be estimated. Any parameters that do not have a specification from which they can be estimated, should be chosen using engineering judgment. The values of the parameters do not need to be exact, as the Bode ideal characteristic is being used to help find the loop transmission that should be required, considering all the trade offs.

All of the following parameters of the Bode ideal characteristic must be calculated:

$M_0$ the low frequency gain

$\omega_0$ the maximum frequency of the low frequency gain

$\Phi_{PM}$ the phase margin

$M_{GM}$ the gain margin

$e$ the pole-zero excess

$k$ the system type

From which the typed Bode ideal characteristic, $H_{TBI}(s)$, can be calculated as discussed in section 6.2 and 6.3.1.1.1.

The first parameter to be estimated is the system type, $k$. As there is no specification on the accuracy to which the pitch of the UAV must be controlled, a type 1 system will be selected as the initial estimate of the desired loop transmission, as this will ensure that there will be a zero steady state error to a constant pitch command. The accuracy will therefore not need to be considered directly. Only the settling time of the response will need to be checked to insure that the pitch of the UAV takes a reasonable time to get close to the steady state pitch.

Now that the system type has been selected the low frequency gain, $M_0$, can be estimated. The type of the system chosen has a phase of -90° at DC. By considering the stability bounds calculated in the previous step, see Figure 6.18, it can be seen some of the stability bounds at the frequencies of 0.5, 1, 2, 5, 10 rad/s have a maximum phase greater than -90°. As these frequencies are expected to lie around the crossover frequency, the value of $M_0$ should be high enough to prevent the phases for the frequencies lower than the maximum frequency of the low frequency gain, $\omega_0$, from falling within these stability bounds. A gain of $M_0=20$ dB will therefore be selected.

The desirable cross-over frequency will be approximately 2 rad/s, which is close to the bandwidth of the plant, see Figure 6.16. This frequency is desirable as it will prevent the loop compensator calculated from having to have high gain near the selected cross-over frequency in order to extend the bandwidth. Assuming the phase of the nominal loop transmission during the roll-off of the magnitude to cross-over will be less than -90°, the magnitude will roll-off at less than 20 dB/decade. So the value of $\omega_0$ should be approximately 10 times smaller than the desired cross-over frequency, which will give a value of approximately 0.1 rad/s.

The phase margin, $\Phi_{PM}$, can now be calculated. The specification that will determine this parameter is the resonant peak specification. As discussed in section 6.3.2.3, the bounds on the loop transmission due to this specification are a set of QFT stability bounds. The phase margin selected should be the minimum phase margin that meets all the stability bounds. The phase margin will lie between approximately 180°-135°=45° and 180°-60°=120°, which are 180° plus the maximum values of the stability bounds which have the minimum maximum phase and the maximum maximum phase respectively. If this calculation is done by hand, a phase margin will be selected between these phases and checked to see if they do not violate the stability bounds. The designer will then need to iterate until the minimum phase margin which meets these bounds is found. In this case the ***calcPhaseMargin*** function of the ***BodeCutoff*** class was used to find the phase margin. See section 12.9.3.7 for a description of this code. A phase margin of $\phi_{PM}=102^o$ was calculated.

The Bode cut-off part of the Bode ideal characteristic with the phase margin calculated is shown in Figure 6.19.

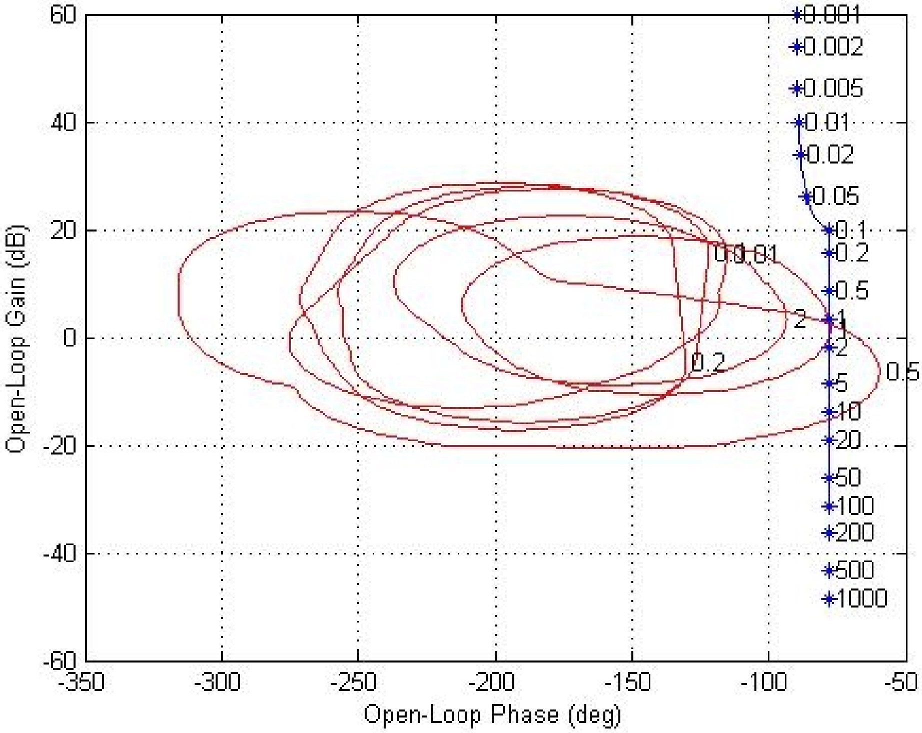


**Figure 6.19: The Bode cut-off part of the Bode ideal characteristic plotted with the stability bounds it must meet.**

The gain margin, $M_{GM}$, must be selected so that the Bode ideal characteristic passes below the stability bounds. As disturbances due to changes in the throttle setting of the engine are expected, the gain margin must be selected such that the disturbances do not cause unacceptable oscillations in pitch angle. So rather than passing below the stability bounds generated for the command input, the stability bounds generated for the disturbance input shall be used. As there is no specification for the disturbance undershoot, the overshoot specification for the command input will be used, so that the disturbance input will not cause the output to oscillate more than is acceptable. As discussed in section 5.2, the resonant peak specification for a disturbance makes use of the inverse Nichols chart, which is simply the Nichols chart rotated by 180° around the point 0 dB and – 180°. This means that if the gain margin is set to the inverse of the maximum magnitude of the stability bounds for the command input, the sensitivity function (which gives the response to a disturbance entering at the output) will have a maximum resonant peak of 1 dB. A gain margin of $M_{GM}=30$ dB will therefore be used.

The pole zero-excess of the plant is 2, so a pole zero excess of $e=3$ will be selected for the nominal loop transmission which will allow the compensator to have a pole zero-excess of 1.

These parameters will give a Bode ideal characteristic which has the Bode plot shown in Figure 6.20.

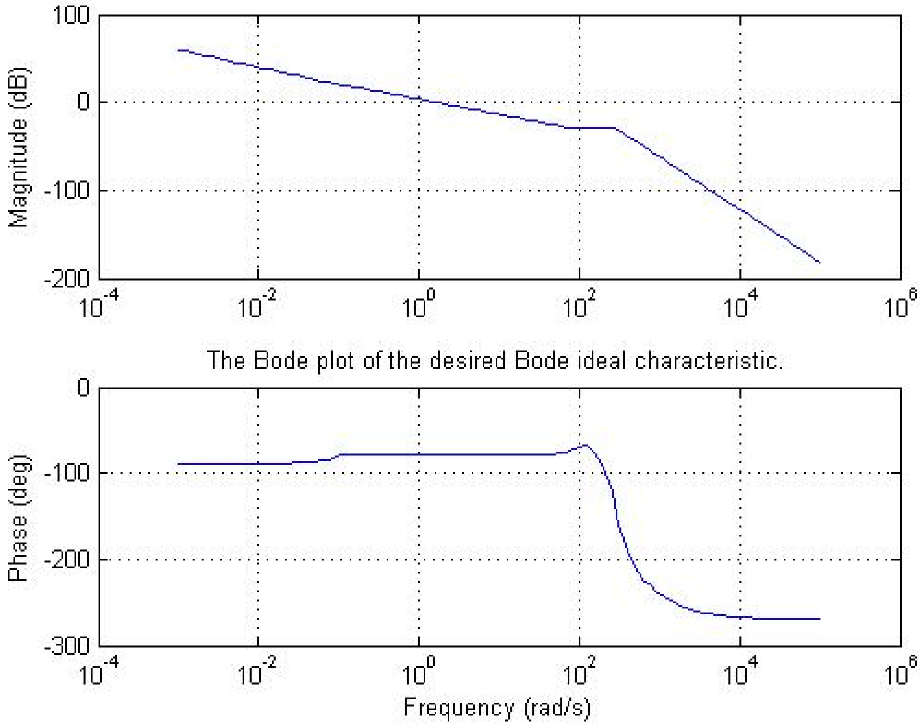


**Figure 6.20: The desired Bode Ideal characteristic.**

Figure 6.21 shows a Bode plot of the fitted nominal loop transmission, $L_0(s)$ , and the typed Bode ideal function, $H_{TBI}(s)$ . Figure 6.21 shows the Bode plot of the fitted nominal loop transmission (in blue) and the typed Bode ideal characteristic (in red). Figure 6.22 shows the Nichols chart plots of the fitted nominal loop transmission (in red) and the typed Bode ideal characteristic (in blue).

The routine ***fit*** function of the ***BodeIdeal*** class was used to fit a rational transfer function to the Bode ideal characteristic ( See section 12.10.3.2 for details on the routine). The following transfer function was obtained

$$L_0(s)=\frac{b_5\cdot s^5+b_4\cdot s^4+b_3\cdot s^3+b_2\cdot s^2+b_1\cdot s+b_0}{s^8+a_7\cdot s^7+a_6\cdot s^6+a_5\cdot s^5+a_4\cdot s^4+a_3\cdot s^3+a_2\cdot s^2+a_1\cdot s}$$

where:

$b_0 = 4.01 \times 10^6$ $a_1 = 5.959 \times 10^6$
$b_1 = 7.728 \times 10^7$ $a_2 = 7.214 \times 10^7$
$b_2 = 5.717 \times 10^8$ $a_3 = 6.084 \times 10^8$
$b_3 = 2.735 \times 10^9$ $a_4 = 2.208 \times 10^9$
$b_4 = 1.587 \times 10^8$ $a_5 = 3.641 \times 10^7$
$b_5 = 1.732 \times 10^6$ $a_6 = 1.899 \times 10^5$
$a_7 = 545.3$

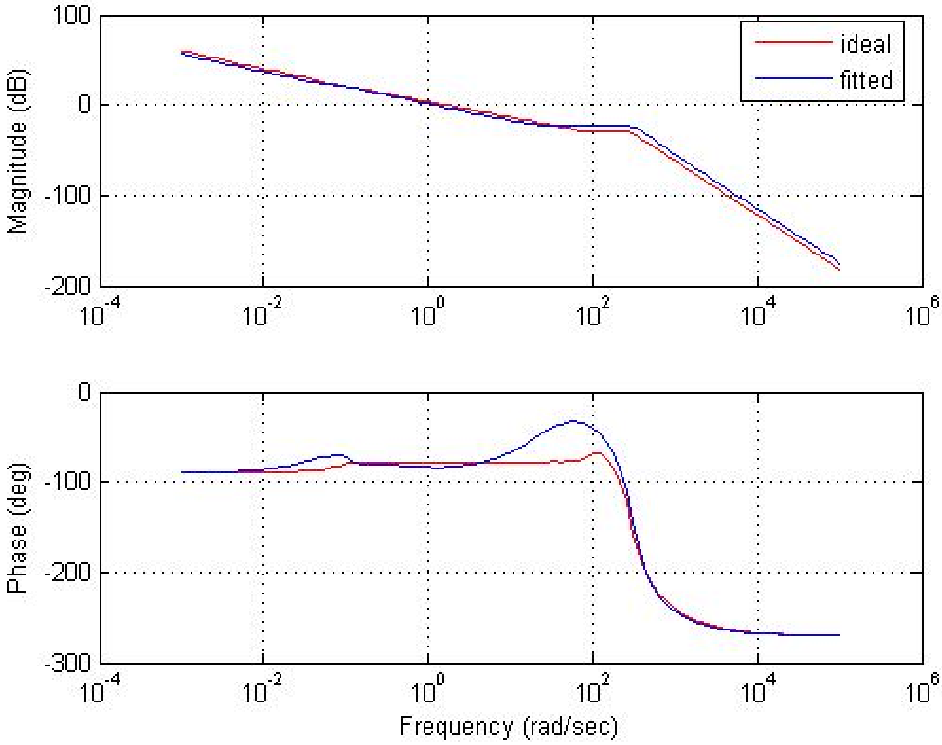


**Figure 6.21: The Bode plot of the rational transfer function fitted to the desired Bode Ideal Characteristic.**

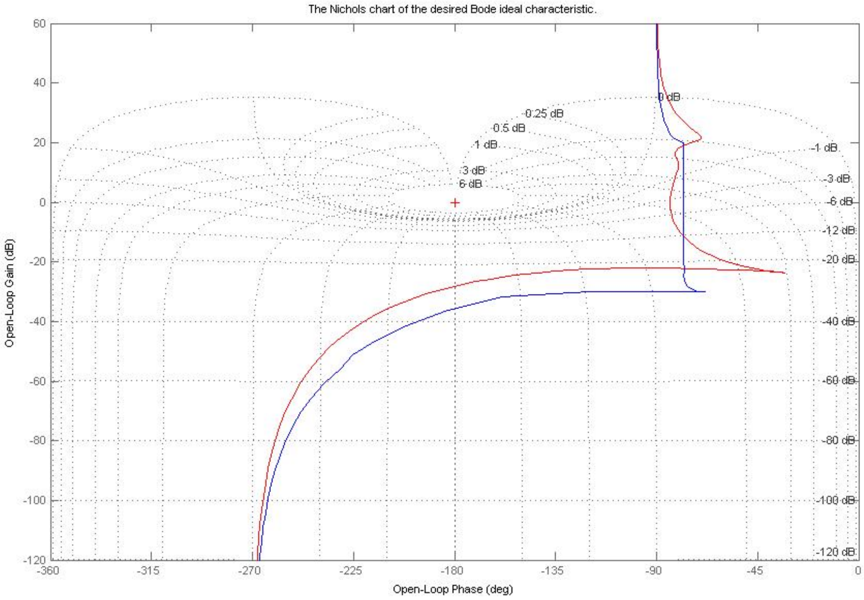


**Figure 6.22: The Nichols chart of the rational transfer function fitted to the desired Bode Ideal Characteristic.**

**Step 5: Calculate the controller**

*Inputs:* The nominal plant, $P_0(s)$ .

The nominal loop transmission, $L_0(s)$ .

*Output:* The loop compensator, $G(s)$ , that gives the nominal loop transmission, $L_0(s)$ , for the nominal plant $P_0(s)$ .

*Procedure:*

The controller is calculated using

$$G(s)=\frac{L_0(s)}{P_0(s)}$$

Figure 6.23 shows a Bode plot of the loop compensator calculated.

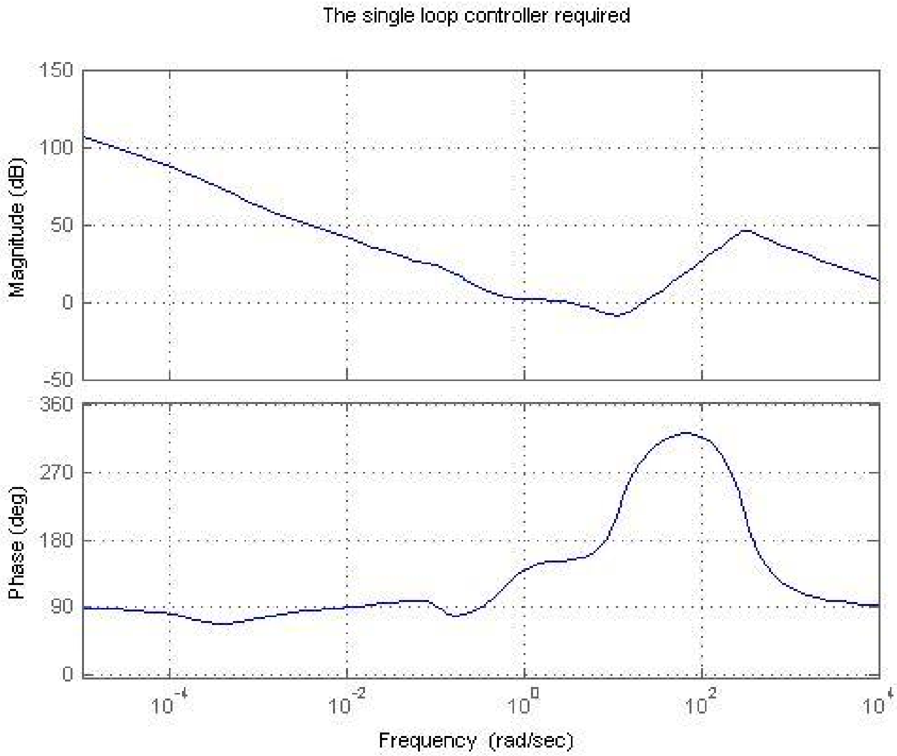


**Figure 6.23: The Bode plot of the required controller.**

**Step 6: Calculate the loop transmission set**

*Inputs:* The loop compensator $G(s)$ .
The plant set, $\tilde{P}(s)$ .

*Output:* The loop transmission set, $\tilde{L}(s)$ .

*Procedure:*
For each plant in the plant set, the loop transmission set is calculated using

$$L_i(s) = P_i(s) G(s)$$

where $i = 1,2,3,\ldots,n$
$n$ is the number of plants in the plant set.

Figure 6.24 shows a Nichols chart plot of the loop transmission set (in blue) and the m-circle related to the resonant peak specification of 1 dB (in red) as calculated in step 1.

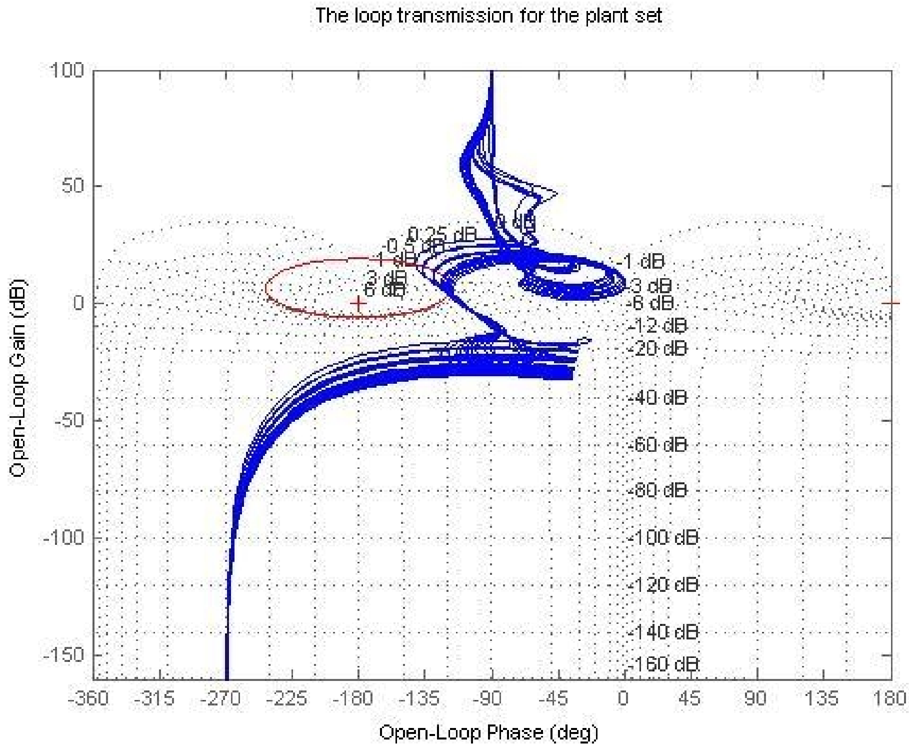


**Figure 6.24: The Nichols chart of the loop transmission.**

Note that due to the inexact fit of the nominal loop transmission, $L_0(s)$ , to the Bode ideal characteristic, the 1 dB resonant peak specification is not met. However, as the Bode ideal characteristic is used only to evaluate prototype models so that a more complicated design can be done, this is not a problem.

Figure 6.25 shows the Bode plot of the loop transmission set.

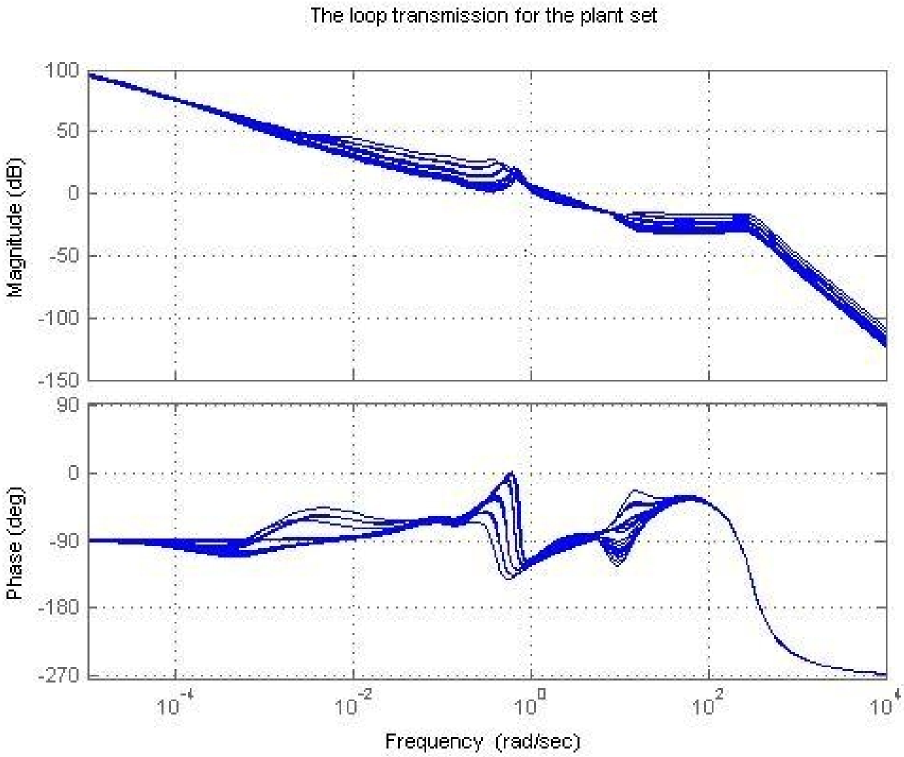


**Figure 6.25: The Bode plot of the loop transmission set.**

**Step 7: Calculate the closed loop transfer function set**

*Input:* The loop transmission set, $\tilde{L}(s)$ .

*Output:* The closed loop frequency response set, $\tilde{T}(s)$ .

*Procedure:*

For each plant in the plant set, the closed loop transfer function set is calculated using

$$T_i(s)=\frac{L_i(s)}{1+L_i(s)}$$

where $i=1,2,3,\ldots,n$

$n$ is the number of plants in the plant set.

Figure 6.26 shows a Bode plot of the closed loop response set.

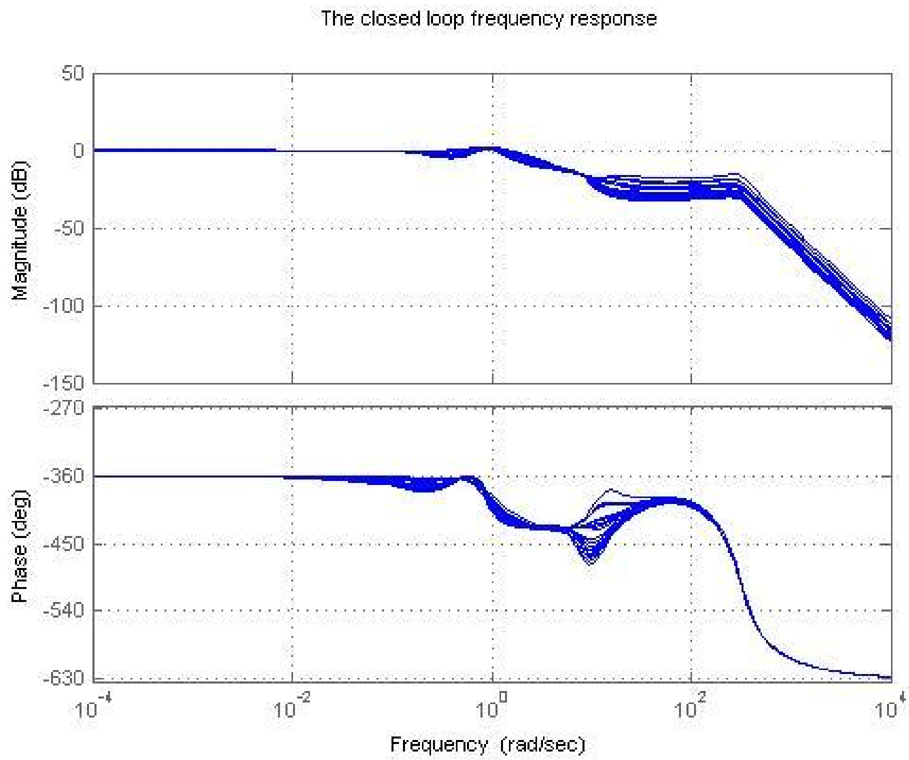


**Figure 6.26: The closed loop frequency response.**

**Step 8: Calculate the closed loop time response set**

*Input:* The closed loop frequency response set, $\tilde{T}(s)$ .

*Output:* The set of system responses to the specified command input. $\tilde{Y}_C(s)$ .

*Procedure:*

For each closed loop transfer function in the closed loop transfer function set, the closed loop response set is calculated using

$$Y_{C_i}(s)=T_i(s)C(s)$$

where $i=1,2,3,\ldots,n$

$n$ is the number of plants in the plant set.

And the time domain response set is given by

$$y_{C_i}(t) = \mathscr{L}^{-1}\{ Y_{C_i}(s) \}$$

Where $\mathscr{L}^{-1}$ denotes the inverse Laplace transform.

The worst case command input that the pitch control system can encounter is a step, as it commands an instantaneous change in pitch which must be held steady thereafter. The command input that the closed loop time response will be calculated for is therefore chosen to be a step input. Figure 6.27 shows the step response calculated.

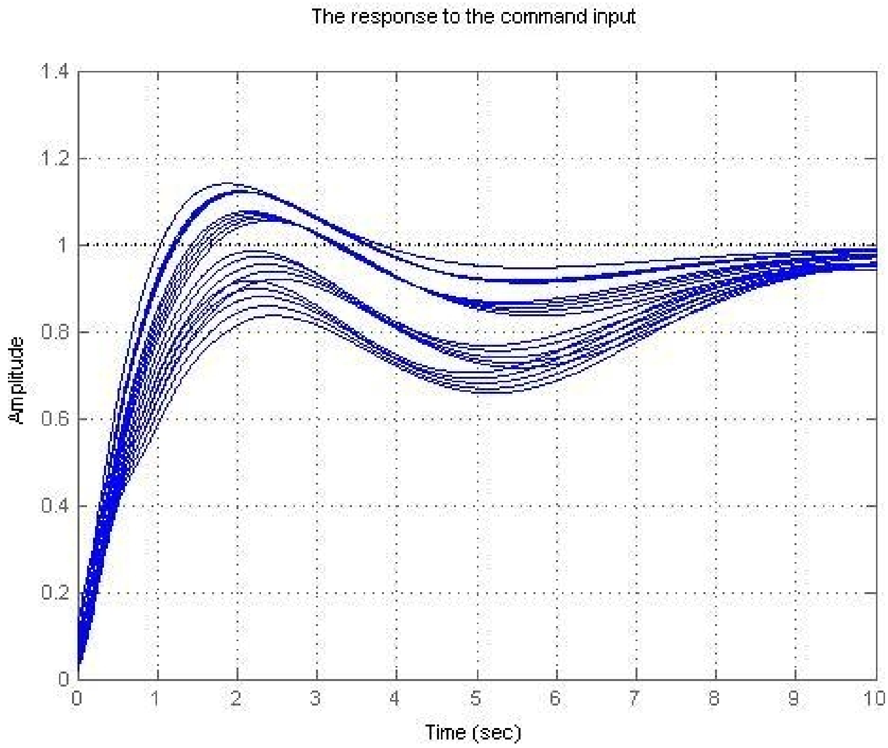


**Figure 6.27: The response to the command input.**

**Step 9: Calculate the plant input set**

*Input:* The loop compensator, $\tilde{G}(s)$ .

The loop transmission set, $\tilde{L}(s)$ .

The plant input, $C(s)$ .

*Output:* The plant input set in the frequency domain $\tilde{U}_C(s)$ .
The plant input set in the time domain $\tilde{u}_C(s)$ .

*Procedure:*

For each closed loop transfer function in the closed loop transfer function set, the closed loop response  set is calculated using

$$U_{C_i}(s) = \frac{G(s)}{1 + L_i(s)} \cdot C(s)$$

where $i = 1,2,3,\ldots,n$

$n$ is the number of plants in the plant set.

And the time domain response set is given by

$$u_{C_i}(t) = \mathscr{L}^{-1}\{ U_{C_i}(s) \}$$

Figure 6.28 and Figure 6.29 shows a Bode plot of the frequency response of the plant input set and the time domain response of the plant input set respectively.

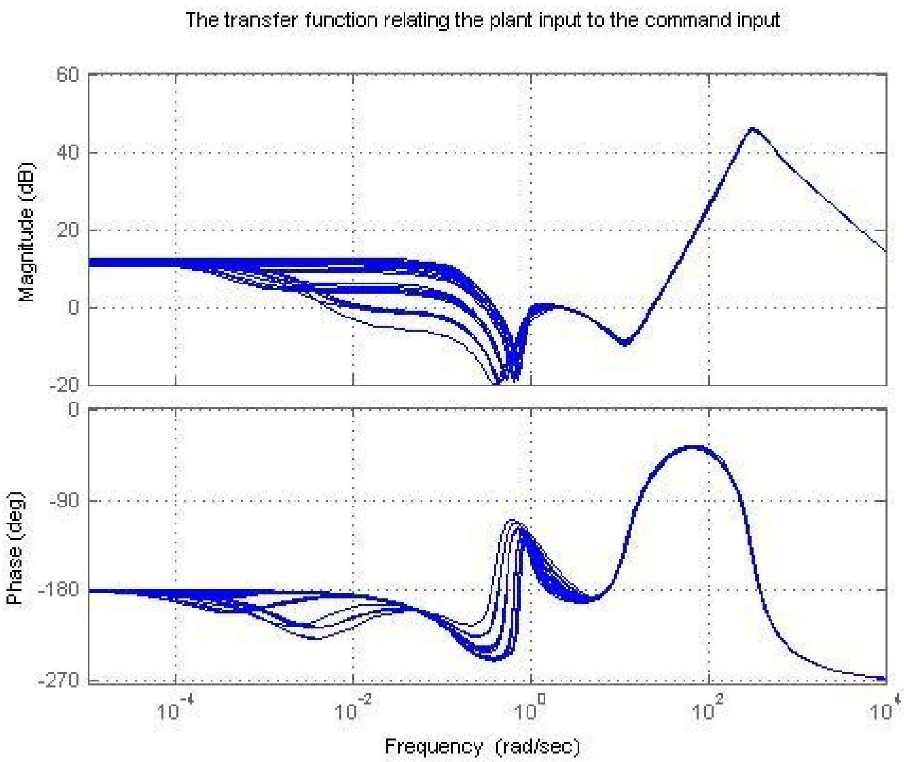


**Figure 6.28: The transfer function relating the plant input to the command input.**

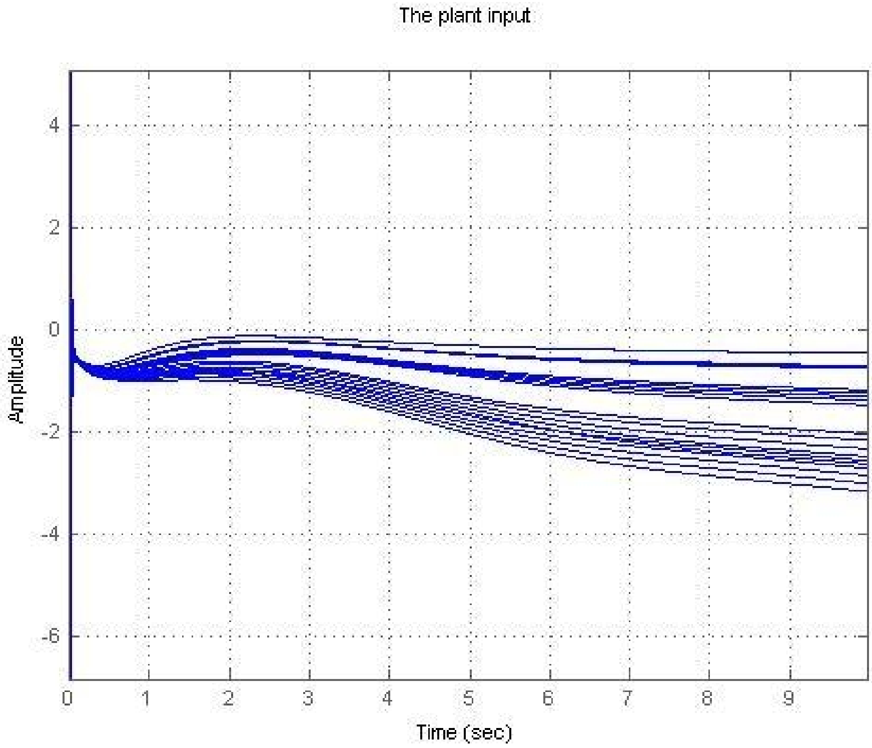


**Figure 6.29: The time response of the plant input.**

**Step 10: Re-iterate the design process in order to optimize the prototype models**

*Input:* All the inputs used in steps 1 to 9.

*Output:* All the outputs generated in steps 1 to 9.

*Procedure:*

The prototype models generated and their responses to the plant input are evaluated in terms of the system performance. The parameters of the Bode ideal characteristic estimated in step 3 are then adjusted to try and improve the systems performance.

Steps 1 to 9 are then recalculated in order to generate a new set of prototype models. These prototype models and their respective responses are compared with the original prototype models to see if the systems performance has been improved. The parameters of the Bode ideal characteristic are adjusted until a satisfactory system performance is achieved.

Now that an initial set of prototype models has been generated, their effectiveness in meeting the specifications must be considered. As the only specification was on the overshoot of the systems response to the command input, it can be seen from Figure 6.27 that the specification is almost met for all the plants in the plant set. However there is still a large variation in the response to the command input. This is highly undesirable and will need to be rectified. The step response also takes a long time, approximately 9 seconds, to settle to an acceptable error of about 10%.

The maximum value, ignoring the high frequency transients, at the plant input is about -3 times the step inputs amplitude (see Figure 6.29), which gives an elevator command of about -30$^{o}$. This is an acceptable command input level. Note that the command input initially has a large oscillation near zero. This is due to the piecewise linear construction of the Bode ideal characteristic. These high frequency transients will not be considered as the Bode ideal characteristic is only being used to get the prototype models and not design the actual compensator. When the design of the compensator is performed these transients will be eliminated by having a smooth loop transmission around the cross-over frequency.

The prototype models obtained therefore imply that the value estimated for $\omega_0$ may have been too conservative and can be increased. This will increase the speed of response of the pitch loop control and help reduce the variation in the step response. So the effect of $\omega_0$ needs to be investigated.

The value of $\omega_0$ was varied until the systems response to a step pitch command is satisfactory. The value of $\omega_0$ which gives a reasonable step response was found to be 0.8.

Note that most pitch control loops make use of gain scheduling based on the aircrafts airspeed and altitude with the intention of reducing the plant uncertainty. So if $\omega_0$ cannot be increased enough to deal with the plant uncertainty due the constraints on the elevator command, gain scheduling may need to be added and the prototype models reevaluated with the gain scheduling added to the plant model.

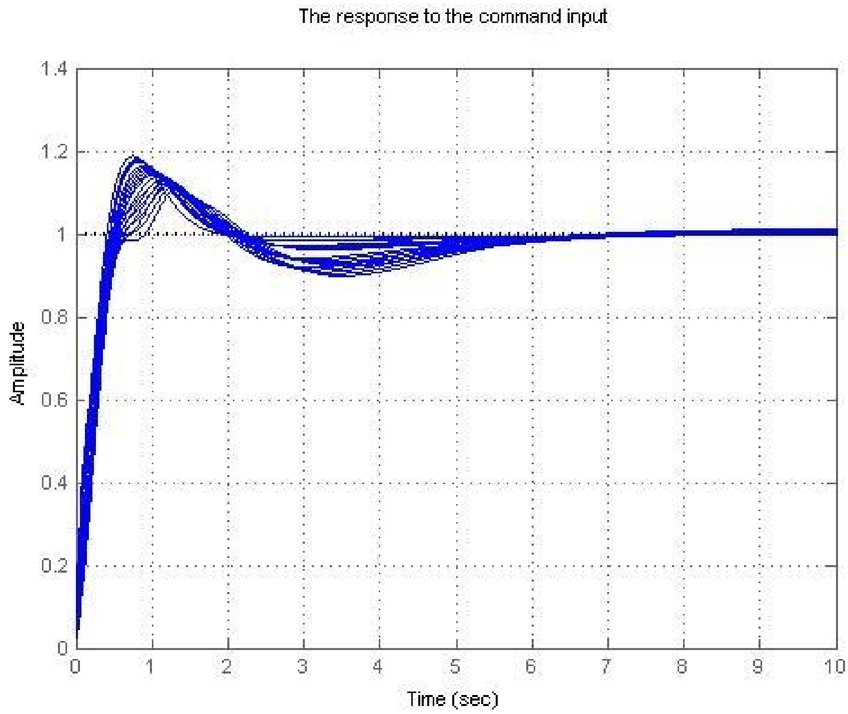


**Figure 6.30: The final response to the command input.**

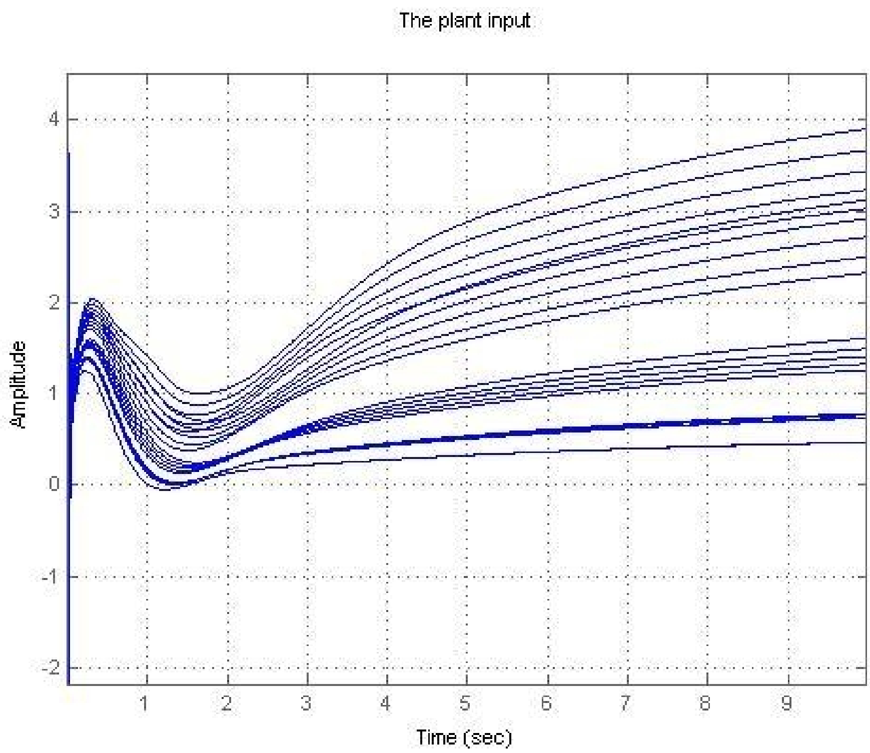


**Figure 6.31: The final time response of the plant input.**

**Step 11: Generate the bounds needed for the control algorithm design process**

*Input:* The closed loop response set, $\tilde{T}(s)$ , or the sensitivity function set, $\tilde{S}(s)$ .

*Output:* The bounds needed as an input to the design algorithm used to design the control algorithm.

*Procedure:*

The procedure used depends on the design algorithm that is used to design the control algorithm.

Now that a reasonable set of prototype models has been generated, these models can be converted into bounds which are required by the design algorithm used to design the control algorithm.

If quantitative feedback theory is to be used to design for a single degree of freedom system, the bounds required are simply the maximum and minimum magnitudes of the closed loop response set, $\tilde{T}(s)$ , at the frequencies of interest. Figure 6.32 shows the upper and lower bounds on the closed loop magnitudes (in red). These bounds are calculated from the set of prototype models of the loop transmission, which means that they have a very high order. They can be simplified by smoothing them before using them in the design process, shown in blue in Figure 6.32.

Even if the smoothed bounds still have a high order, the QFT design process will allow the designer to limit the order of the compensator obtained.

If quantitative feedback theory is to be used to design for a two degree of freedom system, a bound on the maximum uncertainty of the closed loop response must be generated. This bound is obtained from the upper and lower bounds on the closed loop magnitudes ( $T_{MAX}(s)$ and $T_{MIN}(s)$ respectively) in the following manner

$$B_{UNC}(s)=\frac{T_{MAX}(s)}{T_{MIN}(s)}$$

The uncertainty bound obtained is plotted in blue in Figure 6.33, and the smoothed version of these bounds are plotted in red.

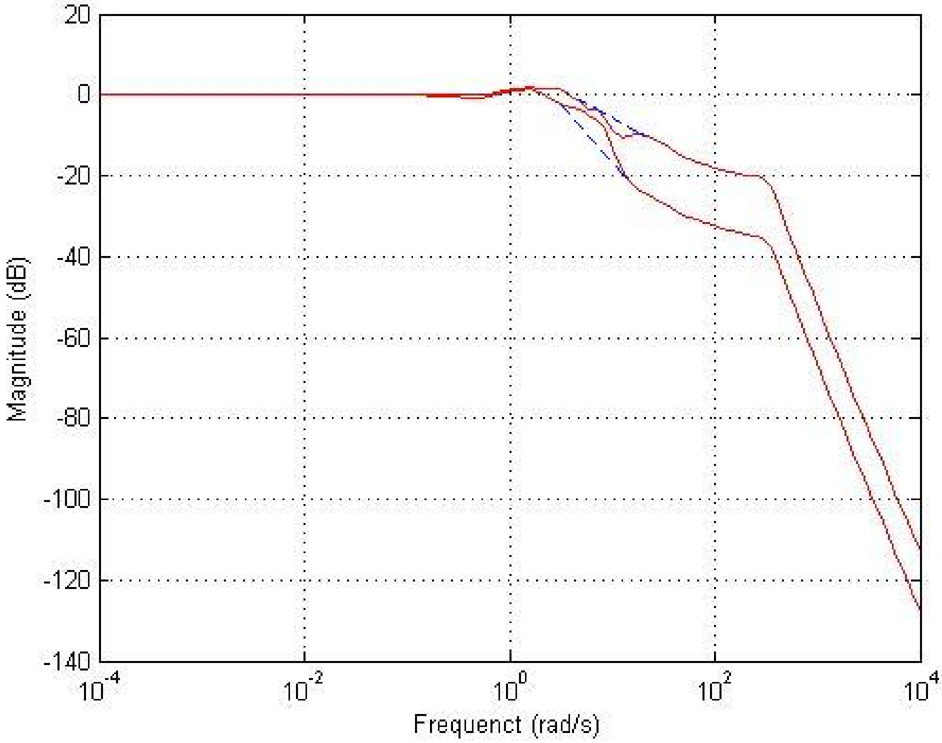


**Figure 6.32: Upper and lower bounds on the systems' closed loop response for single degree of freedom QFT designs.**

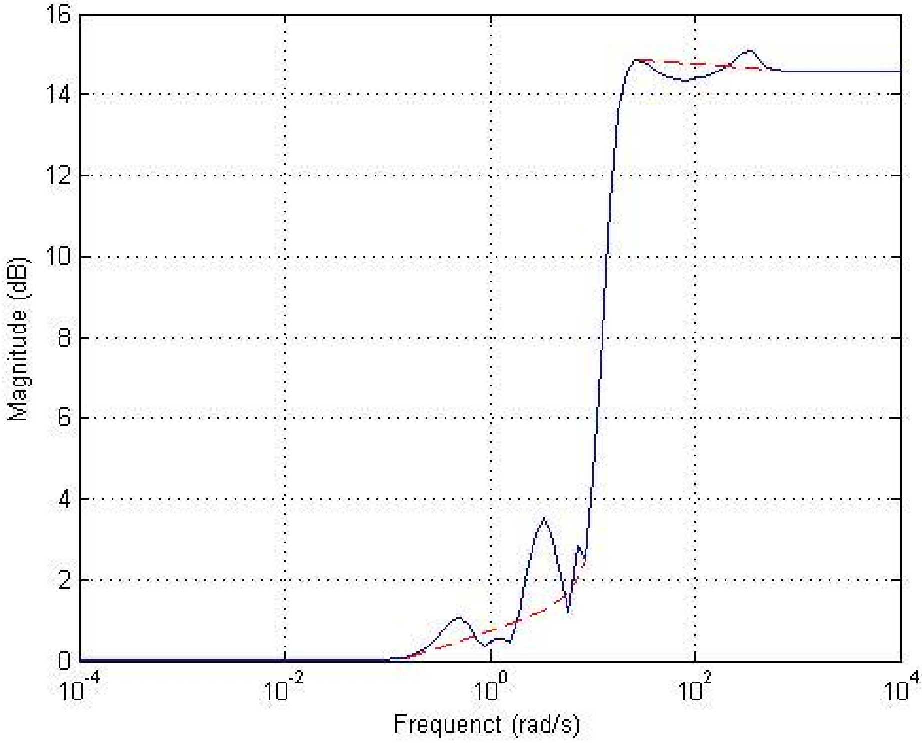


**Figure 6.33: Uncertainty bounds for two degree of freedom QFT designs**

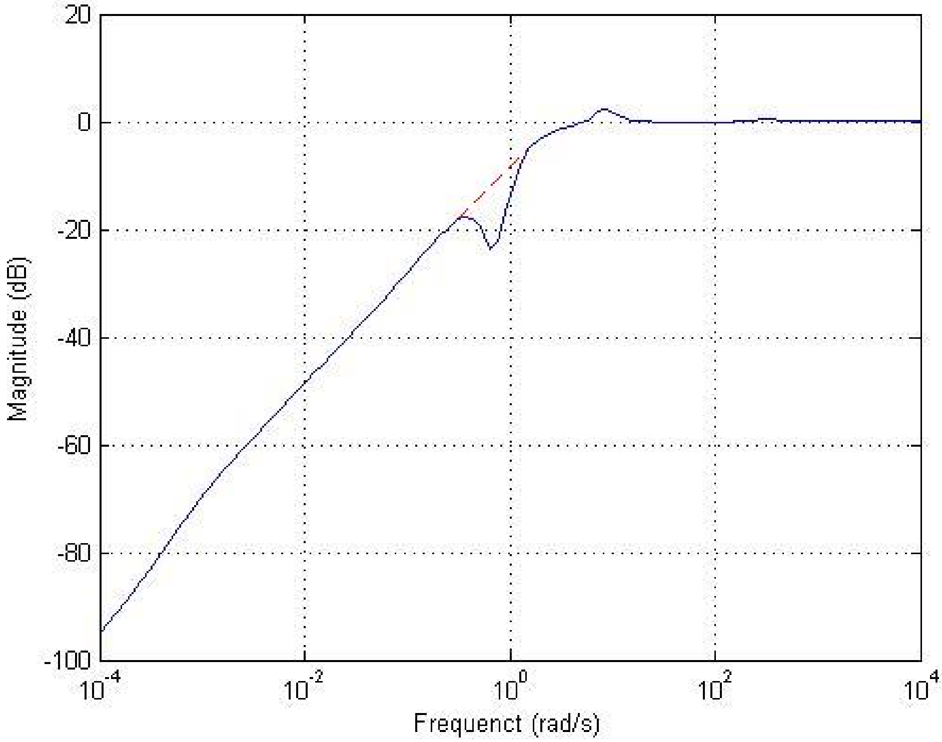


**Figure 6.34: Upper bound on the sensitivity function for use in H-infinity design.**

If H-infinity optimal control is to be used to design for the control algorithm, the performance weighting function needs to be generated from the prototype models. The H-infinity bound placed on the nominal performance normally takes the following form.

$$\|W_1 \cdot S\|_\infty < 1$$

which can be written as

$$\|S\|_\infty < \frac{1}{W_1}$$

So the performance weighting function, $W_1$ , is related to the upper bound of the sensitivity function by

$$W_1 = \frac{1}{S_{MAX}}$$

So for this example the upper bound on the sensitivity function is plotted in blue in Figure 6.34, and the simplified version in red.

Note that all the bounds obtained give a consistent set of bounds while taking into account the plants' characteristics.

# 7 Designing for unstable plants

## 7.1 The suitability of the Bode ideal characteristic for the loop transmission for an unstable plant

As the Bode ideal characteristic can be successfully used for describing the desired loop transmission for a control system with a stable minimum phase plant, the new question is can it be used for the loop transmission of an unstable minimum phase plant?

The simplest way to answer this question is to look at the controller that will be required if the Bode ideal characteristic is used for a plant with unstable poles. So if the nominal plant can be described by the following transfer function:

$$P_0(s)=\frac{P_{SMP}(s)}{D_u(s)}$$

where:

$P_{SMP}(s)$ is the stable minimum phase part of the plant.

$D_u(s)$ are the unstable poles of the plant.

The controller obtained if the loop transmission is the typed Bode ideal characteristic, $H_{TBI}(s)$ , will be:

$$G(s)=\frac{H_{TBI}(s)}{P_0(s)}=\frac{H_{TBI}(s)\cdot D_u(s)}{P_{SMP}(s)}$$

Now, as the typed Bode ideal characteristic has no unstable poles, the controller will have a non-minimum phase zero that cancels out the nominal plant's unstable poles. Although, mathematically this is a solution to the problem, in practice this is not a solution. As the exact frequencies and damping ratios of the unstable poles are not known, the cancellation will not be exact. Therefore, the control system will still contain the unstable poles and the control system may still be unstable.

The loop transmission must therefore contain the unstable poles so that the controller obtained does not cancel them.

## 7.2 The properties of unstable poles

The previous chapter discussed a method for designing for the stable minimum phase part of the plant. What needs to be considered now is the effect the unstable minimum phase part will impact on the design?

In order to answer this question the properties of unstable poles needs to be evaluated and their effect on the desired loop transmission needs to be understood.

### 7.2.1 Unstable real poles

Consider an unstable real pole of the form:

$$H(s)=\frac{\alpha}{s-\alpha}$$

where $\alpha$ is positive, the magnitude and phase of the unstable pole versus frequency is given by

$$|H(j\cdot\omega)|=\frac{\alpha}{\sqrt{\omega^2+\alpha^2}}$$

$$\sphericalangle H(j\cdot\omega)=-\tan^{-1}\left(\frac{\omega}{-\alpha}\right)$$

An unstable pole at a frequency of 1 rad/s will have the Bode plot shown in Figure 7.1. From the plot it can be seen that the phase of a real unstable pole starts at a phase of $-180^{o}$ at low frequencies and increases to a phase of $-90^{o}$ at high frequencies.

Therefore an unstable real pole has extra phase lag at low frequencies than its stable counterpart and the same phase at high frequencies. The magnitude of an unstable pole is exactly the same as its stable counterpart.

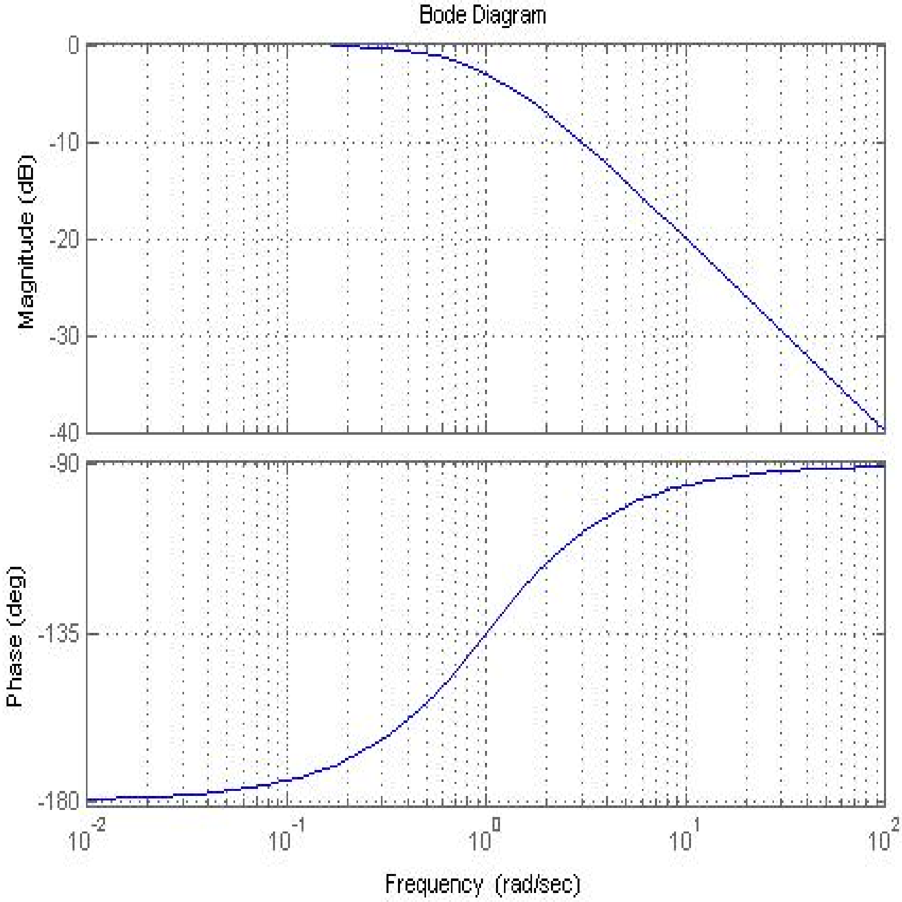


**Figure 7.1: The Bode plot of an unstable pole.**

## 7.2.2 Unstable complex conjugate pole pairs

Consider an unstable complex conjugate pole of the form:

$$H(s)=\frac{\alpha^2+\beta^2}{(s-\alpha+j\cdot\beta)\cdot(s-\alpha-j\cdot\beta)}=\frac{\alpha^2+\beta^2}{s^2-2\cdot\alpha\cdot s+(\alpha^2+\beta^2)}$$

where $\alpha$ and $\beta$ are positive, the magnitude and phase of the unstable pole versus frequency is given by

$$|H(j\cdot\omega)|=\frac{\alpha^2+\beta^2}{\sqrt{[(\alpha^2+\beta^2)-\omega^2]^2+4\cdot\alpha^2\cdot\omega^2}}$$

$$\measuredangle H(j\cdot\omega) = -\tan^{-1}\left(\frac{\beta+\omega}{-\alpha}\right) - \tan^{-1}\left(\frac{\omega-\beta}{-\alpha}\right)$$

An unstable pole at a frequency of 1 rad/s will have the Bode plot shown in Figure 7.2. From the plot it can be seen that the phase of a unstable complex pole pair starts at a phase of $-360^o$ at low frequencies and increases to a phase of $-180^o$ at high frequencies.

Therefore an unstable complex pole pair has extra phase lag at low frequencies than its stable counterpart and the same phase at high frequencies. The magnitude of an unstable complex pole pair is exactly the same as its stable counterpart.

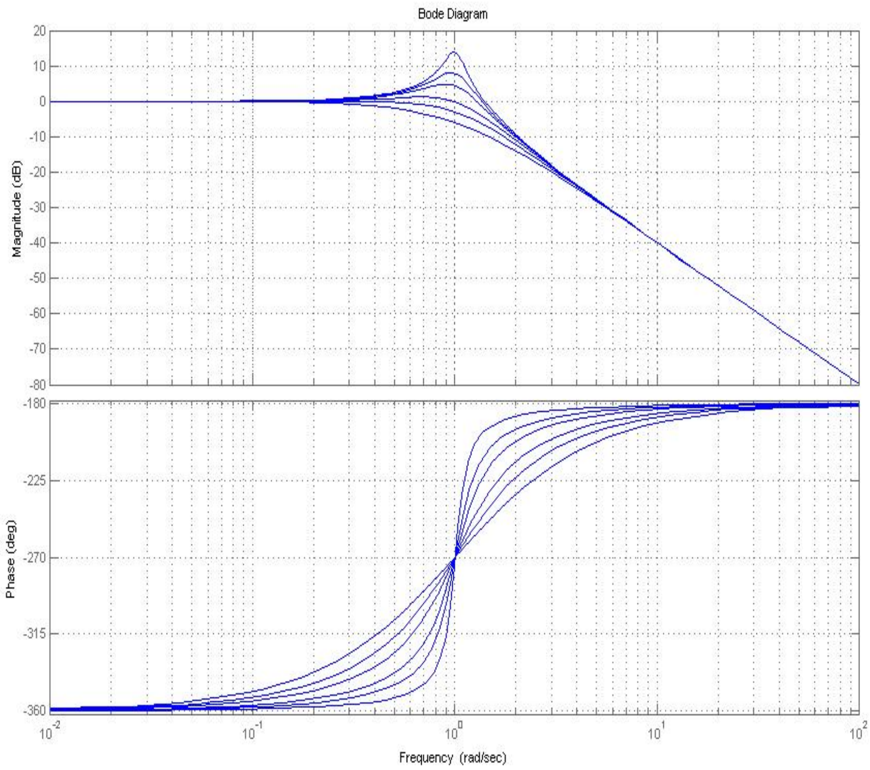


**Figure 7.2: The Bode plot of an unstable complex pole pair.**

### 7.2.3 General properties of unstable poles

For a plant containing only unstable poles, the phase of the plant at 0 rad/s is

$$\measuredangle P(0) = -180^o \cdot n$$

where:

$n$ is the number of unstable poles (complex or real).

The phase of the plant as the frequency tends to infinity is

$$\measuredangle P(\infty) = -90^o \cdot e$$

where:

$e$ is the pole zero excess of the plant.

The magnitude of the plant remains unchanged from a plant which has stable poles instead of the unstable poles at the same frequencies.

## *7.3 Modifying the Bode ideal characteristic*

Incorporating the unstable poles into the loop transmission will now be discussed. It is possible to simply add the unstable poles to the Bode ideal characteristic to obtain the nominal loop transmission. This however has a draw back in that the added unstable poles will not only add extra phase lag at low frequencies, but the magnitude would also be reduced above the frequencies of the unstable poles. A common means of preventing having to deal with a decrease in magnitude with a lead in phase is to define the all-pass function (Horowitz,1993) , which will now be discussed.

### 7.3.1 The all-pass function

The previous discussion shows that the loop transmission must contain the unstable poles. Therefore, the unstable poles must be added to the Bode ideal characteristic. But, this will cause both changes in the magnitude and phase of the typed Bode ideal characteristic.

The all-pass function is defined as

$$H_{AP}(s) = \frac{(s-\bar{p}_1)\cdot(s-\bar{p}_2)\ldots(s-\bar{p}_{n-1})\cdot(s-\bar{p}_n)}{(s+p_1)\cdot(s+p_2)\ldots(s+p_{n-1})\cdot(s+p_n)}$$

where:

$p_{1,} p_{2,} \cdots, p_n$ are the unstable poles of the nominal plant.

Note that the denominator contains the unstable poles of the nominal plant and the numerator their negative complex conjugates. The poles may be complex, but for a practical system the all-pass function will always contain complex-conjugate pairs and not just a single complex pole.

The magnitude of the all-pass function is unity for all frequencies which means that only the phase need be considered.

Figure 7.3 shows the phase of the all-pass function for an unstable real pole and Figure 7.4 the phase of the all-pass function for a number of unstable complex pole pairs with different damping ratios at a natural frequency of 1 rad/s.

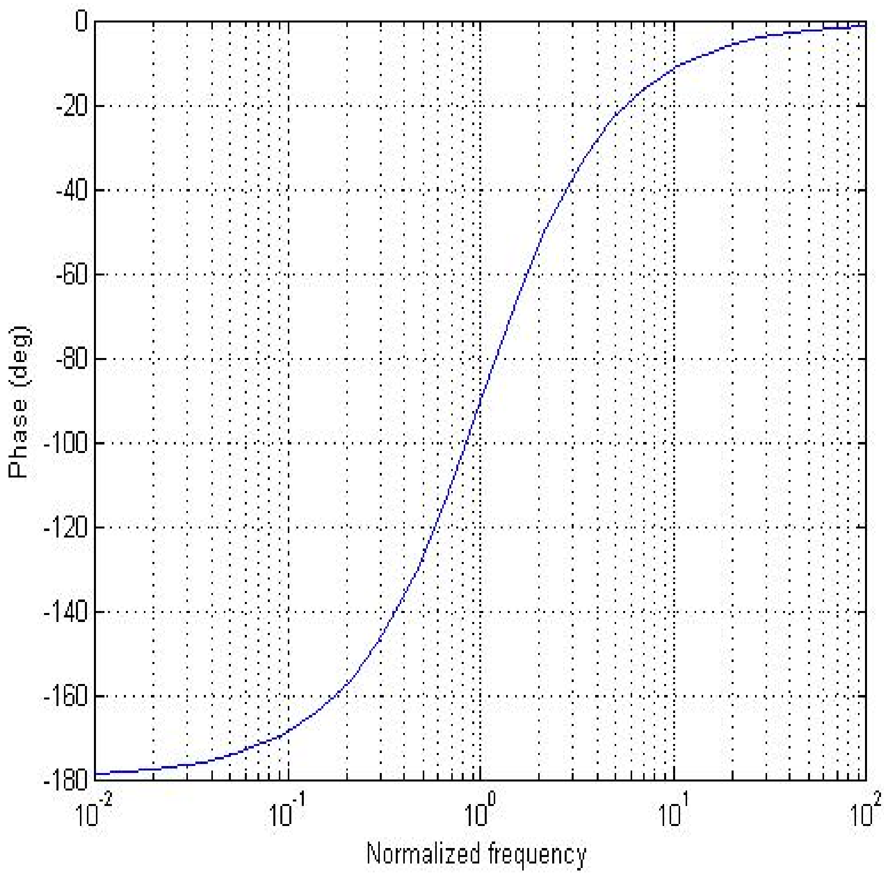


**Figure 7.3: The all-pass function for an unstable real pole.**

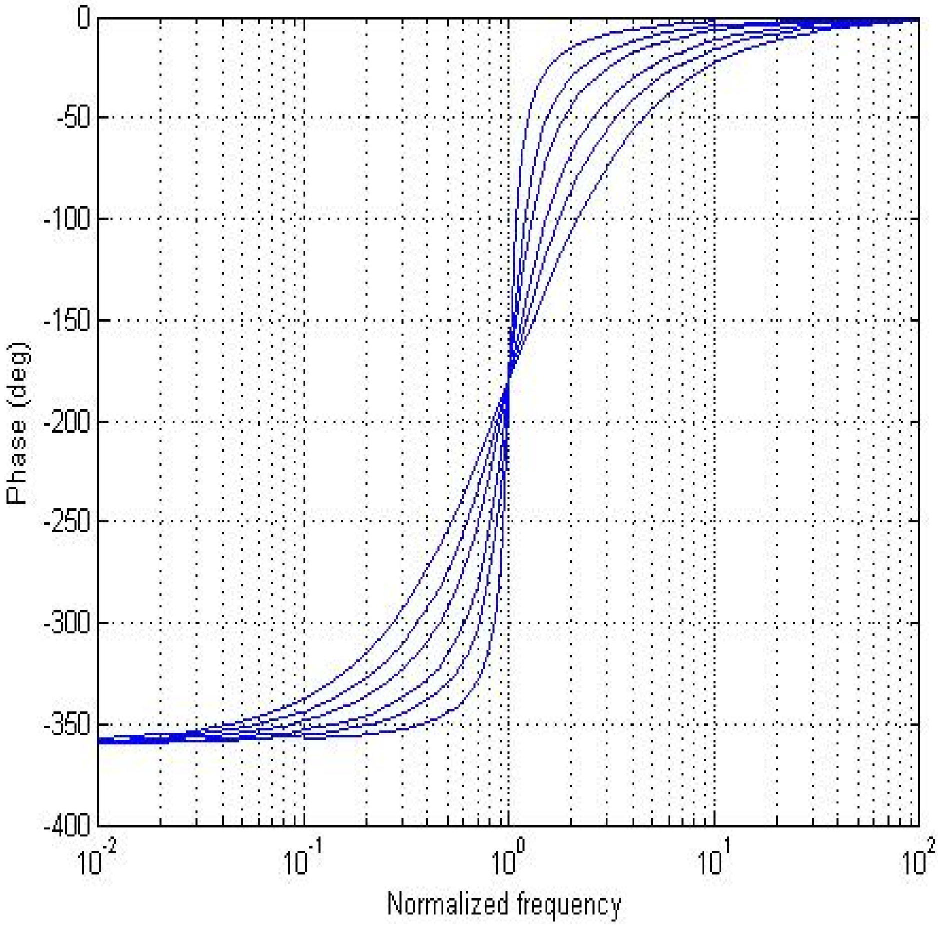


**Figure 7.4: The all-pass function for an unstable complex pole pair.**

## 7.3.2 Modifying the nominal loop transmission to take the unstable poles into account

The complete Bode ideal characteristic modified to allow for a plant with unstable poles is obtained by adding the all-pass function to the Bode ideal characteristic. The nominal loop transmission then becomes

$$|L_0(s)| = |H_{TBI}(s)|$$

$$\sphericalangle L_0(s) = \sphericalangle H_{TBI}(s) + \sphericalangle H_{AP}(s)$$

Now that the nominal loop transmission has been defined, the value of its parameters must be determined such that they meet the specifications that have been specified.

## 7.4 Relating the parameters of the modified Bode ideal characteristic to the specifications

The most direct way of determining the parameters is to recalculate them for the Bode ideal characteristic which has been modified for an unstable plant. This fortunately, is unnecessary as the bounds which were calculated for the nominal loop transmission (see section 5) can simply be modified by taking the all-pass function into account. Then the parameters of the Bode ideal characteristic used for a stable minimum phase plant can be calculated for the bounds modified by the all-pass function as before. The Bode ideal characteristic modified for the unstable plant will then meet the unmodified bounds on the nominal loop transmission.

The nominal loop transmission bounds are modified by shifting them in phase by the negative value of the all-pass function. This means that the bounds at low frequency will be shifted higher in phase and the high frequency bounds will not be shifted much at all.

The bounds which can be modified in this manner are:

1. The specifications on tracking the command input.
2. The relative stability specifications (gain margin, phase margin and resonant peak).
3. The speed of response (bandwidth).

The steady state error specifications for both the steady state response to the command and disturbance inputs cannot however be modified in this manner and will be recalculated in the following section.

The high frequency properties of the closed loop response, such as the magnitude and roll-off of the closed loop magnitude, are not effected at all by the all-pass function as these are only concerned with the magnitude which the all-pass function does not effect.

### 7.4.1 Steady state error

#### 7.4.1.1 The steady state error due to the command input for a system with a Bode ideal characteristic loop transmission

As discussed in section 3.4.1.4, the steady state error can be calculated using the final value theorem. For a unity feedback system with power of time inputs, an error coefficient can be defined which in turn can be used to calculate the steady state error.

Using the error coefficients defined for the steady state error due to a power of time input

$$K_i = \lim_{s \to 0} s^i \cdot L(s)$$

Where the loop transmission is given by

$$L(s) = H_{TBI}(s) = \frac{\omega_0^k}{s^k} \cdot H_{BI}(s) \cdot H_{AP}(s)$$

where:

k is the desired system type

Then the error coefficient becomes

$$K_i = \lim_{s \to 0} \left[ s^{i-k} \cdot \omega_0^k \cdot H_{BI}(s) \cdot H_{AP}(s) \right]$$

As $s \to 0$ the Bode ideal characteristic, $H_{BI}(0) = M_0$ and the all-pass function tends to

$$H_{AP}(0) = (-1)^n$$

where

$n$ is the number of unstable poles.

So the error coefficient can be simplified to

$$K_i = M_0 \cdot \omega_0^k \cdot \left( \lim_{s \to 0} s^{i-k} \right) = \begin{cases} 0 & if \quad k < i \\ M_0 \cdot \omega_0^k \cdot (-1)^n & if \quad k = i \\ \infty & if \quad k > i \end{cases}$$

The steady state error can now be calculated using the error coefficients.

The steady state error to a step input (*i=0*) is given by

$$e(\infty)=\frac{A}{1+K_0}=\begin{cases}\dfrac{A}{1+(-1)^n\cdot M_0} & if \quad k=0\\ 0 & if \quad k>0\end{cases}$$

The steady state error to the other power of time inputs (*i>0*) is given by

$$e(\infty)=\frac{A}{K_i}=\begin{cases}0 & if \quad k<i\\ \dfrac{A}{(-1)^n\cdot M_0\cdot \omega_0^k} & if \quad k=i\\ \infty & if \quad k>i\end{cases}$$

The steady state errors can be summarised in Table 5:

| *Type of L(s)* | *Steady state error to a step input* $r(t)=A\cdot u(t)$ | *Steady state error to a ramp input* $r(t)=A\cdot t$ | *Steady state error to a parabolic input* $r(t)=A\cdot t^2$ |
|---|---|---|---|
| 0 | $\frac{A}{1+(-1)^n\cdot M_0}$ | $\infty$ | $\infty$ |
| 1 | 0 | $\frac{A}{(-1)^n\cdot M_0\cdot\omega_0^k}$ | $\infty$ |
| 2 | 0 | 0 | $\frac{A}{(-1)^n\cdot M_0\cdot\omega_0^k}$ |
| 3 | 0 | 0 | 0 |

**Table 5: The steady state error due to a power of time command input for a system with the Bode ideal characteristic as the loop transmission and unstable poles.**

From the table above it can be seen that the steady state error is not effected by the all-pass function for loop transmissions of type 1 or greater. For type 0 systems the all-pass function increases the steady state error slightly for plants with an odd number of unstable poles.

### 7.4.1.2 The steady state error due to a disturbance input for a system with a Bode ideal characteristic loop transmission

As discussed in the previous section the error coefficient for the Bode ideal characteristic is given by:

$$K_i = \begin{cases} 0 & \text{if} \quad k < i \\ M_0 \cdot \omega_0^k \cdot (-1)^n & \text{if} \quad k = i \\ \infty & \text{if} \quad k > i \end{cases}$$

So for a step disturbance (*i=0*) the steady state error is given by

$$e(\infty) = \begin{cases} \dfrac{A}{1+(-1)^n \cdot M_0} \cdot P_D(0) & \text{if} \quad k = 0 \\ 0 & \text{if} \quad k > 0 \end{cases}$$

The steady state error to the other power of time disturbance inputs (*i>0*) is given by

$$e(\infty) = \frac{A}{K_i} = \begin{cases} 0 & \text{if} \quad k < i \\ \dfrac{A}{(-1)^n \cdot M_0 \cdot \omega_0^k} \cdot P_D(0) & \text{if} \quad k = i \\ \infty & \text{if} \quad k > i \end{cases}$$

The steady state responses to disturbance inputs can be summarised in Table 6:

| *Type of L(s)* | *Steady state error to a step input* $r(t)=A\cdot u(t)$ | *Steady state error to a ramp input* $r(t)=A\cdot t$ | *Steady state error to a parabolic input* $r(t)=A\cdot t^2$ |
|---|---|---|---|
| 0 | $\frac{A}{1+(-1)^n\cdot M_0}\cdot P_D(0)$ | $\infty$ | $\infty$ |
| 1 | 0 | $\frac{A}{(-1)^n\cdot M_0\cdot \omega_0^k}\cdot P_D(0)$ | $\infty$ |
| 2 | 0 | 0 | $\frac{A}{(-1)^n\cdot M_0\cdot \omega_0^k}\cdot P_D(0)$ |
| 3 | 0 | 0 | 0 |

**Table 6: The steady state error due to a power of time disturbance input for a system with the Bode ideal characteristic as the loop transmission and unstable poles.**

### 7.4.2 The trade offs between the parameter values which should be considered

The trade offs discussed for the design for stable minimum phase plants should be considered in this case (See section 6.3.5). The trade-offs are now further complicated by the all-pass function containing the unstable poles. The complication is that the phase margin parameter, $\phi_{PM}$ , must increase as the crossover frequency of the Bode ideal characteristic decreases. This implies a further trade-off between the phase margin and the bandwidth of the system. This is due to the fact that the compensator must be quick enough to counteract the unstable poles.

## *7.5 Example*

The example describes the design of the bank control loop for an Aerosonde unmanned air vehicle (UAV).

### 7.5.1 The plant model

The plant set for the Aerosonde UAV was obtained by linearising the non-linear model of the UAV, implemented using the Aerosim block set for Simulink®, at a number of points in the flight envelope. The set of linear models obtained are of the following form

$$\frac{\phi(s)}{\delta_a(s)} = \frac{K\cdot(s^2/\omega_1^2+2\cdot\zeta_1/\omega_1\cdot s+1)}{(s^2/\omega_r^2+2\cdot\zeta_r/\omega_r\cdot s+1)\cdot(s/\omega_s+1)}$$

where:

- $\phi$ is the bank of the UAV.
- $\delta_a$ is the aileron deflection.
- $K$ is the DC gain of the bank response to the aileron deflection.
- $\omega_1$ is the natural frequency of the zero of the linear model.
- $\zeta_r$ is the damping ratio of the roll subsidence poles.
- $\omega_r$ is the natural frequency of roll subsidence poles.
- $\omega_s$ is the frequency of the spiral divergence pole.

### 7.5.2 The operational requirements

There are no formal specifications for this example, so the design goal will be to find an optimal bank control loop by evaluating the trade offs between the different performance characteristics.

For practical reasons the following informal specifications will be imposed:

1. The allowable range for the bank command will be $-30^o \leqslant \phi_{c_{MAX}} \leqslant +30^o$ .
2. The maximum overshoot will be limited to approximately $\phi_{OV} = 1.5^o$ .

**Step 1: Translate the time domain specifications into the frequency domain**

*Inputs:* All the available time domain specifications on the closed loop control system.

*Outputs:* The frequency domain versions of the time domain specifications.

*Procedure*:

Using the calculations discussed in section 3.4, the frequency domain specifications may be estimated.

The only specification is the maximum overshoot that the closed loop bank control loop may have. The maximum overshoot in bank allowed is $\phi_{OV} = 1.5^o$ . The maximum bank command allowed is $\phi_{c_{MAX}} = 30^o$ . So the maximum percentage overshoot allowed is

$$Percentage\,overshoot = \frac{\phi_{OV}}{\phi_{c_{MAX}}} \cdot 100 = \frac{1.5^o}{30^o} \cdot 100 = 5\%$$

From Figure 3.12 the resonant peak of a second order model that corresponds to a 5% overshoot is approximately

$$M_{RP} = 0\ dB$$

This specification is impossible to achieve as all the transfer functions in the plant set have a phase of -180° at DC due to the unstable pole that each of them contain. In order for the resonant peak specification to be met the magnitude of the loop transmission at DC must be less than -6 dB, which means that the closed loop responses will have unacceptable steady state errors.

Note that if the plant models were not considered when evaluating what reasonable specifications would be for the closed loop systems performance, the original specification of 0 dB would have required an initial design to be done before it was discovered that it was unreasonable.

A resonant peak specification of 1 dB will be used instead and a pre-filter will be required to lower the resonant peak to meet the overshoot specification. From the m-circles plotted in Figure 5.4, it can be seen that this specification puts a requirement that all the loop transmissions magnitudes at DC must have a have a magnitude greater than 19 dB.

**Step 2: Select the nominal plant.**

*Inputs:* The plant set, $P(s)$

*Outputs:* The nominal plant, $P_0(s)$

*Procedure:*

Select the nominal plant to be the transfer function which lies in the middle of each of the plant templates over the range of frequencies of interest.

As the plant in this example has a significant amount of plant uncertainty due to the flight envelope, the first step is to generate the plant template and select the nominal plant.

The nominal plant selected is chosen to lie in the middle of the plant template (as far as practically possible). The Bode plot of the magnitude of the nominal plant selected is shown in Figure 7.6. The nominal plant is given by

$$P_0(s)=\frac{106.7463\cdot(s^2+1.682s+19.53)}{(s+17.67)\cdot(s-0.05785)\cdot(s^2+2.277s+31.65)}$$

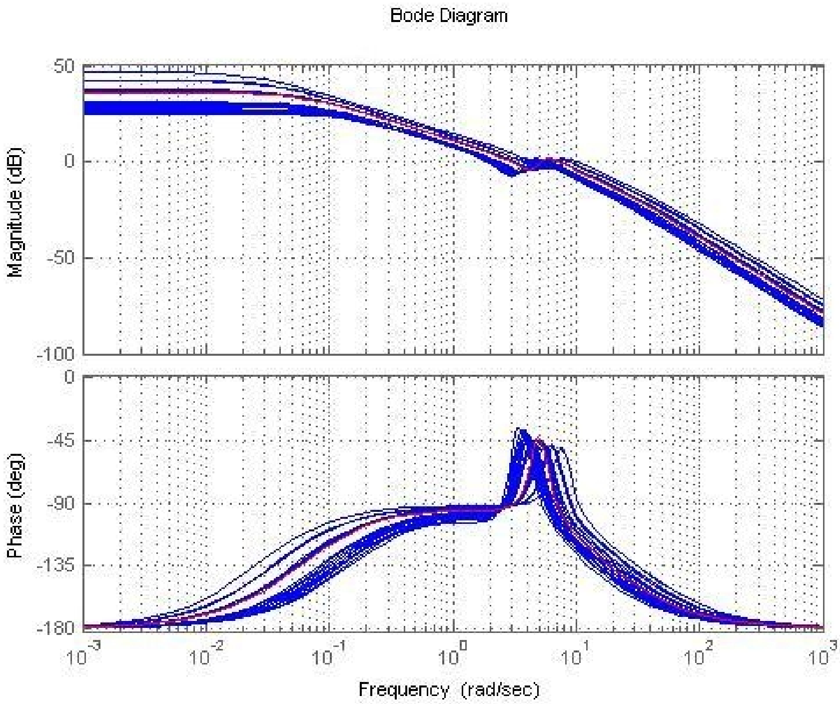


**Figure 7.5: A bode plot of the bank loop.**

**Step 3: Determine the unstable poles and split the nominal plant into the all-pass function and the stable minimum phase part.**

*Inputs:* The nominal plant, $P_0(s)$

*Outputs:* The all-pass function for the nominal plant, $P_{AP}(s)$

The stable minimum phase part of the nominal plant, $P_{SMP}(s)$

*Procedure:*

- Find the poles of the plant.
- All the positive poles are the unstable poles of the plant, $\tilde{p} = p_{1,} p_{2,} \ldots, p_n$
- The all-pass function of the plant is given by:

$$P_{AP}(s) = \frac{(s - \overline{p_1}) \cdot (s - \overline{p_2}) \ldots (s - \overline{p_n})}{(s + p_1) \cdot (s + p_2) \ldots (s + p_n)}$$

- The stable minimum phase part of the plant can be calculated from:

$$P_{SMP}(s) = \frac{P(s)}{P_{AP}(s)}$$

In this example there is a single unstable pole, so the vector of unstable poles is simply:

$\tilde{p} = 0.05785$ rad/s

And the all-pass function is then

$$P_{AP}(s) = \frac{s + 0.05785}{s - 0.05785}$$

The plot of the phase versus frequency of the all pass function is shown in Figure 7.6.

The stable minimum phase part of the plant is

$$P_{SMP}(s) = \frac{106.7463 \cdot (s^2 + 1.682s + 19.53)}{(s + 17.67) \cdot (s + 0.05785) \cdot (s^2 + 2.277s + 31.65)}$$

The Bode plot of the stable minimum phase part of the plant is shown in Figure 7.7 along with the rest of the plant set.

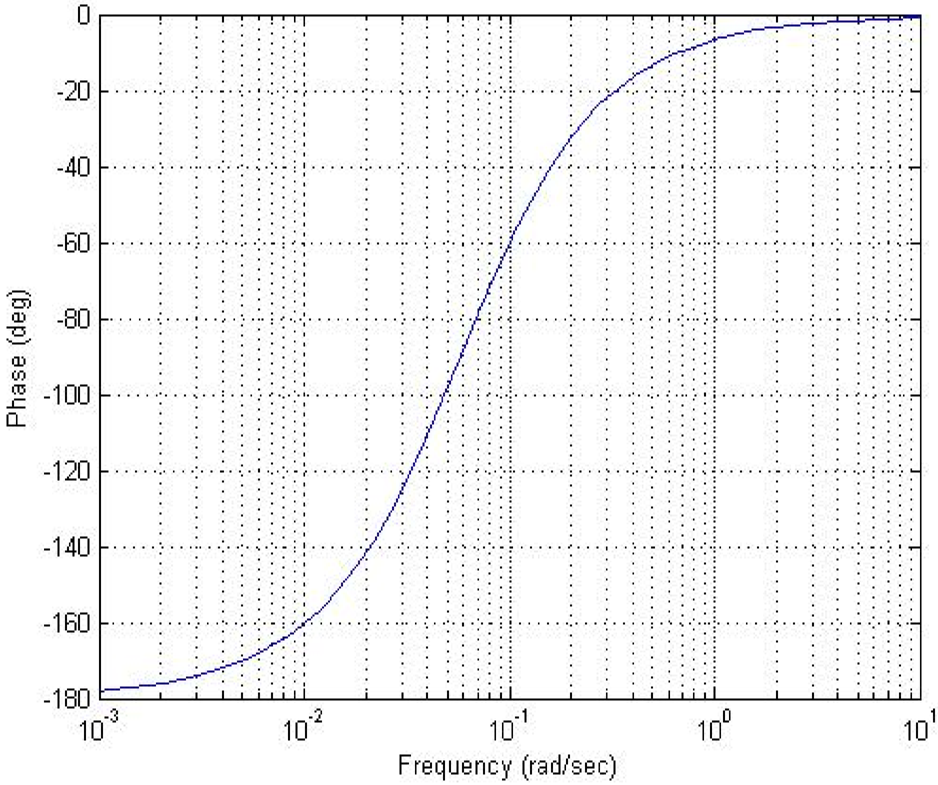


**Figure 7.6: Phase versus frequency of the all-pass function**

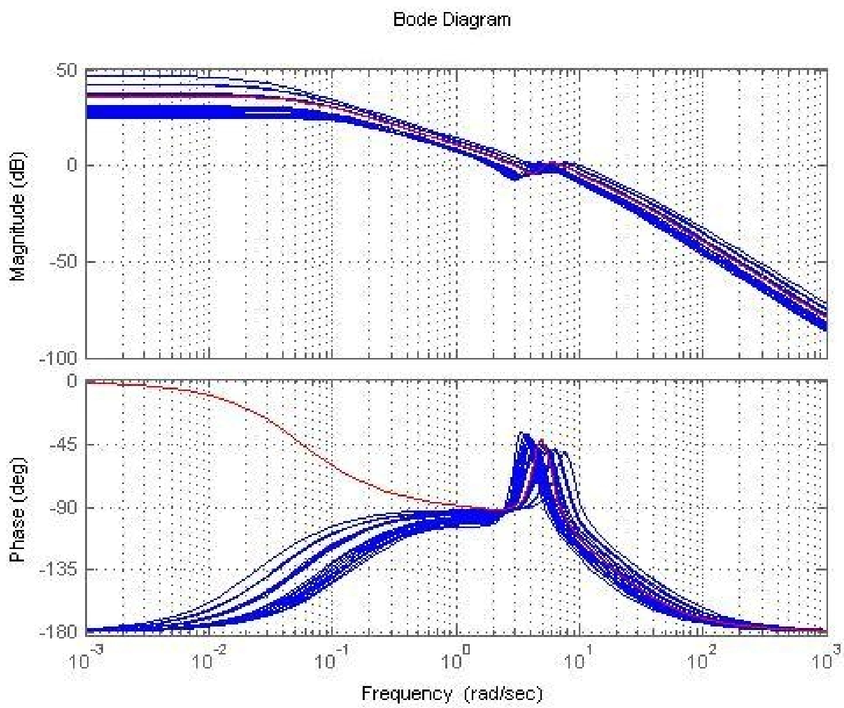


**Figure 7.7: The Bode plot of the new nominal plant and the plant set.**

**Step 4: Translate the frequency domain bounds into bounds on the stable minimum phase part of the nominal loop transmission**

*Inputs:* All the available frequency domain specifications on the closed loop control system.

*Outputs:* The bounds on the stable minimum phase part of the loop transmission.

*Procedure*:

Using the calculations discussed in section **5**, the bounds on the loop transmission obtained from the closed loop frequency domain specifications.

Figure 7.8 shows a Nichols plot of the plant templates at a number of frequencies. Note that for the low frequencies the nominal plant is not contained within the plant templates.

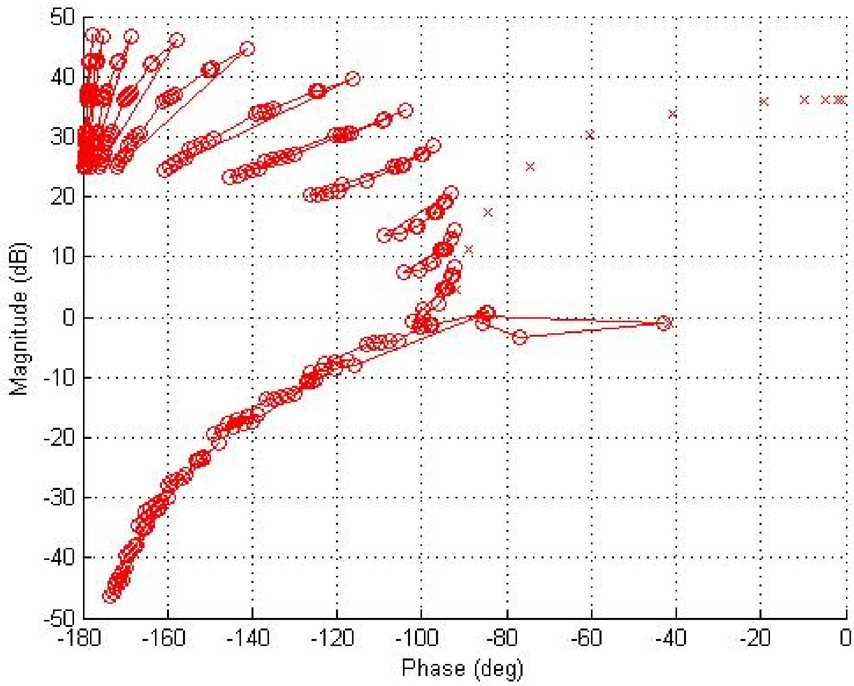


**Figure 7.8: The plant templates for the bank control loop.**

Now that the plant templates have been generated, the frequency domain closed loop specifications can be translated into the bounds on the loop transmission. Figure 7.9 shows a

Nichols plot of the stability bounds calculated from the 1 dB resonant peak specification calculated in the previous step. These stability bounds were calculated using the method discussed in section 5.1.2. The bounds obtained are effectively shifted by the phase of the all-pass function, plotted in Figure 7.6. This is due to the fact that the plant templates contain the unstable poles, but the nominal plant has  a stable version of the all-pass function as part of it. The nominal plant therefore is not contained within the  plant template at low frequencies, but is shifted in phase from the position it would have had, had it simply contained the unstable poles. The phase by which it is shifted is the phase of the all-pass function at the frequency of the plant template. This will have the effect of shifting the stability bound obtained by the negative value of the all pass function, but the bound will still have the same shape. The bounds obtained are therefore bounds on the stable minimum phase nominal loop transmission, $L_0(s)$, which calculated in step 3 and plotted in Figure 7.7.

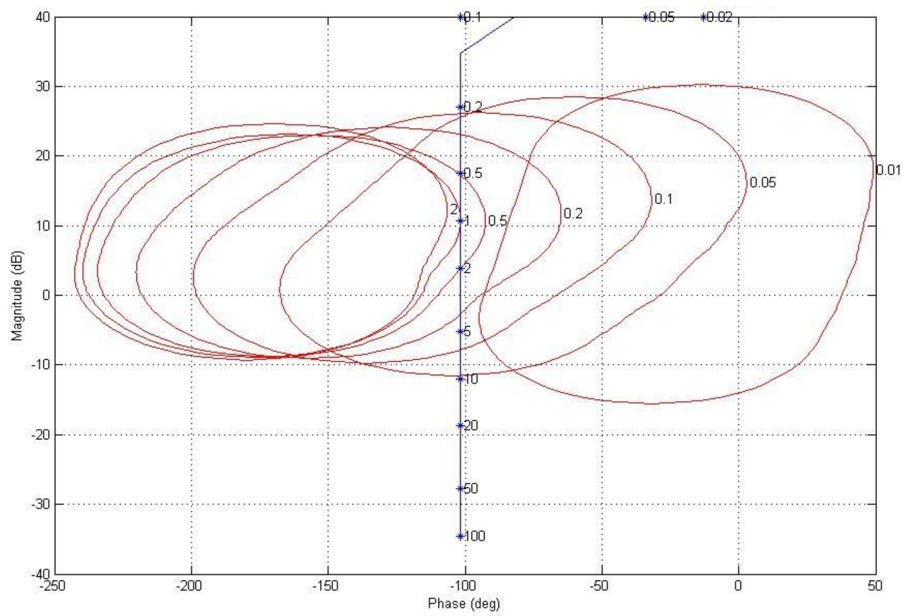


**Figure 7.9: The stability bounds due to the 1 dB resonant peak at a number of frequencies.**

**Step 5: Estimate the loop transmission required to meet the specifications**

*Inputs:* All the bounds placed on the nominal loop transmission.

*Outputs:* The typed Bode ideal characteristic, $H_{TBI}(s)$ .

*Procedure*:

Using the calculations discussed in section 7.4, the parameters required may be estimated. Any parameters that do not have a specification from which they can be estimated, should be chosen using engineering judgment. The values of the parameters do not need to be exact, as the Bode ideal characteristic is being used to help find the loop transmission that should be required, considering all the trade offs.

All of the following parameters of the Bode ideal characteristic must be calculated:

$M_0$ the low frequency gain

$\omega_0$ the maximum frequency of the low frequency gain

$\Phi_{PM}$ the phase margin

$M_{GM}$ the gain margin

$e$ the pole-zero excess

$k$ the system type

From which the typed Bode ideal characteristic, $H_{TBI}(s)$, can be calculated as discussed in section 6.2 and 6.3.1.1.1.

The first parameter to be estimated is the system type, $k$. As the plant contains an unstable pole, a system type greater than 0 will add extra phase lag which will make it more difficult to stabilise the system. Therefore, $k=0$ will be used.

As discussed in step 1, the resonant peak specification places bounds on the minimum magnitude of 19 dB that the loop transmissions may have for all the plants in the plant set. The stability bounds in turn translate this bound into bounds on the stable minimum phase part of the of the nominal loop transmission over the frequency range of interest. The value of the low frequency gain, $M_0$, will therefore need to be chosen to lie above the stability bounds for frequencies less than or equal to the maximum frequency of the low frequency gain, $\omega_0$ .As $\omega_0$ has not been calculated yet, all the stability bounds will be considered. This will not be overly conservative as the maximum magnitude of the stability bounds vary from approximately 25 dB to 30 dB, as shown in Figure 7.9.

The DC gain of the stable minimum phase of the nominal plant is approximately 40 dB this will used for the value of $M_0$ .

The desirable cross-over frequency will be approximately 1 rad/s, which is close to the bandwidth of the plant, see Figure 7.7. This frequency is desirable as it will prevent the loop compensator calculated from having to have high gain near the selected cross-over frequency in order to extend the bandwidth. Assuming the phase of the nominal loop transmission during the roll-off of the magnitude to cross-over will be less than -90$^o$, the magnitude will roll-off at less than 20 dB/decade. So the value of $\omega_0$ should be approximately 10 times smaller than the desired cross-over frequency, which will give a value of approximately 0.1 rad/s.

The phase margin, $\Phi_{PM}$ , will now be estimated. The specification that will determine this parameter is the resonant peak specification. As discussed in section 6.3.2.3, the bounds on the loop transmission due to this specification are a set of QFT stability bounds. The phase margin selected should be the minimum phase margin that meets all the stability bounds. The phase margin will lie between approximately 180$^o$-110$^o$=70$^o$ and 180$^o$+ 50$^o$=230$^o$, which are 180$^o$ plus the maximum values of the stability bounds which have the minimum maximum phase and the maximum maximum phase respectively.

The maximum phase margin of 230$^o$ is obviously impractically high. This highlights the fact that the loop transmission must have a cut-off frequency high enough to allow the phase lag added by the all-pass function (due to the unstable poles) to fall to a small enough value. Fortunately the plant templates with frequencies below $\omega_0$ need not be considered so the maximum phase margin that may be required is approximately 180$^o$-35$^o$=145$^o$.

If this calculation is done by hand, a phase margin will be selected between these phases and checked to see if they do not violate the stability bounds. The designer will then need to iterate until the minimum phase margin which meets these bounds is found. In this case the ***calcPhaseMargin*** function of the ***BodeCutoff*** class was used to find the phase margin. See section 12.9.3.7 for a description of this code. A phase margin of $\phi_{PM}=78.4^o$ was calculated.

The gain margin, $M_{GM}$, must be selected so that the Bode ideal characteristic passes below the stability bounds. As disturbances due to changes in the throttle setting of the engine are expected, the gain margin must be selected such that the disturbances do not cause unacceptable oscillations in bank angle. So rather than passing below the stability bounds generated for the command input, the stability bounds generated for the disturbance input shall be used. As there is no specification for the disturbance undershoot, the overshoot specification for the command input will be used, so that the disturbance input will not cause the output to oscillate more than is acceptable. As discussed in section 5.2, the resonant peak specification for a disturbance makes use of the inverse Nichols chart, which is simply the Nichols chart rotated by 180° around the point 0 dB and – 180°. This means that if the gain margin is set, the inverse of the maximum magnitude of the stability bounds for the command input, the sensitivity function (which gives the response to a disturbance entering at the output) will have a maximum resonant peak of 1 dB. A gain margin of $M_{GM}=16.2$ dB will therefore be used.

The pole zero-excess of the plant is 2, so a pole zero excess of $e=2$ will be selected for the nominal loop transmission.

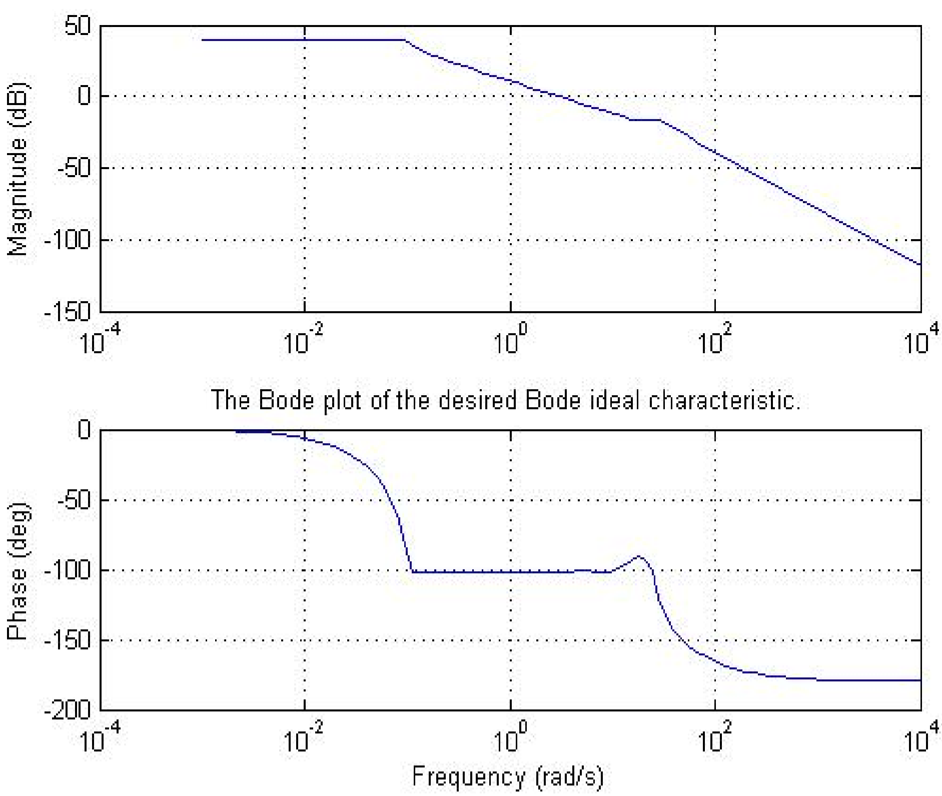


**Figure 7.10: The desired Bode Ideal characteristic.**

These parameters will give a Bode ideal characteristic which has the Bode plot shown in Figure 7.10.

Figure 7.11 shows a Bode plot of the fitted nominal loop transmission, $L_0(s)$, in blue and the typed Bode ideal function, $H_{TBI}(s)$, in red. Figure 7.12 shows the Nichols chart plot of the fitted nominal loop transmission (in red) and the typed Bode ideal characteristic (in blue).

The routine ***fit*** function of the ***BodeIdeal*** class was used to fit a rational transfer function to the Bode ideal characteristic (See section 12.10.3.2 for details on the routine). The following transfer function was obtained

$$L_0(s)=\frac{b_3\cdot s^3+b_2\cdot s^2+b_1\cdot s+b_0}{s^5+a_4\cdot s^4+a_3\cdot s^3+a_2\cdot s^2+a_1\cdot s}$$

where:

$$\begin{array}{ll} b_0=4220 & a_0=61.16 \\ b_1=2.657\times10^4 & a_1=831.8 \\ b_2=2352 & a_2=8444 \\ b_3=119.6 & a_3=1106 \\ & a_4=42.75 \end{array}$$

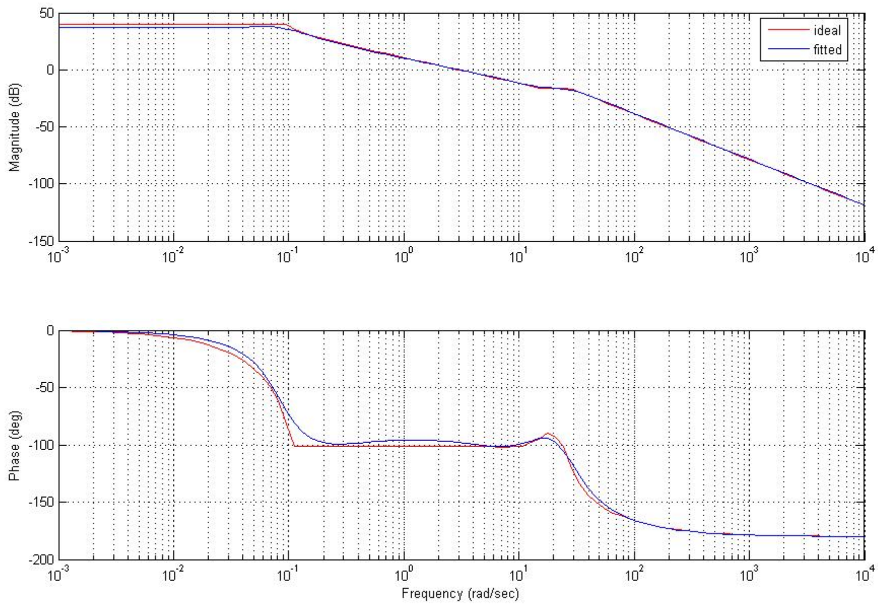


**Figure 7.11: The Bode plot of the rational transfer function fitted to the desired Bode Ideal Characteristic.**

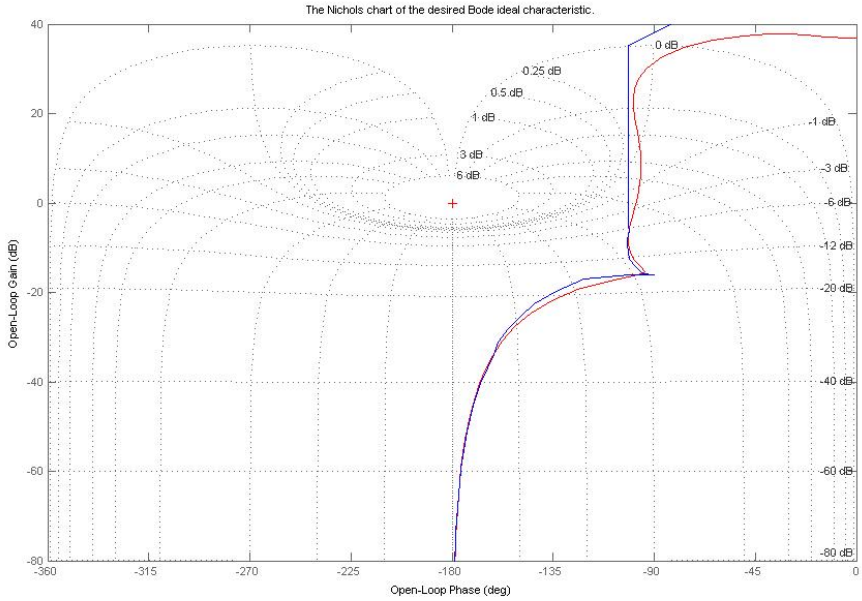


**Figure 7.12: The Nichols chart of the rational transfer function fitted to the desired Bode Ideal Characteristic.**

**Step 6: Calculate the controller**

*Inputs:* The nominal plant, $P_0(s)$ .

The nominal loop transmission, $L_0(s)$ .

*Output:* The loop compensator, $G(s)$ , that gives the nominal loop transmission, $L_0(s)$ , for the nominal plant $P_0(s)$ .

*Procedure:*

The controller is calculated using

$$G(s)=\frac{L_0(s)}{P_0(s)}$$

Figure 7.13 shows a Bode plot of the loop compensator calculated.

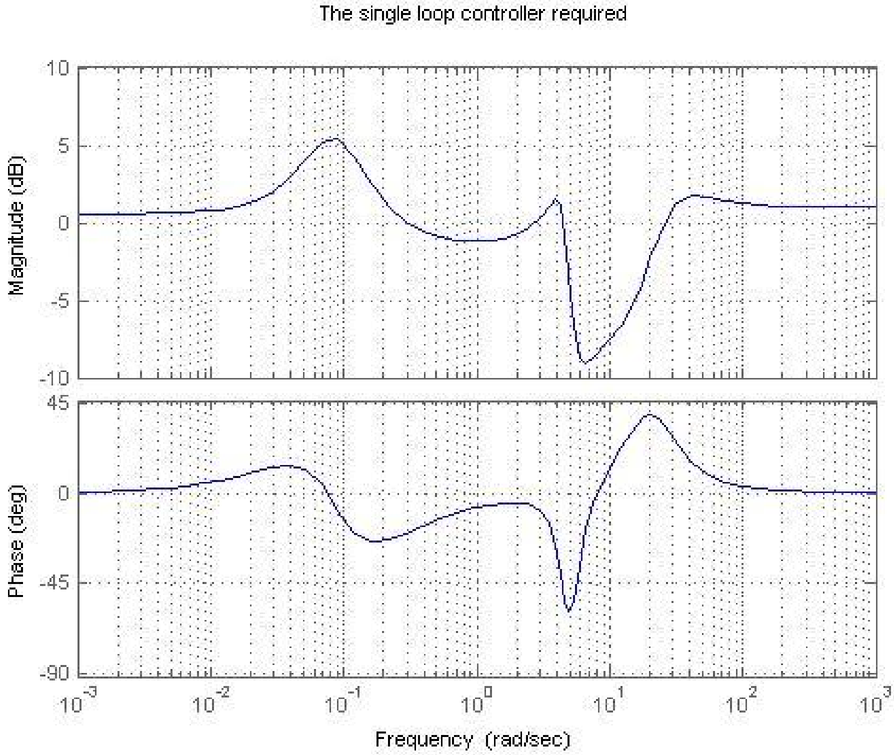


**Figure 7.13: The Bode plot of the required controller.**

**Step 7: Calculate the loop transmission set**

*Inputs:* The loop compensator $G(s)$ .
The plant set, $\tilde{P}(s)$ .

*Output:* The loop transmission set, $\tilde{L}(s)$ .

*Procedure:*

For each plant in the plant set, the loop transmission set is calculated using

$$L_i(s) = P_i(s)G(s)$$

where $i = 1,2,3,\ldots,n$

$n$ is the number of plants in the plant set.

Figure 7.14 shows a Nichols chart plot of the loop transmission set and the m-circle related to the resonant peak specification of 1 dB as calculated in step 1.

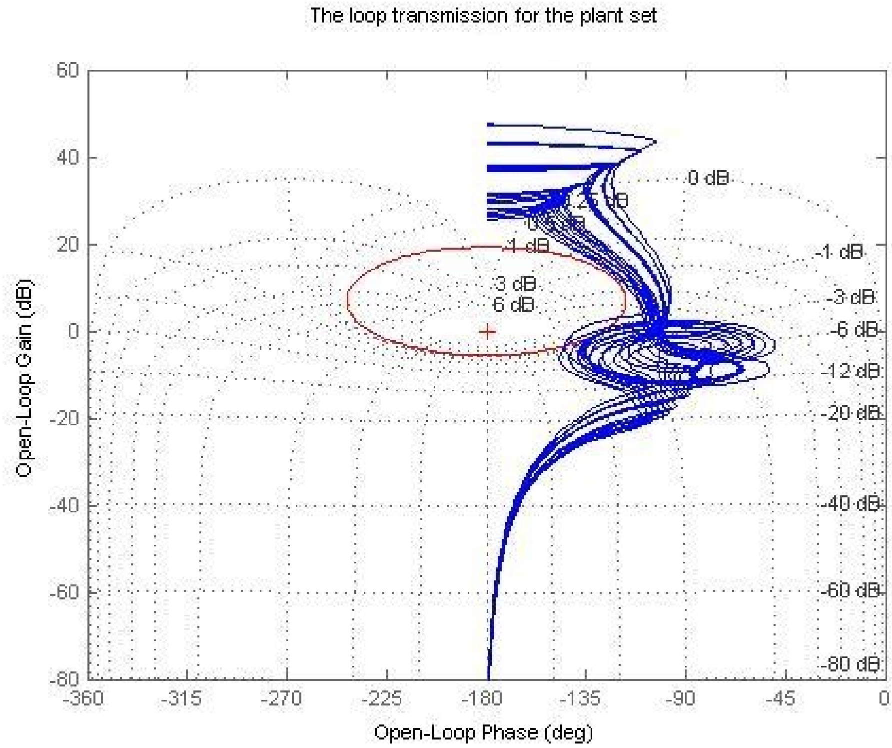


**Figure 7.14: The Nichols chart of the loop transmission.**

Figure 7.15 shows the Bode plot of the loop transmission set.

**Step 8: Calculate the closed loop transfer function set**

*Input:* The loop transmission set, $\tilde{L}(s)$ .

*Output:* The closed loop frequency response set, $\tilde{T}(s)$ .

*Procedure:*

For each plant in the plant set, the closed loop transfer function set is calculated using

$$T_i(s) = \frac{L_i(s)}{1 + L_i(s)}$$

where $i = 1,2,3,\ldots,n$

$n$ is the number of plants in the plant set.

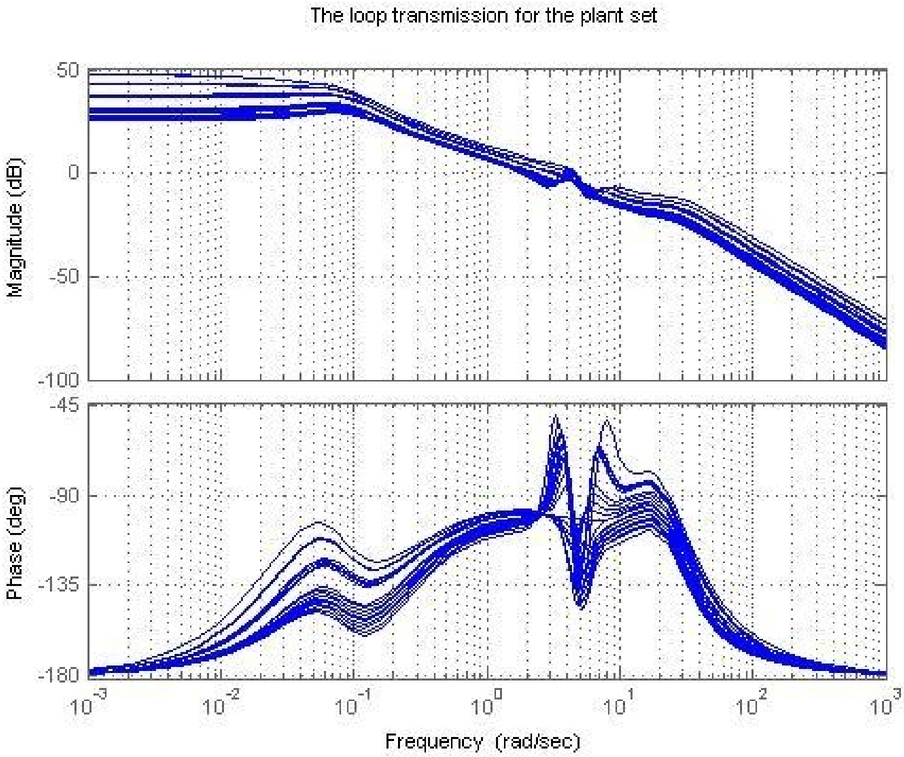


**Figure 7.15: The Bode plot of the loop transmission set.**

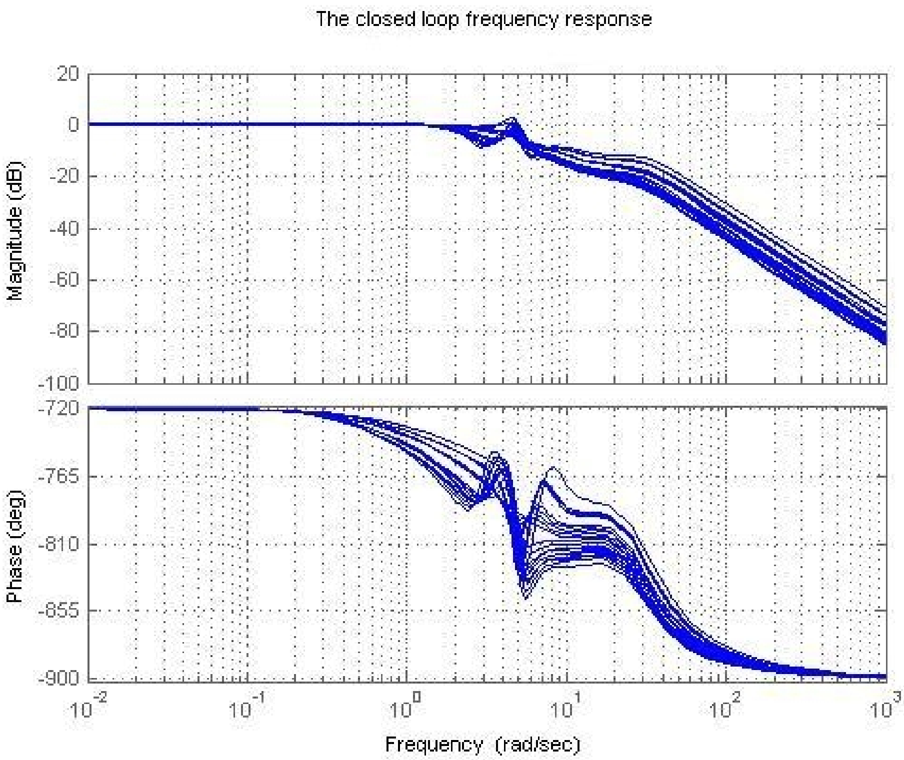


**Figure 7.16: The closed loop frequency response.**

Figure 7.16 shows a Bode plot of the closed loop response set.

**Step 9: Calculate the closed loop time response set**

*Input:* The closed loop frequency response set, $\tilde{T}(s)$ .

*Output:* The set of system responses to the specified command input. $\tilde{Y}_C(s)$ .

*Procedure:*

For each closed loop transfer function in the closed loop transfer function set, the closed loop response set is calculated using

$$Y_{C_i}(s)=T_i(s)C(s)$$

where $i=1,2,3,\ldots,n$

$n$ is the number of plants in the plant set.

And the time domain response set is given by

$$y_{C_i}(t) = \mathscr{L}^{-1}\{ Y_{C_i}(s) \}$$

Where $\mathscr{L}^{-1}$ denotes the inverse Laplace transform.

The worst case command input that the bank control system can encounter is a step, as it commands an instantaneous change in bank which must be held steady thereafter. The command input that the closed loop time response will be calculated for is therefore chosen to be a step input. Figure 7.17 shows the step response calculated.

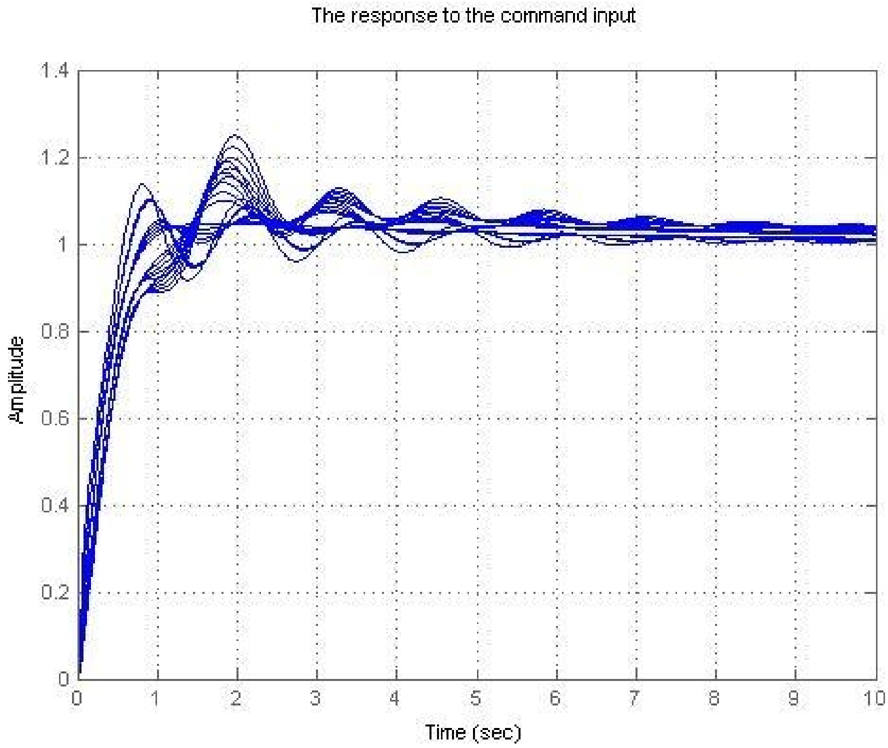


**Figure 7.17: The response to the command input.**

**Step 10: Calculate the plant input set**

*Input:* The loop compensator, $\tilde{G}(s)$ .

The loop transmission set, $\tilde{L}(s)$ .

The plant input, $C(s)$ .

*Output:* The plant input set in the frequency domain $\tilde{U}_C(s)$ .

The plant input set in the time domain $\tilde{u}_C(s)$ .

*Procedure:*

For each closed loop transfer function in the closed loop transfer function set, the closed loop response set is calculated using

$$U_{C_i}(s)=\frac{G(s)}{1+L_i(s)}\cdot C(s)$$

where $i=1,2,3,\ldots,n$

$n$ is the number of plants in the plant set.

And the time domain response set is given by

$$u_{C_i}(t) = \mathscr{L}^{-1}\{ U_{C_i}(s) \}$$

Figure 7.18 and Figure 7.19 shows a Bode plot of the frequency response of the plant input set and the time domain response of the plant input set respectively.

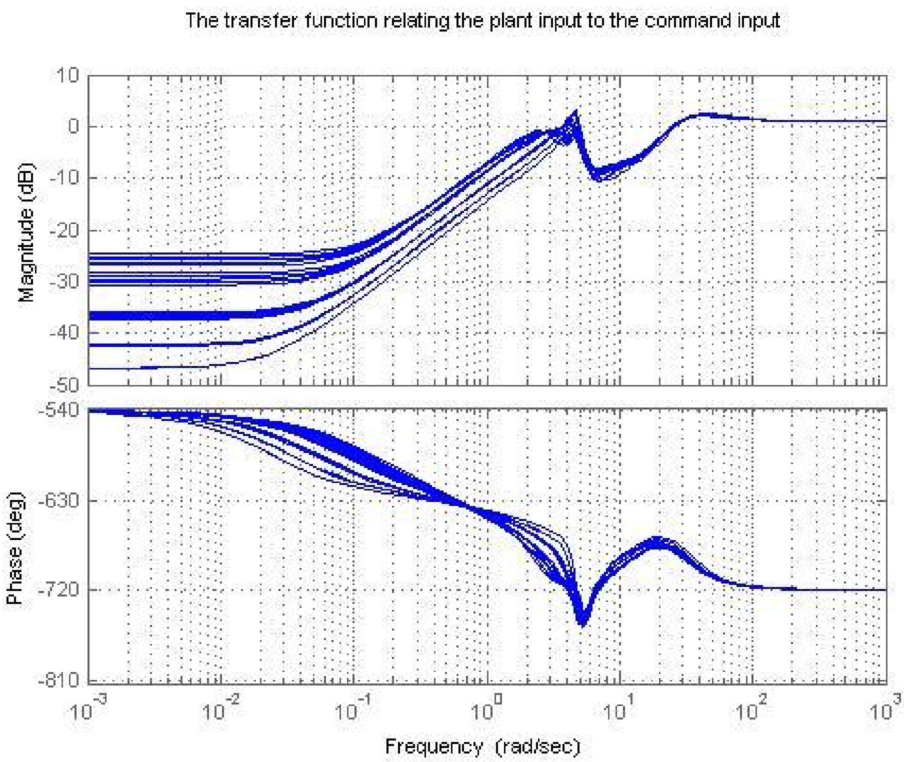


**Figure 7.18: The transfer function relating the plant input to the command input.**

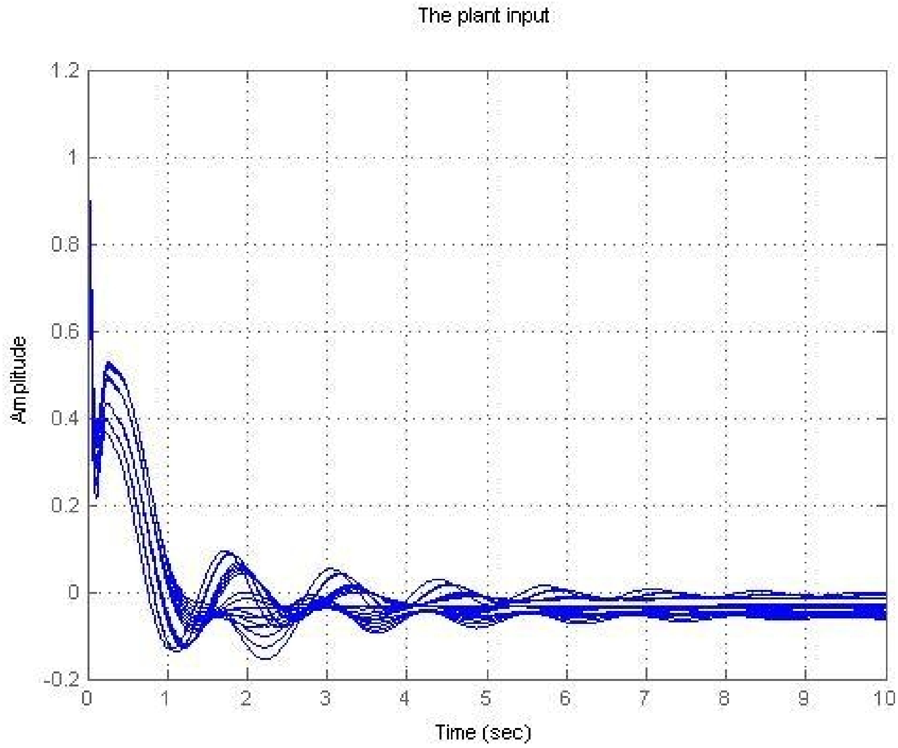


**Figure 7.19: The time response of the plant input.**

**Step 11: Fit a simple transfer function to the controller and recalculate the closed loop transfer function.**

*Inputs:* The loop compensator, $G(s)$
The loop transmission set, $\tilde{L}(s)$ .

*Outputs:* The fitted loop compensator, $G_{fit}(s)$ .
The closed loop frequency response set for the fitted controller, $\tilde{T}_{fit}(s)$ .

*Procedure*:

- Fit a simple transfer function to the compensator calculated in step 4.
- Re-calculate the closed loop transfer function as discussed in step 6.

This step need not be done now, but as the next step will be to calculate a pre-filter to reduce the overshoot it will help to work with a closed loop transfer function of a lower order.

Considering Figure 7.13 it can be seen that the magnitude of the controller is almost constant with frequency except at 1 rad/s and 6 rad/s, which is the frequency of the $\omega_0$ and around the frequencies of the Bode step part of the Bode ideal characteristic. Therefore, the magnitudes at these frequencies can be smoothed out.

The simplified controller transfer function will then be simply a gain of

$$G_{fit}(s) = 1\,dB$$

The loop transmission is recalculated as discussed in step 5 to give Figure 7.20 and the closed loop transfer function recalculated, as discussed in step 6, to give Figure 7.21. The step response to the command input is plotted in Figure 7.22.

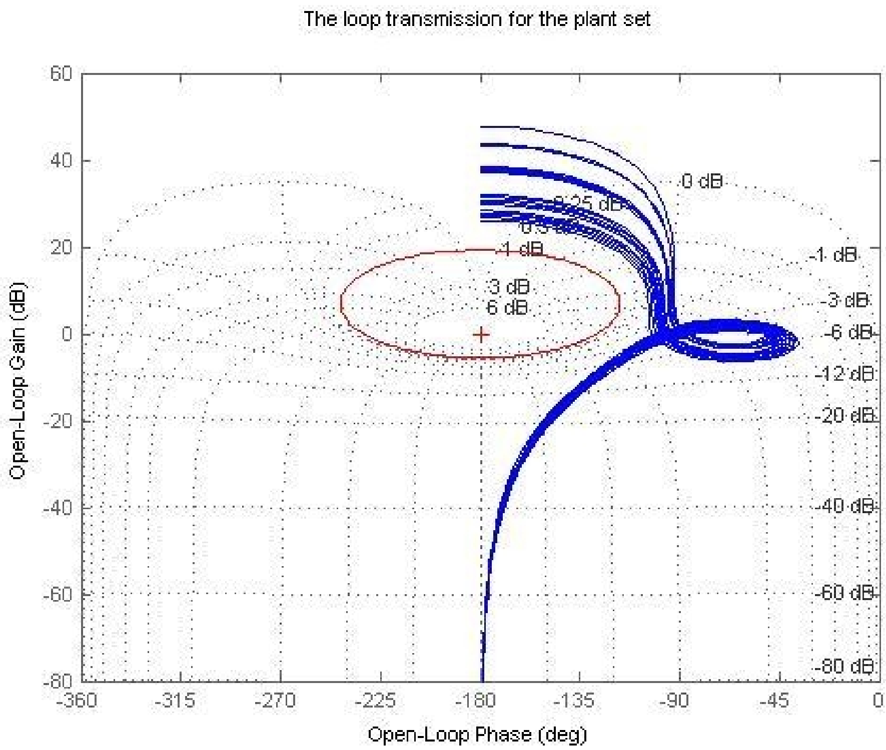


**Figure 7.20: The loop transmission set for the fitted controller.**

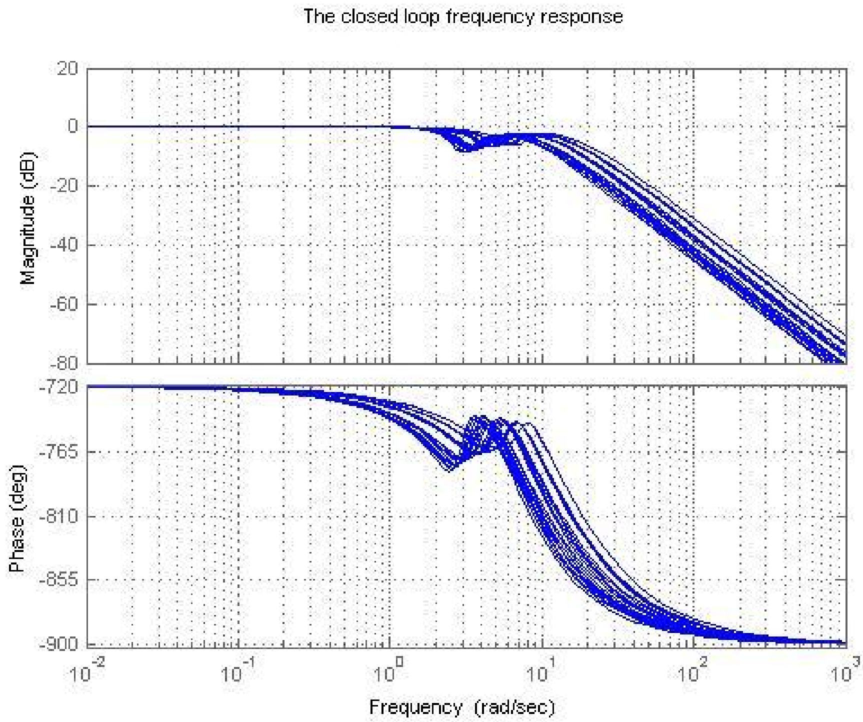


**Figure 7.21: The closed loop response set for the fitted controller.**

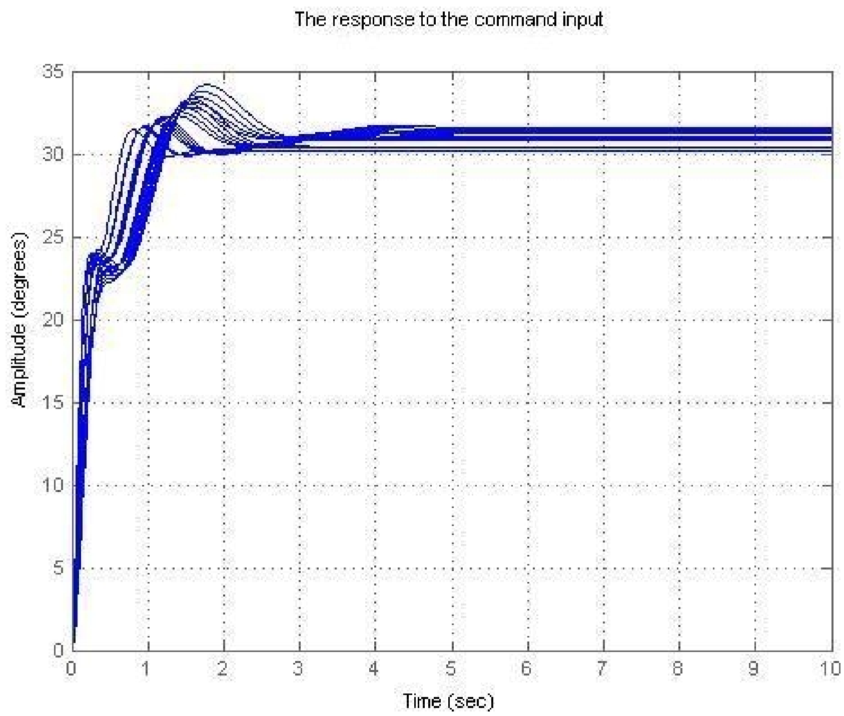


**Figure 7.22: The closed loop step response set.**

**Step 12: Calculate the pre-filter**

*Inputs:* The closed loop frequency response set for the fitted controller, $\tilde{T}_{fit}(s)$ .

*Outputs:* The pe-filter, $F(s)$ .

*Procedure*:

- A simple estimate of the pre-filter is required, so the procedure used will be to add a simple first order low pass filter as the pre-filter.
- The bandwidth of this filter is varied until the desired step response is obtained.

The step response to the command input obtained, see Figure 7.22, has a large steady state error. This will not be effected by the pre-filter. So in order to evaluate the pre-filters effect on the overshoot the step response will be normalised so that all the responses have zero steady state error. This is achieved by dividing each step response by the final value it obtains.

A simple first order low pass filter, of the following form, will now be applied to the normalised step responses

$$F(s)=\frac{\alpha}{s+\alpha}$$

The normalised step response to the command input was obtained with $\alpha=4$ rad/s.

The prototype models obtained give a satisfactory performance for the bank loop. The specification on the overshoot is reached with the help of a pre-filter. The steady state error due to a step command input may however be problematic. This can be reduced by increasing the loop transmissions DC gain, but would however cause the relative stability of the bank control loop to be reduced. It would seem that gain scheduling the gain of the pre-filter based on the UAV's airspeed and altitude would be the best option, as the relative stability can be maintained while reducing the steady state error. This however increases the complexity of the control algorithm.

In this example, the loop compensator obtained when calculating the prototype models was very close to a simple gain (See Figure 7.13). So a simple gain was used to calculate the rest of the example. This negated the need to use a control system design procedure such as QFT or H-infinity. However, if the compensator obtained was not simplified the example could have continued in a similar manner to the stable example, as discussed in section 6.4, and bounds which could be used by the control system design algorithm would have been calculated.

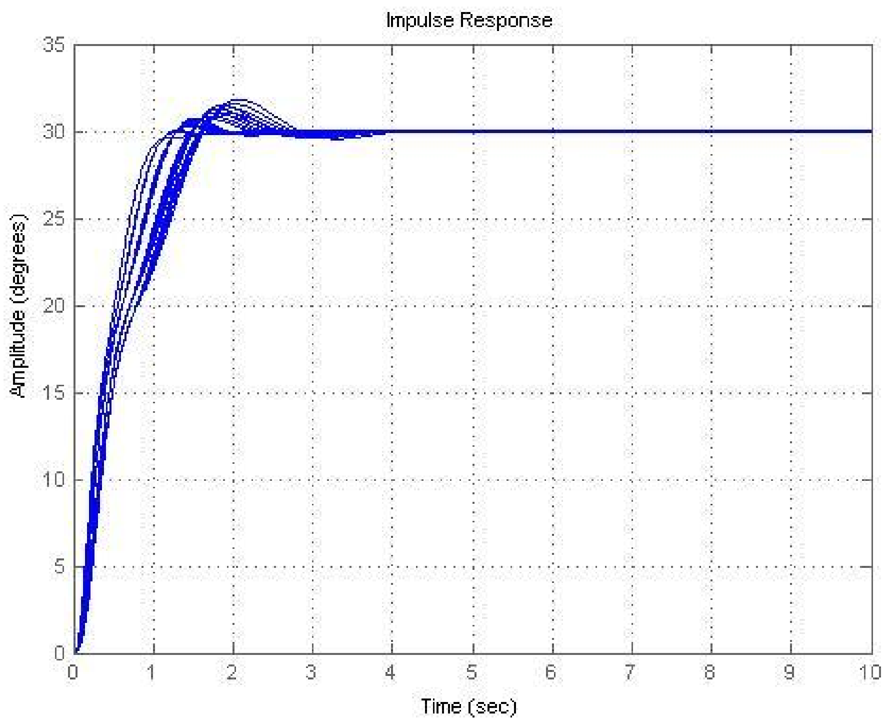


**Figure 7.23: The closed loop step response set with the pre-filter added and the steady state error removed.**

# 8 Designing for non-minimum phase plants

## 8.1 The suitability of the Bode ideal characteristic for the loop transmission for a non-minimum phase plant

As the Bode ideal characteristic can be successfully used for describing the desired loop transmission for a control system with a stable minimum phase plant, the new question is can it be used for the loop transmission of a stable non-minimum phase plant?

The simplest way to answer this question, as in the unstable minimum phase plant case (see section 7), is to look at the controller that will be required if the Bode ideal characteristic is used for a plant with non-minimum phase zeros. So if the nominal plant can be described by the following transfer function:

$$P_0(s)=P_{SMP}(s)\cdot N_{nmp}(s)$$

where:

$P_{SMP}(s)$ is the stable minimum phase part of the plant.

$N_{nmp}(s)$ are the non-minimum phase zeros of the plant.

The controller obtained if the loop transmission is the typed Bode ideal characteristic, $H_{TBI}(s)$ , will be:

$$G(s)=\frac{H_{TBI}(s)}{P_0(s)}=\frac{H_{TBI}(s)}{P_{SMP}(s)\cdot N_{nmp}(s)}$$

Now, as the typed Bode ideal characteristic has no non-minimum phase zeros, the controller will have unstable poles that attempt to cancel out the nominal plant's non-minimum phase zeros . Although, mathematically this is a solution to the problem, as in the case of the unstable plant, the plant uncertainty will ensure that the cancellation will not happen . Therefore, the control system will still contain the unstable poles introduced by the controller and the control system may now be unstable.

The loop transmission must therefore contain the non-minimum phase zeros so that the controller obtained does not cancel them.

## *8.2 The properties of non-minimum phase zeros*

### 8.2.1 Non-minimum phase real zeros

Consider a non-minimum phase real zero of the form:

$$H(s)=\frac{s-\alpha}{\alpha}$$

where $\alpha$ is positive, the magnitude and phase of the non-minimum phase real zero versus frequency is given by

$$|H(j\cdot\omega)|=\frac{\sqrt{\omega^2+\alpha^2}}{\alpha}$$

$$\measuredangle H(j\cdot\omega)=\arctan\left(\frac{\omega}{-\alpha}\right)$$

A non-minimum phase zero at a frequency of 1 rad/s will have the Bode plot shown in Figure 8.1. From the plot it can be seen that the phase of a real non-minimum phase zero starts at a phase of 180$^o$ at low frequencies and decreases to a phase of 90$^o$ at high frequencies.

Therefore a non-minimum phase real zero has extra phase lead at low frequencies than its minimum phase counterpart and the same phase at high frequencies. The magnitude of a non-minimum phase zero is exactly the same as its minimum phase counterpart.

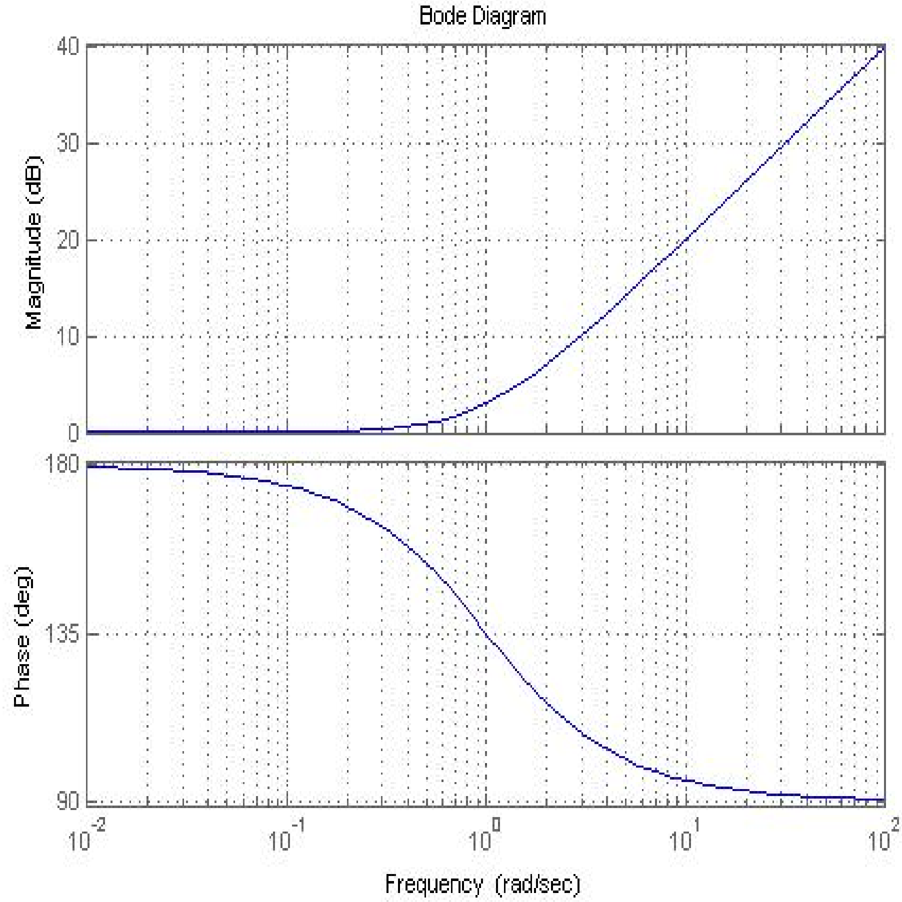


**Figure 8.1: The Bode plot of a non-minimum phase zero.**

### 8.2.2 Non-minimum phase complex conjugate zero pairs

Consider a non-minimum phase complex conjugate zero pair of the form:

$$H(s)=\frac{(s-\alpha+j\cdot\beta)\cdot(s-\alpha-j\cdot\beta)}{\alpha^2+\beta^2}=\frac{s^2-2\cdot\alpha\cdot s+(\alpha^2+\beta^2)}{\alpha^2+\beta^2}$$

where $\alpha$ and $\beta$ are positive, the the magnitude and and phase of the non-minimum phase zero versus frequency is given by

$$|H(j\cdot\omega)|=\frac{\sqrt{\left[(\alpha^2+\beta^2)-\omega^2\right]^2+4\cdot\alpha^2\cdot\omega^2}}{\alpha^2+\beta^2}$$

$$\measuredangle H(j\cdot\omega)=\arctan\left(\frac{\omega+\beta}{-\alpha}\right)+\arctan\left(\frac{\omega-\beta}{-\alpha}\right)$$

A complex conjugate non-minimum phase zero pair at a frequency of 1 rad/s will have the Bode plot shown in Figure 8.2. From the plot it can be seen that the phase of a non-minimum phase complex zero pair starts at a phase of 360° at low frequencies and decreases to a phase of 180° at high frequencies.

Therefore a complex conjugate non-minimum phase zero pair has extra phase lead at low frequencies than its minimum phase counterpart and the same phase at high frequencies. The magnitude of a non-minimum phase zero pair is exactly the same as its minimum phase counterpart.

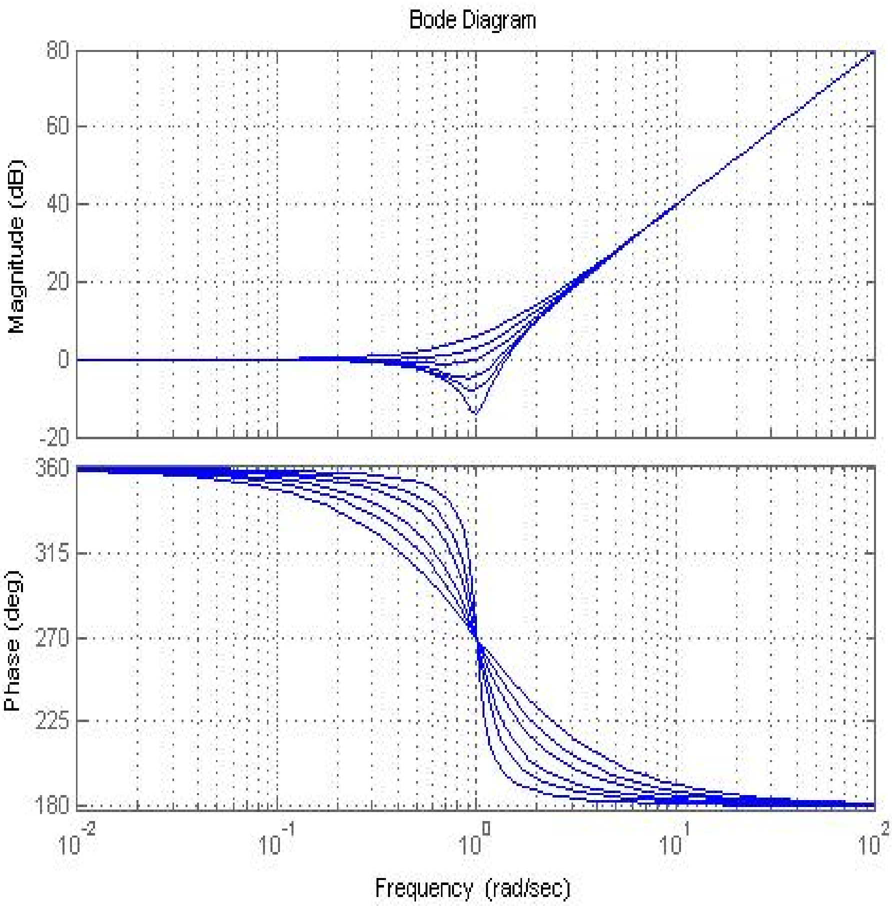


**Figure 8.2: The Bode plot of a non-minimum phase complex zero pair.**

### 8.2.3 General properties of non-minimum phase zeros

For a plant containing only non-minimum phase zeros, the phase of the plant at 0 rad/s is

$$\measuredangle P(0) = 180^{o} \cdot n$$

where:

$n$ is the number of non-minimum phase zeros (complex or real).

The phase of the plant as the frequency tends to infinity is

$$\measuredangle P(\infty) = +90^{o} \cdot e$$

where:

$e$ is the pole-zero excess of the plant.

The magnitude of the plant remains unchanged from a plant which has minimum phase zeros instead of the non-minimum phase zeros at the same frequencies.

## *8.3 Modifying the Bode ideal characteristic*

Incorporating the non-minimum phase zeros into the loop transmission will now be discussed. As in the case of the unstable plant the all-pass function will be defined (Horowitz,1993).

### 8.3.1 The all-pass function

The all-pass function is defined as

$$H_{AP}(s) = \frac{(s+\bar{z}_1) \cdot (s+\bar{z}_2) \ldots (s+\bar{z}_{n-1}) \cdot (s+\bar{z}_n)}{(s-z_1) \cdot (s-z_2) \ldots (s-z_{n-1}) \cdot (s-z_n)}$$

where:

$z_{1,} z_{2,} \ldots, z_n$ are the non-minimum phase zeros of the nominal plant.

The numerator contains the non-minimum phase zeros of the nominal plant and the denominator their negative complex conjugates. The zeros may be complex, but for a practical system the all-pass function will always contain complex-conjugate pairs and not just a single complex zero.

The magnitude of the all-pass function is unity for all frequencies which means that only the phase need be considered.

Figure 8.3 shows the phase of the all-pass function for a non-minimum phase real zero and Figure 8.4 the phase of the all-pass function for a number of non-minimum phase complex zero pairs with different damping ratios at a natural frequency of 1 rad/s.

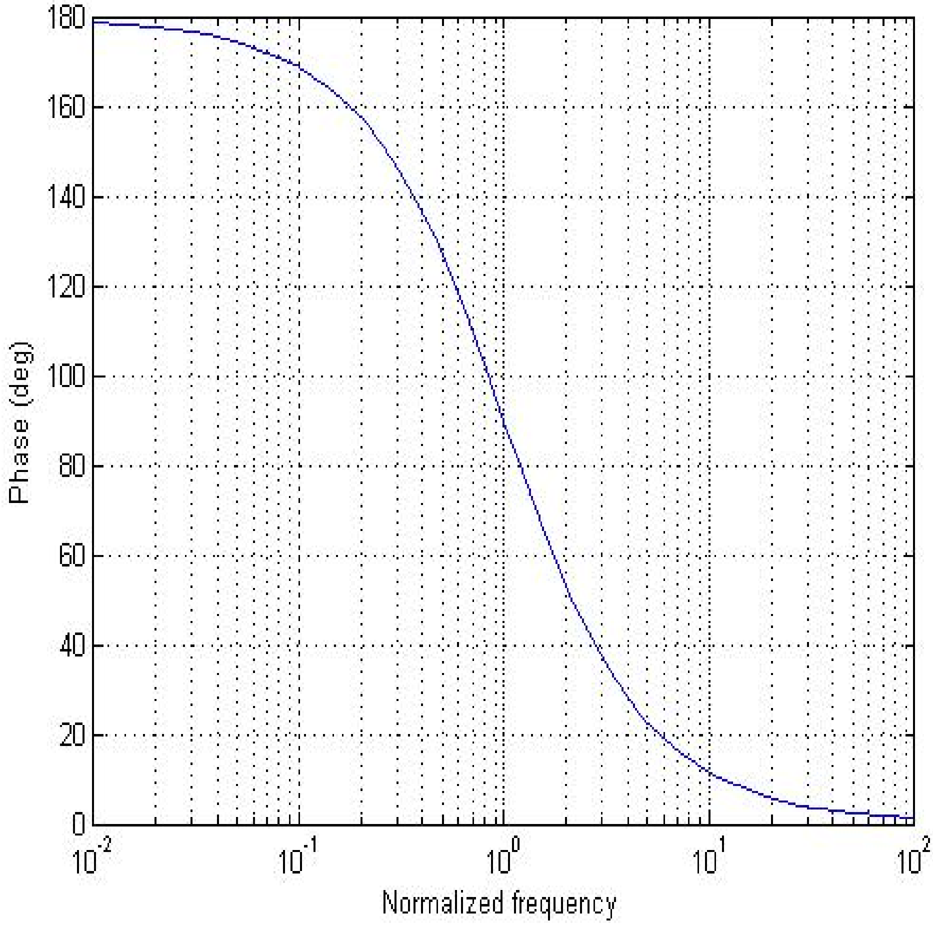


**Figure 8.3: The all-pass function for a non-minimum phase zero.**

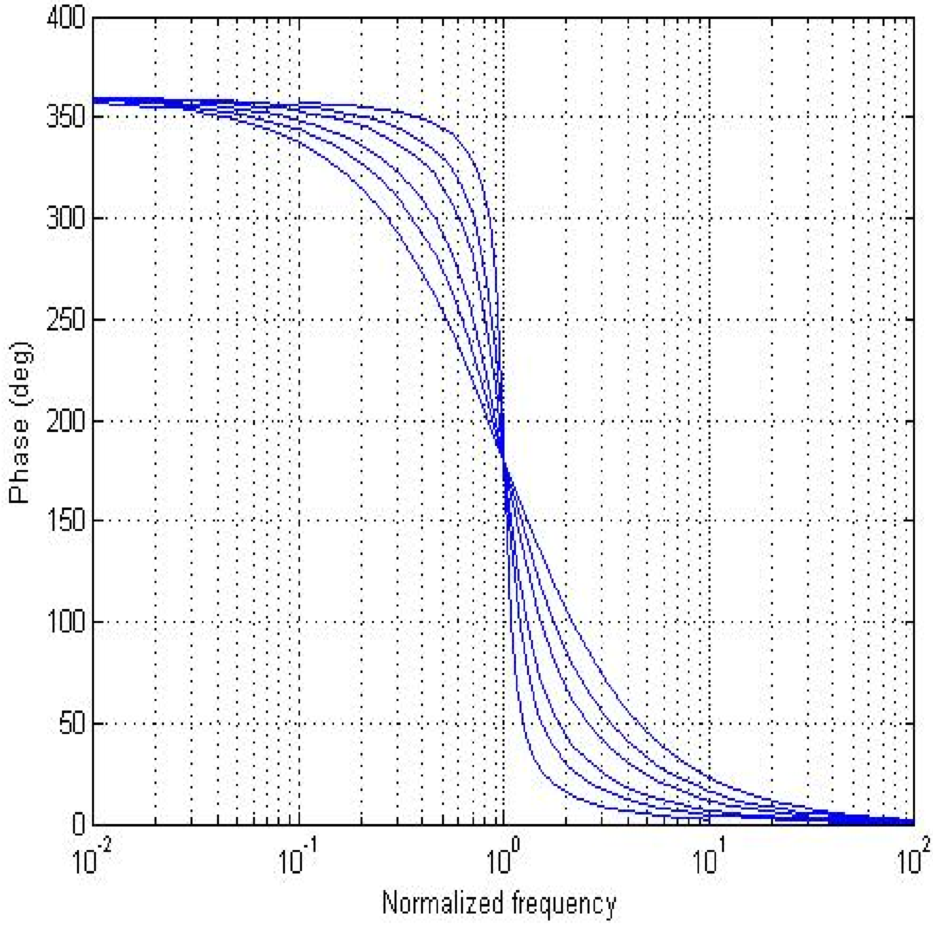


**Figure 8.4: The all-pass function for a non-minimum phase complex zero pair.**

### 8.3.2 Modifying the nominal loop transmission to take the non-minimum phase zeros into account

The complete Bode ideal characteristic can be modified to allow for a plant with non-minimum phase zeros by adding the all-pass function to the Bode ideal characteristic as in the case of the unstable plant. The problem with this is that the nominal loop transmission will not start at a phase of $0^o$ or a multiple of $360^o$. This however is easily remedied by making the compensator have a negative sign for a plant with an odd number of non-minimum phase zeros (Horowitz,1993). The compensator will then have the following sign.

$$G_{SIGN} = (-1)^n$$

where:

$n$ is the number of non-minimum phase zeros (complex or real).

The simplest solution for the design is to normalise the all-pass function such that it starts at 0° (Horowitz,1993). In other words the all pass-function is modified as follows

$$\measuredangle H_{APN}(s) = \measuredangle H_{AP}(s) - 180^o \cdot n$$

The nominal loop transmission then becomes

$$|L_0(s)| = |H_{TBI}(s)|$$

$$\measuredangle L_0(s) = \measuredangle H_{TBI}(s) + \measuredangle H_{APN}(s)$$

Now that the nominal loop transmission has been defined, the value of its parameters must be determined such that they meet the specifications that have been specified.

## 8.4 Relating the parameters of the modified Bode ideal characteristic to the specifications

As in the unstable plant case the bounds which were calculated for the nominal loop transmission (see section 5) can simply be modified by taking the all-pass function into account. So the design procedure will be exactly the same as the unstable case except the all-pass function will modify the high frequency bounds and hardly change the low frequency bounds.

The nominal loop transmission bounds are modified by shifting them in phase by the negative value of the all-pass function. This means that the bounds at high frequency will be shifted higher in phase and the low frequency bounds will not be shifted much at all.

As before the bounds which can be modified in this manner are:

1. The specifications on tracking the command input.
2. The relative stability specifications (gain margin, phase margin and resonant peak).
3. The speed of response (bandwidth).

The steady state error specifications for both the steady state response to the command and disturbance inputs cannot however be modified in this manner and will be recalculated in the following section.

The high frequency properties of the closed loop response, such as the magnitude and roll-off of the closed loop magnitude, are not effected at all by the all-pass function as these are only concerned with the magnitude, which the all-pass function does not effect.

### 8.4.1 Steady state error

#### 8.4.1.1 The steady state error due to the command input for a system with a Bode ideal characteristic loop transmission

As discussed in section 3.4.1.4, the steady state error can be calculated using the final value theorem. For a unity feedback system with power of time inputs, an error coefficient can be defined which in turn can be used to calculate the steady state error.

Using the error coefficients defined for the steady state error due to a power of time input

$$K_i = \lim_{s \to 0} s^i \cdot L(s)$$

where the loop transmission is given by

$$L(s) = H_{TBI}(s) = \frac{\omega_0^k}{s^k} \cdot H_{BI}(s) \cdot H_{APN}(s)$$

where:

k is the desired system type

Then the error coefficient becomes

$$K_i = \lim_{s \to 0} \left[ s^{i-k} \cdot \omega_0^k \cdot H_{BI}(s) \cdot H_{AP}(s) \right]$$

As $s \to 0$ the Bode ideal characteristic, $H_{BI}(0) = M_0$ and the normalised all-pass function tends to

$$H_{AP}(0) = 1$$

where

$n$ is the number of non-minimum phase zeros.

So the error coefficient will the same as in the stable minimum phase case. So the steady state errors can be summarised in Table 7:

| ***Type of L(s)*** | ***Steady state error to a step input*** $r(t)=A\cdot u(t)$ | ***Steady state error to a ramp input*** $r(t)=A\cdot t$ | ***Steady state error to a parabolic input*** $r(t)=A\cdot t^2$ |
|---|---|---|---|
| 0 | $\frac{A}{1+M_0}$ | $\infty$ | $\infty$ |
| 1 | 0 | $\frac{A}{M_0\cdot\omega_0^k}$ | $\infty$ |
| 2 | 0 | 0 | $\frac{A}{M_0\cdot\omega_0^k}$ |
| 3 | 0 | 0 | 0 |

**Table 7: The steady state error due to a power of time command input for a system with the Bode ideal characteristic as the loop transmission and non-minimum phase zeros.**

#### 8.4.1.2 The steady state error due to a disturbance input for a system with a Bode ideal characteristic loop transmission

As discussed in the previous section the error coefficient for the non-minimum phase plant is the same as the stable minimum phase case. So the steady state responses to disturbance inputs can be summarised in Table 8:

| Type of L(s) | Steady state error to a step input $r(t)=A\cdot u(t)$ | Steady state error to a ramp input $r(t)=A\cdot t$ | Steady state error to a parabolic input $r(t)=A\cdot t^2$ |
|---|---|---|---|
| 0 | $\frac{A}{1+M_0}\cdot P_D(0)$ | $\infty$ | $\infty$ |
| 1 | 0 | $\frac{A}{M_0\cdot\omega_0^k}\cdot P_D(0)$ | $\infty$ |
| 2 | 0 | 0 | $\frac{A}{M_0\cdot\omega_0^k}\cdot P_D(0)$ |
| 3 | 0 | 0 | 0 |

**Table 8: The steady state error due to a power of time disturbance input for a system with the Bode ideal characteristic as the loop transmission and non-minimum phase zeros.**

### 8.4.2 The trade offs between the parameter values which should be considered

The trade offs discussed for the design for stable minimum phase plants should be considered in this case (See section 6.3.5). The trade-offs are now further complicated by the noramlised all-pass function containing the non-minimum phase zeros. The complication is that the phase margin parameter, $\phi_{PM}$ , must be increased as the crossover frequency of the Bode ideal characteristic increases. This implies a further trade-off between the phase margin and the bandwidth of the system. This is due to the fact that if the compensator is too quick, the plant output will be delayed enough to force the compensator to over compensate for the error and therefore cause oscillations or even instability.

## *8.5 The design procedure*

The design procedure for plants with non-minimum phase zeros is almost exactly the same as for the plant with unstable poles, which was discussed in the example in section 7.5. The only difference is that the sign of the loop compensator is determined by the number of non-minimum phase zeros. In section 8.3.2, it was shown that if there are an odd number of non-minimum phase zeros, the loop compensator must have a negative sign.

If this is taken into account the rest of the design procedure used for the plant with unstable poles can be used to design for a plant with non-minimum phase zeros. The bounds placed on the loop transmission are also shifted by the phase of the all-pass function as before.

# 9 Conclusion

The goal of this project report was to develop a method that was capable of:

1. Evaluating a set of specifications placed on a control system for consistency.
2. Generate the specifications that are missing, but required by the control system design algorithm which is to be used. Further, these specifications must be consistent with the other specifications placed on the system as well as the plant dynamics.

A method using the Bode ideal characteristic as a nominal loop transmission was developed in order to generate prototype models for use in specifying control system bounds. The goals were:

1. To generate the prototype models which meet the specifications that were provided.
2. To provide the designer with a means of trading off the control loop characteristics which are not covered by the specifications.
3. To bring the plant characteristics into the generation and evaluation of the prototype models.

The Bode ideal characteristic provides a good estimate of the required nominal loop transmission as it has the following properties:

- It has high gain at low frequencies. This high gain is required to meet the tracking, steady state error to a command input and steady state performance to a disturbance input specifications.
- At high frequencies it has a low gain so that any high frequency sensor noise is attenuated and the high frequency plant uncertainty, which often contains unmodelled dynamics, does not effect the stability or performance of the system.
- It has a definable relative stability which allows it to meet the gain margin, phase margin and/or resonant peak specifications.

It also allows trade-offs between the open loop low frequency magnitude, the bandwidth, the closed loop high frequency magnitude and the relative stability of the system to be evaluated. These trade-offs can be evaluated for stable plants, and plants containing unstable poles or non-minimum phase zeros.

The characteristics of the plant is also catered for by using the plant uncertainty to expand the nominal performance of the closed loop system, as provided by the Bode ideal characteristic, into a set of prototype models which give a variation in performance which is realistic for the plant under consideration. The limits on performance of the plant were also evaluated through the generation of the input signals as seen by the plant input.

The only problem that was encountered using the Bode ideal characteristic was the high frequency transients it generated in the plant input signal. This was due to the piece wise linear nature of the Bode ideal characteristic near cross-over. But as the goal was to generate prototype models to be used as inputs to the control algorithm design rather than the generation of the control algorithm itself, this does not cause any major problem as these transients can be ignored. This is achieved by smoothing the bounds (the weighting functions in H-infinity and the upper and lower bounds on the closed loop response in QFT) generated from the prototype models which are used by the design process.

# 10 Appendix A – Relating the open loop and closed loop frequency responses

## 10.1 Relating the closed loop magnitude to the open loop magnitude and phase

The closed loop frequency response is related to the loop transmission by:

$$T(s) = \frac{L(s)}{1 + L(s)}$$

**Equation 10.1**

The closed loop magnitude is given by:

$$|T(s)| = \frac{|L(s)|}{\left|1 + |L(s)| e^{j \measuredangle L(s)}\right|}$$

Using Euler's equation, $e^{j\phi} = \cos\phi + j\sin\phi$ , the closed loop magnitude can be written as:

$$|T(s)| = \frac{|L(s)|}{\left|1 + |L(s)| \cos \measuredangle L(s) + j\, |L(s)| \sin \measuredangle L(s)\right|}$$

$$|T(s)| = \frac{|L(s)|}{\sqrt{1 + 2\,|L(s)| \cos \measuredangle L(s) + |L(s)^2| (\cos^2 \measuredangle L(s) + \sin^2 \measuredangle L(s))}}$$

As $\cos^2\phi + \sin^2\phi = 1$ , the closed loop magnitude can be written as:

$$|T(s)| = \frac{|L(s)|}{\sqrt{1 + 2\,|L(s)| \cos \measuredangle L(s) + |L(s)|^2}}$$

**Equation 10.2**

## 10.2 Relating the open loop magnitude to the closed loop magnitude and the open loop phase

Squaring Equation 10.2 gives:

$$|T(s)|^2 \cdot \left(1 + 2 \cdot |L(s)| \cos \sphericalangle L(s) + |L(s)|^2\right) = |L(s)|^2$$

$$1 + 2 \cdot |L(s)| \cos \sphericalangle L(s) + \left(1 - \frac{1}{|T(s)|^2}\right) |L(s)|^2 = 0$$

Let $\beta = 1 - \frac{1}{|T(s)|^2}$ then the equation may be simplified to,

$$1 + 2 \cdot |L(s)| \cos \sphericalangle L(s) + \beta \cdot |L(s)|^2 = 0$$

**Equation 10.3:**

The roots of the equation are then given by,

$$|L(s)| = \frac{1}{\beta} \cdot \left(-\cos(\sphericalangle L(s)) \pm \sqrt{\cos^2(\sphericalangle L(s)) - \beta}\right)$$

**Equation 10.4**

From Equation 10.4 it can be seen that the open loop magnitude is dependent on $\beta$ (which is dependent on the closed loop magnitude) and the open loop phase. It is very useful to relate the open loop magnitude to the open loop phase for a constant $\beta$ (closed loop magnitude).

Also Equation 10.4 gives an infinite result for $\beta = 0$ , which is incorrect. The correct equation may be found by substituting $\beta = 0$ into Equation 10.3 , which gives

$$1 + 2 \cdot |L(s)| \cos \sphericalangle L(s) = 0$$

Solving for $|L(s)|$ gives,

$$\left|L(s)\right| = \frac{-1}{2 \cdot \cos \sphericalangle L(s)}$$

**Equation 10.5:**

In summary the open loop magnitude is given by,

$$\left|L(s)\right| = \begin{cases} \frac{1}{\beta} \cdot \left(-\cos \sphericalangle L(s) \pm \sqrt{\cos \sphericalangle L(s) - \beta}\right) & for \quad \beta < 0 \\ \frac{-1}{2 \cdot \cos \sphericalangle L(s)} & for \quad \beta = 0 \\ \frac{1}{\beta} \cdot \left(-\cos \sphericalangle L(s) + \sqrt{\cos \sphericalangle L(s) - \beta}\right) & for \quad \beta > 0 \end{cases}$$

**Equation 10.6:**

## 10.3 Relating the open loop phase to the open loop magnitude and closed loop magnitude

Starting with Equation 10.3,

$$1 + 2 \cdot \left|L(s)\right| \cdot \cos \sphericalangle L(s) + \beta \cdot \left|L(s)\right|^2 = 0$$

Solving for the closed loop phase gives,

$$\sphericalangle L(s) = \cos^{-1}\left(\frac{-1 - \beta \cdot \left|L(s)\right|^2}{2 \cdot \left|L(s)\right|}\right)$$

**Equation 10.7:**

## 10.4 Finding the minimum and maximum open loop magnitudes for a specific closed loop magnitude

The maximum and minimum magnitudes of the m-circle for the case where $\beta < 0$ ( $|T(s)| > 1$ ) is calculated by differentiating Equation 10.6 with respect to the open loop phase and setting it to zero. The derivative is given by

$$\frac{d|L(s)|}{d\sphericalangle L(s)} = \frac{1}{\beta} \cdot \sin\sphericalangle L(s) \cdot \left[ 1 \mp \frac{2 \cdot \cos\sphericalangle L(s)}{\sqrt{\cos^2\sphericalangle L(s) - \beta}} \right]$$

Set the derivative to zero and solve for $\sphericalangle L(s)$ to find the phases at which the open loop magnitude is at its maximum or minimum values.

$$\frac{1}{\beta} \cdot \sin\sphericalangle L(s) = 0 \quad \text{or} \quad \left[ 1 \mp \frac{2 \cdot \cos\sphericalangle L(s)}{\sqrt{\cos^2\sphericalangle L(s) - \beta}} \right] = 0$$

$$\sin\sphericalangle L(s) = 0 \quad \text{or} \quad \cos^2\sphericalangle L(s) = \frac{-\beta}{3}$$

As the phase can not have a complex solution, the phase at the the minimum or maximum magnitude are given by

$$\sphericalangle L(s) = 0^o, \pm 180^o, \pm 360^o, \ldots$$

At these phase $\cos\sphericalangle L(s) = \pm 1$ , so the minimum open loop magnitudes is given when $\cos\sphericalangle L(s) = +1$ is substituted in Equation 10.2 which gives

$$|L(s)|_{MIN} = \frac{|T(s)|}{|T(s)| + 1}$$

and the maximum open loop magnitudes is given when $\cos\sphericalangle L(s) = -1$ is substituted in Equation 10.2 which gives

$$|L(s)|_{MAX} = \frac{|T(s)|}{|T(s)| - 1}$$

The maximum and minimum magnitudes of the m-circle for the case where $\beta > 0$ ( $|T(s)| < 1$ ) is calculated in the similar manner as the $\beta < 0$ case. Which gives the minimum open loop magnitude of

$$|L(s)|_{MIN} = \frac{|T(s)|}{1 + |T(s)|}$$

and the maximum open loop magnitudes of

$$|L(s)|_{MAX} = \frac{|T(s)|}{1 - |T(s)|}$$

The maximum and minimum magnitudes of the m-circle for the case where $\beta = 0$ ( $|T(s)| = 1$ ) is calculated by differentiating Equation 10.5 with respect to the open loop phase and setting it to zero. The derivative is given by

$$\frac{d|L(s)|}{d \sphericalangle L(s)} = -\frac{1}{2} \cdot \frac{\sin \sphericalangle L(s)}{\cos^2 \sphericalangle L(s)}$$

Solving for the open loop phase when the derivative is set to zero is given by

$$\sin \sphericalangle L(s) = 0$$

which gives

$$\sphericalangle L(s) = 0^o, \pm 180^o, \pm 360^o, \ldots$$

Substituting the open loop phases into Equation 10.5 gives

$$|L(s)| = \pm \frac{1}{2}$$

As the open loop magnitude must be positive the open loop magnitude in this case only has a minimum value of

$$|L(s)|_{MIN} = \frac{1}{2}$$

## 10.5 Finding the minimum and maximum open loop phases for a specific closed loop magnitude

Differentiating Equation 10.7 gives,

$$\frac{d \measuredangle L(s)}{d\,|L(s)|} = \frac{1-\beta\cdot|L(s)|^2}{|L(s)|\cdot\sqrt{4\cdot|L(s)|^2-\left(-1-\beta\cdot|L(s)|^2\right)}}$$

Set the derivative to zero and solve for $|L(s)|$ to find the magnitudes at which the open loop phase is at its maximum or minimum values.

$$1-\beta\cdot|L(s)|^2 = 0$$

$$|L(s)| = \frac{\pm 1}{\sqrt{\beta}}$$

But the magnitude is always positive, so solution is

$$|L(s)| = \frac{1}{\sqrt{\beta}}$$

**Equation 10.8:**

Substitute Equation 10.8 into equation Equation 10.7 to find the maximum or minimum phases,

$$\measuredangle L(s) = \cos^{-1}\left(-\sqrt{\beta}\right) = \cos^{-1}\left(\frac{\sqrt{|T(s)|^2-1}}{-|T(s)|}\right)$$

## 10.6 Relating the gain and phase margins specifications to the resonant peak specification

### 10.6.1 Relating the gain margin specifications to the resonant peak specification

The gain margin is defined at an open loop phase, $\measuredangle L(s)$, of -180$^o$ so $\cos\measuredangle L(s) = -1$. Then the closed loop magnitude is given by,

$$|T(s)| = \frac{|L(s)|}{\sqrt{1 - 2|L(s)| + |L(s)|^2}} = \frac{|L(s)|}{1 - |L(s)|}$$

Solving for the magnitude of the loop transmission gives,

$$|L(s)| = \frac{|T(s)|}{|T(s)| + 1}$$

### 10.6.2 Relating the phase margin specifications to the resonant peak specification

The phase margin is defined at the loop transmission magnitude of 1, so substituting $|T(s)| = 1$ in Equation 10.2 gives

$$|T(s)| = \frac{1}{\sqrt{2 + 2\cos\measuredangle L(s)}}$$

Solving for the open loop phase gives,

$$\measuredangle L(s) = \cos^{-1}\left(\frac{1}{2|T(s)|^2} - 1\right)$$

The phase margin is then given by

$$\phi_{PM} = \cos^{-1}\left(\frac{1}{2|T(s)|^2} - 1\right) + 180^o$$

# 11 Appendix B – Useful equations

## *11.1 Calculating the range of a point on a line given the angle*

The equation of a line is given by:

$$y = m \cdot x + c$$

**Equation 11.1:**

The constants of the equation, *m* and *c*, can be found if a set of two points on the line are known.

If the points are $(x_1, y_1)$ and $(x_2, y_2)$ then the constants are given by,

$$m = \frac{y_1 - y_2}{x_1 - x_2}$$

**Equation 11.2:**

and

$$c = \frac{x_1 \cdot y_2 - x_2 \cdot y_1}{x_1 - x_2}$$

**Equation 11.3:**

But the position of a point in polar co-ordinates relates to a point in cartesian co-ordinates as follows,

$$(x, y) = (r \cdot \cos(\theta), r \cdot \sin(\theta))$$

**Equation 11.4:**

Substituting Equation 11.4 into Equation 11.1 gives a formula for the line in polar co-ordinates as follows,

$$r \cdot \sin(\theta) = m \cdot r \cdot \cos(\theta) + c$$

Solving for *r*,

$$r = \frac{c}{\sin(\theta) - m \cdot \cos(\theta)}$$

**Equation 11.5:**

Substituting Equation 11.2 and Equation 11.3 into Equation 11.5 gives the equation of a line in polar c-ordinates given two points on the line in cartesian co-ordinates,

$$r = \frac{x_1 \cdot y_2 - x_2 \cdot y_1}{(x_1 - x_2) \cdot \sin(\theta) - (y_1 - y_2) \cdot \cos(\theta)}$$

**Equation 11.6:**

# 12 Appendix C – Matlab Code

Matlab code was generated so that the examples using the Bode ideal characteristic to evaluate the prototype models for a stable minimum phase plant and the unstable minimum phase plant could be given (See sections 6.4 and 7.5 respectively). The examples for the stable minimum phase plant and the unstable minimum phase plant are implemented in the Matlab m-files Stable_Example.m and Unstable_Example.m respectively.

As Matlabs' standard tool boxes did not have all the functionality required, extra code was necessary in order to implement the examples. The following extra functionality was required:

1. The functionality to generate an m-circle (See section 12.2).
2. The functionality to generate a rotated m-circle (See section 12.3).
3. The functionality to generate and evaluate the QFT bounds needed. The following classes implement the required functionality:
   (a) The plant template class (See section 12.4).
   (b) The stability bound class (See section 12.5).
   (c) The uncertainty bound class (See section 12.6).
4. The functionality to generate and evaluate a Bode ideal characteristic. This is implemented by the Bode ideal characteristic class (See section 12.10) which makes use of the following classes:
   (a) Bode semi-infinite characteristic class (See section 12.7).
   (b) Bode step class (See section 12.8).
   (c) Bode cut-off characteristic class (See section 12.9).
5. The functionality to generate and evaluate a second order model (See section 12.11).

The Matlab code generated in order to implement the examples is given an a CD at the back of this research report.

## 12.1 Architecture

The functionality discussed in the previous section was implemented using Matlab objects and functions.

Each class has a constructor which has the same name as the class, which is capable of creating a new instance of the class or a copy of an existing object of the same type.

When the constructor creates anew instance of a class, it will require the user to specify all the attribute values of the class required to allow all the methods to work correctly.

Each class also has a ***display*** function, which the user or Matlab can use to display the values of the attributes of the class.

There are also utility functions which supply functionality required by the classes which do not belong to a class, see section 12.12.

## *12.2 M-circle class*

### 12.2.1 Overview

The m-circle defines a locus of open loop magnitudes and phases which have a constant closed loop magnitude.

The m-circle class has the following functionality:

1. It is capable of calculating the following properties of the m-circle:
    - The maximum magnitude of the m-circle and the phase at which it occurs.
    - The minimum magnitude of the m-circle bound and the phase at which it occurs.
    - The maximum phase of the m-circle and the magnitude at which it occurs.
    - The minimum phase of the m-circle bound and the magnitude at which it occurs.
    - The gain margin that the loop transmission must have in order for the for the closed loop magnitude to have the maximum value as specified by the m-circle.
    - The phase margin that the loop transmission must have in order for the for the closed loop magnitude to have the maximum value as specified by the m-circle.
2. It is capable of calculating the minimum and maximum phases of the m-circle at a specified magnitude.
3. It is capable of calculating the minimum and maximum magnitudes of the m-circle at a specified phase.
4. It is capable of plotting the m-circle on the Nichols chart.

### 12.2.2 Attributes

In order for the m-circle class to be capable of performing the above mentioned functionality it must keep track of the following attribute:

1. $|T(s)|$ , the closed loop magnitude of the m-circle. This attribute is stored as an absolute value. As the user specifies the desired closed loop magnitude of the m-circle in decibels, it is converted to an absolute value before being stored.

### 12.2.3 Methods

#### 12.2.3.1 calcMag

This method calculates the magnitudes of the m-circle given a vector of phases for which the magnitudes are required.

**Inputs**

The algorithm requires the following input:

$\measuredangle L(s)$ , the vector of open loop phases of the m-circle, at which the open loop magnitude is required (deg).

**Outputs**

The algorithm produces the following output:

$|L(s)|$ , the open loop magnitude of the m-circle (dB).

**Pre-conditions**

The phases specified are checked to see if they are in the acceptable range of values for the m-circle. Any phases which are outside this range have their corresponding magnitudes assigned the value *NaN* (not a number).

The acceptable range is calculated by calling the m-circle functions ***getMinPhase*** and ***getMaxPhase*** functions (see sections 12.2.3.6 and 12.2.3.7 respectively).

**Algorithm**

The open loop magnitude, $|L(s)|$ , is calculated from the following formula (see 10.1 for the derivation of the formula),

$$|L(s)| = \begin{cases} \frac{1}{\beta} \cdot \left( -\cos \measuredangle L(s) \pm \sqrt{\cos \measuredangle L(s) - \beta} \right) & for \quad \beta < 0 \\ \frac{-1}{2 \cdot \cos \measuredangle L(s)} & for \quad \beta = 0 \\ \frac{1}{\beta} \cdot \left( -\cos \measuredangle L(s) + \sqrt{\cos \measuredangle L(s) - \beta} \right) & for \quad \beta > 0 \end{cases}$$

where

$$\beta = 1 - \frac{1}{|T(s)|^2}$$

$|T(s)|$ is the attribute which stores the closed loop magnitude of the m-circle.

The open loop magnitude, $|L(s)|$, is converted to decibels before being returned.

## 12.2.3.2 calcPhase

This method calculates the phases of the m-circle given a vector magnitudes for which the phases are required.

**Inputs**

The algorithm requires the following inputs:

$|L(s)|$, the open loop magnitude of the m-circle (dB).

**Outputs**

The algorithm produces the following output:

$\measuredangle L(s)$, the open loop phase of the m-circle (deg).

**Pre-conditions**

The magnitudes specified are checked to see if they are in the acceptable range of values for the m-circle. Any magnitudes which are outside this range have the corresponding phases assigned the value *NaN* (not a number).

The acceptable range is calculated by calling the m-circle functions ***getMinMag*** and ***getMaxMag*** functions (see sections 12.2.3.4 and 12.2.3.5 respectively).

**Algorithm**

The open loop magnitude, $\measuredangle L(s)$, is calculated from the following formula (see 10.3 for the derivation of the formula), after the open loop magnitude inputs, $|L(s)|$, are converted to an absolute value:

$$\measuredangle L(s) = \cos^{-1}\left(\frac{-1-\beta\cdot|L(s)|^2}{2\cdot|L(s)|}\right)$$

where

$$\beta = 1 - \frac{1}{|T(s)|^2}$$

$|T(s)|$ is the attribute which stores the closed loop magnitude of the m-circle.

### 12.2.3.3getClosedLoopMag

This method retrieves the value of the closed loop magnitude of the m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$|T(s)|$, the closed loop magnitude of the m-circle (dB).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The value of the closed loop magnitude attribute, $|T(s)|$, is returned.

### 12.2.3.4 getMinMag

This method calculates the minimum magnitude of the m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$|L(s)|_{MIN}$ , the minimum open loop magnitude of the m-circle (dB).

$\measuredangle L(s)_{|L(s)|=|L(s)|_{MIN}}$ , the phase of the m-circle at the minimum magnitude (deg).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The minimum open loop magnitude, $|L(s)|_{MIN}$, is calculated from the following formula (see section 10.4 for the derivation of the formula),

$$|L(s)|_{MIN} = 20 \cdot \log\left(\frac{|T(s)|}{|T(s)|+1}\right)$$

where

$|T(s)|$ is the attribute which stores the closed loop magnitude of the m-circle.

And the open loop phase at which the magnitude is a minimum is given by,

$$\measuredangle L(s)\big|_{|L(s)|=|L(s)|_{MIN}} = -180^o$$

## 12.2.3.5 getMaxMag

This method calculates the maximum magnitude of the m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$|L(s)|_{MAX}$ , the maximum open loop magnitude m-circle (dB).

$\measuredangle L(s)\big|_{|L(s)|=|L(s)|_{MAX}}$ , the phase/s of the m-circle at the maximum magnitude (deg).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The maximum open loop magnitude, $|L(s)|_{MAX}$ , is calculated from the following formula (see section 10.4 for the derivation of the formula),

$$|L(s)|_{MAX} = \begin{cases} \dfrac{|T(s)|}{1-|T(s)|} & for \quad |T(s)|<1 \\ \infty & for \quad |T(s)|=1 \\ \dfrac{|T(s)|}{|T(s)|-1} & for \quad |T(s)|>1 \end{cases}$$

where

$|T(s)|$ is the attribute which stores the closed loop magnitude of the m-circle.

And the open loop phase at which the magnitude is a maximum is given by,

$$\measuredangle L(s)_{|L(s)|=|L(s)|_{MAX}} = \begin{cases} 0^o & for \quad |T(s)|<1 \\ [-270^o \quad -90^o] & for \quad |T(s)|=1 \\ -180^o & for \quad |T(s)|>1 \end{cases}$$

The maximum open loop magnitude, $|L(s)|_{MAX}$, calculated is converted to decibels before being returned.

### 12.2.3.6 getMinPhase

This method calculates the minimum phase of the m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$\measuredangle L(s)_{MIN}$, the minimum open loop phase of the m-circle (deg).

$|L(s)|_{\measuredangle L(s)=\measuredangle L(s)_{MIN}}$, the open loop magnitude at the minimum open loop phase (dB).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The minimum open loop phase, $\measuredangle L(s)_{MIN}$, is calculated from the following formula (see section 10.5 for the derivation of the formula),

$$\measuredangle L(s)_{MIN} = \begin{cases} NaN & for \quad |T(s)| < 1 \\ -270^o & for \quad |T(s)| = 1 \\ -360^o + \cos^{-1}\left(\frac{\sqrt{|T(s)|^2 - 1}}{-|T(s)|}\right) & for \quad |T(s)| > 1 \end{cases}$$

where

$|T(s)|$ is the attribute which stores the closed loop magnitude of the m-circle.

And the open loop phase at which the magnitude is a maximum is given by,

$$|L(s)|_{\measuredangle L(s) = \measuredangle L(s)_{MIN}} = \begin{cases} NaN & for \quad |T(s)| < 1 \\ \infty & for \quad |T(s)| = 1 \\ 20 \cdot \log\left(\frac{|T(s)|}{\sqrt{|T(s)|^2 - 1}}\right) & for \quad |T(s)| > 1 \end{cases}$$

### 12.2.3.7 getMaxPhase

This method calculates the maximum phase of the m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$\measuredangle L(s)_{MAX}$ , the maximum open loop phase (deg).

$|L(s)|_{\measuredangle L(s) = \measuredangle L(s)_{MAX}}$ , the open loop magnitude at the maximum open loop phase (dB).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The maximum open loop phase, $\measuredangle L(s)_{MAX}$, is calculated from the following formula (see section 10.5 for the derivation of the formula),

$$\measuredangle L(s)_{MAX} = \begin{cases} NaN & for \quad |T(s)| < 1 \\ -90^{o} & for \quad |T(s)| = 1 \\ \cos^{-1}\left( \dfrac{\sqrt{|T(s)|^{2}-1}}{-|T(s)|} \right) & for \quad |T(s)| > 1 \end{cases}$$

where

$|T(s)|$ is the attribute which stores the closed loop magnitude of the m-circle.

And the open loop phase at which the magnitude is a maximum is given by,

$$|L(s)|_{\measuredangle L(s) = \measuredangle L(s)_{MAX}} = \begin{cases} NaN & for \quad |T(s)| < 1 \\ \infty & for \quad |T(s)| = 1 \\ 20 \cdot \log\left( \dfrac{|T(s)|}{\sqrt{|T(s)|^{2}-1}} \right) & for \quad |T(s)| > 1 \end{cases}$$

## 12.2.3.8 getGainMargin

This method calculates the gain margin that the loop transmission must have in order for the for the closed loop magnitude to have the maximum value as specified by the m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$\left|L(s)\right|_{GM}$ , the gain margin related to the m-circle (dB).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The gain margin is simply the negative value of the minimum magnitude of the m-circle (in decibels). The m-circles' ***getMinMag*** function is used to calculate the minimum magnitude of the m-circle (see section 12.2.3.4).

### 12.2.3.9 getPhaseMargin

This method calculates the phase margin that the loop transmission must have in order for the for the closed loop magnitude to have the maximum value as specified by the m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$\measuredangle L(s)_{PM}$ , the phase margin related to the m-circle.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

For m-circles with closed loop magnitudes greater than or equal to unity (0 dB), the phase margin is simply the phase of the m-circle at 0 dB plus 180$^{o}$. The m-circles' ***calcPhase*** function is used to calculate the minimum phase of the m-circle (see section ) by passing it an open loop magnitude of 0 dB.

For m-circles with closed loop magnitudes smaller than unity (0 dB), not a number,*NaN*, is returned as a phase margin in this case makes little sense.

#### 12.2.3.10 nichols

This function plots the m-circle in Nichols chart format.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The acceptable range of magnitudes for the m-circle is calculated by calling the m-circle classes functions ***getMinMag*** and ***getMaxMag*** functions (see sections 12.2.3.4 and 12.2.3.5 respectively).

This range of magnitudes is sampled and phases relating to these magnitudes are then calculated using the m-circle classes ***calcPhase*** function (see section 12.2.3.2).

Matlabs plotting routines are then used to plot the m-circle using the magnitudes and phases calculated.

## *12.3 The rotated M-circle class*

### 12.3.1 Overview

As discussed in section 5.2, the rotated m-circle defines a locus of open loop magnitudes and phases which have a constant magnitude for the sensitivity function. The loop transmission of the rotated m-circle, $l(s)$, is related to the loop transmission of the m-circle, $L(s)$, by

$$l(s)=\frac{1}{L(s)}$$

Then the magnitude of the rotated m-circle is related to the m-circle's magnitude by

$$|l(s)|=\frac{1}{|L(s)|}$$

The code only calculates the phase of the m-circle and rotated m-circles in the range from greater than -360° to 0°. As the rotated m-circle's phase is simply the negative of the m-circles phase and the rotated m-circle is symmetrical around -180°, the rotated m-circle's phase and the m-circle's phase are equal in this range.

These relationships are used by the rotated m-circle class to provide the functionality required of it, by making use of the m-circle classes' functionality.

The rotated m-circle class provides the following functionality:

1. It is capable of calculating the following properties of the rotated m-circle:
   - The maximum magnitude of the rotated m-circle and the phase at which it occurs.
   - The minimum magnitude of the rotated m-circle bound and the phase at which it occurs.
   - The maximum phase of the rotated m-circle and the magnitude at which it occurs.
   - The minimum phase of the rotated m-circle bound and the magnitude at which it occurs.
   - The gain margin that the loop transmission must have in order for the for the closed loop magnitude to have the maximum value as specified by the rotated m-circle.
   - The phase margin that the loop transmission must have in order for the for the closed loop magnitude to have the maximum value as specified by the rotated m-circle.
2. It is capable of calculating the minimum and maximum phases of the rotated m-circle at a specified magnitude.

3. It is capable of calculating the minimum and maximum magnitudes of the rotated m-circle at a specified phase.
4. It is capable of plotting the rotated m-circle on the Nichols chart.

## 12.3.2 Attributes

In order for the rotated m-circle class to be capable of performing the above mentioned functionality it must keep track of the following attribute:

1. An instance of the MCircle class (see section 12.2).

## 12.3.3 Methods

### 12.3.3.1calcMag

This method calculates the magnitudes of the rotated m-circle given a vector of phases for which the magnitudes are required.

**Inputs**

The algorithm requires the following input:

$\measuredangle l(s)$ , the vector of open loop phases of the rotated m-circle, at which the open loop magnitude is required (deg).

**Outputs**

The algorithm produces the following output:

$|l(s)|$ , the open loop magnitude of the rotated m-circle (dB).

**Pre-conditions**

The ***calcMag*** method of the m-circle class, which is an attribute of this class, is used to check the phases specified. See section 12.2.3.1 for a discussion on the ***calcMag*** method.

**Algorithm**

The open loop magnitude of the m-circle, $|L(s)|$, is calculated using the ***calcMag*** method of the m-circle class (see section 12.2.3.1).

The magnitude of the rotated m-circle is then calculated from the m-circles open loop magnitude as follows

$$|l(s)| = \frac{1}{|L(s)|}$$

### 12.3.3.2calcPhase

This method calculates the phases of the rotated m-circle given a vector magnitudes for which the phases are required.

**Inputs**

The algorithm requires the following input:

$|l(s)|$, the open loop magnitude of the rotated m-circle (dB).

**Outputs**

The algorithm produces the following output:

$\sphericalangle l(s)$, the open loop phase of the rotated m-circle (deg).

**Pre-conditions**

The ***calcPhase*** method of the m-circle class, which is an attribute of this class, is used to check the magnitudes specified. See section 12.2.3.2 for a discussion on the ***calcPhase*** method.

**Algorithm**

The vector of rotated m-circle magnitudes first needs to be transformed into a vector of m-circle class magnitudes as follows

$$|L(s)| = \frac{1}{|l(s)|}$$

The open loop magnitude, $\measuredangle l(s)$ , is calculated using the ***calcMag*** method of the m-circle class (see section 12.2.3.2).

### 12.3.3.3getMinMag

This method calculates the minimum magnitude of the rotated m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$|l(s)|_{MIN}$ , the minimum open loop magnitude/s (dB).

$\measuredangle l(s)_{|l(s)|=|l(s)|_{MIN}}$ , the phase/s of the rotated m-circle at the minimum magnitude (deg).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

As the magnitude of the rotated m-circle is the inverse of the m-circle's magnitude, the m-circle classes' ***getMaxMag*** method (see section 12.2.3.5) will give the inverse of the minimum magnitude. The minimum magnitude (in decibels) is then calculated from

$$|l_{MIN}(s)|=-|L_{MAX}(s)|$$

The phases at the minimum magnitude are given by the ***getMaxMag*** method.

### 12.3.3.4getMaxMag

This method calculates the maximum magnitude of the m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$|l(s)|_{MAX}$ , the maximum open loop magnitude (dB).

$\measuredangle l(s)_{|l(s)|=|l(s)|_{MAX}}$ , the phase/s of the rotated m-circle at the maximum magnitude (deg).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

As the magnitude of the rotated m-circle is the inverse of the m-circle's magnitude, the m-circle classes' ***getMinMag*** method (see section 12.2.3.4) will give the inverse of the maximum magnitude. The maximum magnitude (in decibels) is then calculated from

$$|l_{MAX}(s)|=-|L_{MIN}(s)|$$

The phases at the maximum magnitude are given by the ***getMinMag*** method.

## 12.3.3.5getMinPhase

This method calculates the minimum phase of the rotated m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$\measuredangle l(s)_{MIN}$ , the minimum open loop phase of rotated m-circle (deg).

$|l(s)|_{\measuredangle l(s)=\measuredangle l(s)_{MIN}}$ , the open loop magnitude at the minimum open loop phase (dB).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The minimum phase of the rotated m-circle, $\measuredangle l(s)_{MIN}$ , is given be the m-circle classes' ***getMinPhase*** method, see section 12.2.3.6. The magnitude, in decibels, at which the minimum phase occurs for the rotated m-circle is given by

$$|l(s)|_{\measuredangle l(s)=\measuredangle l(s)_{MIN}} = -|L(s)|_{\measuredangle L(s)=\measuredangle L(s)_{MIN}}$$

where $|L(s)|_{\measuredangle L(s)=\measuredangle L(s)_{MIN}}$ is the magnitude at which the minimum phase occurs for the m-circle, which is returned by the m-circle classes' ***getMinPhase*** method.

### 12.3.3.6 getMaxPhase

This function calculates the maximum phase of the m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$\measuredangle L(s)_{MAX}$ , the maximum open loop phase.

$|L(s)|_{\measuredangle L(s)=\measuredangle L(s)_{MAX}}$ , the open loop magnitude at the maximum open loop phase.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The maximum phase of the rotated m-circle, $\sphericalangle l(s)_{MAX}$ , is given be the m-circle classes' ***getMaxPhase*** method, see section 12.2.3.7. The magnitude, in decibels, at which the maximum phase occurs for the rotated m-circle is given by

$$|l(s)|_{\sphericalangle l(s)=\sphericalangle l(s)_{MAX}}=-|L(s)|_{\sphericalangle L(s)=\sphericalangle L(s)_{MAX}}$$

where $|L(s)|_{\sphericalangle L(s)=\sphericalangle L(s)_{MAX}}$ is the magnitude at which the minimum phase occurs for the m-circle, which is returned by the m-circle classes' ***getMaxPhase*** method.

## 12.3.3.7getGainMargin

This function calculates the gain margin that the loop transmission must have in order for the for the closed loop magnitude to have the maximum value as specified by the rotated m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$|l(s)|_{GM}$ , the gain margin related to the rotated m-circle (dB).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The gain margin is simply the negative value of the minimum magnitude of the rotated m-circle (in decibels). The rotated m-circle's ***getMinMag*** function is used to calculate the minimum magnitude of the rotated m-circle (see section 12.3.3.3) for rotated m-circles with a closed loop magnitude less than or equal to 0 dB.

For rotaed m-circles with closed loop magnitudes smaller than 0 dB, not a number,*NaN*, is returned as a gain margin in this case makes little sense.

## 12.3.3.8getPhaseMargin

This method calculates the phase margin that the loop transmission must have in order for the for the closed loop magnitude to have the maximum value as specified by the rotated m-circle.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$\sphericalangle l(s)_{PM}$ , the phase margin related to the rotated m-circle (deg).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

For rotated m-circles with closed loop magnitudes greater than or equal to unity, the phase margin is simply the maximum phase of the rotated m-circle (in degrees) plus $180^{o}$. The rotated m-circle's ***getMaxPhase*** function is used to calculate the maximum phase of the rotated m-circle(see section 12.3.3.6).

For m-circles with closed loop magnitudes smaller than 0 dB, not a number,*NaN*, is returned as a phase margin in this case makes little sense.

## 12.3.3.9nichols

This method plots the rotated m-circle in Nichols chart format.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The acceptable range of magnitudes for the rotated m-circle is calculated by calling the rotated m-circle classes functions ***getMinMag*** and ***getMaxMag*** functions (see sections 12.2.3.4 and 12.2.3.5).

This range of magnitudes is sampled and phases relating to these magnitudes are then calculated using the m-circle classes ***calcPhase*** function (see section 12.2.3.2).

Matlabs plotting routines are then used to plot the m-circle using the magnitudes and phases calculated.

## *12.4 QFT template class*

### 12.4.1 Overview

The QFT template defines a region on the Nichols chart which contains all the magnitudes and phases the plants in the plant set may have, at a specified frequency, $f_T$ . Figure 12.1 shows an example of a plant template. The plants in the plant set are also shown as pale blue circles.

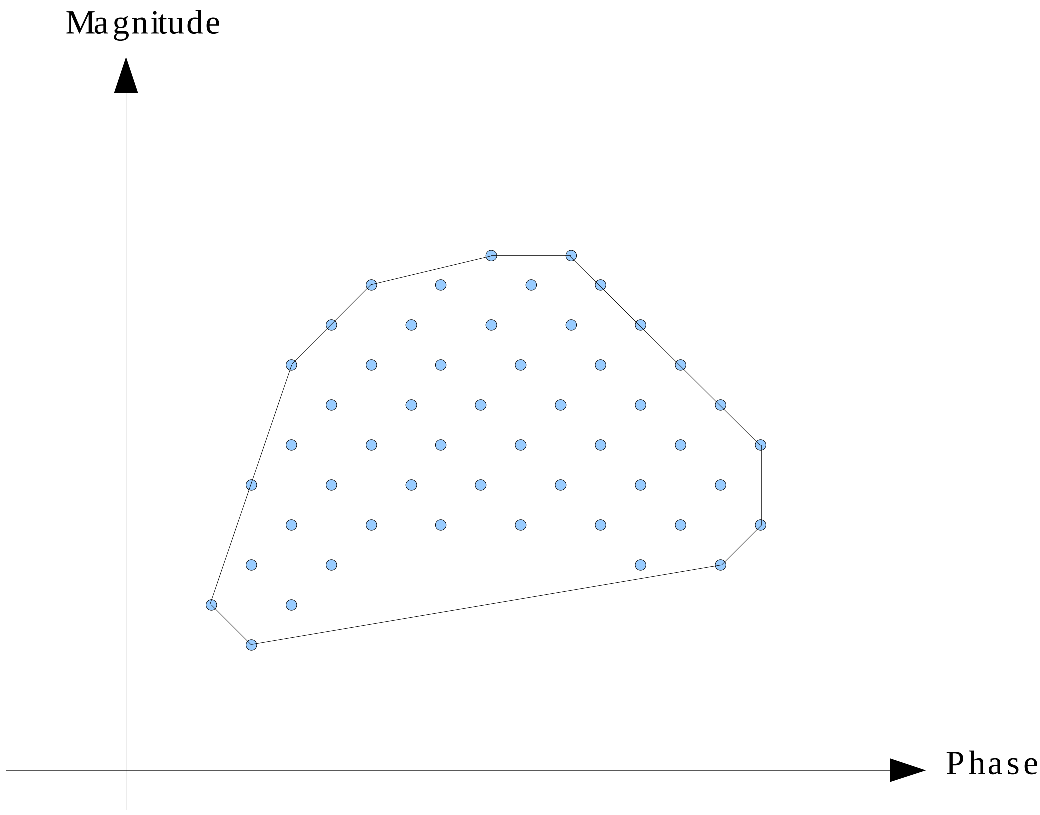


**Figure 12.1: The QFT plant template.**

The QFT template class has the following functionality:

1. It is capable of calculating boundary of the plant template given the plant set and the frequency of the template.
2. It is capable of calculating the following properties of the template:
   - The maximum magnitude of the template and the phase at which it occurs.
   - The minimum magnitude of the template bound and the phase at which it occurs.
   - The maximum phase of the template and the magnitude at which it occurs.
   - The minimum phase of the template bound and the magnitude at which it occurs.
3. It is capable of returning the following properties of the template:
   - The boundary of the template.
   - The magnitude and phase of the nominal plant.
   - The frequency of the template.
4. It is capable of calculating the minimum and maximum phases of the template at a specified magnitude.
5. It is capable of calculating the minimum and maximum magnitudes of the template at a specified phase.

6. It is capable of sampling the closed loop magnitude of the template boundary given a desired nominal magnitude and phase. These samples also have a maximum difference in magnitude (in decibels) between each consecutive point on the boundary a specified by the user.
7. It is capable of plotting the stability bound on the Nichols chart.

## 12.4.2 Attributes

In order for the QFT template class to be capable of performing the above mentioned functionality it must keep track of the following attributes:

1. $f_T$ , the frequency of the template (rad/s).
2. $\left(\left|\tilde{Q}(s)\right|, \measuredangle\tilde{Q}(s)\right)$ , the magnitudes (absolute value) and phase (deg) co-ordinates that define the boundary of the template.
3. $\left(\left|P_0(s)\right|, \measuredangle P_0(s)\right)$ , the magnitude (absolute value) and phase (deg) of the nominal plant at the frequency of the template.

## 12.4.3 Methods

### 12.4.3.1 getMinMag

This function retrieves the minimum magnitude of the template and the phase associated with it.

**Inputs**

The algorithm requires the following input:

reference , a flag used to define the reference frame to be used.

'absolute' The magnitude and phase returned are the absolute magnitude and phases. This is the default value.

'relative' The magnitude and phase returned are the relative to the magnitude and phase of the nominal plant.

**Outputs**

The algorithm produces the following outputs:

$\left|Q(s)\right|_{MIN}$ , the minimum magnitude of the plant template (dB).

$\measuredangle Q(s)_{|Q(s)|=|Q(s)|_{MIN}}$ , the phase at which the plant template has its minimum magnitude (deg).

**Pre-conditions**

The reference flag is checked to see that it has a valid value (i.e. *'absolute'* or *'relative'*). If the reference frame has not been specified it is set to the default value of *'absolute'*.

**Algorithm**

The minimum magnitude, $|Q(s)|_{MIN}$ , and its corresponding phase, $\measuredangle Q(s)_{|Q(s)|=|Q(s)|_{MIN}}$ , are determine from the magnitude and phase co-ordinates of the templates boundary, $\left(\widetilde{|Q(s)|}, \widetilde{\measuredangle Q(s)}\right)$ , using Matlabs' ***min*** function.

The minimum magnitude and its corresponding phase are converted to be relative to the nominal plant if the *reference* flag is set to *'relative'*.

## 12.4.3.2 getMaxMag

This function retrieves the maximum magnitude of the template.

**Inputs**

The algorithm requires the following input:

reference , a flag used to define the reference frame to be used.

'absolute' The magnitude and phase returned are the absolute magnitude and phases. This is the default value.

'relative' The magnitude and phase returned are the relative to the magnitude and phase of the nominal plant.

**Outputs**

The algorithm produces the following outputs:

$|Q(s)|_{MAX}$ , the maximum magnitude of the plant template (dB).

$\sphericalangle Q(s)_{|Q(s)|=|Q(s)|_{MAX}}$ , the phase at which the plant template has its maximum magnitude (deg).

**Pre-conditions**

The reference flag is checked to see that it has a valid value (i.e. *'absolute'* or *'relative'*). If the reference frame has not been specified it is set to the default value of *'absolute'*.

**Algorithm**

The maximum magnitude, $|Q(s)|_{MAX}$ , and its corresponding phase, $\sphericalangle Q(s)_{|Q(s)|=|Q(s)|_{MAX}}$ , are determine from the magnitude and phase co-ordinates of the templates boundary, $\left(\widetilde{|Q(s)|}, \widetilde{\sphericalangle Q(s)}\right)$ , using Matlabs' ***max*** function.

The maximum magnitude and its corresponding phase are converted to be relative to the nominal plant if the *reference* flag is set to *'relative'*.

## 12.4.3.3 getMinPhase

This function retrieves the minimum phase of the template.

**Inputs**

The algorithm requires the following input:

reference , a flag used to define the reference frame to be used.

'absolute' The magnitude and phase returned are the absolute magnitude and phases. This is the default value.

'relative' The magnitude and phase returned are the relative to the magnitude and phase of the nominal plant.

**Outputs**

The algorithm produces the following outputs:

$|Q(s)|_{\sphericalangle Q(s)=\sphericalangle Q(s)_{MIN}}$ , the magnitude at which the plant template has its minimum phase (dB).

$\measuredangle Q(s)_{MIN}$ , the minimum phase of the plant template (deg).

**Pre-conditions**

The reference flag is checked to see that it has a valid value (i.e. *'absolute'* or *'relative'*). If the reference frame has not been specified it is set to the default value of *'absolute'*.

**Algorithm**

The minimum phase, $\measuredangle Q(s)_{MIN}$ , and its corresponding magnitude, $|Q(s)|_{\measuredangle Q(s)=\measuredangle Q(s)_{MIN}}$ , are determine from the magnitude and phase co-ordinates of the templates boundary, $\left(\widetilde{|Q(s)|}, \widetilde{\measuredangle Q(s)}\right)$ , using Matlabs' ***min*** function.

The minimum phase and its corresponding magnitude are converted to be relative to the nominal plant if the *reference* flag is set to *'relative'*.

### 12.4.3.4 getMaxPhase

This function retrieves the maximum phase of the template.

**Inputs**

The algorithm requires the following input:

reference , a flag used to define the reference frame to be used.
'absolute' The magnitude and phase returned are the absolute magnitude and phases. This is the default value.
'relative' The magnitude and phase returned are the relative to the magnitude and phase of the nominal plant.

**Outputs**

The algorithm produces the following output:

$|Q(s)|_{\measuredangle Q(s)=\measuredangle Q(s)_{MAX}}$ , the magnitude at which the plant template has its maximum phase (dB).

$\measuredangle Q(s)_{MAX}$ , the maximum phase of the plant template (deg).

**Pre-conditions**

The reference flag is checked to see that it has a valid value (i.e. *'absolute'* or *'relative'*). If the reference frame has not been specified it is set to the default value of *'absolute'*.

**Algorithm**

The maximum phase, $\measuredangle Q(s)_{MAX}$ , and its corresponding magnitude, $|Q(s)|_{\measuredangle Q(s)=\measuredangle Q(s)_{MAX}}$ , are determine from the magnitude and phase co-ordinates of the templates boundary, $\left(\widetilde{|Q(s)|}, \widetilde{\measuredangle Q(s)}\right)$ , using Matlabs' ***max*** function.

The maximum phase and its corresponding magnitude are converted to be relative to the nominal plant if the *reference* flag is set to *'relative'*.

### 12.4.3.5 getBound

This method retrieves the boundary calculated and stored as an attribute when the object was created.

**Inputs**

The algorithm requires the following input:

reference , a flag used to define the reference frame to be used.

'absolute' The magnitude and phase returned are the absolute magnitude and phases. This is the default value.

'relative' The magnitude and phase returned are the relative to the magnitude and phase of the nominal plant.

**Outputs**

The algorithm produces the following outputs:

$\left(\widetilde{|Q(s)|}, \widetilde{\measuredangle Q(s)}\right)$ , the magnitude (dB) and phase (deg) co-ordinates of the templates boundary.

**Pre-conditions**

The reference flag is checked to see that it has a valid value (i.e. *'absolute'* or *'relative'*). If the reference frame has not been specified it is set to the default value of *'absolute'*.

**Algorithm**

A copy of the magnitude and phase co-ordinates which define the templates boundary, $\left(\widetilde{|Q(s)|}, \widetilde{\measuredangle Q(s)}\right)$, are returned after the magnitudes has been converted to decibels.

### 12.4.3.6 getNom

This method retrieves the magnitude and phase of the nominal plant.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$\left(|P_0(s)|, \measuredangle P_0(s)\right)$, the magnitude (dB) and phase (deg) of the nominal plant at the frequency of the template.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

A copy of the attribute containing the nominal magnitude and phase, $\left(|P_0(s)|, \measuredangle P_0(s)\right)$, are returned after the magnitude has been converted to decibels.

### 12.4.3.7 getFreq

This method retrieves the frequency of the template.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$f_T$ , the frequency of the template (rad/s).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

A copy of the attribute containing the template's frequency, $f_T$ , is returned.

### 12.4.3.8 calcPhase

This function calculates the maximum and minimum phase of the plant template at a specified magnitude. See Figure 12.2.

**Inputs**

The algorithm requires the following inputs:

$|Q(s)|$ , the magnitude for which the boundary phases are required (dB).

reference , a flag used to define the reference frame to be used.

'absolute' The phases returned are the absolute phases. This is the default value.

'relative' The phases returned are the relative to the magnitude and phase of the nominal plant.

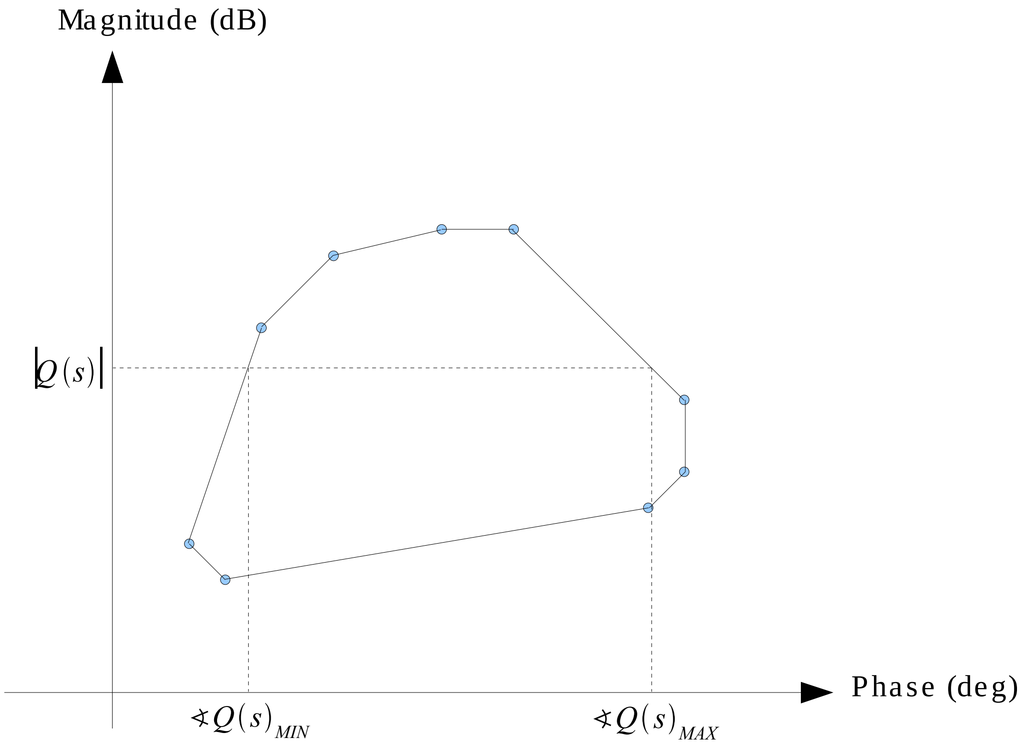


**Figure 12.2: Calculating the minimum and maximum phase of a QFT template at a specified magnitude.**

**Outputs**

The algorithm produces the following outputs:

$\measuredangle Q(s)_{MIN}$ , the minimum phase of the template at the magnitude specified (deg).

$\measuredangle Q(s)_{MAX}$ , the maximum phase of the template at the magnitude specified (deg).

**Pre-conditions**

The reference flag is checked to see that it has a valid value (i.e. *'absolute'* or *'relative'*). If the reference frame has not been specified it is set to the default value of *'absolute'*.

**Algorithm**

The minimum and maximum phases of the template at the magnitude specified are determined as follows:

1. The magnitude specified, $|Q(s)|$ , is converted to an absolute value.

2. If the reference flag has value of *'relative'*, the magnitude of the nominal plant is added back into the magnitude specified as follows:

$$|Q(s)| = |Q(s)| \cdot |P_0(s)|$$

3. The minimum and maximum magnitudes of the template ( $|Q(s)|_{MIN}$ and $|Q(s)|_{MAX}$ ) are determined from the magnitude and phase co-ordinates which define the templates boundary, $\left(\widetilde{|Q(s)|}, \widetilde{\measuredangle Q(s)}\right)$ , using Matlab's ***min*** and ***max*** functions respectively.
4. If the magnitude specified, $|Q(s)|$ , is greater than the templates maximum magnitude, $|Q(s)|_{MAX}$ , or smaller than the templates minimum magnitude, $|Q(s)|_{MIN}$ , not a number (*NaN*) is returned for both the minimum and maximum phases of the template ( $\measuredangle Q(s)_{MIN}$ and $\measuredangle Q(s)_{MAX}$ ).
5. If the magnitude specified, $|Q(s)|$ , is equal to the templates maximum magnitude, $|Q(s)|_{MAX}$ , then the phase of the point on the boundary which gives the templates maximum magnitude is returned for both the minimum and maximum phases of the template at the magnitude specified ( $\measuredangle Q(s)_{MIN}$ and $\measuredangle Q(s)_{MAX}$ ).
6. If the magnitude specified, $|Q(s)|$ , is equal to the templates minimum magnitude, $|Q(s)|_{MIN}$ , then the phase of the point on the boundary which gives the templates minimum magnitude is returned for both the minimum and maximum phases of the template ( $\measuredangle Q(s)_{MIN}$ and $\measuredangle Q(s)_{MAX}$ ).
7. If the magnitude specified, $|Q(s)|$ , is smaller the templates maximum magnitude, $|Q(s)|_{MAX}$ , and greater than the templates minimum magnitude, $|Q(s)|_{MIN}$ , then the minimum and maximum phases of the template ( $\measuredangle Q(s)_{MIN}$ and $\measuredangle Q(s)_{MAX}$ ) are calculated as follows:
   a) The magnitude and phase points which define the templates boundary, $\left(\widetilde{|Q(s)|}, \widetilde{\measuredangle Q(s)}\right)$ , are cycled through to find the consecutive points on the boundary which bracket the magnitude specified, $|Q(s)|$ . When such points are found a line is fitted to them using the ***fit_line*** utility function, see section 12.12.1. The phase of the line at the magnitude specified is then added to a vector of possible solutions, $\measuredangle Q(s)_{BND}$ .

b) Step a) is repeated until all the pairs of consecutive points have been considered.

c) The minimum and maximum phases of the template ( $\measuredangle Q(s)_{MIN}$ and $\measuredangle Q(s)_{MAX}$ ) are then determined using Matlab's ***min*** and ***max*** functions on the vector of possible solutions, $\measuredangle Q(s)_{BND}$ .

8. If the reference flag has value of *'relative'*, the phase of the nominal plant is removed from the minimum and maximum phases of the template as follows:

$$\measuredangle Q(s)_{MIN} = \measuredangle Q(s)_{MIN} - \measuredangle P_0(s)$$

$$\measuredangle Q(s)_{MAX} = \measuredangle Q(s)_{MAX} - \measuredangle P_0(s)$$

### 12.4.3.9 calcMag

This method calculates the maximum and minimum magnitude of the plant template at a specified phase. See Figure 12.3.

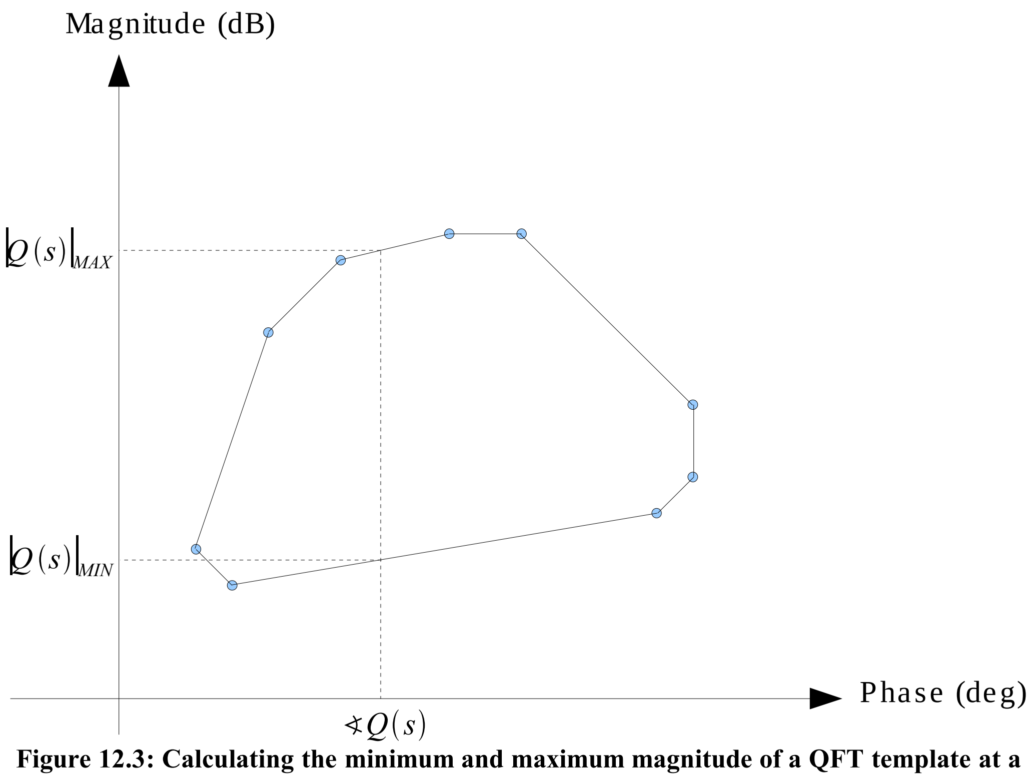


**Figure 12.3: Calculating the minimum and maximum magnitude of a QFT template at a specified phase.**

**Inputs**

The algorithm requires the following inputs:

$\measuredangle Q(s)$ , the phase for which the boundary magnitudes are required (deg).

reference , a flag used to define the reference frame to be used.

'absolute' The phases returned are the absolute phases. This is the default value.

'relative' The phases returned are the relative to the magnitude and phase of the nominal plant.

**Outputs**

The algorithm produces the following outputs:

$|Q(s)|_{MIN}$ , the minimum magnitude of the template at the phase specified (dB).

$|Q(s)|_{MAX}$ , the maximum magnitude of the template at the phase specified (dB).

**Pre-conditions**

The reference flag is checked to see that it has a valid value (i.e. *'absolute'* or *'relative'*). If the reference frame has not been specified it is set to the default value of *'absolute'*.

**Algorithm**

The minimum and maximum magnitudes of the template at the phase specified are determined as follows:

1. If the reference flag has value of *'relative'*, the phase of the nominal plant is added back into the phase specified as follows:

$$\measuredangle Q(s) = \measuredangle Q(s) + \measuredangle P_0(s)$$

2. The minimum and maximum phases of the template ( $\measuredangle Q(s)_{MIN}$ and $\measuredangle Q(s)_{MAX}$ ) are determined from the magnitude and phase co-ordinates which define the templates boundary, $\left(\widetilde{|Q(s)|}, \widetilde{\measuredangle Q(s)}\right)$ , using Matlab's ***min*** and ***max*** functions respectively.
3. If the phase specified, $\measuredangle Q(s)$ , is greater than the templates maximum phase, $\measuredangle Q(s)_{MAX}$ , or smaller than the templates minimum phase, $\measuredangle Q(s)_{MIN}$ , not a

number (*NaN*) is returned for both the minimum and maximum magnitudes of the template ( $|Q(s)|_{MIN}$ and $|Q(s)_{MAX}|$ ).

4. If the phase specified, $\measuredangle Q(s)$ , is equal to the templates maximum phase, $\measuredangle Q(s)_{MAX}$ , then the magnitude of the point on the boundary which gives the templates maximum phase is returned for both the minimum and maximum magnitudes of the template ( $|Q(s)|_{MIN}$ and $|Q(s)_{MAX}|$ ).
5. If the phase specified, $\measuredangle Q(s)$ , is equal to the templates minimum phase, $\measuredangle Q(s)_{MIN}$ , then the magnitude of the point on the boundary which gives the templates minimum phase is returned for both the minimum and maximum magnitudes of the template ( $|Q(s)|_{MIN}$ and $|Q(s)_{MAX}|$ ).
6. If the phase specified, $\measuredangle Q(s)$ , is smaller the templates maximum phase, $\measuredangle Q(s)_{MAX}$ , and greater than the templates minimum phase, $\measuredangle Q(s)_{MIN}$ , then the minimum and maximum magnitudes of the template ( $|Q(s)|_{MIN}$ and $|Q(s)_{MAX}|$ ) are calculated as follows:
   a) The magnitude and phase points which define the templates boundary, $\left(\widetilde{|Q(s)|}, \widetilde{\measuredangle Q(s)}\right)$ , are cycled through to find the consecutive points on the boundary which bracket the phase specified, $\measuredangle Q(s)$ . When such points are found a line is fitted to them using the ***fit_line*** utility function, see section 12.12.1. The magnitude of the line at the phase specified is then added to a vector of possible solutions, $|Q(s)|_{BND}$ .
   b) Step a) is repeated until all the pairs of consecutive points have been considered.
   c) The minimum and maximum magnitudes of the template ( $|Q(s)|_{MIN}$ and $|Q(s)|_{MAX}$ ) are then determined using Matlab's ***min*** and ***max*** functions on the vector of possible solutions, $|Q(s)|_{BND}$ .
7. If the reference flag has value of *'relative'*, the magnitude of the nominal plant is removed from the minimum and maximum magnitudes of the template as follows:

$$|Q(s)|_{MIN} = |Q(s)|_{MIN} - |P_0(s)|$$

$$|Q(s)|_{MAX} = |Q(s)|_{MAX} - |P_0(s)|$$

Note that the magnitudes are in decibels.

### 12.4.3.10 sampleClosedLoopMag

This method samples the closed loop magnitude of the QFT template which has the magnitude and phase of the nominal plant specified. The method ensures that maximum difference between each consecutive magnitude sample is not greater than a specified value.

**Inputs**

The algorithm requires the following inputs:

$\measuredangle P_N(s)$ , the desired nominal phase of the plant template (deg).

$|P_N(s)|$ , the desired nominal magnitude of the plant template (dB).

$|\Delta T|$ , the maximum difference between each consecutive magnitude sample (dB).

**Outputs**

The algorithm requires the following inputs:

$|T(s)|$ , the closed loop magnitude samples (dB).

$\measuredangle T(s)$ , the closed loop phase at each of the closed loop magnitude samples (deg)

**Pre-conditions**

As the inputs can have any value, nothing needs to be checked.

**Algorithm**

The closed loop magnitude of the QFT template is sampled in the following manner:

1. The minimum and maximum open loop magnitudes relative to the nominal plant ( $|Q(s)|_{MIN}$ and $|Q(s)|_{MAX}$ ) are calculated by calling this class's ***getMinMag*** and ***getMaxMag*** methods respectively. See sections 12.4.3.1 and 12.4.3.2.
2. The number of samples required to sample the open loop magnitude to the accuracy specified can be estimated by

$$n = ceil\left(\frac{|Q(s)|_{MAX} - |Q(s)|_{MIN}}{|\Delta T|}\right)$$

Now that the number of samples required has been estimated, Matlab's ***linspace*** function can be used to sample the open loop magnitude.

3. The open loop phases of the template relative to the nominal plant at the sampled open loop magnitudes are calculated using this class's ***calcPhase*** method, see section 12.4.3.8.
4. The closed loop magnitudes and phases are then calculated from the open loop magnitude and phase samples.
5. The difference between consecutive closed loop magnitude samples is then calculated. For samples with a magnitude difference greater than the maximum specified difference, $|\Delta T|$ , an extra open loop sample is added midway (in decibels) between the related open loop magnitude samples.
6. Steps 3 to 5 are repeated until all the differences between consecutive closed loop magnitude samples are smaller than the maximum specified difference, $|\Delta T|$ .
7. The closed loop magnitude, $|T(s)|$ , and phase, $\sphericalangle T(s)$ , samples are returned.

### 12.4.3.11 nichols

This function plots the QFT template in Nichols chart format.

**Inputs**

The algorithm requires the following input:

reference , a flag used to define the reference frame to be used.

'absolute' The phases returned are the absolute phases. This is the default value.

'relative' The phases returned are the relative to the magnitude and phase of the nominal plant.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

The reference flag is checked to see that it has a valid value (i.e. *'absolute'* or *'relative'*). If the reference frame has not been specified it is set to the default value of *'absolute'*.

**Algorithm**

Matlabs plotting routines are used to plot the the magnitudes and phases points of the templates boundary. These points are joined by lines.

Note that if the reference flag has a value of *'relative'* the boundary plotted is relative to the nominal plant.

### 12.4.3.12 init

This method initialises the class by performing the preliminary calculations necessary to be able to perform the functionality required of the QFT template class.

**Inputs**

The algorithm requires the following inputs:

$f$ , the frequency of the template (rad/s).

$\tilde{P}(s)$ , the vector of the plants in the plant set.

$P_0(s)$ , the nominal plant.

**Outputs**

The algorithm produces no outputs.

**Pre-conditions**

The frequency specified for the template, $f$ , is checked to see that it is real and greater than or equal to 0 rad/s.

The vector of plants,, and the nominal plant,, are checked to see that the are Matlab ***lti*** classes.

**Algorithm**

This method performs the following calculations:

1. The frequency of the template attribute, $f_T$ , is set to the frequency specified, $f$ .

2. The magnitudes and phases of all the plants in the plant set are calculated at the frequency of the template, $f_T$, using the utility function ***safe_freqresp***, see section 12.12.3.
3. The plants that sit on the boundary of the template are calculated by calling the template class's ***calcBoundary*** function, see section 12.4.3.13.

## 12.4.3.13 calcBoundary

This method finds the boundary of the plant template. The method used is based on a method used to find the boundary of an image as discussed by Todd (1986).

**Inputs**

The algorithm requires the following inputs:

$|\tilde{P}(s)|$ , the magnitudes of the plants in the plant set at the frequency of the template (absolute value).

$\measuredangle\tilde{P}(s)$ , the phases of the plants in the plant set at the frequency of the template (deg).

**Outputs**

The algorithm produces no outputs.

**Pre-conditions**

As this method is a private method its inputs are generated by the QFT template class and therefore do not need to be checked.

**Algorithm**

The boundary of the template is determined as follows:

1. A reference point is found that is in the interior of the plant template, see Figure 12.4. This is achieved by finding the mid-points between the maximum and minimum magnitudes, and the maximum and minimum phases of the plant template. Before these points can be found the magnitudes of the plants in the plant set must be converted to decibels. Then the maximum and minimum magnitudes of the plants in the plant set are determined, $|P(s)|_{MAX}$ and $|P(s)|_{MIN}$ respectively. Similarly, the maximum and

minimum phases of the plants in the plant set are determined, $\measuredangle P(s)_{MAX}$ and $\measuredangle P(s)_{MIN}$ respectively. The magnitude and phase of the reference point is then calculated from

$$|P(s)|_{REF} = \frac{|P(s)|_{MIN} + |P(s)|_{MAX}}{2}$$

and

$$\measuredangle P(s)_{REF} = \frac{\measuredangle P(s)_{MIN} + \measuredangle P(s)_{MAX}}{2}$$

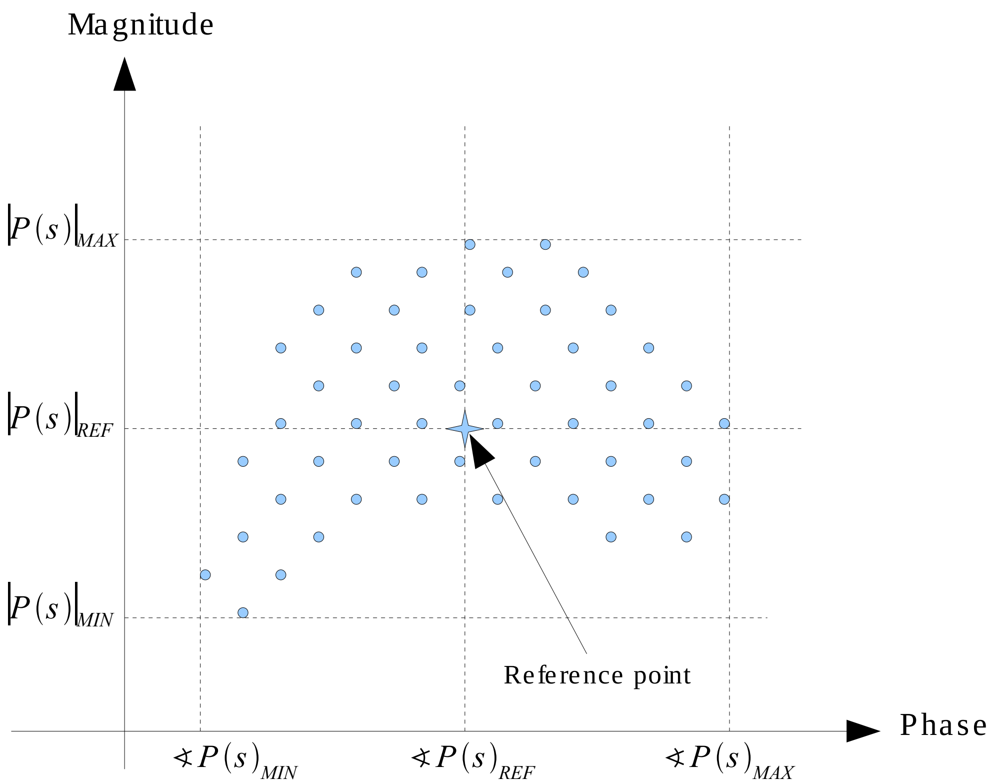


**Figure 12.4: Calculating a reference point for the plant template.**

2. The magnitude and phase of each plant is set relative to the reference point. This is achieved in the following manner

$$|P_i(s)| = |P_i(s)| - |P_0(s)|$$

and

$$\measuredangle P_i(s) = \measuredangle P_i(s) - \measuredangle P_0(s)$$

where the plant set is given by

$$\tilde{P}(s) = P_1(s), P_2(s), \ldots, P_n(s)$$

3. Convert the relative magnitudes and phases of the plants in the plant set to polar co-ordinates (see ).

$$d_i = \sqrt{|P(s)|_i^2 + \measuredangle P(s)_i^2}$$

and

$$\theta_i = \tan^{-1}\left(\frac{|P(s)|_i}{\measuredangle P(s)_i}\right)$$

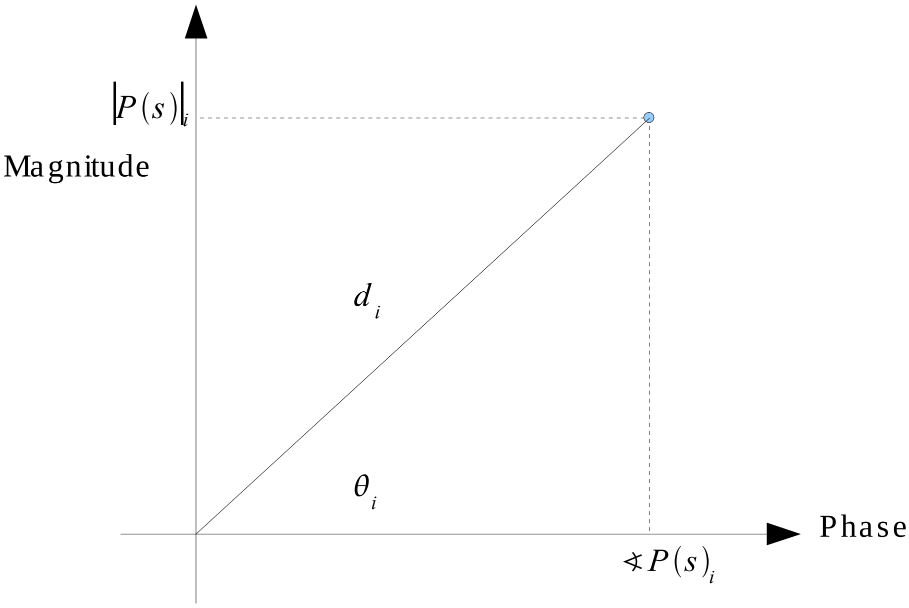


**Figure 12.5: Converting the position of a plant in the plant set to polar co-ordinates.**

4. The relative magnitudes and phases of the plants are used to calculate the distance from the reference point, $d_i$ , and the angle, $\theta_i$ , as follows

$$d_i = \sqrt{(\measuredangle P_i(s))^2 + |P_i(s)|_2}$$

and

$$\theta_i = \tan^{-1}\left(\frac{|P_i(s)|}{\measuredangle P_i(s)}\right)$$

5. The plants are ordered in order of ascending angle, $\theta_i$ .
6. Find the plant furthest from the reference point i.e. The plant with the maximum distance, $d_k$ . Set the starting point as the furthest plant from the origin as a starting point (let *i* = *k*).
7. Using the current plant, $P(s)_k$ , fit a line between it and the plant after the next plant, $P(s)_{k+2}$ , shown in Figure 12.6. Find the distance from the origin that of the point on the line, $l_{k+1}$ , at the angle of the next plant, $\measuredangle P(s)_{k+1}$ (Using the utility function ***calcPolarLine***, see section 12.12.2).
8. If the distance from the reference point to the point on the line at an angle, $l_{k+1}$ , is greater than the current plants distance from the reference point, $d_{k+1}$ , the current plant under consideration, $P(s)_{k+1}$ , is not on the boundary and is discarded.
9. Steps 6 and 7 are repeated for all plants in the plant set. If plants have been discarded in step 7, then steps 6 and 7 are again repeated for all plants in the plant set, until no further plants are discarded during step 7.
10. The magnitudes and phases of the remaining plants define the boundary of the template, $(|\tilde{Q}(s)|, \measuredangle \tilde{Q}(s))$ .

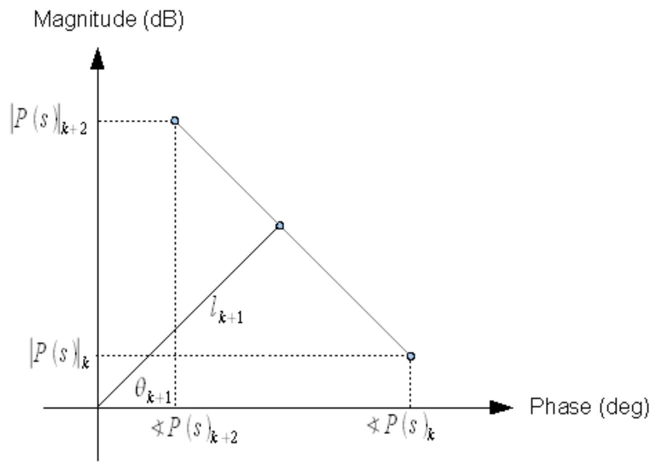


**Figure 12.6: Fitting a line between two plants in the QFT templates plant set.**

## *12.5 Stability bound class*

### 12.5.1 Overview

The stability bound defines a region on the Nichols chart into which the nominal loop transmission may not cross if the closed loop magnitude of the control system is to remain below a user defined maximum magnitude for a particular frequency. The stability bound takes into account the uncertainty of the plant.

The stability bound class has the following functionality:

1. It is capable of calculating the stability bound given the maximum magnitude allowed and the plant template. Note that the plant template specifies both the frequency of the stability bound as well as the uncertainty in the plant at that frequency.
2. It is capable of calculating the following properties of the stability bound:
   - The maximum magnitude of the stability bound and the phase at which it occurs.
   - The minimum magnitude of the stability bound and the phase at which it occurs.
   - The maximum phase of the stability bound and the magnitude at which it occurs.
   - The minimum phase of the stability bound and the magnitude at which it occurs.
3. It is capable of calculating the minimum and maximum phases of the stability bound at a specified magnitude.

4. It is capable of calculating the minimum and maximum magnitudes of the stability bound at a specified phase.
5. It is capable of plotting the stability bound on the Nichols chart.

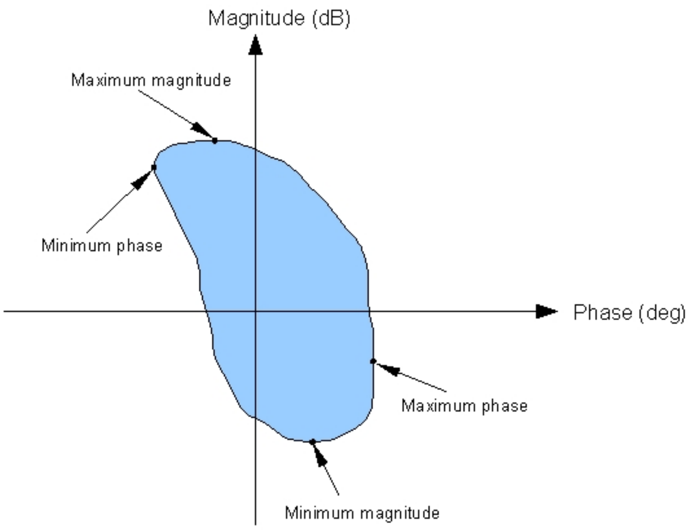


**Figure 12.7: The stability bound and its properties.**

## 12.5.2 Attributes

In order for the stability bound class to be capable of performing the above mentioned functionality it must keep track of the following attributes:

1. $M$ , the m-circle which defines the maximum magnitude specified.
2. $Q$ , the plant template which defines the frequency of the stability bound and the plant uncertainty at that frequency.

## 12.5.3 Methods

### 12.5.3.1 calcPhase

This function calculates the maximum and minimum phase of the stability bound at a specified magnitude.

**Inputs**

The algorithm requires the following input:

$|B(s)|$ , the magnitude for which the stability bounds minimum and maximum phases are required (dB).

**Outputs**

The algorithm produces the following outputs:

$\sphericalangle B(s)_{MIN}$ , the minimum phase of the stability bound at the magnitude specified (deg).

$\sphericalangle B(s)_{MAX}$ , the maximum phase of the stability bound at the magnitude specified (deg).

**Pre-conditions**

The magnitude for which the phase of the m-circle is required, $|B|$ , must be greater than zero. If the magnitude is not part of the stability bound, *NaN* (not a number) is returned to indicate this fact.

**Algorithm**

The algorithm used to calculate the phase of the stability bound at a given magnitude uses the following steps:

1. The phase of the m-circle is sampled. The number of samples required in order to keep the difference between them below the maximum phase error is given by:

$$n = \left\lceil \frac{\sphericalangle M_{MAX} - \sphericalangle M_{MAX}}{\sphericalangle \Delta M} \right\rceil$$

where $\sphericalangle \Delta M$ is the maximum phase error which is hard coded to 0.1 dB. Then the sampled phases are given by:

$$\sphericalangle M_i = \left( \frac{\sphericalangle M_{MAX} - \sphericalangle M_{MIN}}{n} \right) i + \sphericalangle M_{MIN} \quad \text{where} \quad i = 1, 2, \ldots n$$

This is achieved in code by calling Matlab's **linspace** function.

2. The magnitude of the m-circle is calculated at each phase sample. This is achieved by calling the m-circle attribute's **calcMag** function (see section 12.2.3.1) with the sampled phases. The sampled m-circle phases are shown in Figure 12.8.

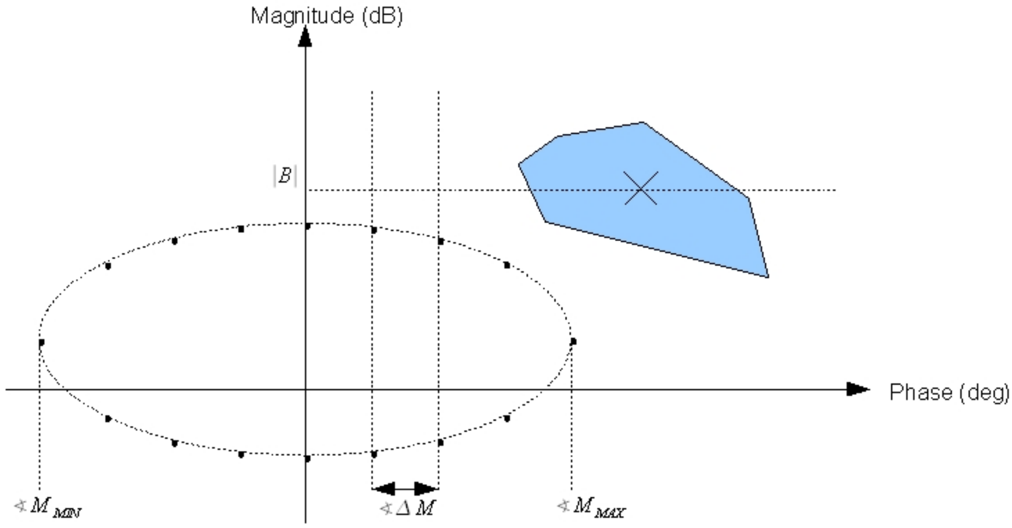


**Figure 12.8: Sampling of the m-circles phase.**

3. The sampled magnitudes of the m-circle (calculated in the previous step) are then shifted by the magnitude for which the phases are required, as follows:

$$|M| = \frac{|M|}{|B|}$$

This transforms the magnitude to an axis relative to the nominal magnitude of the plant template.

4. The nominal phase of the QFT template which places its boundary on one of the m-circle samples is now calculated, see Figure 12.9. Note that as the are two phases samples for each magnitude of the m-circle, there are two nominal phase of the QFT template at each magnitude. This is shown in Figure 12.9 with a solid light blue and a dashed outlined QFT template. The phases of the QFT template at the magnitude of the i$^{th}$ m-circle sample magnitude, $\sphericalangle Q_{i_{|Q|=|M|}}$, is calculated using the QFT template's

***calcPhase*** method, see section 12.4.3.8. The nominal phase of the QFT template is then given by

$$\measuredangle B_i = \measuredangle M_i - \measuredangle Q_{i_{|Q|=|M_i|}}$$

where $\measuredangle M_i$ is the phase of the i[th] m-circle sample.

5. Step 4 is repeated for all the m-circle samples, and the nominal phases of the QFT template calculated are stored in a vector of possible solutions, $\measuredangle \tilde{B}_{BND}$ .

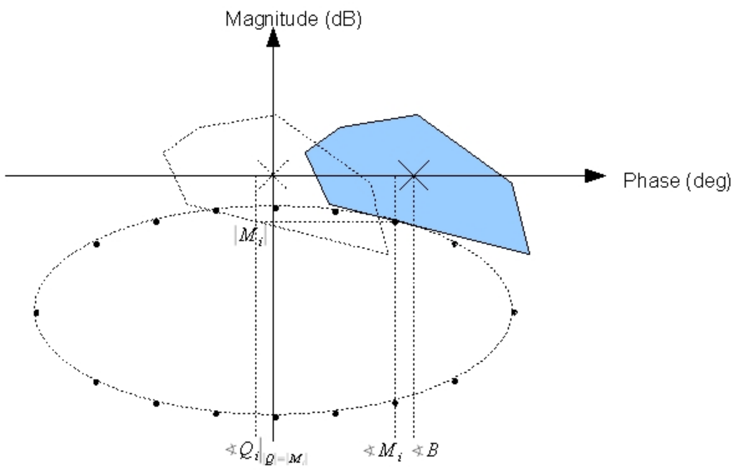


**Figure 12.9: Finding the nominal phase of the QFT template which places its boundary on one of the m-circle samples.**

6. The nominal phase of the QFT template that places one of its boundary points on the m-circles boundary is now calculated, see Figure 12.10. The boundary magnitudes and phases of the QFT template relative to the nominal plant are retrieved using the QFT template's ***getBound*** method, see section 12.4.3.5. The nominal phase of the QFT template at its i[th] boundary point is then given by

$$\measuredangle B_i = \measuredangle M_{i_{|M|=|Q_i|}} - \measuredangle Q_i$$

where $\measuredangle M_i$ is the phase of the i[th] m-circle sample.

7. Step 6 is repeated for all the QFT templates boundary points, and the nominal phases of the QFT template calculated are stored in a vector of possible solutions, $\measuredangle \tilde{B}_{BND}$ .
8. The minimum and maximum phases of the stability bound at the magnitude specified are then given by

$$\measuredangle B(s)_{MIN} = min\left(\measuredangle \tilde{B}_{BND}\right)$$

$$\measuredangle B(s)_{MAX} = max\left(\measuredangle \tilde{B}_{BND}\right)$$

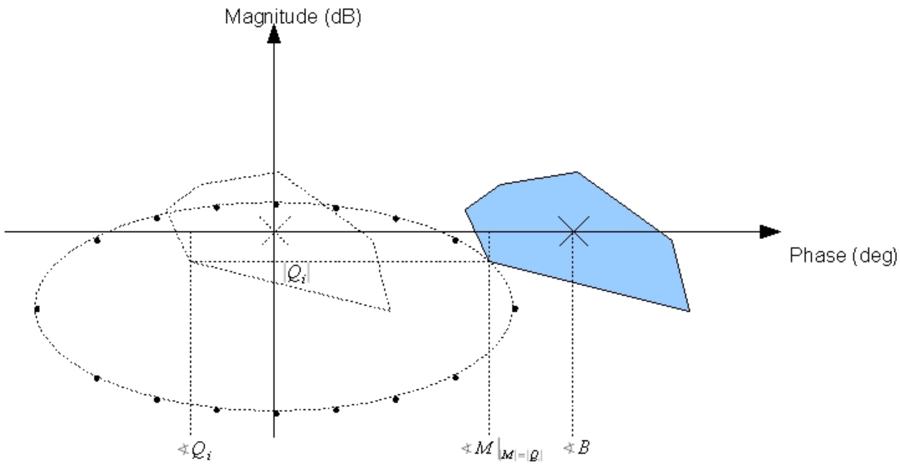


**Figure 12.10: Finding the nominal phase of the QFT template that places one of its boundary points on the m-circles boundary.**

### 12.5.3.2 getMinMag

This method calculates the minimum magnitude of the stability bound and the phase at which it occurs.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$|B|_{MIN}$ , the minimum magnitude of the stability bound (dB).

$\measuredangle B|_{|B|=|B|_{MIN}}$ , the phase of the stability bound at its minimum magnitude (deg).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The minimum magnitude of the stability bound is calculated as follows:

1. The minimum magnitude of the m-circle, $|M|_{MIN}$ , and the phase at which it occurs, $\measuredangle M|_{|M|=|M|_{MIN}}$ , are determined. Figure 12.11 shows the m-circle with the minimum magnitude and phase. The ***getMinMag*** method of the m-circle attribute, $M$ , calculates these magnitude and phase. See section 12.2.3.4 or section 12.3.3.3 depending on whether the m-circle attribute contains a m-circle object or rotated m-circle object respectively.
2. The maximum magnitude of the plant template, $|Q|_{MAX}$ , and the phase at which it occurs, $\measuredangle Q|_{|Q|=|Q|_{MAX}}$ , are determined. Figure 12.11 shows the plant template positioned so that the has the nominal plant at the origin (0 dB, -180°). The plant template is shown as dashed and the maximum magnitude and its corresponding phase are also shown. The ***getMaxMag*** method of the m-circle attribute, $Q$ , is used to calculate the maximum magnitude and its corresponding phase. See section 12.4.3.2.
3. The minimum magnitude of the stability bound is then given by

$$|B|_{MIN} = |M|_{MIN} - |Q|_{MAX}$$

where the magnitudes are in decibels, and its corresponding phase are given by

$$\measuredangle B|_{|B|=|B|_{MIN}} = \measuredangle M|_{|M|=|M|_{MIN}} - \measuredangle Q|_{|Q|=|Q|_{MAX}}$$

where the angles are in degrees.

Figure 12.11 shows the plant template positioned so that the nominal plants position gives the minimum magnitude of the stability bound and its corresponding phase. The plant template is shown in pale blue.

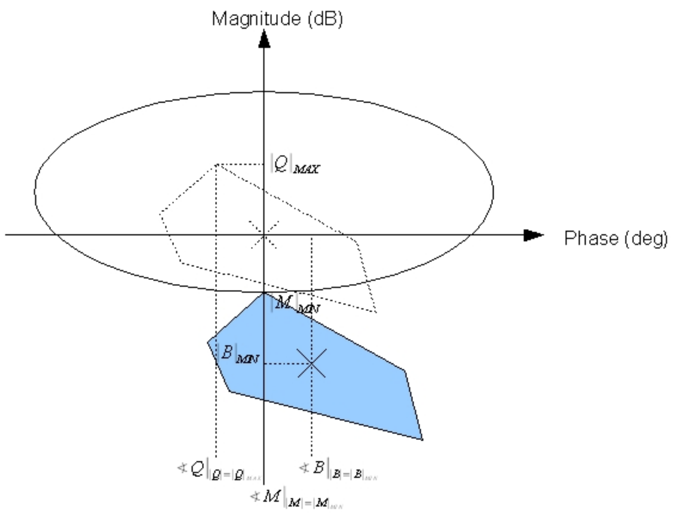


**Figure 12.11: Calculating the minimum magnitude of the stability bound.**

### 12.5.3.3 getMaxMag

This method calculates the maximum magnitude of the stability bound and the phase at which it occurs.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$|B|_{MAX}$ , the maximum magnitude of the stability bound (dB).

$\measuredangle B\big|_{|B|=|B|_{MAX}}$ , the phase of the stability bound at its maximum magnitude (deg).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The minimum magnitude of the stability bound is calculated as follows:

1. The maximum magnitude of the m-circle, $|M|_{MAX}$ , and the phase at which it occurs, $\measuredangle M\big|_{|M|=|M|_{MAX}}$ , are determined. Figure 12.12 shows the m-circle with the maximum magnitude and phase. The ***getMaxMag*** method of the m-circle attribute, $M$ , calculates these magnitude and phase. See section 12.2.3.5 or section 12.3.3.4 depending on whether the m-circle attribute contains a m-circle object or rotated m-circle object respectively.
2. The minimum magnitude of the plant template, $|Q|_{MIN}$ , and the phase at which it occurs, $\measuredangle Q\big|_{|Q|=|Q|_{MIN}}$ , are determined. Figure 12.12 shows the plant template positioned so that the has the nominal plant at the origin (0 dB, -180$^{o}$). The plant template is shown as dashed and the minimum magnitude and its corresponding phase are also shown. The ***getMinMag*** method of the m-circle attribute, $Q$ , is used to calculate the minimum magnitude and its corresponding phase. See section 12.4.3.1.
3. The maximum magnitude of the stability bound is then given by

   $$|B|_{MAX} = |M|_{MAX} - |Q|_{MIN}$$

   where the magnitudes are in decibels, and its corresponding phase are given by

   $$\measuredangle B\big|_{|B|=|B|_{MAX}} = \measuredangle M\big|_{|M|=|M|_{MAX}} - \measuredangle Q\big|_{|Q|=|Q|_{MIN}}$$

   where the angles are in degrees.
   Figure 12.12 shows the plant template positioned so that the nominal plants position gives the maximum magnitude of the stability bound and its corresponding phase. The plant template is shown in pale blue.

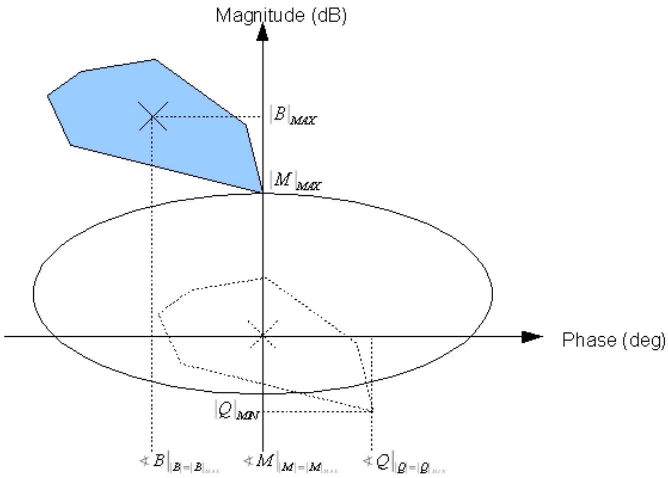


**Figure 12.12: Calculating the maximum magnitude of the stability bound.**

## 12.5.3.4getMinPhase

This method calculates the minimum phase of the stability bound and the magnitude at which it occurs.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$|B|_{\sphericalangle B = \sphericalangle B_{MIN}}$ , the magnitude of the stability bound at its minimum phase (dB).

$\sphericalangle B|_{MIN}$ , the minimum phase of the stability bound (deg).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The minimum phase of the stability bound is calculated as follows:

1. The minimum phase of the m-circle, $\measuredangle M|_{MIN}$, and the magnitude at which it occurs, $|M|_{\measuredangle M=\measuredangle M_{MIN}}$, are determined. Figure 12.13 shows the m-circle with the minimum phase and its corresponding magnitude. The ***getMinPhase*** method of the m-circle attribute, $M$, calculates these magnitude and phase. See section 12.2.3.6 or section 12.3.3.5 depending on whether the m-circle attribute contains a m-circle object or rotated m-circle object respectively.
2. The maximum phase of the plant template, $\measuredangle Q|_{MAX}$, and the magnitude at which it occurs, $|Q|_{\measuredangle Q=\measuredangle Q_{MAX}}$, are determined. Figure 12.13 shows the plant template positioned so that the has the nominal plant at the origin (0 dB, -180$^{o}$). The plant template is shown as dashed and the maximum phase and its corresponding magnitude are also shown. The ***getMaxPhase*** method of the m-circle attribute, $Q$, is used to calculate the maximum phase and its corresponding magnitude. See section 12.4.3.4.
3. The minimum phase of the stability bound is then given by

$$\measuredangle B|_{MIN}=\measuredangle M|_{MIN}-\measuredangle Q|_{MAX}$$

where the phases are in degrees, and its corresponding magnitude is given by

$$|B|_{\measuredangle B=\measuredangle B_{MIN}}=|M|_{\measuredangle M=\measuredangle M_{MIN}}-|Q|_{\measuredangle Q=\measuredangle Q_{MAX}}$$

where the magnitudes are in decibels.

Figure 12.13 shows the plant template positioned so that the nominal plants position gives the minimum phase of the stability bound and its corresponding magnitude . The plant template is shown in pale blue.

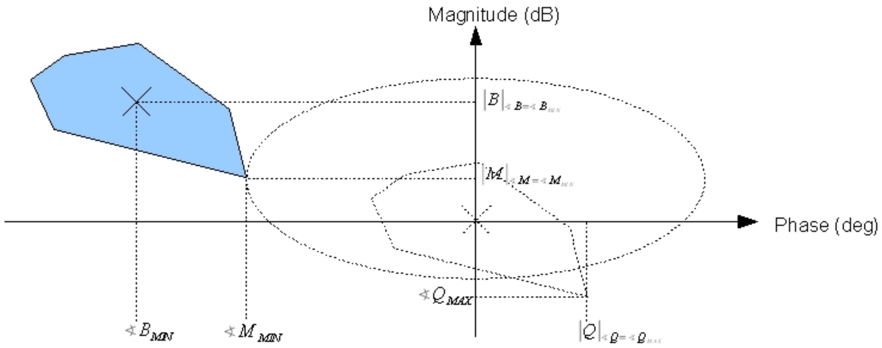


**Figure 12.13: Calculating the minimum phase of the stability bound.**

## 12.5.3.5getMaxPhase

This method calculates the maximum phase of the stability bound and the magnitude at which it occurs.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$|B|_{\measuredangle B=\measuredangle B_{MIN}}$ , the magnitude of the stability bound at its maximum phase (dB).

$\measuredangle B|_{MIN}$ , the maximum phase of the stability bound (deg).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The maximum phase of the stability bound is calculated as follows:

1. The maximum phase of the m-circle, $\measuredangle M|_{MAX}$, and the magnitude at which it occurs, $|M|_{\measuredangle M = \measuredangle M_{MAX}}$, are determined. Figure 12.14 shows the m-circle with the maximum phase and its corresponding magnitude. The ***getMaxPhase*** method of the m-circle attribute, $M$, calculates these magnitude and phase. See section 12.2.3.7 or section 12.3.3.6 depending on whether the m-circle attribute contains a m-circle object or rotated m-circle object respectively.
2. The minimum phase of the plant template, $\measuredangle Q|_{MIN}$, and the magnitude at which it occurs, $|Q|_{\measuredangle Q = \measuredangle Q_{MIN}}$, are determined. Figure 12.14 shows the plant template positioned so that the has the nominal plant at the origin (0 dB, -180$^{o}$). The plant template is shown as dashed and the minimum phase and its corresponding magnitude are also shown. The ***getMinPhase*** method of the m-circle attribute, $Q$, is used to calculate the minimum phase and its corresponding magnitude. See section 12.4.3.3.
3. The maximum phase of the stability bound is then given by

$$\measuredangle B|_{MAX} = \measuredangle M|_{MAX} - \measuredangle Q|_{MIN}$$

where the phases are in degrees, and its corresponding magnitude is given by

$$|B|_{\measuredangle B = \measuredangle B_{MAX}} = |M|_{\measuredangle M = \measuredangle M_{MAX}} - |Q|_{\measuredangle Q = \measuredangle Q_{MIN}}$$

where the magnitudes are in decibels.

Figure 12.14 shows the plant template positioned so that the nominal plants position gives the maximum phase of the stability bound and its corresponding magnitude . The plant template is shown in pale blue.

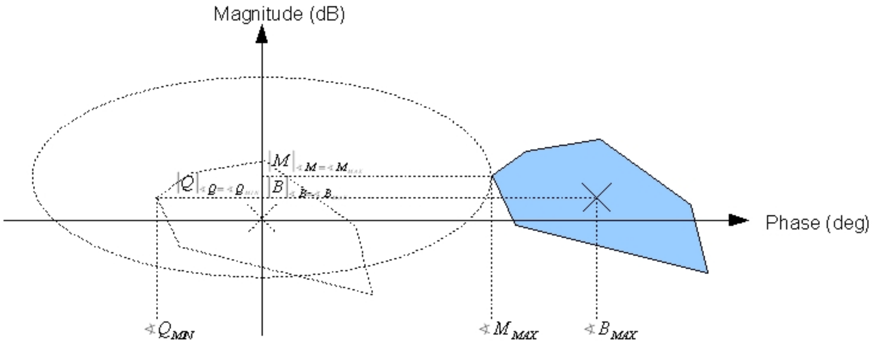


**Figure 12.14: Calculating the maximum phase of the stability bound.**

## 12.5.3.6getFreq

This method retrieves the frequency of the stability bound.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$f_T$ , the frequency of the stability bound (rad/s).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The frequency that must be returned is the frequency of the QFT template attribute, $Q$ . This is achieved by calling the template attributes' ***getFreq*** method, section 12.4.3.7.

#### 12.5.3.7 nichols

This function plots the stability bound in Nichols chart format.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The stability bound is plotted on the Nichols chart as follows:

1. The range of magnitudes required to plot the stability bound is calculated using the stability bound classes ***getMinMag*** and ***getMaxMag*** methods. See sections 12.5.3.2 and 12.5.3.3.
2. The range of magnitudes are sampled using Matlabs' ***linspace*** function.
3. The phases of the stability bound are calculated at the sampled magnitudes using the stability bound classes' ***calcPhase*** method. See section 12.5.3.1.
4. The stability bound is plotted by plotting the sampled magnitudes and the phases calculated using Malabs' plotting functions.

## *12.6 Uncertainty bound class*

### 12.6.1 Overview

The uncertainty bound defines the minimum open loop magnitude that the nominal loop transmission may have versus open loop phase and still meet the uncertainty specification. The uncertainty specification is the maximum allowable variation in magnitude that the closed loop response set may have. The uncertainty bound is defined at a particular frequency.

The uncertainty bound class has the following functionality:

1. It is capable of calculating the uncertainty bound given the uncertainty specification and the plant template. Note that the plant template specifies both the frequency of the stability bound as well as the uncertainty in the plant at that frequency.
2. It is capable of plotting the uncertainty bound on the Nichols chart.

## 12.6.2 Attributes

In order for the uncertainty bound class to be capable of performing the above mentioned functionality it must keep track of the following attributes:

1. $|\Delta T_{REQ}|$ , the uncertainty specification (dB).
2. $Q$ , the plant template which defines the frequency of the uncertainty bound and the plant uncertainty at that frequency.

## 12.6.3 Methods

### 12.6.3.1calcMag

This method calculates the magnitude of the uncertainty bound at the phase specified.

**Inputs**

The algorithm requires the following input:

$\measuredangle L_N(s)$ , the phase for which the magnitude of the uncertainty bound is required (deg).

**Outputs**

The algorithm produces the following output:

$|L_N(s)|$ , the magnitude of the uncertainty bound at the phase specified (dB).

**Pre-conditions**

As the phase of the nominal plant may have any value, it input does not need to be checked.

**Algorithm**

The uncertainty bound is calculated in the following manner:

1. A numerical algorithm will be used to solve for the magnitude of the uncertainty bound, $|L_N(s)|$ , at the phase specified, $\measuredangle L_N(s)$ . Matlab's ***fzero*** function is used to solve for the uncertainty using the following internal function:
   (a) The closed loop magnitude of the plant template is sampled, with open loop magnitude and phase of the nominal plant set to the specified phase, $\measuredangle L_N(s)$ , and the magnitude that Matlab's ***fzero*** function is using in its current iteration. The closed loop magnitude is sampled by using the ***sampleCloseLoopMag*** method of the QFT template attribute, $Q$ . This method is discussed in section 12.4.3.10.
   (b) The minimum and maximum closed loop magnitudes of the QFT template, $|T(s)|_{MIN}$ and $|T(s)|_{MIN}$ respectively, are determined using Matlab's ***min*** and ***max*** functions on the sampled closed loop magnitudes.
   (c) The maximum variation in the closed loop magnitude of the QFT template is given by

   $$|\Delta T(s)| = |T(s)|_{MAX} - |T(s)|_{MIN}$$

   (d) The difference between the desired closed loop uncertainty and the uncertainty value at the current nominal magnitude is given by

   $$\Delta = |\Delta T_{REQ}| - |\Delta T(s)|$$

   Matlab's ***fzero*** function solves for the magnitude of the nominal plant, $|L_N(s)|$ , and at the specified phase, $\measuredangle L_N(s)$ , such that the difference, $\Delta$ , is equal to 0 dB.

## 12.6.3.2nichols

This function plots the uncertainty bound in Nichols chart format.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The uncertainty bound is plotted on the Nichols chart as follows:

1. The phase is sampled from -180º to 0º using Matlabs' ***linspace*** function.
2. The magnitude of the uncertainty bound is calculated at each of these phase samples. This is achieved using the uncertainty bound classes' ***calcMag*** method, see section 12.6.3.1.
3. The sampled phases and the magnitudes calculated are plotted on the Nichols chart using Matlabs' plotting routines.

## *12.7 Bode semi-Infinite characteristic class*

### 12.7.1 Overview

The Bode semi-infinite characteristic allows the user to define the magnitude of a transfer function in a piece wise linear fashion. The Bode semi-infinite characteristic has a magnitude of 0 dB up until the cut-off frequency, $\omega_0$, after which it has a magnitude with a slope of $20 \cdot k\ dB/decade$ (see section 6.2.2). The phase is calculated from the magnitude specified.

The Bode semi-infinite characteristic class has the following functionality:

1. It is capable of calculating the semi-infinite characteristic at any specified frequency.
2. It is capable of fitting a rational transfer function to the semi-infinite characteristic.
3. It is capable of plotting semi-infinite characteristic on a Bode plot.
4. It is capable of plotting the semi-infinite characteristic on the Nichols chart.

## 12.7.2 Attributes

In order for the Bode semi-infinite characteristic class to be capable of performing the above mentioned functionality it must keep track of the following attributes:

1. $k$ , the slope of the magnitude greater than the cut-off frequency, $\omega_0$ , (dB/decade).
2. $\omega_0$ , the cut-off frequency (rad/s).

## 12.7.3 Methods

### 12.7.3.1calcResponse

This method calculates the magnitude and phase of the Bode semi-infinite characteristic for a specified set of frequencies.

**Inputs**

The algorithm requires the following input:

$\omega$ , the frequencies at which the magnitudes and phases of the semi-infinite characteristic are required (rad/s).

**Outputs**

The algorithm produces the following outputs:

$|H_{SI}(j\omega)|$ , the magnitudes of the semi-infinite characteristic for the frequencies specified (dB).

$\measuredangle H_{SI}(j\omega)$ , the phases of the semi-infinite characteristic for the frequencies specified (deg).

**Pre-conditions**

The frequencies specified, $\omega$ , are checked that they are real and greater than zero.

**Algorithm**

The magnitude of the Bode semi-infinite characteristic is given by:

$$\left|H_{SI}(j\omega)\right|=\begin{cases} 1 & for \quad \omega \leqslant \omega_0 \\ \left(\dfrac{\omega}{\omega_0}\right)^k & for \quad \omega > \omega_0 \end{cases}$$

And the phase (in radians) is given by:

For $0 < x_0^C \leqslant 0.414$ ,

$$\measuredangle H_{SI}(j\omega)=k\frac{2}{\pi}\left[x_0^C+\frac{1}{9}\left(x_0^C\right)^3+\frac{1}{25}\left(x_0^C\right)^5+\frac{1}{49}\left(x_0^C\right)^7\right]$$

For $0.414 < x_0^C \leqslant 1.0$ ,

$$\measuredangle H_{SI}(j\omega)=k\left[\frac{\pi}{4}-\frac{1}{\pi}\ln\left(x_0^C\right)\ln\left(y_0^C\right)-\frac{2}{\pi}\left(y_0^C+\frac{1}{9}\left(y_0^C\right)^3+\frac{1}{25}\left(y_0^C\right)^5+\frac{1}{49}\left(y_0^C\right)^7\right)\right]$$

For $x_0^C > 1.0$ ,

$$\measuredangle H_{SI}(j\omega)=k\frac{\pi}{2}-\measuredangle H_{SI}\left(\frac{1}{x_0^C}\right)$$

where :

$$x_0^C=\frac{\omega}{\omega_0}$$

$$y_0^C=\frac{\omega_0-\omega}{\omega_0+\omega}$$

The magnitude and phases are of the semi-infinite characteristic are converted to decibels and degrees respectively before being returned by the function.

### 12.7.3.2set

This method allows the user to set the properties of the semi-infinite characteristic class.

**Inputs**

The algorithm requires the following input:

$\left(property\ , value\right)$, the property-value pair. The *property* part specifies the property that needs to be set to the *value* part of the property-value pair.

**Outputs**

The algorithm generates no outputs.

**Pre-conditions**

The property part of the property-value pair is checked to see that it has one of the following values:

1. *'slope'*
2. *'w0'*

If the property part is *'slope'*, then the value part is checked to see that it is a scalar real value.

If the property part is *'w0'*, then the value part is checked to see that it is a scalar real value greater than zero.

**Algorithm**

The method runs through all the property-value pairs specified and sets the attributes to the values specified.

If the property part is *'slope'*, then the slope of the magnitude greater than the cut-off frequency, $k$, is set to the value specified.

If the property part is *'w0'*, then the cut-off frequency, $\omega_0$, is set to the value specified.

## 12.7.3.3get

This method allows the user to retrieve the values of the properties of the semi-infinite characteristic class.

**Inputs**

The algorithm requires the following input:

$property$ , the property for which the value is required. If the

**Outputs**

The algorithm produces the following output:

$value$ , the value of the property specified.

**Pre-conditions**

The property part of the property-value pair is checked to see that it has one of the following values:

1. *'slope'*
2. *'w0'*

**Algorithm**

If the value of the property specified is *'slope'* the attribute $k$ is returned.

If the value of the property specified is *'w0'* the attribute $\omega_0$ is returned.

## 12.7.3.4fit

This method fits a rational transfer function to the semi-infinite characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$N(s)$ , the numerator of the rational transfer function.

$D(s)$ , the denominator of the rational transfer function.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The rational transfer function is fitted to the semi-infinite characteristic in the following manner:

1. The highest power of the numerator is calculated as follows
$$n = \begin{cases} 0 & \text{if } k<0 \\ round\left(\frac{|k|}{20}\right) & \text{otherwise} \end{cases}$$
2. Similarly, the highest power of the denominator is calculated as follows
$$m = \begin{cases} round\left(\frac{|k|}{20}\right) & \text{if } k<0 \\ 0 & \text{otherwise} \end{cases}$$
3. A range of frequencies, which ranges from two decades below and above the cut-off frequency, is generated and sampled. The frequency is sampled using Matlabs ***logspace*** function.
4. The magnitudes and phases of the semi-infinite characteristic are calculated at the frequencies sampled in the previous step, using the semi-infinite characteristic classes' ***calcResponse*** method (see section 12.7.3.1).
5. The magnitudes and phases are then converted to a complex number format.
6. A rational transfer function with the a numerator of order $n$ and a denominator of order $m$ is fitted to these complex numbers using Matlabs ***invfreqs*** function. The numerator, $N(s)$ , and denominator, $D(s)$ obtained are returned by the method.

## 12.7.3.5bode

This method generates a Bode plot of the semi-infinite characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The Bode plot is generated in the following manner:

1. A range of frequencies, which ranges from two decades below and above the cut-off frequency, is generated and sampled. The frequency is sampled using Matlabs ***logspace*** function.
2. The semi-infinite characteristic classes ***calcResponse*** method, see section 12.7.3.1, is used to calculate the magnitudes and phases of the semi-infinite characteristic at the sampled frequencies.
3. Matlabs plotting functions are used to plot the Bode plot of the semi-infinite characteristic using the magnitudes, phases and frequencies calculated.

### 12.7.3.6nichols

This method generates a Nichols chart plot of the semi-infinite characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The Nichols chart plot is generated in the following manner:

1. A range of frequencies, which ranges from two decades below and above the cut-off frequency, is generated and sampled. The frequency is sampled using Matlabs ***logspace*** function.
2. The semi-infinite characteristic classes ***calcResponse*** method, see section 12.7.3.1, is used to calculate the magnitudes and phases of the semi-infinite characteristic at the sampled frequencies.
3. Matlabs plotting functions are used to plot the Nichols chart plot of the semi-infinite characteristic using the magnitudes, phases and frequencies calculated.

## *12.8 Bode step class*

### 12.8.1 Overview

The Bode step combines two semi-infinite characteristics with the goal of obtaining the desired high frequency roll-off without adding extra phase lag at frequencies lower than the lowest frequency corner frequency. See section 6.2.3.

The Bode step class has the following functionality:

1. It is capable of calculating the Bode step at a specified frequency.
2. It is capable of fitting a rational transfer function to the Bode step characteristic.
3. It is capable of plotting Bode step on a Bode plot.
4. It is capable of plotting the Bode step on the Nichols chart.

### 12.8.2 Attributes

In order for the Bode step class to be capable of performing the above mentioned functionality it must keep track of the following attributes:

1. $H_{SI1}(j\omega)$ , the instance of the semi-infinite characteristic class which implements the first corner frequency, $\omega_1$ .
2. $H_{SI2}(j\omega)$ , the instance of the semi-infinite characteristic class which implements the second corner frequency, $\omega_2$ .

### 12.8.3 Methods

#### 12.8.3.1calcResponse

This method calculates the magnitude and phase of the Bode step for a specified set of frequencies.

**Inputs**

The algorithm requires the following input:

$\omega$ , the frequencies at which the magnitudes and phases of the Bode step are required (rad/s).

**Outputs**

The algorithm produces the following outputs:

$|H_{BS}(j\omega)|$ , the magnitudes of the Bode step for the frequencies specified (dB).

$\measuredangle H_{BS}(j\omega)$ , the phases of the Bode step for the frequencies specified (deg).

**Pre-conditions**

The frequencies specified, $\omega$ , are checked by the two Bode semi-infinite classes ***calcResponse*** methods, see section 12.7.3.1. So no checking needs to be done by this method.

**Algorithm**

The magnitudes and phases of the Bode step at the frequencies specified are calculated in the following manner:

1. The magnitudes and phases of the semi-infinite characteristic class which implements the first corner frequency, $H_{SI1}(j\omega)$ , are calculated using its ***calcResponse*** method, see section 12.7.3.1.
2. The magnitudes and phases of the semi-infinite characteristic class which implements the second corner frequency, $H_{SI2}(j\omega)$ , are calculated using its ***calcResponse*** method, see section 12.7.3.1.

3. The magnitude of the Bode step, in decibels, is given by:

$$|H_{BS}(j\omega)| = |H_{SI1}(j\omega)| + |H_{SI2}(j\omega)|$$

And the phase (in radians) is given by:

$$\measuredangle H_{BS}(j\omega) = \measuredangle H_{SI1}(j\omega) + \measuredangle H_{SI2}(j\omega)$$

### 12.8.3.2set

This method allows the user to set the properties of the Bode step class.

**Inputs**

The algorithm requires the following input:

$(property\ , value)$ , the property-value pair. The *property* part specifies the property that needs to be set to the *value* part of the property-value pair.

**Outputs**

The algorithm generates no outputs.

**Pre-conditions**

The property part of the property-value pair is checked to see that it has one of the following values:

1. *'slope1'*
2. *'w1'*
3. *'slope2'*

If the property part is *'slope1'* or *'slope2'*, then the value part is checked to see that it is a scalar real value.

If the property part is *'w1'*, then the value part is checked to see that it is a scalar real value greater than zero.

**Algorithm**

The method runs through all the property-value pairs specified and sets the attributes to the values specified.

If the property part is *'slope1'*, then the value part of the property-value pair is used to set the '*slope'* parameter of of the semi-infinite characteristic class which implements the first corner frequency, $H_{SI1}(j\omega)$ . This achieved by using the classes' ***set*** method, see section 12.7.3.2.

If the property part is *'w1'*, then the value part of the property-value pair is used to set the 'w0′ parameter of of the semi-infinite characteristic class which implements the first corner frequency, $H_{SI1}(j\omega)$ . This achieved by using the classes' ***set*** method, see section 12.7.3.2.

If the property part is *'slope2'*, then the value part of the property-value pair is used to set the '*slope'* parameter of of the semi-infinite characteristic class which implements the second corner frequency, $H_{SI2}(j\omega)$ . This achieved by using the classes' ***set*** method, see section 12.7.3.2.

The second corner frequency is calculated using the Bode step classes' ***calcSecondCornerFreq*** function. The second corner frequency is used to set the 'w0′ parameter of of the semi-infinite characteristic class which implements the second corner frequency, $H_{SI2}(j\omega)$ . This achieved by using the classes' ***set*** method as before, see section 12.7.3.2.

### 12.8.3.3get

This method allows the user to retrieve the values of the properties of the Bode step class.

**Inputs**

The algorithm requires the following input:

$property$ , the property for which the value is required.

**Outputs**

The algorithm produces the following outputs:

$value$ , the value of the property specified.

**Pre-conditions**

The property part of the property-value pair is checked to see that it has one of the following values:

1. *'slope1'*
2. *'w1'*
3. *'slope2'*
4. *'w2'*

**Algorithm**

If the property part of the property-value pair is *'slope1'* or *'w1'* the semi-infinite characteristics' ***get*** method , see section 12.7.3.3, is used to retrieve its *'w0'* or *'slope'* property. which implements the first corner frequency, $H_{SI1}(j\omega)$ .

If the property part of the property-value pair is *'slope2'* or *'w2'* the semi-infinite characteristics' ***get*** method , see section 12.7.3.3, is used to retrieve its *'w0'* or *'slope'* property. which implements the second corner frequency, $H_{SI2}(j\omega)$ .

## 12.8.3.4fit

This method fits a rational transfer function to the Bode step.

**Inputs**

The algorithm requires the following input:

plot_style , defines the plots that must be generated.

**Outputs**

The algorithm produces the following output:

$N(s)$ , the numerator of the rational transfer function.

$D(s)$ , the denominator of the rational transfer function.

**Pre-conditions**

The *plot_style* input is checked to see that it has one of the following values:

1. *'none'*
2. *'detail'*
3. *'fit'*

**Algorithm**

The rational transfer function is fitted to the Bode step in the following manner:

1. A rational transfer function is fitted to the semi-infinite characteristic class which implements the first corner frequency, $H_{SI1}(j\omega)$ . This is achieved by calling the semi-infinite characteristic classes' ***fit*** method, see section 12.7.3.4. The method returns a numerator, $N_1(s)$ , and denominator, $D_1(s)$ .
2. A rational transfer function is fitted to the semi-infinite characteristic class which implements the second corner frequency, $H_{SI2}(j\omega)$ . This is achieved by calling the semi-infinite characteristic classes' ***fit*** method, see section 12.7.3.4. The method returns a numerator, $N_2(s)$ , and denominator, $D_2(s)$ .
3. The numerators obtained, $N_1(s)$ and $N_2(s)$ , are multiplied as polynomials using Matlabs' ***conv*** function.
4. The denominators obtained, $D_1(s)$ and $D_2(s)$ , are multiplied as polynomials using Matlabs' ***conv*** function.
5. If the *plot_style* input is *'detail'* or *'fit'* the Bode step and the rational transfer function calculated are plotted. This is achieved as follows:
   (a) The frequencies used to plot the the Bode step and the fitted rational transfer function are generated using the Bode step classes' ***calcFreqRange*** method, see section 12.8.3.7.
   (b) The magnitudes and phases of the Bode step at the frequencies used for plotting are then calculated using the Bode step classes' ***calcResponse*** method, see section 12.8.3.1.
   (c) The magnitudes and phases of the fitted rational transfer function at the frequencies used for plotting are calculated using Matlabs' ***bode*** function.
   (d) Matlabs' plotting routines are used to plot both the Bode step and the fitted rational transfer function using the magnitudes and phases calculated.

### 12.8.3.5bode

This method generates a Bode plot of the Bode step characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The Bode plot is generated in the following manner:

1. The Bode step classes ***calcFreqRange*** method, see section 12.8.3.7, is used to calculate the range of frequencies needed to plot the Bode step characteristic and generate a sample of frequencies over the range.
2. The Bode step classes ***calcResponse*** method, see section 12.8.3.1, is used to calculate the magnitudes and phases of the Bode step characteristic at the sampled frequencies.
3. Matlabs plotting functions are used to plot the Bode plot of the Bode step characteristic using the magnitudes, phases and frequencies calculated.

### 12.8.3.6nichols

This method generates a Nichols chart plot of the Bode step characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The Nichols chart plot is generated in the following manner:

1. The Bode step classes ***calcFreqRange*** method, see section 12.8.3.7, is used to calculate the range of frequencies needed to plot the Bode step characteristic and generate a sample of frequencies over the range.
2. The Bode step classes ***calcResponse*** method, see section 12.8.3.1, is used to calculate the magnitudes and phases of the Bode step characteristic at the sampled frequencies.
3. Matlabs plotting functions are used to plot the Nichols chart plot of the Bode step characteristic using the magnitudes, phases and frequencies calculated.

### 12.8.3.7 calcFreqRange

This method calculates the range of frequencies required to plot the Bode step.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm requires the following output:

$\omega$ , the frequencies which can be used to plot the Bode step (rad/s).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The range of frequencies is generated in the following manner:

2. The first and second corner frequencies ( $\omega_1$ and $\omega_2$ respectively) are retrieved using the ***get*** methods of the semi-infinite characteristic classes which implement the first and second corner frequencies, see section 12.7.3.3.
3. The minimum frequency in the range is calculated from

$$\omega_{MIN} = floor\left(\log(\omega_1)\right) - 2$$

4. The maximum frequency in the range is calculated from

$$\omega_{MAX} = ceil\left(\log(\omega_2)\right) + 2$$

5. Matlabs' ***logspace*** function is used to generate the sample of frequencies in the range.

### 12.8.3.8calcSecondCornerFreq

This method calculates the second corner frequency from the slopes $k_1$ and $k_2$ and first corner frequency, $\omega_1$ , provided.

**Inputs**

The algorithm requires the following inputs:

$k_1$ , the slope of the magnitude greater than the first corner frequency, $\omega_1$ , and smaller than the second corner frequency, $\omega_2$ , (dB/decade).

$\omega_1$ , the first corner frequency (rad/s).

$k_2$ , the slope of the magnitude greater than the second corner frequency, $\omega_2$ , (dB/decade).

**Outputs**

The algorithm produces the following output:

$\omega_2$ , the second corner frequency (rad/s).

**Pre-conditions**

As this is a private method the class generates the inputs, therefore nothing needs to be checked.

**Algorithm**

The second corner frequency is calculated as follows:

$$\omega_2 = \left( \frac{k_1 - k_2}{k_1} \right) \cdot \omega_1$$

## *12.9 Bode cut-off class*

### 12.9.1 Overview

The Bode cut-off characteristic allows the designer to specify the magnitude and phase in different frequency ranges. For a defined cut-off frequency, $\omega_0$ , the magnitude is specified for frequencies lower than $\omega_0$ and the phase for frequencies greater than $\omega_0$ . See section 6.2.1 for a discussion on the Bode cut-off characteristic.

The Bode cut-off characteristic class has the following functionality:

1. It is capable of calculating the Bode cut-off characteristic at a specified frequency.
2. It is capable of fitting a rational transfer function to the Bode cut-off characteristic.
3. It is capable of calculating the frequency at which the Bode cut-off characteristic has a specified magnitude.
4. It is capable of calculating the frequency at which the Bode cut-off characteristic has a specified phase.
5. It is capable of allowing the user to set and retrieve the attributes of the Bode cut-off characteristic.
6. It is capable of calculating the following properties of the Bode cut-off characteristic:
   a) The bandwidth of the Bode cut-off characteristic.
   b) The cross-over frequency of the Bode cut-off characteristic.
   c) The phase margin of the Bode cut-off characteristic.

7. It is capable of plotting Bode cut-off characteristic on a Bode plot.
8. It is capable of plotting the Bode cut-off characteristic on the Nichols chart.

### 12.9.2 Attributes

In order for the Bode cut-off class to be capable of performing the above mentioned functionality it must keep track of the following attributes:

1. $M_0$ , the magnitude required at the cut-off frequency, $\omega_0$ , in decibels.
2. $\phi_{PM}$ , the phase margin required (deg).
3. $\omega_0$ , the cut-off frequency (rad/s).
4. $k$ , the type of the system required.

### 12.9.3 Methods

#### 12.9.3.1set

This method allows the user to set the properties of the Bode cut-off class.

**Inputs**

The algorithm requires the following input:

$(property\ ,value)$ , the property-value pair. The *property* part specifies the property that needs to be set to the *value* part of the property-value pair.

**Outputs**

The algorithm generates no outputs.

**Pre-conditions**

The property part of the property-value pair is checked to see that it has one of the following values:

1. *'m0'*
2. *'phasemargin'*
3. *'w0'*
4. *'type'*

If the property part is *'m0'* or *'w0'*, then the value part is checked to see that it is a scalar real value greater than 0.

If the property part is *'phasemargin'*, then the value part is checked to see that it is a scalar real value greater than 0° and smaller than 180°.

If the property part is *'type'*, then the value part is checked to see that it is a scalar integer value greater than or equal to 0.

**Algorithm**

The method runs through all the property-value pairs specified and sets the attributes to the values specified.

If the property part is *'m0'*, then the value part of the property-value pair is used to set the magnitude required at the cut-off frequency, $M_0$ .

If the property part is *'phasemargin'*, then the value part of the property-value pair is used to set the phase margin, $\phi_{PM}$ .

If the property part is *'w0'*, then the value part of the property-value pair is used to set the cut-off frequency, $\omega_0$ .

If the property part is *'m0'*, then the value part of the property-value pair is used to set the type of the system, $k$ .

## 12.9.3.2get

This method allows the user to retrieve the values of the properties of the Bode cut-off class.

**Inputs**

The algorithm requires the following input:

$property$ , the property for which the value is required.

**Outputs**

The algorithm produces the following outputs:

$value$ , the value of the property specified.

**Pre-conditions**

The property part of the property-value pair is checked to see that it has one of the following values:

1. *'m0'*
2. *'w0'*
3. *'phasemargin'*
4. *'type'*

**Algorithm**

If the property part of the property-value pair is *'m0'* the value of the magnitude required at the cut-off frequency, $M_0$ , is returned.

If the property part of the property-value pair is *'w0'* the cut-off frequency, $\omega_0$ , is returned.

If the property part of the property-value pair is *'phasemargin'* the phase margin, $\phi_{PM}$ , is returned.

If the property part of the property-value pair is *'type* the type of the system, $k$ , is returned.

### 12.9.3.3calcBandwidth

This method calculates the bandwidth of the Bode cut-off characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$\omega_B$ , the bandwidth of the Bode cut-off (rad/s).

**Pre-conditions**

As there are no inputs nothing needs to be checked.

**Algorithm**

The bandwidth is calculated as followings:

1. The Bode cut-offs' phase related to the phase margin is calculated using

$$\phi = \phi_{PM} - 180^o$$

2. The magnitude of the -3 dB m-circle is then calculated at the Bode cut-offs' phase. This is achieved by instantiating an instance of the m-circle class with a closed loop magnitude of -3 dB, and using its method ***calcMag***, see section 12.2.3.1.
3. The frequency at which the Bode cut-off characteristic crosses the -3 dB m-circle gives the bandwidth of the Bode cut-off characteristic. This is calculated using the Bode cut-off characteristic classes' ***calcFreqAtMag*** method (see section 12.9.3.5) and passing it the magnitude calculated in the previous step.

### 12.9.3.4calcCrossoverFreq

This method calculates the crossover frequency of the Bode cut-off characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following output:

$\omega_C$ , the crossover frequency of the Bode cut-off (rad/s).

**Pre-conditions**

As there are no inputs nothing needs to be checked.

**Algorithm**

The crossover frequency is the frequency at which the Bode cut-off characteristic is equal to 0 dB. This is calculated using the Bode cut-off characteristic classes' ***calcFreqAtMag*** method (see section 12.9.3.5) and passing it a magnitude of 0 dB.

### 12.9.3.5calcFreqAtMag

This method calculates the frequencies at which the Bode cut-off characteristic has the magnitudes specified.

**Inputs**

The algorithm requires the following input:

$|H_{BC}(j\omega)|$ , the magnitudes for which frequencies of the Bode cut-off characteristic are required (dB).

**Outputs**

The algorithm produces the following output:

$\omega$ , the frequencies at which the Bode cut-off characteristic has the magnitudes specified (rad/s).

**Pre-conditions**

The magnitudes specified for the Bode cut-off characteristic, $|H_{BC}(j\omega)|$ , must be real.

**Algorithm**

The frequencies at which the Bode cut-off has the magnitudes are specified are calculated as follows:

1. If the type of the Bode cut-off characteristic is zero, then frequencies are given by

$$\omega = \begin{cases} NaN & for \quad |H_{BC}(j\omega)| > M_0 \\ \frac{\omega_0}{2} \cdot \left(k - \frac{1}{k}\right) & for \quad |H_{BC}(j\omega)| \leq M_0 \end{cases}$$

where

$$k = \left(\frac{M_0}{|H_{BC}(j\omega)|}\right)^{\frac{1}{2\cdot\alpha}} \quad \text{and} \quad \alpha = 1 - \frac{\phi_{PM}}{180^o} - \frac{k}{2}$$

2. If the type of the Bode cut-off characteristic is greater than zero, then frequencies are given by

$$\omega = \omega_0 \cdot \left(\frac{M_0}{|H_{BC}(j\omega)|}\right) \quad \text{for} \quad |H_{BC}(j\omega)| \geq M_0$$

If $|H_{BC}(j\omega)| < M_0$ , the frequencies need to be calculated numerically. This is achieved using Matlab's ***fzero*** function and the Bode cut-off characteristic class's ***calcResponse*** method, see section 12.9.3.8.

## 12.9.3.6calcFreqAtPhase

This method calculates the frequencies at which the Bode cut-off characteristic has the phases specified.

**Inputs**

The algorithm requires the following input:

$\sphericalangle H_{BC}(j\omega)$ , the phases for which frequencies of the Bode cut-off are required (deg).

**Outputs**

The algorithm produces the following output:

$\omega$ , the frequencies at which the Bode cut-off has the phases specified (rad/s).

**Pre-conditions**

The phases are checked to see that they lie within the range of frequencies that is appropriate to the Bode cut-off characteristic. The minimum and maximum phases of the Bode ideal characteristic are given by

$$\measuredangle H_{BC_{MIN}}(j\omega)=\begin{cases}-180^{o}+\phi_{PM} & for\, k\leqslant 1\\ -90\cdot k & otherwise\end{cases}$$

$$\measuredangle H_{BC_{MAX}}(j\omega)=\begin{cases}-90\cdot k & for\, k\leqslant 1\\ -180^{o}+\phi_{PM} & otherwise\end{cases}$$

If any of the phases specified are smaller than the minimum phase, $\measuredangle H_{BC_{MIN}}(j\omega)$ , or greater than the maximum phase, $\measuredangle H_{BC_{MAX}}(j\omega)$ , *NaN* will be returned for that frequency.

**Algorithm**

The frequencies at which the Bode cut-off has the phases specified, $\measuredangle H_{BC}(j\omega)$ , are given by

$$\omega=\frac{\omega_0}{2\cdot\alpha}\cdot\sin\left(\measuredangle H_{BC}(j\omega)+90\cdot k\right)$$

where

$$\alpha=1-\frac{\phi_{PM}}{180^{o}}-\frac{k}{2}$$

## 12.9.3.7calcPhaseMargin

This method calculates the phase margin required of the Bode cut-off characteristic in order to meet a set of stability bounds.

**Inputs**

The algorithm requires the following input:

$\tilde{B}$ , the vector of stability bounds which the Bode cut-off characteristic must meet.

**Outputs**

The algorithm produces the following output:

$\phi_{PM}$ , the required phase margin of the Bode cut-off characteristic (rad/s).

**Pre-conditions**

As the input is a vector of stability bound classes which does its own checking, nothing needs to be checked.

**Algorithm**

The phase margin required of the Bode cut-off characteristic in order to meet a set of stability bounds is calculated as follows:

1. As a numerical method is going to be used to calculate the phase margin required, bounds must be calculated which bracket the phase margin. The phase margin itself is not bracketed, but the minimum phase of the Bode cut-off characteristic is. This is achieved as follows:
   a) The maximum phase of each stability bound in the vector of stability bounds, $\tilde{B}$ , are calculated. This is done using the stability bound class's ***getMaxPhase*** method, see section 12.5.3.5.
   b) The minimum and maximum phase values for the minimum phase of the Bode cut-off characteristic are then minimum and maximum phases in the vector of maximum phases of the stability bounds, calculated in the previous step.
2. Matlab's ***fminbnd*** function is used to calculate the minimum phase of the Bode cut-off characteristic which meets the stability bounds. The minimum and maximum phase values for the minimum phase of the Bode cut-off characteristic, and the private function ***calcPhaseError*** are used by the ***fminbnd*** function. The private function ***calcPhaseError*** is discussed in section 12.9.3.13.

3. The phase margin required is then calculated from the minimum phase of the Bode cut-off characteristic returned by the ***fminbnd*** function, by adding 180°.

### 12.9.3.8calcResponse

This method calculates the magnitude and phase of the Bode cut-off for a specified set of frequencies.

**Inputs**

The algorithm requires the following input:

$\omega$ , the frequencies at which the magnitudes and phases of the Bode cut-off are required (rad/s).

**Outputs**

The algorithm produces the following outputs:

$|H_{BC}(j\omega)|$ , the magnitudes of the Bode cut-off for the frequencies specified (dB).

$\sphericalangle H_{BC}(j\omega)$ , the phases of the Bode cut-off for the frequencies specified (deg).

**Pre-conditions**

The frequencies specified, $\omega$ , are checked that they are real and greater than zero.

**Algorithm**

The magnitude of the bode ideal Bode cut-off characteristic is given by:

$$|H_{BC}(j\omega)|=\begin{cases}\left(\frac{\omega_0}{\omega}\right)^k\cdot M_0 & for \quad \omega\leqslant\omega_0\\ \left(\frac{\omega_0}{\omega}\right)^k\cdot\frac{M_0}{\left(\frac{\omega}{\omega_0}+\sqrt{\left(\frac{\omega}{\omega_0}\right)^2-1}\right)^{2\alpha}} & for \quad \omega>\omega_0\end{cases}$$

and the phase is given by:

$$\measuredangle H_{BC}(j\omega)=\left\{\begin{matrix} -2\alpha\tan^{-1}\left(\dfrac{\omega}{\sqrt{\omega_0^2-\omega^2}}\right)-\dfrac{\pi}{2}\cdot k & for & \omega\leqslant\omega_0 \\ -\alpha\pi-\dfrac{\pi}{2}\cdot k & for & \omega>\omega_0 \end{matrix}\right\}\ \text{radians}$$

where

$$\alpha=1-\frac{\phi_{PM}}{180^o}-\frac{k}{2}$$

The magnitude and phases are of the Bode cut-off characteristic are converted to decibels and degrees respectively before being returned by the function.

### 12.9.3.9fit

This method fits a rational transfer function to the Bode cut-off characteristic.

**Inputs**

The algorithm requires the following input:

$\omega_{MAX}$ , the maximum frequency of the Bode cut-off characteristic to which the rational transfer function must be fitted (rad/s)

$e$ , the pole-zero excess that is required of the fitted response.

plot_style , defines the plots that must be generated.

**Outputs**

The algorithm produces the following output:

$N(s)$ , the numerator of the rational transfer function.

$D(s)$ , the denominator of the rational transfer function.

**Pre-conditions**

The *plot_style* input is checked to see that it has one of the following values:

1. *'none'*
2. *'detail'*
3. *'fit'*

**Algorithm**

The fitting of the rational transfer function to the Bode cut-off characteristic is split into two parts. The first part approximates the Bode ideal cut-off characteristic using semi-infinite characteristic segments with slopes which are multiples of 20 dB/decade. The second part fits a lead-lag segment to the difference between the Bode cut-off characteristic and the semi-infinite segments so that the phase between the corner frequency attribute, $\omega_0$, and the maximum frequency to which the rational transfer function must be fitted, $\omega_{MAX}$, can be approximated.

An extra semi-infinite characteristic is added with a corner frequency at $\omega_{MAX}$ and a slope which gives the desired pole-zero excess, $e$.

If plot_style input is 'detail' or 'fit' Bode plots of the semi-infinite characteristic segments and the lead-lag segments are plotted on a Bode plot using Matlab's plotting routines.

### 12.9.3.10bode

This method generates a Bode plot of the Bode cut-off characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The Bode plot is generated in the following manner:

1. The Bode cut-off classes ***calcFreqRange*** method, see section 12.9.3.12, is used to calculate the range of frequencies needed to plot the Bode cut-off characteristic and generate a sample of frequencies over the range.
2. The Bode cut-off classes ***calcResponse*** method, see section 12.9.3.8, is used to calculate the magnitudes and phases of the Bode cut-off characteristic at the sampled frequencies.
3. Matlabs plotting functions are used to plot the Bode plot of the Bode cut-off characteristic using the magnitudes, phases and frequencies calculated.

#### 12.9.3.11 nichols

This method generates a Nichols chart plot of the Bode cut-off characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The Nichols chart plot is generated in the following manner:

1. The Bode cut-off classes ***calcFreqRange*** method, see section 12.9.3.12, is used to calculate the range of frequencies needed to plot the Bode cut-off characteristic and generate a sample of frequencies over the range.
2. The Bode cut-off classes ***calcResponse*** method, see section 12.9.3.8, is used to calculate the magnitudes and phases of the Bode cut-off characteristic at the sampled frequencies.

3. Matlabs plotting functions are used to plot the Nichols chart plot of the Bode cut-off characteristic using the magnitudes, phases and frequencies calculated.

### 12.9.3.12calcFreqRange

This method calculates the range of frequencies required to plot the Bode cut-off characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates the following output:

$\omega$ , the frequencies which can be used to plot the Bode cut-off characteristic (rad/s).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The range of frequencies is generated in the following manner:

1. The crossover frequency, $\omega_C$ is calculated using the ***calcCrossoverFreq*** method of the Bode cut-off class, see section 12.9.3.4.
2. The minimum frequency in the range is calculated using the Bode cut-off characteristic's cut-off frequency, $\omega_0$ , as follows

$$\omega_{MIN} = floor\left(\log(\omega_0)\right) - 2$$

3. The maximum frequency in the range is calculated from

$$\omega_{MAX} = ceil\left(\log(\omega_C)\right) + 2$$

4. Matlabs' ***logspace*** function is used to generate the sample of frequencies in the range.

## 12.9.3.13calcPhaseError

This method calculates the difference between the minimum phase of the Bode cut-off characteristic and the phase required to meet the stability bounds. This method is private and is used by the ***calcPhaseMargin*** method to calculate the phase margin required of the Bode cut-off characteristic in order to meet a set of stability bounds, see section 12.9.3.7.

**Inputs**

The algorithm requires the following inputs:

$\measuredangle H_{BC}(j\omega)\big|_{MIN}$ , the minimum phase of the Bode cut-off characteristic (deg).

$\tilde{B}$ , the vector of stability bounds which the Bode cut-off characteristic must meet.

$\tilde{\omega}_B$ , the vector containing the frequencies of the stability bounds (rad/s).

$M_0$ , the magnitude required at the cut-off frequency, $\omega_0$ , in decibels.

$\omega_0$ , the cut-off frequency (rad/s).

$k$ , the type of the system required.

**Outputs**

The algorithm generates the following output:

$\Delta\phi$ , the difference between the minimum phase of the Bode cut-off characteristic and the phase required to meet the stability bounds (deg).

**Pre-conditions**

As all the inputs are generated by the Bode cut-off characteristic class, nothing needs to be checked.

**Algorithm**

The difference between the minimum phase of the Bode cut-off characteristic and the phase required to meet the stability bounds is calculated as follows:

1. The phase margin of the Bode cut-off characteristic, $\phi_{PM}$, is calculated from the minimum phase of the Bode cut-off characteristic as follows:

$$\phi_{PM} = \sphericalangle H_{BC}(j\omega)\big|_{MIN} + 180^o$$

2. The magnitudes and phases of the Bode cut-off characteristic, which has the phase margin calculated, at the frequencies of the stability bounds are calculated using the classes ***calcResponse*** method.
3. The maximum phase of the stability bounds, $\sphericalangle \tilde{B}_{MAX}$, at the magnitudes of the Bode cut-off characteristic are calculated using the ***calcPhase*** method of each stability bound class in the vector of stability bounds, $\tilde{B}$.
4. The difference between the minimum phase of the Bode cut-off characteristic and the phase required to meet the stability bounds is then given by

$$\Delta\phi = min\left(\left|\sphericalangle H_{BC}(j\tilde{\omega}_B) - \sphericalangle \tilde{B}_{MAX}\right|\right)$$

## *12.10 Bode ideal characteristic class*

### 12.10.1Overview

The Bode ideal characteristic defines a loop transmission that has a high low frequency gain, for good low frequency performance, and has definable stability characteristics. The Bode ideal characteristic is discussed in section 6.2.

The Bode ideal characteristic class has the following functionality:

1. It is capable of calculating the Bode ideal characteristic at a specified frequency.
2. It is capable of fitting a rational transfer function to the Bode ideal characteristic.
3. It is capable of plotting Bode ideal characteristic on a Bode plot.
4. It is capable of plotting the Bode ideal characteristic on the Nichols chart.

### 12.10.2Attributes

In order for the Bode ideal characteristic class to be capable of performing the above mentioned functionality it must keep track of the following attributes:

1. $H_{TBC}(j\omega)$ , the instance of the Bode cut-off class used to implement the typed Bode cut-off part of the Bode ideal characteristic.
2. $H_{BS}(j\omega)$ , the instance of the Bode step class used to implement the Bode step part of the Bode ideal characteristic.

These objects are instantiated by the Bode ideal characteristic classes' constructor. The arguments required these objects' constructors is also calculated by the Bode ideal characteristics' constructor. This is achieved in the following manner:

1. The arguments passed to the Bode ideal characteristic class's constructor which are passed directly to the Bode cut-off classes' constructor are the following:
   a) $M_0$ , the magnitude required at the cut-off frequency, $\omega_0$ , in decibels.
   b) $\phi_{PM}$ , the phase margin required (deg).
   c) $\omega_0$ , the cut-off frequency (rad/s).
   d) $k$ , the type of the system required.
2. The arguments passed to the Bode ideal characteristic class's constructor which are used to calculate the arguments required by the Bode step class's constructor are:
   a) $\phi_{PM}$ , the phase margin required (deg).
   b) $M_{GM}$ , the gain margin required (dB).
   c) $e$ , the pole-zero excess required.
3. The arguments required by the Bode step class's constructor are calculated as follows:
   a) $k_1$ , the desired slope of the magnitude between the first and second corner frequencies is given by

   $$k_1 = \frac{180^o - \phi_{PM}}{90^o}$$

   b) $\omega_1$ , the first corner frequencies is calculated by the Bode cut-off characteristic class's ***calcFreqAtMag*** method using the gain margin, $M_{GM}$ ,see section .
   c) $k_2$ , the desired slope of the magnitude after the second corner frequency is given by

   $$k_2 = -e - k$$

## 12.10.3Methods

### 12.10.3.1calcResponse

This method calculates the magnitude and phase of the Bode ideal characteristic for a specified set of frequencies.

**Inputs**

The algorithm requires the following input:

$\omega$ , the frequencies at which the magnitudes and phases of the Bode ideal characteristic are required (rad/s).

**Outputs**

The algorithm produces the following outputs:

$|H_{TBI}(j\omega)|$ , the magnitudes of the Bode ideal characteristic for the frequencies specified (dB).

$\measuredangle H_{TBI}(j\omega)$ , the phases of the Bode ideal characteristic for the frequencies specified (deg).

**Pre-conditions**

The frequencies specified, $\omega$ , are checked that they are real and greater than zero.

**Algorithm**

The magnitudes and phases of the Bode ideal characteristic at the frequencies specified are calculated in the following manner:

1. The magnitudes and phases of the Bode cut-off class which implements the Bode cut-off part of the Bode ideal characteristic, $H_{BC}(j\omega)$ , are calculated using its ***calcResponse*** method, see section 12.9.3.8.
2. The magnitudes and phases of the Bode step class which implements the Bode step part of the Bode ideal characteristic, $H_{BS}(j\omega)$ , are calculated using its ***calcResponse*** method, see section 12.8.3.1.
3. The magnitude (in decibels) of the Bode ideal characteristic is given by:

$$|H_{TBI}(j\omega)|=|H_{TBC}(j\omega)|+|H_{BS}(j\omega)|$$

And the phase (in degrees) is given by:

$$\sphericalangle H_{TBI}(j\omega)=\sphericalangle H_{TBC}(j\omega)+\sphericalangle H_{BS}(j\omega)$$

### 12.10.3.2fit

This method fits a rational transfer function to the Bode ideal characteristic.

**Inputs**

The algorithm requires the following input:

plot_style , defines the plots that must be generated.

**Outputs**

The algorithm produces the following output:

$N(s)$ , the numerator of the rational transfer function.

$D(s)$ , the denominator of the rational transfer function.

**Pre-conditions**

The *plot_style* input is checked to see that it has one of the following values:

1. *'none'*
2. *'detail'*
3. *'fit'*

**Algorithm**

The rational transfer function is fitted to the Bode ideal characteristic in the following manner:

1. A rational transfer function is fitted to the Bode cut-off characteristic class attribute by calling the its ***fit*** method, see section 12.9.3.9. The method returns a numerator, $N_{BC}(s)$ , and denominator, $D_{BC}(s)$ . The roll-off specified for the fit above the

corner frequency $\omega_1$ as calculated in this classes constructor is specified as 0 dB/decade. This allows a semi-infinite characteristic to be used to fit the frequencies of he Bode ideal characteristic higher than $\omega_1$ .

2. A rational transfer function is fitted to the semi-infinite characteristic class which implements the frequencies of the Bode ideal characteristic higher than $\omega_1$ . This is achieved by instantiating a the semi-infinite characteristic class with a corner frequency, $\omega_2$ , and a roll-off specified by the pole zero excess, $e$ . The corner frequency, $\omega_2$ , is retrieved from the Bode step characteristic class attribute using its ***get*** method, see section 12.9.3.2. The the semi-infinite characteristic class's ***fit*** method, see section 12.7.3.4, is then used to fit a rational transfer function to itself. The method returns a numerator, $N_{SI}(s)$ , and denominator, $D_{SI}(s)$ .
3. The numerators obtained, $N_{BC}(s)$ and $N_{SI}(s)$ , are multiplied as polynomials using Matlabs' ***conv*** function to give the complete numerator $N(s)$ .
4. The denominators obtained, $D_{BC}(s)$ and $D_{SI}(s)$ , are multiplied as polynomials using Matlabs' ***conv*** function to give the complete denominator $D(s)$ .
5. If the *plot_style* input is *'detail'* or *'fit'* the Bode ideal characteristic and the rational transfer function calculated are plotted on both the Bode and Nichols chart.

### 12.10.3.3bode

This method generates a Bode plot of the Bode ideal characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The Bode plot is generated in the following manner:

1. The Bode ideal classes ***calcFreqRange*** method, see section 12.10.3.5, is used to calculate the range of frequencies needed to plot the Bode ideal characteristic and generate a sample of frequencies over the range.
2. The Bode ideal classes ***calcResponse*** method, see section 12.10.3.1, is used to calculate the magnitudes and phases of the Bode ideal characteristic at the sampled frequencies.
3. Matlabs plotting functions are used to plot the Bode plot of the Bode ideal characteristic using the magnitudes, phases and frequencies calculated.

### 12.10.3.4nichols

This method generates a Nichols chart plot of the Bode ideal characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm generates no outputs besides the plot.

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The Nichols chart plot is generated in the following manner:

1. The Bode ideal classes ***calcFreqRange*** method, see section 12.10.3.5, is used to calculate the range of frequencies needed to plot the Bode ideal characteristic and generate a sample of frequencies over the range.
2. The Bode ideal classes ***calcResponse*** method, see section 12.10.3.1, is used to calculate the magnitudes and phases of the Bode ideal characteristic at the sampled frequencies.
3. Matlabs plotting functions are used to plot the Nichols chart plot of the Bode ideal characteristic using the magnitudes, phases and frequencies calculated.

### 12.10.3.5calcFreqRange

This method calculates the range of frequencies required to plot the Bode ideal characteristic.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm requires the following output:

$\omega$ , the frequencies which can be used to plot the Bode ideal characteristic (rad/s).

**Pre-conditions**

As there are no inputs, nothing needs to be checked.

**Algorithm**

The range of frequencies is generated in the following manner:

1. The cut-off frequency of the Bode cut-off part of the Bode ideal characteristic, $\omega_0$ , is retrieved using the ***get*** method of the Bode cut-off class, see section 12.9.3.2.
1. The second corner frequency of the Bode step part of the Bode ideal characteristic, $\omega_2$ , is retrieved using the ***get*** method of the Bode step class, see section 12.8.3.3.
2. The minimum frequency in the range is calculated from

$$\omega_{MIN} = floor\left(\log(\omega_0)\right) - 2$$

3. The maximum frequency in the range is calculated from

$$\omega_{MAX} = ceil\left(\log(\omega_2)\right) + 2$$

4. Matlabs' ***logspace*** function is used to generate the sample of frequencies in the range.

## 12.11 The second order model class

### 12.11.1 Overview

The second order model class calculates the properties of a second order model which is described by the following differential equation:

$$\frac{1}{\omega_n^2}\cdot\frac{d^2 y(t)}{dt^2}+\frac{2\cdot\zeta}{\omega_n}\cdot\frac{dy(t)}{dt}+y(t)=c(t)$$

where:

| | |
|---|---|
| $y(t)$ | is the model output with respect to time. |
| $c(t)$ | is the model command input with respect to time. |
| $\zeta$ | is the damping ratio. |
| $\omega_n$ | is the natural frequency. |

The second order model is discussed in section 3.4.1.

The second order model m-circle class has the following functionality:

1. It is capable of calculating the following properties of the second order time response to a step input:
   - The magnitude of the step response at a specified time.
   - The overshoot.
   - The rise time.
   - The settling time.
   - The turning points.
2. It is capable of calculating the following properties of the second order frequency response:
   - The resonant peak of the frequency response.
3. It is capable of allowing the user to set and retrieve the following attributes of the second order model class:
   - The damping ratio, $\zeta$ .
   - The natural frequency, $\omega_n$ .

## 12.11.2Attributes

In order for the second order model class to be capable of performing the above mentioned functionality it must keep track of the following attributes:

1. $\zeta$ , the damping ratio of the second order model.
2. $\omega_n$ , the natural frequency of the second order model.

## 12.11.3Methods

### 12.11.3.1calcResponse

This method calculates the magnitude of the second order model's step response at a specified time.

**Inputs**

The algorithm requires the following input:

$t$ , the times at which the magnitude of the second order model is required (s).

**Outputs**

The algorithm produces the following output:

$y$ , the magnitude of the second order model at the times specified.

**Pre-conditions**

The times specified, $t$ , are checked that they are real and greater than zero.

**Algorithm**

The magnitude of the second order model is given by

$$y(t)=\begin{cases} 1-e^{-\zeta\cdot\omega_n\cdot t}\cdot\left(\cos(\omega_0\cdot t)+\dfrac{\zeta\cdot\omega_n}{\omega_0}\cdot\sin(\omega_0\cdot t)\right) & for \quad \zeta<1 \\ 1-e^{-\zeta\cdot\omega_n\cdot t}\cdot(1+\omega_n\cdot t) & for \quad \zeta=1 \\ 1-e^{-\zeta\cdot\omega_n\cdot t}\cdot\left(\cosh(\omega_x\cdot t)+\dfrac{\zeta\cdot\omega_n}{\omega_x}\cdot\sinh(\omega_x\cdot t)\right) & for \quad \zeta>1 \end{cases}$$

where:

$$\omega_0=\omega_n\cdot\sqrt{1-\zeta^2}$$

$$\omega_x=\omega_n\cdot\sqrt{\zeta^2-1}$$

### 12.11.3.2getOvershoot

This method calculates the percentage overshoot of the second order model's time response.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

- $t_{os}$ , the time at which the maximum overshoot occurs (s).
- $y_{os}$ , the magnitude of the overshoot expressed as a percentage of the second order models final value (%).

**Pre-conditions**

As there are no inputs nothing needs to be checked.

**Algorithm**

If the damping ratio of the second order model, $\zeta$ , is smaller than 1 the overshoot is calculated as follows

1. The overshoot of a second order model is given by the first turning point. The time at which the overshoot occurs, $t_{os}$ , and the amplitude of the overshoot, $y_p$ , can be calculated using the second order model classes ***getTurningPoint*** method (see section 12.11.3.6).
2. The overshoot calculated can then be converted to a percentage as follows

$$y_{ov}=(y_p-1)\cdot 100$$

If the damping ratio of the second order model, $\zeta$ , is greater than or equal to 1 *NaN* is returned.

### 12.11.3.3getResonantPeak

This method calculates the resonant peak of the second order model's frequency response.

**Inputs**

The algorithm requires no inputs.

**Outputs**

The algorithm produces the following outputs:

$\omega_p$ , the frequency at which the resonant peak occurs (rad/s).
$M_p$ , the magnitude of the resonant peak (dB).

**Pre-conditions**

As there are no inputs nothing needs to be checked.

**Algorithm**

If the damping ratio of the second order model, $\zeta$ , is smaller than 1 the resonant peak is calculated as follows

1. The frequency at which the resonant peak occurs is given by

$$\omega_p = \omega_n \cdot \sqrt{1 - 2 \cdot \zeta^2}$$

2. The magnitude of the resonant peak is given by

$$M_p = 20 \cdot \log\left(\frac{1}{2 \cdot \zeta \cdot \sqrt{1 - \zeta^2}}\right)$$

If the damping ratio of the second order model, $\zeta$ , is greater than or equal to 1, *NaN* is returned.

## 12.11.3.4 getRiseTime

This method calculates the rise time of the second order model's time response.

**Inputs**

The algorithm requires the following input:

$x_r$ , the magnitude of the second order models' step response for which the rise time is required. The value is expressed as a percentage of the second order model step responses' final value.

**Outputs**

The algorithm produces the following output:

$t_r$ , the rise time (s).

**Pre-conditions**

The magnitude of the second order models' step response for which the rise time is required, $x_r$ , is checked to see that it is greater than 0 and smaller than 1.

**Algorithm**

The rise time is calculated in the following manner:

2. As a numerical algorithm will be used to solve for the rise time, the rise time needs to be bracket.
   (a) If the second order models damping ratio is smaller than 1, the rise time can be bracket by a time of 0 and the time at which the second order model has its maximum overshoot value. The time to the maximum overshoot value is calculated using the second order models ***calcOvershoot*** method, see section 12.11.3.2.
   (b) If the second order models damping ratio is greater than or equal to 1, the rise time can be estimated by

$$t'_r = \frac{\log\left(\frac{100}{x_r}\right)}{\zeta \cdot \omega_n}$$

   In order to ensure that the rise time is bracket a time of 0 and 10 times the estimated rise time value, $t'_r$ , is used.
3. Matlabs' ***fzero*** function is used to solve for the rise time using the bracketed rise times calculated and the second order model classes' ***calcResponse*** method, see section 12.11.3.1.

#### 12.11.3.5 getSettlingTime

This method calculates the settling time of the second order model's time response.

**Inputs**

The algorithm requires the following input:

$x_s$ , the difference in magnitude of the second order model step responses' final value and the value of the step response at the settling time required. The value is expressed as a percentage of the second order model step responses' final value.

**Outputs**

The algorithm produces the following output:

$t_s$ , the settling time (s).

**Pre-conditions**

The magnitude of the second order model's step response for which the settling time is required, $x_s$ , is checked to see that it is greater than 0 and smaller than 1.

**Algorithm**

The settling time is calculated in the following manner:

1. As a numerical algorithm will be used to solve for the settling time, the settling time needs to be bracket.
   (a) If the second order models damping ratio is smaller than 1, the settling time must be bracketed by finding the turning points whose magnitude is just greater than the settling time magnitudes specified. The next turning points difference in magnitude from the step responses final value will be smaller than the settling time magnitude specified. The turning points are calculated using the second order models ***getTurningPoints*** method, see section 12.11.3.6.
   (b) If the second order models damping ratio is greater than or equal to 1, the settling time can be estimated by

$$t'_s = \frac{\log\left(\frac{100}{x_s}\right)}{\zeta \cdot \omega_n}$$

   In order to ensure that the rise time is bracket a time of 0 and 10 times the estimated settling time value, $t'_s$ , is used.
2. Matlabs' ***fzero*** function is used to solve for the settling time using the bracketed settling times calculated and the second order model classes' ***calcResponse*** method, see section 12.11.3.1.

## 12.11.3.6getTurningPoints

This method calculates the turning points of the time response of the second order model.

**Inputs**

The algorithm requires the following input:

$n$ , the turning point required. As there are theoretically an infinite number of turning points, the required turning point must be specified.

**Outputs**

The algorithm produces the following outputs:

$t_p$ , the time at which the $n^{th}$ turning point occurs (s).

$y_p$ , the magnitude of the $n^{th}$ turning point.

**Pre-conditions**

The turning point required, $n$ , is checked to see that it is an integer greater than or equal to 1.

**Algorithm**

The time of the $n^{th}$ turning point is given by

$$t_p = \frac{n \cdot \pi}{\omega_n \cdot \sqrt{1 - \zeta^2}}$$

The magnitude of the second order model, $y_p$ , can then be calculated by calling the second order model classes' ***calcResponse*** method, see section 12.11.3.1.

### 12.11.3.7set

This method allows the user to set the properties of the second order model class.

**Inputs**

The algorithm requires the following input:

$(property, value)$ , the property-value pair. The *property* part specifies the property that needs to be set to the *value* part of the property-value pair.

**Outputs**

The algorithm generates no outputs.

**Pre-conditions**

The property part of the property-value pair is checked to see that it has one of the following values:

1. *'zeta'*
2. *'w_n'*

If the property part is *'zeta'*, then the value part is checked to see that it is a scalar real value greater than 0.

If the property part is *'w_n'*, then the value part is checked to see that it is a scalar real value greater than zero.

**Algorithm**

The method runs through all the property-value pairs specified and sets the attributes to the values specified.

If the property part is *'zeta'*, then the value part of the property-value pair is used to set the damping ratio attribute, $\zeta$ , is set to the value part.

If the property part is *'w_n'*, then the value part of the property-value pair is used to set the natural frequency attribute, $\omega_n$ , is set to the value part.

## 12.11.3.8get

This method allows the user to retrieve the values of the attributes of the second order model class.

**Inputs**

The algorithm requires the following input:

$property$ , the property for which the value is required.

**Outputs**

The algorithm produces the following outputs:

$value$ , the value of the property specified.

**Pre-conditions**

The property part of the property-value pair is checked to see that it has one of the following values:

1. *'zeta'*
2. *'w_n'*

**Algorithm**

If the property part of the property-value pair is *'zeta'* the value of the damping ratio attribute, $\zeta$ , is returned.

If the property part of the property-value pair is *'w_n'* the value of the natural frequency attribute, $\omega_n$ , is returned.

## *12.12 Utility functions*

### 12.12.1 Fit a line to two sets of cartesian co-ordinates

This function calculates the coefficients of the equation of line which passes through a pair of points $(x_1, y_1)$ and $(x_2, y_2)$ , given in cartesian co-ordinates.

The name of the function generated to implement this functionality is ***fit_line*** and is stored in an m-file of the same name.

**Inputs**

The algorithm requires the following input:

$(x_1, y_1)$ , the first point on the line.

$(x_2, y_2)$ , the second point on the line.

**Outputs**

The algorithm produces the following outputs:

$[m, c]$ , the coefficients of the line given as a vector which can be used to calculated any point on the line using Matlabs' ***polyval*** function. The vector contains the slope, $m$ , and offset, $c$ , of the line.

**Pre-conditions**

As the co-ordinates may have any value, they need not be checked. Matlab is also capable of dealing with dealing with infinite solutions and solutions which give not a number, *NaN*.

**Algorithm**

The slope of the line is given by

$$m=\frac{y_1-y_2}{x_1-x_2}$$

The offset of the line is given by

$$c=\frac{x_1 \cdot y_2-x_2 \cdot y_1}{x_1-x_2}$$

The vector of coefficients is then simply, $[m, c]$ .

### 12.12.2 Calculate a point on a line using polar co-ordinates

This function fits a line to the a pair of points $(x_1, y_1)$ and $(x_2, y_2)$ , in cartesian co-ordinates, and calculates the range (distance from the origin) of a point on the fitted line at the angle specified in polar co-ordinates.

The name of the function generated to implement this functionality is ***calcPolarLine*** and is stored in an m-file of the same name.

**Inputs**

The algorithm requires the following input:

$(x_1, y_1)$ , the first point on the line.

$(x_2, y_2)$ , the second point on the line.

$\theta$ , the angle at which the range is required for a point on the line (deg).

**Outputs**

The algorithm produces the following outputs:

$r$ , the range (distance from the origin) of a point on the fitted line at the angle specified.

**Pre-conditions**

The points specified are checked to see that the do not define a line passing through the origin. If the line passes through the origin the following equation is true,

$$x_1 \cdot y_2 - x_2 \cdot y_1 = 0$$

If this is the case not a number, *NaN*, is returned.

**Algorithm**

The range of the point on the line is given by,

$$r = \frac{x_1 \cdot y_2 - x_2 \cdot y_1}{(x_1 - x_2) \cdot \sin(\theta) - (y_1 - y_2) \cdot \cos(\theta)}$$

The above equation is derived in section 11.1.

### 12.12.3 Calculating the frequency response of a rational transfer function

This function calculates the magnitude and phase of a rational transfer function at frequencies specified. Although Matlab has functions to do this, they give an incorrect result for the phase if the transfer function contains non-minimum phase zeros or unstable poles, when specifying a set of frequencies.

The name of the function generated to implement this functionality is ***safe_freqresp*** and is stored in an m-file of the same name.

**Inputs**

The algorithm requires the following input:

$H(j\omega)$ , the rational transfer function.

$\omega$ , the frequencies at which the magnitudes and phases of the transfer function are required (rad/s).

**Outputs**

The algorithm produces the following outputs:

$|H(j\omega)|$ , the magnitudes of the transfer function at the frequencies specified (absolute value).

$\measuredangle H(j\omega)$ , the phases of the transfer function at the frequencies specified (deg).

**Pre-conditions**

The frequencies specified, $\omega$ , are checked that they are real and greater than zero.

**Algorithm**

The magnitudes and phases of the rational transfer functions are calculated in the following manner:

1. The rational transfer function is broken up into:
   a) its zeros, $\tilde{z}=z_1, z_2, \ldots, z_n$ where $n$ is the number of zeros
   b) poles, $\tilde{p}=p_1, p_2, \ldots, z_m$ where $m$ is the number of poles
   c) and gain, $k$ .

   This is achieved by using Matlab's ***zpkdata*** function.
2. The magnitudes and phases are initialised as follows

$$|H(j\omega)| = 1$$

$$\sphericalangle H(j\omega) = 0^o$$

3. The magnitudes, $|z_1|, |z_2|, \ldots, |z_n|$, and phases, $\sphericalangle z_1, \sphericalangle z_2, \ldots, \sphericalangle z_n$, of the zeros are calculated using the utility function ***rootfreqresp***, see section 12.12.4.
4. The magnitudes, $|p_1|, |p_2|, \ldots, |p_n|$, and phases, $\sphericalangle p_1, \sphericalangle p_2, \ldots, \sphericalangle p_n$, of the poles are calculated using the utility function ***rootfreqresp***, see section 12.12.4.
5. The magnitudes and phases due to all the zeros is given by

$$|\tilde{z}| = \prod_{i=1}^{n} |z_i|$$

$$\sphericalangle \tilde{z} = \sum_{i=1}^{n} \sphericalangle z_i$$

6. The magnitudes and phases due to all the poles is given by

$$|\tilde{p}| = \prod_{i=1}^{m} |p_i|$$

$$\sphericalangle \tilde{p} = \sum_{i=1}^{m} \sphericalangle p_i$$

7. The magnitude of the transfer function is given by

$$|H(j\omega)| = |k| \cdot \frac{|\tilde{z}|}{|\tilde{p}|}$$

8. The phase of the transfer function is given by

$$\sphericalangle H(j\omega) = \begin{cases} \sphericalangle \tilde{z} - \sphericalangle \tilde{p} - 180^o & \mathit{for}\ k < 0 \\ \sphericalangle \tilde{z} - \sphericalangle \tilde{p} & \mathit{otherwise} \end{cases}$$

## 12.12.4 Calculating the frequency response of a root

This function calculates the magnitude and phase of a root at a frequency specified.

The name of the function generated to implement this functionality is ***rootfreqresp*** and is stored in an m-file of the same name.

**Inputs**

The algorithm requires the following input:

$r$ , the corner frequency of the root (rad/s).

$\omega$ , the frequencies at which the magnitudes and phases of the root are required (rad/s).

**Outputs**

The algorithm produces the following outputs:

$|R(j\omega)|$ , the magnitudes of the root at the frequencies specified (absolute value).

$\measuredangle R(j\omega)$ , the phases of the root at the frequencies specified (deg).

**Pre-conditions**

The frequencies specified, $\omega$ , are checked that they are real and greater than zero.

**Algorithm**

The magnitude of the root is given by

$$|R(j\omega)|=\sqrt{\omega^2+r^2}$$

The phase of the root is given by

$$\measuredangle R(j\omega)=\tan^{-1}\left(\frac{\omega}{r}\right)$$

The phase must be calculated in the range, $0^o \leqslant \measuredangle R(j\omega) < 360^o$ .

# 13 References

Franklin, G.F. , Powell, J.D. and Emami-Naeini, A. (1986) Feedback control of dynamic systems, Addison-Wesley publishing company, Reading Massachusetts, USA.

James, J.R., Frederick, D.K., and Taylor, J.H. (1987) Use of expert-systems programming techniques for the design of lead lag compensators, IEE Proceedings, Vol. 134, Pt. D, No. 3, May 1987.

Horowitz, I.M. (1993) Quantitative feedback theory (QFT), Volume 1, QFT Publications, Boulder Colorado, USA.

Stevens, B.L. and Lewis, F.L. (2003) Aircraft control and simulation, Second edition, John Wiley and Sons, Inc., Hoboken, New Jersey, USA.

Lurie, B.J. and Enright, P.J. (2000) Classical feedback control with Matlab, Marcel Dekker Inc., New York, USA.

Houpis, C.H. and Rasmussen, S.J. (1999) Quantitative feedback theory: Fundamentals and applications, Marcel Dekker Inc., New York, USA.

Houpis, C.H. and D'Azzo, J.J. (1966) Feedback control system analysis and synthesis, Second edition, McGraw-Hill Kogakusha, Tokyo, Japan.

Toivonen, H.T. (1998) Advanced control methods, Department of engineering, Abo Akademi University.

Doyle, J., Francis, B. and Tannenbaum, A. (1990) Feedback control theory, Macmillan Publishing Co.

Stanley, W.D., Dougherty, G.R. and Dougherty, R. (1984) Digital signal processing, Second edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, USA.

Pritchard, C.J. (1995) Determining signal levels in robustly controlled plants using frequency domain closed loop system specifications, PhD thesis, University of the Witwatersrand, Johannesburg, South Africa, pp. 138-156.

Todd, D.J (1986) Fundamentals of robot technology, Kogan Page Ltd, London, UK, pp. 102-103.

# 14 Bibliography

Bronson, R. and Costa, G. (2006) Differential equations, Shaum's outlines, Third edition, McGraw-Hill Companies, Inc., New York, USA.

Distefano, J.J., and Stubberud, A.R. and Williams, I.J. (1990) Feedback and control systems, Shaum's outlines, Second edition, McGraw-Hill Companies, Inc., New York, USA.

Hanselman, D. and Littlefield (1997) The student edition of Matlab, Prentice-Hall, Upper Saddle River, New Jersey, USA.

Houtstetter, G.H., Savant, Jr, C.J. and Stefani, R.T. (1989) Design of feedback control system, Second edition, Holt, Rinehart and Winston, Inc., New York, USA.

Kreyzig, E. (1983) Advanced engineering mathematics, Fifth edition,John Wiley and Sons, New York, USA.

Spiegel, M.R. and Liu, J. (2001) Mathematical handbook of formulas and tables, Shaum's easy outlines, McGraw-Hill, New York, USA.

Stroud, K.A. (1987) Engineering mathematics, Third edition, MacMillan Education Ltd., Houndmills, UK.