

The copyright of this thesis vests in the author. No quotation from it or information derived from it is to be published without full acknowledgement of the source. The thesis is to be used for private study or non-commercial research purposes only.

Published by the University of Cape Town (UCT) in terms of the non-exclusive license granted to UCT by the author.

Optimal Controller Comparison using Pareto Fronts

Thesis presented for the degree of Master of Science
in the Department of Electrical Engineering
UNIVERSITY OF CAPE TOWN
South Africa

by

David Moore

August, 2010



I know the meaning of plagiarism and declare that all the work in this document, save for that which is properly acknowledged, is my own.

David Moore

2010

TABLE OF CONTENTS

1	Introduction	1
2	Methods	7
2.1	Sliding Mode Control	7
2.1.1	Integrated Sliding Mode Control	9
2.1.2	Boundary Layer Sliding Mode Control	10
2.2	Proportional Integral Control	11
2.2.1	Pole-Zero Cancellation	12
2.3	Multi-Objective Optimisation	13
2.3.1	Differential Evolution	15
2.4	Level Diagrams	16
2.5	Performance Measures	21
3	The Control Systems and Objectives	29
3.1	The Simulations	29
3.2	The Plant Models	30
3.2.1	First Order Plant	30
3.2.2	Other Plants	31
3.3	Optimisation Objectives	32
3.3.1	Error Costs	32
3.3.2	Controller Input Costs	34
3.3.3	Robustness Cost	34

3.4	Pareto Differential Evolution Optimiser	36
3.4.1	Optimiser Parameters	36
3.4.2	Controller Parameters	37
4	Results	39
4.1	Region of Interest	39
4.2	Level Diagram Results	41
4.2.1	Use of Different Norms	41
4.2.2	Number of Objectives	44
4.2.3	PI versus ISMC	46
4.2.4	Integrated SMC versus Boundary Layer SMC	59
4.3	Performance Measure Results	66
4.3.1	PI versus ISMC	66
4.3.2	PI versus PI Pole-Zero Cancellation	69
4.3.3	Integrated SMC versus Boundary Layer SMC	71
5	Other Plant Results	73
5.1	Input Saturating Plant	73
5.1.1	PI versus ISMC	73
5.1.2	ISMC versus bSMC	94
5.2	Parasitic Dynamic Plant	102
5.2.1	PI versus ISMC	102
5.2.2	ISMC versus bSMC	114
6	SMC Decision Space Investigation	122

6.1	Parameter Reduction	122
7	Conclusions	126
7.1	General Conclusions	126
7.2	Comparison Conclusions	127
7.3	Further Work	129
	References	131
A	Sliding Mode Control Design Method	134
B	Parallel Pareto Differential Evolution	139
C	Plant Modelling and System Identification	144
D	Performance Measure Calculation	149
D.1	Hypervolume	149
D.1.1	Unary Hypervolume	149
D.1.2	Binary Hypervolume	151
E	PDE Optimiser Parameter Testing	153
E.1	Number of Generations	153
E.2	Population Size	156
F	Performance Measure Relationships	159
F.1	Dominance Relationships	159
F.2	Quality Indicators and Interpretation Functions	163

LIST OF FIGURES

2.1	SMC controller with integrator	10
2.2	2-D Pareto Front	20
2.3	Level Diagrams	20
2.4	cf. [35] Examples of dominance relations on objective vectors.	23
2.5	cf. [22] 2-D example of unary and binary hypervolume measures	26
2.6	cf. [35] 2-D example of the ϵ -indicator, ϵ values indicated for A_1	27
3.1	General Control Setup	30
4.1	ISU_d Level Diagrams using the one, two and infinity norms	40
4.2	ISU_d Level Diagrams using the one, two and infinity norms using maximum norm limiting.	41
4.3	$\ \cdot\ _1$ Cost Level Diagrams for PI, PI Cancel and SMC	42
4.4	$\ \cdot\ _2$ Cost Level Diagrams for PI, PI Cancel and SMC	42
4.5	$\ \cdot\ _\infty$ Cost Level Diagrams for PI, PI Cancel and SMC	43
4.6	U_d Level Diagrams using the one, two and infinity norms	43
4.7	$\ \cdot\ _2$ ISE Cost Level Diagrams for PI and PI Cancel using two costs	44
4.8	$\ \cdot\ _2$ ISE Cost Level Diagrams for PI and PI Cancel using four costs	45
4.9	$\ \cdot\ _2$ EST Cost Level Diagrams for PI and PI Cancel using two costs	45
4.10	$\ \cdot\ _2$ EST Cost Level Diagrams for PI and PI Cancel using four costs	46
4.11	$\ \cdot\ _2$ ISE Cost Level Diagrams for PI, PI Cancel and ISMC	47
4.12	$\ \cdot\ _2$ ISE Input Level Diagrams for PI, PI Cancel and ISMC	48
4.13	$RobustnessDelta$ for PI, PI Cancel and ISMC	48

4.14	ISE_d, ISU_d, ISE_r and ISU_r for PI, PI Cancel and ISMC	49
4.15	ISE_v and ISU_v for PI, PI Cancel and ISMC	50
4.16	λ, ϕ and K for ISMC	51
4.17	K_p and K_i for PI	52
4.18	K_p and K_i for PI cancellation set	52
4.19	Lowest norm step tests for PI and ISMC for ISE objectives.	53
4.20	$\ \cdot\ _2$ EST Cost Level Diagrams for PI, PI Cancel and ISMC	54
4.21	$\ \cdot\ _2$ EST Input Level Diagrams for PI, PI Cancel and ISMC	54
4.22	$RobustnessDelta$ for PI, PI Cancel and ISMC	55
4.23	EST_d, ISU_d, EST_r and ISU_r for PI, PI Cancel and ISMC	56
4.24	E_v and U_v for PI, PI Cancel and ISMC	56
4.25	K_p and K_i for PI	57
4.26	K_p and K_i for PI Cancel	57
4.27	λ, ϕ and K for ISMC	58
4.28	Lowest norm step tests for PI and ISMC for EST objectives.	58
4.29	$\ \cdot\ _2$ ISE Cost Level Diagrams for ISMC and bSMC	60
4.30	$\ \cdot\ _2$ ISE Input Level Diagrams for ISMC and bSMC	60
4.31	ISE_v and ISU_v for ISMC and bSMC	61
4.32	ISE_r, ISE_d, ISE_r and ISU_d for ISMC and bSMC	62
4.33	$RobustnessDelta$ for ISMC and bSMC	62
4.34	λ, ϕ and K for ISMC	63
4.35	λ, ϕ, BL and K for bSMC	64
4.36	Lowest norm step tests for ISMC and bSMC for ISE objectives.	65

5.1	$\ \cdot\ _2$ <i>ISE</i> Cost Level Diagrams for PI, PI Cancel and ISMC	74
5.2	$\ \cdot\ _2$ <i>ISE</i> Input Level Diagrams for PI, PI Cancel and ISMC	75
5.3	$\ \cdot\ _2$ <i>ISE</i> Cost Level Diagrams for PI, PI Cancel and ISMC	76
5.4	$\ \cdot\ _2$ <i>ISE</i> Input Level Diagrams for PI, PI Cancel and ISMC	76
5.5	<i>RobustnessDelta</i> for PI, PI Cancel and ISMC	77
5.6	ISE_v and ISU_v for PI, PI Cancel and ISMC	78
5.7	ISE_d , ISU_d , ISE_r and ISU_r for PI, PI Cancel and ISMC	78
5.8	Separate Inputs: PI	79
5.9	Separate Inputs: PI cancel	79
5.10	Separate Inputs: ISMC	80
5.11	Lowest norm step tests for PI and ISMC for <i>ISE</i> objectives.	81
5.12	$\ \cdot\ _2$ <i>EST</i> Cost Level Diagrams for PI, PI Cancel and ISMC	82
5.13	$\ \cdot\ _2$ <i>EST</i> Input Level Diagrams for PI, PI Cancel and ISMC	83
5.14	EST_v and ISU_v for PI, PI Cancel and ISMC	83
5.15	<i>RobustnessDelta</i> for PI, PI Cancel and ISMC	84
5.16	EST_d , ISU_d , EST_r and ISU_r for PI, PI Cancel and ISMC	84
5.17	K_p and K_i for PI	85
5.18	K_p and K_i for PI cancellation set.	85
5.19	λ , ϕ and K for ISMC	86
5.20	Lowest norm step tests for PI and ISMC for <i>EST</i> objectives.	87
5.21	$\ \cdot\ _2$ <i>ISE</i> Cost Level Diagrams for ISMC and bSMC	94
5.22	$\ \cdot\ _2$ <i>ISE</i> Input Level Diagrams for ISMC and bSMC	95
5.23	ISE_v and ISU_v for ISMC and bSMC	96

5.24	<i>RobustnessDelta</i> for ISMC and bSMC	96
5.25	ISE_d , ISU_d , ISE_r and ISU_r for ISMC and bSMC	97
5.26	λ , ϕ , BL and K for bSMC	98
5.27	λ , ϕ and K for ISMC	99
5.28	Lowest norm step tests for ISMC and bSMC for <i>ISE</i> objectives.	100
5.29	$\ \cdot\ _2$ <i>ISE</i> Cost Level Diagrams for PI, PI Cancel and ISMC	103
5.30	$\ \cdot\ _2$ <i>ISE</i> Input Level Diagrams for PI, PI Cancel and ISMC	104
5.31	ISE_v and ISU_v for PI, PI Cancel and ISMC	104
5.32	<i>RobustnessDelta</i> for PI, PI Cancel and ISMC	105
5.33	ISE_d , ISU_d , ISE_r and ISU_r for PI, PI Cancel and ISMC	105
5.34	K_p and K_i for PI cancellation set.	106
5.35	K_p and K_i for PI	106
5.36	λ , ϕ and K for ISMC	107
5.37	Lowest norm step tests for PI and ISMC for <i>ISE</i> objectives.	108
5.38	$\ \cdot\ _2$ <i>EST</i> Cost Level Diagrams for PI, PI Cancel and ISMC	109
5.39	$\ \cdot\ _2$ <i>EST</i> Input Level Diagrams for PI, PI Cancel and ISMC	110
5.40	Lowest norm step tests for PI and ISMC for <i>EST</i> objectives.	111
5.41	$\ \cdot\ _2$ <i>ISE</i> Cost Level Diagrams for ISMC and bSMC	115
5.42	$\ \cdot\ _2$ <i>ISE</i> Input Level Diagrams for ISMC and bSMC	115
5.43	ISE_v and ISU_v for ISMC and bSMC	116
5.44	<i>RobustnessDelta</i> for ISMC and bSMC	116
5.45	ISE_d , ISU_d , ISE_r and ISU_r for ISMC and bSMC	117
5.46	λ , ϕ , BL and K for bSMC	118

5.47	λ , ϕ and K for ISMC	118
5.48	Lowest norm step tests for ISMC and bSMC for <i>ISE</i> objectives.	119
6.1	$\ \cdot\ _2$ <i>ISE</i> Cost Level Diagrams for bSMC, fbSMC and fKbSMC	123
6.2	$\ \cdot\ _2$ <i>ISE</i> Input Level Diagrams for bSMC, fbSMC and fKbSMC	123
C.1	Graph of Measured $y[V]$ and Simulated DC motor y_{mt} for a step input . .	145
C.2	General Control Setup	146
D.1	Hypervolume Example	150
E.1	Cost Level Diagrams of the Comparison of the Number of Generations. . .	154
E.2	Input Level Diagrams of the Comparison of the Number of Generations. . .	155
E.3	Cost Level Diagrams of the Comparison of Population Sizes. Population = 200 (light grey diamond), 300 (dark grey cross), 1000 (black dot).	157
F.1	Example of $A \succ \succ B$	160
F.2	Example of $A \succ B$	161
F.3	Example of $A \triangleright B$	161
F.4	Example of $A \parallel B$	162
F.5	Example of $A \parallel B$	162
F.6	$I_{H2}(A, B)$ Example: $A \succ \succ B$	163
F.7	$I_{H2}(A, B)$ Example: $A \succ B$	164
F.8	$I_{H2}(A, B)$ Example: $A \triangleright B$	164
F.9	$I_{H2}(A, B)$ Example: $A \parallel B$	165
F.10	$I_{H2}(A, B)$ Example: $A \parallel B$	166

LIST OF TABLES

2.1	cf.[35] Relations on Objective Vectors and Approximation Sets.	23
2.2	cf.[35] Indicator Interpretation Function Relations	28
4.1	Performance Measure Results for PI versus ISMC	67
4.2	Performance Measure Results for PI versus PI Pole-Zero Cancellation . . .	69
4.3	Performance Measure Results for ISMC versus bSMC	71
5.1	Performance Measures: PI vs ISMC	89
5.2	Performance Measures: PI vs PI Cancel	92
5.3	Performance Measures: ISMC vs bSMC	101
5.4	Performance Measures: PI vs ISMC	112
5.5	Performance Measures: PI vs PI Cancel	113
5.6	Performance Measures: ISMC vs bSMC	120
D.1	Empirical testing of I_H	151
E.1	Performance Measure Results for 30, 20 and 10 Generations	155
E.2	Performance Measure Results for Population Size 1000, 300 and 200	157
F.1	cf.[35] Relations on Objective Vectors and Approximation Sets.	159

ACKNOWLEDGMENTS

Special thanks to Prof. Martin Braae for his invaluable assistance and the many hours of discussion we had. Also thanks to Jaston Sikaundi and Chris Cecchini for keeping me sane through my Masters.

University of Cape Town

ABSTRACT OF THE THESIS

Optimal Controller Comparison using Pareto Fronts

by

David Moore

Master of Science in Electrical Engineering

University of Cape Town, South Africa, 2010

New design methods in Control Systems are regularly proposed. These new methods are typically compared to existing methods in a focused manner that highlights certain criteria, but may neglect other criteria. This thesis investigates the usefulness of Level Diagram visualisation and Performance Measures for Pareto fronts as tools for comparing modern control schemes in a more robust and objective manner.

CHAPTER 1

Introduction

New methods are constantly being proposed for the design of Control Systems (e.g. Variable Structure Control (VSC) in 1970s, H-infinity 1980s, Model Predictive Control (MPC) in 1990s, etc.). When a new method is proposed, current practice is to present the mathematical theory of the method, then compare the new method to its predecessor, alternative or competing method to determine its competitiveness.

Some examples of this include [18] where a new variable structure proportional-integral-derivative (PID) controller tested with both normal and boundary layer chatter reduction is compared to PID and another paper's formulation of a variable structure PID controller for robot motion control. The presented comparison is done by two means, first by assessing the advantages of each of the proposed methods, and then by analysis of the control performance through a time based simulation of the sinusoidal tracking of the joint positions. The comparison is done by inspection of the various response graphs; no quantitative analysis is done.

Reference [25] applies another variable structure controller using an integral sliding surface to the adaptive field-oriented control of an induction motor. The method is compared to PI; this is done by comparison of various step speed responses for a number of test situations.

Similarly, in [19] a modified proportional-integral sliding mode controller is proposed for direct current (DC) motor speed control. It is compared to proportional-integral (PI), PID and an integral sliding mode controller. The comparison is achieved through direct

comparison of a step response plot and through analysis of the advantages and disadvantages of each method; again there is no quantitative analysis. Further, several methods are presented for reducing chatter in the sliding mode controllers, and are included in the comparisons. Again, there is no quantitative analysis of the various controllers' performances.

Reference [15] compares a new sliding mode model predictive controller to model predictive control. It specifically looks at each controller's suitability for controlling non-minimum phase systems. The example used is a non-linear, non-minimum phase process. The comparison of the methods is done by means of step response plots; different plots looking at the response for setpoint changes, model changes and for disturbances in the process.

In some cases, benchmark problems are used as a basis for comparison of control design methods. Reference [16] is a good example of this; it reviews the results of a series of papers where the same benchmark problem was used by various groups to design controllers for the benchmark problems. Each group then presented a paper on their design and findings. In each case the primary method of comparison was direct comparison of position or velocity error step responses to setpoint changes.

One of the shortfalls of most of these design methods comparisons is that they compare a single controller against a single controller, without the explicit verification of the quality of the specific designs used. Reference [11] has shown that optimisation is a viable means for off-line design of controllers, finding optimal controller designs. This could be used in order to address this quality concern.

In [24] synthesis techniques for second-order plus dead-time problems were investigated using Pareto fronts¹. It was found that the comparisons used in control engineering

¹A Pareto efficient or optimal situation is one where any change to make an objective better off is impossible without making some other objective worse off [20]. The set of Pareto-optimal solutions is referred to as the Pareto front [8].

frequently highlight a limited set of criteria and neglect other important considerations. This can result in comparisons that are often incomplete and practically infeasible.

The use of Multi-Objective Optimisation (MOO) and Pareto fronts for control system design is discussed in [12], where the high dimensional nature of control designs is stressed as an issue in terms of the usability of these methods.

The use of MOO in control engineering design is highlighted in [14], where the application of MOO to the design of PID and MPC controllers is investigated. The author finds the use of MOO in parameter optimisation of controllers to be very attractive, for both on-line and off-line applications. Specifically, it is noted that "decision making for control applications is also a field that needs more ideas and considerable more research."

Traditional approaches such as Linear Quadratic Regulator can fail to fully explore the Pareto front [12], due to the fact that many of these approaches are incapable of generating solutions in non-convex regions of the Pareto front. As such the use of Evolutionary Multi-Objective Optimisation methods is suggested as a means to find the relevant trade-off front for the engineering designer. One of the problems noted with this application of Multi-Objective Optimisation is the high dimensional nature inherent in control design problems. This high dimensionality makes analysis of the resulting Pareto fronts difficult. One of the possible measures suggested in [12] to address this issue is the use of high dimensional visualisation techniques.

Reference [5] presents a novel method to deal with the visualisation of high dimensional Pareto fronts called Level Diagrams. In this thesis Level Diagrams are investigated as a possible means for visualisation of these Pareto fronts for use in controller comparisons.

In [12] the "region of interest" (ROI) concept is presented, it is the sub-region of objective or decision space that is of interest to the designer. For example, if the designer needs a system that settles within 3 seconds, it is unnecessary and often beneficial to exclude values outside the ROI from any analyses. The ROI concepts are investigated

and applied in this thesis as a means to reasonably reduce the amount of data that needs to be analysed.

Another shortfall of the control design methods comparisons discussed above was the lack of quantitative analysis of the results: Each of these comparisons uses the visual comparison of response data as the primary means of comparison.

During investigation of the use of Level Diagrams, and the use of Pareto fronts for control comparisons, it became further apparent that it would be useful to quantify these comparisons. In the case of Level Diagrams, this would mean quantifying some of the characteristics of the Pareto fronts for comparison.

Another field where the comparison of Pareto fronts is used is the field of Multi-Objective Optimisation (where the Pareto fronts are typically generated). The interest here is in assessing the quality of approximations of the Pareto fronts resulting from different algorithms for the same problem [35] as a means to evaluate those algorithms. These comparisons are typically done quantitatively by means of Performance Measures.

Performance Measures² are a field of quantitative methods used for comparing Pareto fronts (and approximate Pareto fronts) [35]. The use of Performance Measures has become common practice in the field of Evolutionary Multi-Objective Optimisation, as a means to evaluate new optimisation algorithms. An example of this is [17], where Performance Measures are applied to analyse the performance of the Pareto Archived Evolution Strategy. Various statistical measures (read: Performance Measures) were applied and used to compare the Pareto fronts resulting from a number of different Multi-Objective Optimisation methods as a means to compare the methods.

Reference [35] presents a theoretical and mathematically robust review of the various

²Performance Measures is the name used in this thesis for the field of quantitative methods used in comparison of Pareto fronts. The field is known by many other names such as: Performance Indices, Performance Assessment Metrics, Quality Indicators, etc. This field is still very active and no fixed nomenclature has been adopted. Performance Measures should not be confused with "optimisation objectives" which are the metrics used as part of Multi-Objective Optimisation methods.

Performance Measures applied in recent years. The paper analyses their application and makes suggestions as to the applicability of various of the proposed Performance Measures. Further, [13] presents a tutorial in the use of Performance Measures and related functions, it suggests the use of empirical attainment functions (which are not directly applicable to this thesis' problem) and Dominance-compliant quality indicators (read: Performance Measures).

This thesis investigates an alternate use for Performance Measures, namely for the quantitative high dimensional comparison of control systems. Instead of being used to investigate different Multi-Objective Optimisation methods being applied to the same problem, they are applied to the case where the Pareto fronts in question arise from two different control design methods applied to the same problem using the same optimisation algorithm. In this way, any differences in the Pareto-fronts will result from the control design methods.

Thus, Level Diagrams and Performance Measures are investigated in the thesis as two separate, but complementary methods for comparison of control design methods: Level Diagrams as a high dimensional visual comparison technique, and Performance Measures as a quantitative comparison technique.

The thesis resulted in the design and implementation of a simulation framework that can handle various control design methods (including different control structures). It can find, using various objectives, the approximate Pareto front of optimal controllers using Pareto Differential Evolution (a Multi-Objective Optimisation technique) for each control design method. The framework can generate the Level Diagrams and various Performance Measures for use in comparing different control design methods. The application of these techniques (Level Diagrams and Performance Measures) and the resulting data are the primary focus and are discussed in the thesis.

Sliding Mode Control (SMC) is used as a test case control design method for com-

parisons using the Level Diagrams and Performance Measures being investigated. Two SMC controllers are tested; each implementing a different simple chatter reduction technique. As in other engineering comparisons, the two-term Proportional Integral controller is used as a base case against which the Sliding Mode Control designs are compared. In addition, avoidance of proportional integral control pole-zero cancellation in optimisation is investigated.

These control design methods are then applied to three different plants. The first is a simple first order plant to get a baseline for the Level Diagrams and Performance Measures. The second applies an input saturation limit to the first order, which is a common non-linear limitation in industry. The final plant includes a non-dominant parasitic dynamic, which is another common phenomenon in control applications, which can cause closed-loop instability.

The remainder of the thesis is structured as follows: Chapter 2 covers the various methods and techniques used in the thesis. Chapter 3 looks at some of the specifics of the control set up, simulator, optimiser and parameter selection. Chapters 4 and 5 include the results of various tests of Level Diagrams and Performance Measures. Chapter 6 presents a brief investigation into over-parametrisation. Finally, Chapter 7 draws conclusions on the work in the thesis and makes suggestions for future work.

CHAPTER 2

Methods

The thesis makes use of many methods, some that are common in control engineering, such as Proportional Integral Control, and others that are not commonly encountered in control engineering. The following topics are discussed:

- Sliding Mode Control
- Proportional Integral Control
- Multi-Objective Optimisation
- Level Diagrams
- Performance Measures

2.1 Sliding Mode Control

Sliding Mode Control (SMC) is a non-linear control design technique. It is a form of Variable Structure Control in that it alters the structure of the control law, typically through non-linear switching, to achieve both stability and desirable dynamics [27].

SMC is of interest because of its purported robustness to parameter variations and invariance to a class of bounded disturbances [31][10]. Also, another advantage of SMC's non-linear nature is that its Sliding Mode is reached in finite instead of asymptotic time [10].

There are a number of negatives that have prevented greater adoption of SMC. It is inherently mathematically complicated and thus difficult to implement. There are many different approaches to implementing SMC further complicating the choice of what is the best method to use. Beyond the implementation difficulties, one of the largest complaints against SMC is the introduction of chattering due to non-ideal switching, which can lead to actuator damage, energy loss and excitation of unmodelled dynamics. Various methods have been proposed to deal with the phenomenon [31], with varying success.

A simple SMC control law is used for this thesis. Design of the control law follows the method that is described in [27] with a modification from [10] to force exponential instead of asymptotic Lyapunov stability.

Given a system of the form:

$$x^{(n)} = f(\mathbf{x}) + b(\mathbf{x})u \quad (2.1)$$

where x is the output, u is the control input and $\mathbf{x} = [x, \dot{x}, \dots, x^{(n-1)}]^T$ is the state vector. The control problem is to get the state \mathbf{x} to track a desired state trajectory (time varying signal) $\mathbf{x}_d = [x_d, \dot{x}_d, \dots, x_d^{(n-1)}]^T$.

A sliding surface is then defined using (2.2), which depends on the degree of the system n and the states \mathbf{x} . λ is a positive design constant that sets the sliding rate, and $\tilde{x} = x - x_d$ is the tracking error for variable x .

$$s(\mathbf{x}, t) = \left(\frac{d}{dt} + \lambda \right)^{n-1} \tilde{x} \quad (2.2)$$

By applying Lyapunov stability theory, the state tracking problem is replaced by a first order stability problem. That is, the control aim is to keep state vector \mathbf{x} on the trajectory $s(\mathbf{x}, t)$. The trajectory is then defined in such a manner that it settles to the desired operating point.

The general control action used in this thesis is computed from 2.3 as explained in Appendix A, where a detailed explanation of SMC for SISO (single input single output) systems is given.

$$u = \hat{b}^{-1} [\hat{u} - k \operatorname{sgn}(s) - \Phi s] \quad (2.3)$$

where \hat{b} is the estimate of the control gain; \hat{u} is the equivalent control; k is the switching gain, s the sliding surface, and Φ the reaching rate.

The allowable values of λ and Φ are limited by various practical considerations [27], primarily relating to the speed. The main consideration that affects these simulations is the sampling speed t_s , which effectively sets an upper limit on the value of λ and Φ . Beyond this limit, the control action is faster than the sampling and can enter a limit cycle or become unstable. Consideration needs to be taken when selecting these parameters.

As chatter is considered unacceptable, two different simple implementations of chatter reduction techniques are investigated as part of the comparisons in this thesis: Integrated Sliding Mode Control (ISMC) and Boundary Layer Sliding Mode Control (bSMC).

2.1.1 Integrated Sliding Mode Control

One of the methods used to deal with chatter is a pragmatic engineering approach, called Integrated SMC. The method works by augmenting the SMC controller with an integrator on its output as shown in Figure 2.1.

The output of this integrator is the manipulated variable u for the plant. There are three main reasons for the integrator's inclusion: Firstly, SMC design techniques call for systems with greater than first order dynamics (otherwise the resulting control is the well known bang-bang controller, resulting from Equation 2.2 with $n = 1$). Secondly, the integrator acts as a filter, removing the need for any other chatter correction. And finally, it eliminates the finite offset error that would occur with first order plant models due to

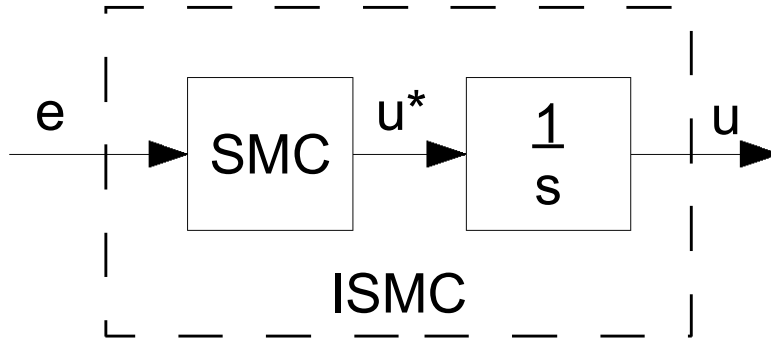


Figure 2.1: SMC controller with integrator

input disturbances.

One expected result of this modification is that integrating the SMC output will reduce the controller's effectiveness at rejecting disturbances due to the delayed response that the integrator introduces.

2.1.2 Boundary Layer Sliding Mode Control

Boundary Layer, or *Pseudo* Sliding Mode Control is one of the more commonly used means of chatter reduction [31][27][10]. This technique works by replacing the switching function $\text{sgn}(s)$ with a smooth function in a thin boundary layer neighbouring the sliding surface [27].

There are many different smoothing functions defined in literature, examples being the saturation function [27] and the signum function [10]. For this thesis the saturation function is used to smooth with a boundary layer of thickness BL .

$$\text{sat}(x) = \begin{cases} 1 & \text{if } x > 1 \\ -1 & \text{if } x < -1 \\ x & \text{otherwise} \end{cases} \quad (2.4)$$

The modified control law using Boundary Layer SMC is:

$$u = \hat{b}^{-1} \left[\hat{u} - k \operatorname{sat} \left(\frac{s}{BL} \right) - \Phi s \right] \quad (2.5)$$

where BL now defines the width of the boundary layer.

This formulation of Boundary Layer SMC successfully reduces chattering in most cases, but can have issues in practical implementation. Due to the linearisation within the boundary layer, it is possible with parasitic sensor dynamics for the controller to enter a limit cycle should the resulting linear control be unstable [31]. This can cause the system to still chatter, even using Boundary Layer SMC. Care must be taken during design to ensure the effective linear control within the Boundary Layer is stable, otherwise limit cycling will occur.

Another issue with the Boundary Layer SMC is that the resulting control is of type zero, the implication being that it cannot track step setpoints with zero error. There will always be a finite (if small) offset error.

2.2 Proportional Integral Control

The two-term PI controller is an industry standard control method developed in the 1940s. It is very popular in industry due to its good performance coupled with its relative simplicity.

It has been shown in [16] that PI is extremely powerful, being competitive when compared to leading robust and adaptive control design methods for time-invariant systems.

Accordingly, PI is often used as a traditional approach to provide base cases for comparison. Further examples of this can be seen in [25],[18] and [15] where PI is compared to various new controller design methods.

As such, PI is used in this thesis in some of the comparisons made later as a base case,

against which other techniques are compared.

A complete analysis and development of design techniques is given in [28]. The format shown below is used in the thesis for programming simulation.

$$K(s) = K_p + \frac{K_i}{s} \quad (2.6)$$

Apart from its infinite gain at zero frequency, PI Control has no implicit definition for robustness. This is usually carefully considered and special care is taken during the design process.

Optimisation has been applied to PI controller design [11] with varied results. One concern with this use of optimisation/tuning techniques applied to Control Engineering is the avoidance of pole-zero cancellation. This is due to the resulting undesirable slow mode behaviour (see Appendix C), and to a lesser extent the overall robustness of the resulting control.

2.2.1 Pole-Zero Cancellation

In applications of PI control to dominantly first order plants the control law may induce poor internal performance [26] in which the open loop poles (usually from the process) are cancelled by zeros (usually from the controller) in the nominal control loop [23]. This cancellation of poles by zeroes is referred to as Pole-Zero cancellation (see Appendix C for a simple analysis of this phenomenon), which results in slow modes in the control loop.

Due to the possibility of poor internal performance, pole-zero cancellation is undesirable in practice and thus should be avoided.

Optimisation frequently does not take account of this effect and depends only on the objectives (costs) being used. Pole-Zero cancellation typically results in very good output-disturbance behaviour [23], but poor input-disturbance behaviour. Depending on

the optimisation objectives used, this good rejection of output-disturbances often results in controllers that are Pareto-optimal. Common cost functions (also known simply as costs or objectives) used in Control Engineering optimisation, such as the Integral Square Plant Error *ISE* [11] (see section 3.3.1) do not account for this issue of internal performance. As such, optimisers can often optimise onto these values where pole-zero cancellation occurs, unless special care is taken in their design.

This thesis looks at the effect of objective selection and the number of objectives on pole-zero cancellation. It examines whether it is possible to avoid this phenomenon by using different objectives, or more objectives.

2.3 Multi-Objective Optimisation

To ensure that the comparisons are as unbiased as possible, the framework uses the best controllers that each method can produce. The performance of the controllers is quantified by objectives (or cost functions) that are assumed to identify the best controllers, where the best is defined in terms of optimality. These objectives are frequently and typically mutually exclusive, resulting in a trade-off situation among various objectives.

Multi-Objective Optimisation (MOO) methods are utilised to find optimal (best) solutions to a given problem [20] where *optimal* is defined in terms of the trade-offs between computed objectives and the concept of Pareto Efficiency [20].

A Pareto Efficient or Optimal situation is one where any change to make an objective better off is impossible without making some other objective worse off [20]. Solutions that are Pareto-optimal are described as non-dominant [8][2]. In Multi Objective problems, the set of Pareto-optimal solutions is referred to as the Pareto front [8].

MOO is a large field with many different methods and ideologies [20]. These methods can roughly be classified into four groups that are defined by when the decision maker

adds preference information to the optimisation process, These four groups are priori (before), progressive (during), posteriori (after), and no preference [3].

One popular group of MOO methods is Evolutionary Algorithms (EA). Some examples of where EA's are seeing use include various engineering fields, scheduling and management, and diverse scientific fields [8]. These methods are advantageous over other mathematical programming techniques as they don't require differentiability of the objective functions and the constraints. Further, EA's are less susceptible to the shape or continuity of the Pareto Front [8].

Within EA there are many different competing algorithms using different schemes for elitism, selection of parents, how parents compete with child members, and other concerns such as crowding and population density. Three of the leading recent Multi Objective EA methods are the SPEA2 [34], PAES [17] and NSGA-II [9] methods. These have all seen success in the field of MOO and are often used as benchmark algorithms against which other multi objective EAs are compared to.

Further there are other EA bio-inspired heuristics including Particle Swarm Optimisation and Differential Evolution [29], both of which are becoming increasingly popular in MOO [8].

Multi-Objective Optimisation finds the best costs by simultaneously optimising two or more conflicting objectives subject to certain constraints [20]. It can be formalised mathematically as follows:

$$\begin{aligned}\theta &= [\theta_1, \dots, \theta_i] \in \Omega, \\ \mathbf{J}(\theta) &= [J_1(\theta), \dots, J_k(\theta)], \\ \min_{\theta \in \Omega} \mathbf{J}(\theta)\end{aligned}$$

where θ is the input or decision vector, Ω is the decision space and $J(\theta)$ is the cost or objective vector. This optimisation can then be explained as simultaneously minimising all objectives of $J_i(\theta)$ for every J_i . Multi-Objective Optimisation yields a set of mutually optimal solutions Ω_P . It is unique for a given set of costs.

2.3.1 Differential Evolution

Differential Evolution (DE) was selected for use in this thesis as it has a number of desirable characteristics. Firstly it has shown it is competitive with other leading EA's [2]. It does not use binary encoding like many other EA's, instead dealing with real numbers aiding with ease of implementation. Finally, there was a base of knowledge on DE in the University department.

DE as an EA is a stochastic population based optimiser. It uses a special crossover-mutation operator replacement scheme with deterministic tournament selection between parent and child. DE is an *a posteriori* method, where as little subjective inference as possible is added before the decision maker is presented with information.

There are many different versions of Differential Evolution, the thesis uses a modified version of the Pareto Differential Evolution Algorithm (PDE)[1], which uses Pareto-optimality as its elitist criteria in a multi-objective DE algorithm. The PDE was modified by the author to allow the use of distributed computing in order to allow faster computation and thus larger population sizes. This modification is detailed in Appendix B.

The PDE algorithm finds a population of n members making up a discrete approximation Ω_P^* of the Pareto front Ω_P .

2.4 Level Diagrams

Visualisation is a useful and viable means to help deal with analysis of Pareto fronts in higher dimensional space [12]. As such, it is one of the methods investigated for use in comparison of control design methods in this thesis.

Traditionally Pareto fronts are represented on objective versus objective 2-D plots (e.g. [35] [9] [17]). For Pareto fronts in higher dimension (typically meaning dimension $n > 2$) it can be difficult to read useful information from these plots.

Various alternative visualisation methods have been proposed, such as Scatter Diagrams, which are the normal objective versus objective plots extended to n dimensions, I.E. an $n \times n$ matrix where each objective is represented on a separate column and row. The issue being that the number of plots can become cumbersome with complexity $O(n^2)$. Another method, Parallel Coordinates used in [12], is plotted in two dimensions where each dimension of the original data is translated to an x-coordinate on the plot. They are a very compact representation technique, but become very difficult to analyse with large data populations [5], and can require different objective ordering to fully visualise each trade-off.

Level Diagrams are a recent (in 2008) alternative technique presented in [5] for visualisation of high dimensional Pareto fronts such that useful information can still be visually interpreted from them. Hence, they are investigated as part of this thesis to establish their usefulness to control-engineering design-method comparisons.

Level Diagrams make use of a synchronising norm as the y-axis of each plot (both decision and objective plots). Thus the number of axes needed for Level Diagrams is of complexity $O(n)$. Traditional 2-D plots can be used to look at high dimensional data, but the number of plots increases by order $O((n - 1)!)$.

This norm is used both as a tool for comparison between each plots, but also as a

numerical (and thus visual) mean for evaluation of the visualised Pareto front approximations.

It would be preferable to use a norm that has a direct real world meaning, such as converting to a cost in power or currency, which would then give more useful information to the decision maker.

However, this is frequently very difficult to accomplish. Calculating a norm, such as total power usage, requires a real world application and very detailed specifics of the problem being analysed, as well as information on the resource production and consumption which may not be available at the time of design.

As such, another norm is needed, one that can be applied in situations where such specifics are not available but which still is able to provide useful synchronisation between plots.

In [5], three such norms are presented, the one, two (Euclidean) and infinity norms. Each provides a different perspective and weighting to the synchronising norm.

In terms of the application of the norm as in [5], the various objectives were computed and then offset and scaled to a unit range before each norm was applied. This is based on the assumption that each objective is of equal importance (but of different size, including order of magnitude).

A weighted normalisation could be used if it was suitable to the application.

In the case of the test setups used in this thesis, it was deemed that each of the seven objectives once calculated would be given equal importance.

The normalisation was achieved as described in [5]: Each objective is normalised based on its proximity to an ideal point i , in this case the objective minima for a given Pareto front approximation; and a worst point r , the objective maxima for a given Pareto front approximation. When more than one Pareto front is present, the ideal point is the objective minima of all of the Pareto front approximations. Similarly, the worst point is

the objective maxima of all of the Pareto front approximations.

Each objective is first normalised with respect to its minimum and maximum values, such that each has equal weighting.

$$J_i^M = \max_{\theta \in \Omega_P^*} J_i(\theta),$$

$$J_i^m = \min_{\theta \in \Omega_P^*} J_i(\theta),$$

$$i = 1, \dots, k$$

$$\bar{J}_i = \frac{J_i(\theta) - J_i^m}{J_i^M - J_i^m}$$

$$0 \leq \bar{J}_i(\theta) \leq 1$$

One of the norms is then computed for the normalised objectives to evaluate the distance to the ideal point. The three different norms presented in [5]: The one norm, the Euclidean norm and the infinity norm, defined respectively by:

$$\|\bar{\mathbf{J}}(\theta)\|_1 = \sum_{i=1}^n |\bar{J}_i(\theta)| \quad (2.7)$$

$$\|\bar{\mathbf{J}}(\theta)\|_2 = \sqrt{\sum_{i=1}^n \bar{J}_i(\theta)^2} \quad (2.8)$$

$$\|\bar{\mathbf{J}}(\theta)\|_\infty = \max \{ \bar{J}_i(\theta) \} \quad (2.9)$$

These norms are bounded depending on the norm and the dimension of the objective

vector:

$$0 \leq \|\bar{\mathbf{J}}(\theta)\|_1 \leq n, \quad (2.10)$$

$$0 \leq \|\bar{\mathbf{J}}(\theta)\|_2 \leq \sqrt{n}, \quad (2.11)$$

$$0 \leq \|\bar{\mathbf{J}}(\theta)\|_\infty \leq 1. \quad (2.12)$$

Each norm gives a different view of the Pareto front shape. The Euclidean norm gives the closest representation of the conventional geometric distance, which is the best view of the real shape of the front [5]. The ∞ -norm gives a view of the worst cost. This is good for trade-off analysis. Any worsening in the ∞ -norm can be directly related to a worsening in at least one cost.

In order to plot Level Diagrams, each objective (\mathbf{J}_i) and input (θ_i) is plotted on a separate set of axes. They are plotted on the X-axis against their corresponding synchronising norm $\|\cdot\|_n$ on the Y axis. This synchronises all the plots, effectively placing all corresponding information for \mathbf{J}_i and θ_i at the same position on the Y-axis.

An example of a 2-D Pareto Front is given below in Figure 2.2, in this case showing a discrete approximation Ω_p^* of the true Pareto front Ω_p for two costs.

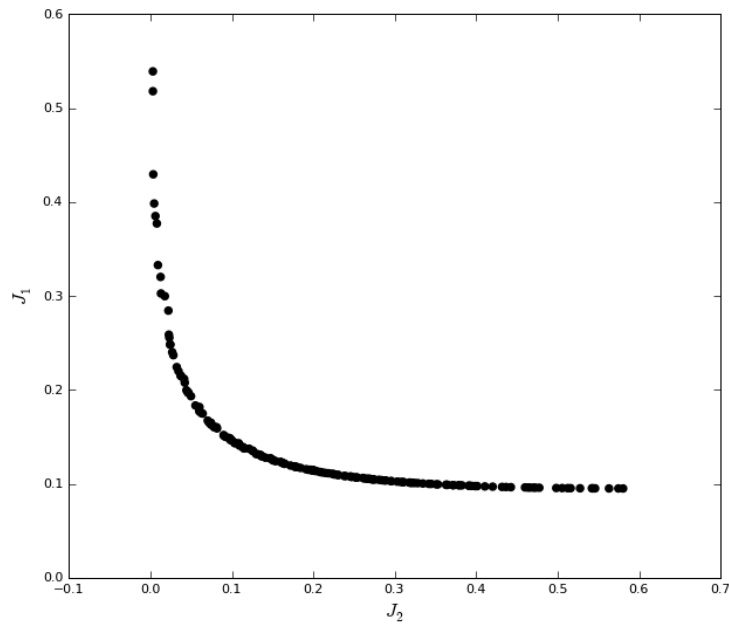


Figure 2.2: 2-D Pareto Front

Using the Euclidean norm and the same data, the corresponding Level Diagrams are shown in Figure 2.3.

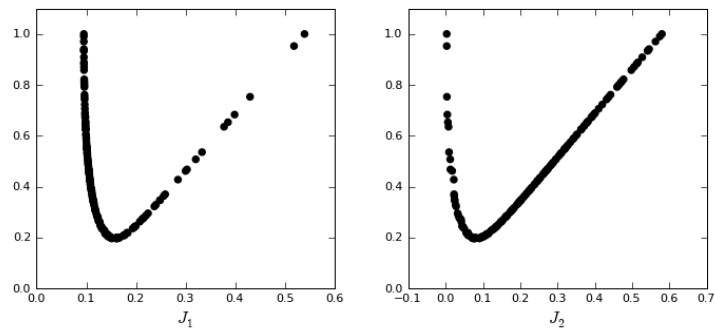


Figure 2.3: Level Diagrams

Common sense makes it important to note that if the norm used is inappropriate, it can highlight the wrong part of the Pareto front, thus making further analysis inaccurate.

The principal advantage of Level Diagrams is their ability to visualise high dimensional

data in a compact format ($O(n)$), while still providing useful visual information. Further, Level Diagrams provide a simple means (through the synchronising norm) to associate inputs (design parameters) with objectives, allowing combined analysis of both objectives and design parameters. Finally, unlike some other visualisation methods, there is only one "view" or perspective of the Level Diagrams and it does not require the use of different orderings (as in Parallel Coordinates) to see different trade-offs.

2.5 Performance Measures

Initial testing of using Pareto fronts for comparison on control design methods revealed that it could be useful to quantify the comparisons of the Pareto fronts.

Quantitative comparison of algorithms is an important issue in Multi-Objective Optimisation [35][8]. Many recent methods assess the quality of the approximations of the Pareto front (which are a typical result from Multi-Objective Optimisation) as a means to evaluate this comparison.

This thesis investigates the usefulness of Performance Measures as a means to quantitatively compare the Pareto fronts resulting from different control design methods (instead of from different optimisation algorithms). It is a separate method from Level Diagrams, but complementary in that it also uses the Pareto fronts for comparison of methods.

Performance Measures combine the use of 3 concepts: quality indicators, interpretation functions and comparison methods in order to compare various Pareto fronts. These are discussed below.

Various measures have been developed to compare different Pareto fronts resulting from different optimisation algorithms. For example [17] makes use of a statistical line intersection method to find what fraction of the combined Pareto front for two methods, belongs to each method. In [35], many of the methods proposed in other research are

reviewed and assessed. While [13] suggests specific methods to use for comparing different fronts. These measures are referred to as quality indicators.

Rigorous assessment and review of existing quality indicators is carried out in [35]. Wherein, a theoretical framework is developed to judge the usefulness, accuracy and inference ability of the various quality indicators proposed in research. The following explanation covers the key concepts relating to the use of quality indicators and follows closely the work in [35].

Performance Measures are intended for comparison of approximate Pareto fronts resulting from different Multi-Objective Optimisation algorithms applied to a given problem. This thesis investigates their usefulness when applied to the case where different control methods are applied to some problem using the same Multi-Objective Optimisation algorithm.

As such, some of the quality indicators have direct real world analogies, while other don't relate to any practical realities. [6] suggests care be taken when using indicators when the Pareto front is not known as their results can be misleading.

Given the solution of an optimisation problem, which is described in terms of a *decision vector* in the *decision space* X . The function $\mathbf{f} : X \rightarrow Z$ evaluates the quality of a solution by assigning it an *objective vector* in the *objective space* Z .

For minimisation problems, an objective vector \mathbf{J}^1 (like those used in Multi-Objective Optimisation) dominates (in a Pareto optimality sense) another objective vector \mathbf{J}^2 if it is not greater than \mathbf{J}^2 in all components and smaller in at least one component. (Note: \mathbf{J}^1 and \mathbf{J}^2 refer to two distinct objective vectors, each with component objectives $\mathbf{J}_1^1, \dots, \mathbf{J}_n^1$).

An objective vector is then Pareto-optimal if it is not dominated by another vector. The set of all Pareto-optimal objective vectors form the Pareto-optimal set or Pareto front in the objective space.

This dominance relationship can be further expanded into a hierarchy of dominance

relationships where $\mathbf{J}^1 \succ\succ \mathbf{J}^2 \Rightarrow \mathbf{J}^1 \succ \mathbf{J}^2 \Rightarrow \mathbf{J}^1 \succeq \mathbf{J}^2$.

These relationships are not commonly used in engineering relationships. In order to help with understanding of what each operator means the relationships are shown visually in Figure 2.4 which is copied from [35].

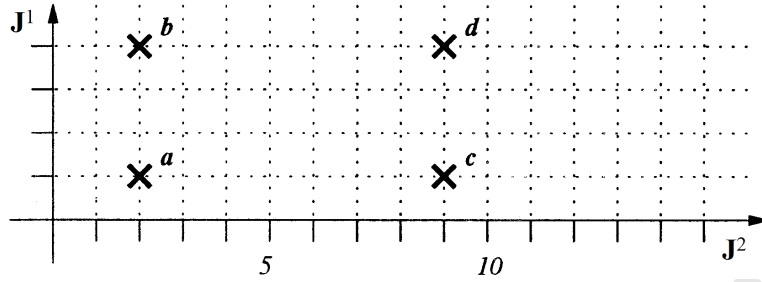


Figure 2.4: cf. [35] Examples of dominance relations on objective vectors.

Assuming that two objectives are to be minimised, it holds that $a \succ b$, $a \succ c$, $a \succ d$, $b \succ d$, $c \succ d$, $a \succ\succ d$, $a \succeq a$, $a \succeq b$, $a \succeq c$, $a \succeq d$, $b \succeq b$, $b \succeq d$, $c \succeq c$, $c \succeq d$, $d \succeq d$, and $b \parallel c$.

These relationships and their interpretations are shown in Table 2.1 which is taken from [35]. They can be naturally extended to dominance comparison of approximation sets A and B, which are also shown in the Table. (see Appendix F for examples of this extension).

relation	objective vectors		approximation sets	
strictly dominates	$\mathbf{J}^1 \succ\succ \mathbf{J}^2$	\mathbf{J}^1 is better than \mathbf{J}^2 in all objectives	$A \succ\succ B$	every $\mathbf{J}^2 \in B$ is strictly dominated by at least one $\mathbf{J}^1 \in A$
dominates	$\mathbf{J}^1 \succ \mathbf{J}^2$	\mathbf{J}^1 is not worse than \mathbf{J}^2 in all objectives and better in at least one objective	$A \succ B$	every $\mathbf{J}^2 \in B$ is dominated by at least one $\mathbf{J}^1 \in A$
better			$A \triangleright B$	every $\mathbf{J}^2 \in B$ is weakly dominated by at least one $\mathbf{J}^1 \in A$ and $A \neq B$
weakly dominates	$\mathbf{J}^1 \succeq \mathbf{J}^2$	\mathbf{J}^1 is not worse than \mathbf{J}^2 in all objectives	$A \succeq B$	every $\mathbf{J}^2 \in B$ is weakly dominated by at least one $\mathbf{J}^1 \in A$
incomparable	$\mathbf{J}^1 \parallel \mathbf{J}^2$	neither \mathbf{J}^1 weakly dominates \mathbf{J}^2 nor \mathbf{J}^2 weakly dominates \mathbf{J}^1	$A \parallel B$	neither A weakly dominates B nor B weakly dominates A

Table 2.1: cf.[35] Relations on Objective Vectors and Approximation Sets.

The relationship and signs are all commutative. Thus, $A \succ B$ is equivalent to $B \prec A$.

For this thesis, we use the terminology of objective vector (cost) $\mathbf{J}(\theta) \in Z$ corresponds to a particular solution (decision vector) $\theta \in X$ in the decision space, that (in the case of this thesis) is typically the parameters of the control law.

Definition 1 (Approximate Set): Let $A \subseteq Z$ be a set of objective vectors. A is called an approximation set if an element of A does not weakly dominate any other objective vector in A . This set is sometimes given the symbol Ω_P^* .

The dominant subset of all approximation sets, i.e. the real Pareto front, is then denoted by Ω_P .

In the case of this thesis, A and B denote the different approximate Pareto fronts for the control design methods being tested.

Quality indicators build on the desire to apply quantitative measures to the comparison of two approximate Pareto-optimal fronts. Instead of saying that one algorithm is better than another, one would be able to say how much better. Or in the case where neither is strictly better than the other, to be able to comment on which aspects of each are better.

The concept is analogous to concepts such as the mean and standard deviation of probability functions. By reducing the various facets of the Pareto fronts to single quantities, information is lost but useful characteristics can still be inferred from the resulting measures.

Various measures have been proposed and assessed in research [35]. In this thesis, three quality indicators are investigated: Two for their direct real world interpretation, and the third at the suggestion of [35] [13].

Definition 2 (Quality Indicator): an m -ary quality indicator I is a function $I : \Omega^m \rightarrow \mathbb{R}$, which assigns each vector (A_1, A_2, \dots, A_m) of m approximation sets a real value $I(A_1, \dots, A_m)$.

The three quality indicators used in this thesis are the unary hypervolume indicator $I_H(A)$, the binary hypervolume indicator $I_{H2}(A, B)$ (which is also referred to as the Coverage difference indicator [32]), and the binary ϵ -indicator $I_\epsilon(A, B)$.

$I_H(A)$, the unary hypervolume indicator, is a measure of the percentage of the hypervolume within a bounded region that the approximation set A dominates. In terms of control engineering, this can be seen as the amount of the objective space that can be achieved or bettered by a given control design method.

The $I_H(A)$ makes use of a bounding region in its calculation (as described in Appendix D). The selection of this bounding region is determined by two points: the ideal point i and the worst point r . These two points have an effect on the value of $I_H(A)$. For this thesis, i and r are the minima and maxima of all objective vectors in the sets being investigated, as used for normalisation with the Level Diagrams.

$I_{H2}(A, B)$, the binary hypervolume indicator [32][35], is the percentage hypervolume of A that is weakly dominated by A but not by B . It can be shown that:

$$I_{H2}(A, B) = I_H(A \cup B) - I_H(B)$$

This can be thought of in an engineering sense as the percentage of the possible control designs that are unique to one method.

Figure 2.5 (copied from [22]), shows a 2-D representation of the I_H and I_{H2} for two approximation sets S_1 and S_2 .

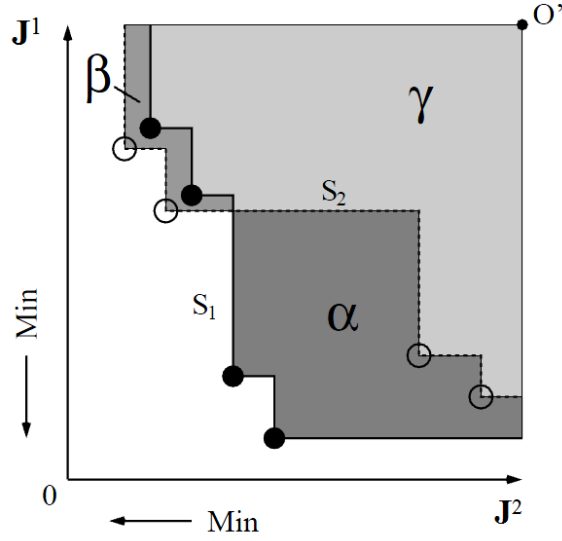


Figure 2.5: cf. [22] 2-D example of unary and binary hypervolume measures

In this case, for the unary hypervolume measures: $I_H(S_1) = \alpha + \gamma$ and $I_H(S_2) = \beta + \gamma$. And for the binary hypervolume measures, $I_{H2}(S_1, S_2) = \alpha$ and $I_{H2}(S_2, S_1) = \beta$.

$I_\epsilon(A, B)$ is the binary multiplicative ϵ -indicator [35]. It can be interpreted as the factor by which each objective value in $\mathbf{J}^2(\theta)$ must be multiplied by to dominate $\mathbf{J}^1(\theta)$. This ϵ -dominance is defined by:

$$\mathbf{J}^1 \succeq_\epsilon \mathbf{J}^2 \text{ iff } \forall 1 \leq i \leq n : \mathbf{J}_i^1 \leq \epsilon \cdot \mathbf{J}_i^2 \quad (2.13)$$

for a given $\epsilon > 0$. Then the $I_\epsilon(A, B)$ is defined as

$$I_\epsilon(A, B) = \inf_{\epsilon \in \mathbb{R}} \{ \forall \mathbf{J}^2 \in B \exists \mathbf{J}^1 \in A : \mathbf{J}^1 \succeq_\epsilon \mathbf{J}^2 \} \quad (2.14)$$

with approximation sets $A, B \in \Omega_P$

Figure 2.6 (copied from [35]), shows a 2-D example of the binary ϵ -indicator. Where the ϵ -indicator is applied to the approximation set A_1 .

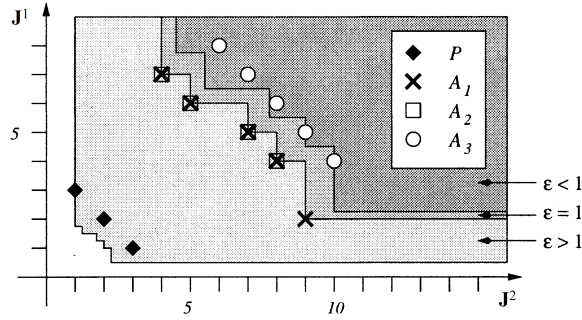


Figure 2.6: cf. [35] 2-D example of the ϵ -indicator, ϵ values indicated for A_1

The amount by which A_1 needs to be scaled by in order to dominate the other sets P , A_2 and A_3 is shown on the right hand side of the figure.

I_ϵ does not have a direct relation with any engineering meaning, but is investigated as it does measure the maximum relative difference in two Pareto fronts.

In order for these quality indicators to be useful, a means for interpreting them is needed. This is achieved with interpretation functions E . These functions relate the quality indicator with some Boolean relationship. The combination of a quality indicator and an interpretation function is called a comparison method [35].

Definition 3 (Comparison Method): Let $A, B \subset \Omega_P^*$ be two approximation sets. $\mathbf{I} = (I_1, I_2, \dots, I_k)$ a combination of quality indicators, and $\mathbf{E} : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \{true, false\}$ an interpretation function which maps two real vectors of length k to a Boolean value. Then $C_{I,E}$ is a comparison method with \mathbf{I} quality indicators and based on \mathbf{E} interpretation functions.

For example, if all indicators are unary, then $C_{I,E}$ defined by \mathbf{I} and \mathbf{E} is of the form:

$$C_{I,E}(A, B) = E(I(A), I(B)) \quad (2.15)$$

Similarly for binary indicators

$$C_{I,E}(A, B) = E(I(A, B), I(B, A)) \quad (2.16)$$

One of the key results of [35] is the linking of comparison methods with dominance relationships described above by means of various interpretation function relationships. These are summarised in the Table 2.2.

indicator	name	relation					
		\succ	\succ	\triangleright	\succeq	$=$	\parallel
I_H	unary hypervolume indicator	-	-	-	-	-	-
I_{H2}	binary hypervolume indicator	-	-	$I_{H2}(A, B) > 0$ $I_{H2}(B, A) = 0$	$I_{H2}(A, B) \geq 0$ $I_{H2}(B, A) = 0$	$I_{H2}(A, B) = 0$ $I_{H2}(B, A) = 0$	$I_{H2}(A, B) > 0$ $I_{H2}(B, A) > 0$
I_ϵ	binary epsilon indicator	$I_\epsilon(A, B) < 1$	-	$I_\epsilon(A, B) \leq 1$ $I_\epsilon(B, A) > 1$	$I_\epsilon(A, B) \leq 1$	$I_\epsilon(A, B) = 1$ $I_\epsilon(B, A) = 1$	$I_\epsilon(A, B) > 1$ $I_\epsilon(B, A) > 1$

Table 2.2: cf.[35] Indicator Interpretation Function Relations

The boolean relations shown in Table 2.2 show which dominance relationships can be inferred by using different interpretation functions. Both I_{H2} and I_ϵ are able to infer various levels of set dominance when comparing different approximation sets.

The unary hypervolume indicator I_H is not able to infer any of these relationships, but is of interest because of two characteristics. Firstly, it has a reasonable real world analogy: The volume or number of possible controllers a method can design in some bounded decision space. Secondly, it is the only known unary indicator that is capable of detecting that A is not worse than B for pairs $A \triangleright B$ [35]. This means that if $I_H(A) > I_H(B)$ (the hypervolume dominated by A is greater than the hypervolume dominated by B) then A is not worse than B in a Pareto sense.

These Performance Measures (including quality indicators, comparison methods and their relating interpretation functions) are used as a means for quantitative analysis of the Pareto front comparisons in the chapters that follow.

CHAPTER 3

The Control Systems and Objectives

This thesis focuses on the speed control of a DC motor using various control methods in order to compare the control methods using Level Diagrams and Performance Measures. These comparisons are based on various analyses of their Pareto fronts. These fronts are in this case approximation sets of the Pareto front Ω_P that are made up of various objectives \mathbf{J} . The PDE algorithm used in this thesis uses various objectives in order to generate the approximate Pareto fronts. These objectives are generated in a purpose coded ODE simulator.

3.1 The Simulations

Many SISO control systems can be approximated by the block diagram in Figure 3.1 which in this case assumes that measurement noise is negligible. $K(s)$ and $G(s)$ are the controller and plant respectively. v is the input disturbance and d the output disturbance that affect the plant, which are assumed to be step disturbances for this thesis. r is the setpoint for the speed control.

For the framework, step signals for r , v and d were used. While somewhat unrealistic, they provide a sound basis for testing the Level Diagram and Performance Measure methods.

The Runge-Kutta 4 algorithm is used to deduce the continuous-time signals u , e and y , used to calculate the various optimisation objectives described below.

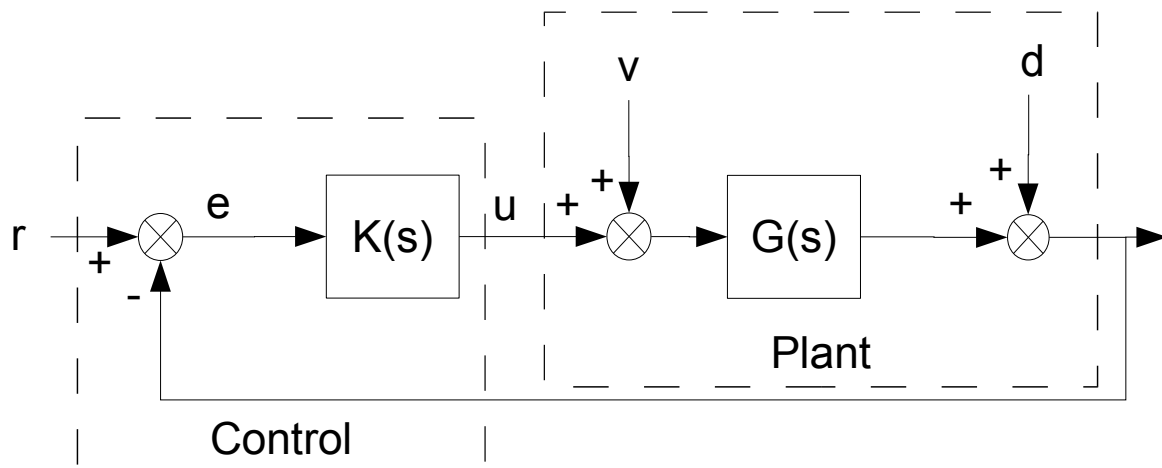


Figure 3.1: General Control Setup

3.2 The Plant Models

Three different plants are used as test cases for the thesis. The first is a simple first-order DC motor model which is used as a baseline for further tests. The next two plants introduce realistic limitations to the first-order plant: the first introduces an input saturation limitation on the plant, and the second has a non-dominant parasitic dynamic introduced to simulate sensor dynamics.

3.2.1 First Order Plant

The test plant $G_{motor}(s)$ is a small sized DC motor typically represented by a model:

$$\begin{aligned}
 G_{motor}(s) &= \frac{A}{(1 + sT_e)(1 + sT_m)} \\
 T_e &= \frac{L}{R} \\
 T_m &= \frac{J}{B}
 \end{aligned} \tag{3.1}$$

Where T_e and T_m are the electrical and mechanical time constants respectively. Often the electrical time constant is far smaller than the mechanical time constant for motors $T_e \ll T_m$, as such the model is often approximated by the first order dynamic system:

$$G_{motor}(s) = \frac{A}{1 + Ts} \quad (3.2)$$

where the parameters are approximately $A = 3.97$ and $T = 0.87$ for this test. (See Appendix C for details of the system identification)

3.2.2 Other Plants

Further testing was done on various plants with non-linearities and limitations as described below. They use the same base model as the first order DC motor above, but in each case include a common control phenomenon. They are theoretical in nature, included to test "what if" scenarios.

3.2.2.1 Input Saturation

The first of the non-linear plants includes input saturation. That is the input u is unable to exceed ± 10 [Volts]. This is realistic limitation based on the digital to analogue converter used to generate the control signal.

3.2.2.2 Parasitic Dynamics

The parasitic dynamics plant includes an extra dynamics term included to simulate parasitic sensor dynamics. As it is a theoretical phenomenon, the parasitic term was chosen to be twice as fast as the plant dynamic. Thus the model for the resulting plant is:

$$G_{pd}(s) = \frac{A}{(1 + T_{motor}s)(1 + T_{parasitic}s)} \quad (3.3)$$

The Nyquist diagram for the parasitic dynamics plant with PI control crosses the negative real axis. This implies that if the system gain is high enough, the resulting control will be unstable. This will test the optimiser as it should balance stability with speed of response.

3.3 Optimisation Objectives

The Pareto Differential Evolution algorithm works by optimising the objectives \mathbf{J} (costs or cost functions) for a given set of decision variables θ (i.e. inputs). The choice of these cost functions affects the outcome of the optimisation.

The resulting Pareto fronts are dependent on the specific cost functions used. In order to keep the comparison as general as possible, the costs were chosen to be as application non-specific as possible.

3.3.1 Error Costs

The first three proposed costs are chosen to give a metric of the plant error e (3.4) resulting from changes in the setpoint r , input v and output d disturbances introduced to the system. Where the plant error is the difference between the plant output and the setpoint:

$$e = r - y \quad (3.4)$$

These costs are intended to cover control engineering specific design requirements of settling time, oscillation and setpoint tracking for each of the disturbances. Two measures are applied (in separate comparisons), with different results.

The first measure $\mathbf{E} = (ISE_r, ISE_v, ISE_d)^T$ is based on the Integral Square Error (3.5) [11]. It is normalised for the magnitude of the perturbation.

$$\begin{aligned} ISE_r &= \int_{t_{perturb}}^{t_{final}} \left(\frac{e}{r}\right)^2 dt \\ ISE_v &= \int_{t_{perturb}}^{t_{final}} \left(\frac{e}{v}\right)^2 dt \\ ISE_d &= \int_{t_{perturb}}^{t_{final}} \left(\frac{e}{d}\right)^2 dt \end{aligned} \quad (3.5)$$

The second measure $\mathbf{E} = (EST_r, EST_v, EST_d)^T$ is a basic approximation of the settling time (3.6) of the dynamic response for a given disturbance. The settling time is defined as the time it takes for the system transient to enter and remain within a 5% band of the final value after perturbation. It does not take account of oscillations nor does it account for finite offset error in the setpoint tracking provided the error settles to within 5% of the desired value.

$$\begin{aligned} EST_r &= T_{r,5\%} \\ EST_v &= T_{v,5\%} \\ EST_d &= T_{d,5\%} \end{aligned} \quad (3.6)$$

The EST calculation used in the thesis is an approximate method. While it is possible in simulation to fit exponential decay curves to the envelope of the simulation data in order to find a more precise settling time, this was deemed too time consuming, considering the rest of the ODE and MOO framework.

An alternative (though less accurate) method was devised that estimates when the step response enters and remains within the 5% band around the final value. The inaccuracy in this method is due to a combination of: Sample time, usually very small and thus negligible (typically less than $\frac{1}{10}$ of the plant time); The frequency of oscillation, a function

can enter the 5% band well before its true decay curve does so; and situations where there is non-negligible tracking error, i.e. the system does not settle to the necessary final value, but has some finite offset. Further, in the case of SMC, the non-linear switching (chatter) can mean that the output never enters the 5% band 'for the last time'. In this case a midpoint average of the output is used as the equivalent linear approximation of the SMC output.

3.3.2 Controller Input Costs

The next three proposed costs $\mathbf{U} = (ISU_r, ISU_v, ISU_d)^T$ are chosen as a measure of total controller effort required by the proposed controller to reject each of the disturbances r, v, d .

They are based on the Integral Square Difference for the plant control u (ISU). The integral square error is modified such that the resulting cost is a finite integral, with the assumption that the controller settles to some final value, u_∞ , within the test period. This is done by subtracting u 's final value u_∞ from u at each time step. It is also normalised for the magnitude of disturbance r, v and d .

$$\begin{aligned}
 ISU_r &= \int_{t_{perturb}}^{t_{final}} \left(\frac{u - u_\infty}{r} \right)^2 dt \\
 ISU_v &= \int_{t_{perturb}}^{t_{final}} \left(\frac{u - u_\infty}{v} \right)^2 dt \\
 ISU_d &= \int_{t_{perturb}}^{t_{final}} \left(\frac{u - u_\infty}{d} \right)^2 dt
 \end{aligned} \tag{3.7}$$

3.3.3 Robustness Cost

The last cost included is a rough measure of the robustness of the controlled system to model changes. It finds the largest positive percentage change for each of the 6 other costs when the plant parameters A and T are varied by 50%.

While it would be preferable to use actual disturbance data from an actual plant, in this simulation case it is not possible.

The DC motor selected for the test case, due to the nature and setup of the motor, does not experience any significant input or output disturbances; nor does its model vary significantly. But, we wish to test the capabilities of the various control design methods and thus theoretical disturbances are applied.

The modification of the A and T values is done on a theoretical "what if" basis. It would be preferable to use actual plant variation data (which does not strictly apply to this plant), or to assume a certain amount of model inaccuracy.

For the metric a maximum mismatch of 50% of the actual value was assumed. This is based on the pragmatic assumption that some value is necessary. One would hope that modelling found a model less than 50% out from the actual value. Further, while far larger parameter variation is possible due to a changing plant and environment (motor wear or temperature variation for example), 50% was judged a reasonable test value.

Due to the way in which the Robustness Delta metric is calculated, it is possible for an extremely large range of values to be achieved (from testing, anything from 0 to $1e24$ was seen). While this value range is acceptable in isolation, it breaks the Level Diagrams used in the thesis. As most of the resulting controllers are fairly robust (i.e. no significant change to performance with model changes), the majority of values are around 1. When the values are normalised in the Level Diagrams, this effectively removes any impact the Robustness Delta metric would have on any given norm, except for the extremely large values.

Thus, in order to have the Robustness Delta metric be more useful when used with Level Diagrams, the base ten logarithm was used in order to spread the values on a more useful scale.

$$\text{Robustness Delta} = \log_{10} \max \left\{ \begin{array}{l} \frac{\mathbf{E}_{A \pm 50\%, T \pm 50\%}}{\mathbf{E}} \\ \frac{\mathbf{U}_{A \pm 50\%, T \pm 50\%}}{\mathbf{U}} \end{array} \right\} \geq 0 \quad (3.8)$$

where \mathbf{E} is either the *ISE* or *EST* measure (both are used in separate tests) and \mathbf{U} is the *ISU* measure.

3.4 Pareto Differential Evolution Optimiser

3.4.1 Optimiser Parameters

The PDE Optimiser has various parameters that affect its function. These include: The number of generations, the population size, the mutation rate and the crossover rate. The selection of these parameters has a direct effect on the accuracy and speed of the optimiser and as such need their specification needs to be investigated.

3.4.1.1 Number of Generations

The number of generations used in the PDE optimiser is one of the variables affecting the overall accuracy of the resulting Pareto front. Empirical testing (see Appendix E) was done to evaluate the number of generations needed to achieve a suitably accurate Pareto front. There is a trade-off between optimisation length and accuracy; the number of generations directly affects the length of the optimisation.

The PDE optimiser was run for 30 generations, this was chosen based on examination of the resulting Pareto fronts: in general for the proposed test problem, there was little variation after 10 generations, but 30 was chosen to give fair leeway for unexpected variation and it would allow the optimiser to find a closer approximation of the Pareto front.

3.4.1.2 Population Size

As with the number of generations, population size has a direct effect on the optimisation length, while affecting the overall accuracy of the resulting Pareto front approximation.

Larger population sizes are obviously preferable in approximating the Pareto front, especially in high dimensional problems. Yet, during testing (see Appendix E) it was found that a smaller population was able to reveal the general shape and bounds of the Pareto front, within a bounded region.

Population Sizes of 300 were used for most optimisations, except in a few cases where the resulting Pareto front approximations appeared to require more.

3.4.1.3 Crossover and Mutation Rates

The crossover rate and mutation rate were both set to 0.15. This was taken from [1], where it was found that lower crossover rates resulted in a larger number of non-dominated solutions.

3.4.2 Controller Parameters

The PDE optimiser uses a bounded range for the decision space for each optimisation. These bounds are selected as follows.

3.4.2.1 Parameter Ranges for PI Control

The input parameters for the PI controller are K_p and K_i as shown in (3.9).

$$K(s) = K_p + \frac{K_i}{s} \quad (3.9)$$

The parameters are bounded to the range $K_p \in [0, 10]$ and $K_i \in [0, 10]$. This range

was chosen such that the resulting zero is stable, and such that the gain of the resulting control is limited to a reasonable range for the DC motor system.

For the parasitic dynamics plant example, the ranges $K_p \in [0, 10]$ and $K_i \in [0, 10]$ allow for system gains that (according to the Nyquist diagram) could result in both stable and unstable systems. Thus, the optimiser will be tested that it is able to avoid these gains.

A set of PI controllers was included (and is used later) with each controller having pole - zero cancellation for comparison purposes. This "PI Cancel" set's parameters are specified such that on the range $[0,10]$ pole - zero cancellation occurs for all controllers, but with each controller having a different gain. It is not optimised, but is rather a fixed set.

3.4.2.2 Parameter Ranges for SMC

Similarly for both SMC, the input parameters are λ , Φ and K . They are bounded to the ranges $\lambda \in [0, 4]$, $\Phi \in [0, 6]$ and $K \in [0, 10]$.

These limitations on λ and Φ are introduced based on the analysis of the eigenvalues of the resulting system dynamics. They are primarily introduced as a result of limited sampling speed and the requirement that the system should be on its sliding mode before settling otherwise its beneficial characteristics of invariance and robustness are not achieved.

For Boundary Layer SMC, the thickness of the boundary layer is specified by BL . The layer needs to be wide enough to avoid chattering, but not be so wide that the control is effectively linear. Thus $BL \in [0, 10]$, gives a large range of allowable thicknesses.

CHAPTER 4

Results

The primary focus of this thesis is determining the usefulness and applicability of Level Diagrams and Performance Measures in comparing different control design methods quantitatively. The results are separated by these methods, with each control design comparison discussed under these broad topics.

This chapter deals with a first-order plant model (as discussed in Chapter 2). Many of the observations and results are obvious in nature, but are intended to draw attention to what can be seen and what conclusions can be drawn, from the methods used.

4.1 Region of Interest

Reference [12] raises an interesting point regarding the pertinence of the solutions being shown to the decision maker. As a means to reduce the amount of information being presented, and make that information clearer, the concept of region of interest ROI is introduced: This is the region of the objective space (and possibly the decision space) that is useful or important to the decision maker.

The PDE method used to optimise each design in this thesis uses a bounded decision space for selecting decision variables, thus effectively creating a region of interest for the decision space.

Level Diagrams provide a simple means to reduce the size of the objective space presented to the decision maker: the synchronising norm. Limiting the value of the norm

that is actually plotted on the Level Diagram effectively applies a region of interest to the Pareto fronts shown. This is done on the assumption that it will remove the extremes of the Pareto front, effectively presenting the best and more balanced cases. The one advantage of this method of region of interest specification is that the decision maker frequently does not know what range of values will result from the various objectives being used, and is thus unable to choose reasonable bounds for them.

A different maximum norm value is applied for the three different norms used. The infinity norm was set to 0.4 based on the general shape and position of the Pareto fronts in the Level Diagrams. The one norm and two norm were then calculated to have the equivalent maximum in their norms. That is the two norm maximum was set to 0.56 and the one norm maximum set to 0.8.

As an example, the one, two and infinity norm Level Diagrams for a single objective, in this case ISU_d , are examined in more detail. These Level Diagrams show three different approximate Pareto fronts: PI in black, ISMC in dark grey, and the PI pole - zero cancellation set in light grey.

First the Level Diagrams are shown without any maximum norm being applied in Figure 4.1. The maximum norm limits are shown on Level Diagrams as a horizontal black line.

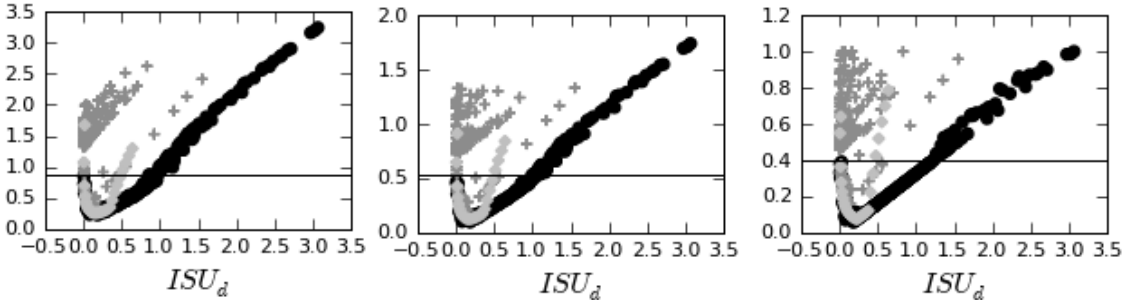


Figure 4.1: ISU_d Level Diagrams using the one, two and infinity norms

In the second figure, Figure 4.2, the same one, two and infinity norm ISU_d Level

Diagrams are now shown, but this time using the maximum norm limits described above, thus creating the same region of interest in each norm.

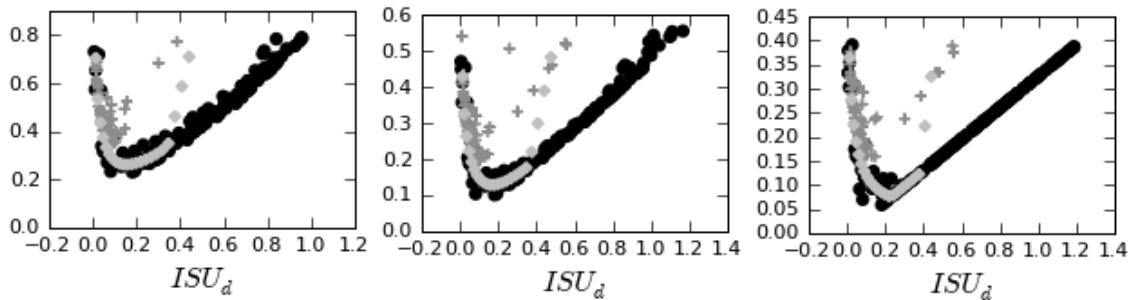


Figure 4.2: ISU_d Level Diagrams using the one, two and infinity norms using maximum norm limiting.

Using this norm reduction technique, it is possible to focus the visualisation on the region of interest such that the relevant information is now much clearer.

4.2 Level Diagram Results

The results pertaining to Level Diagrams for the first-order example are presented below.

4.2.1 Use of Different Norms

As described in Section 2.4, there are 3 different synchronisation norms suggested in [5]: The $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ norms. Figures 4.3, 4.4 and 4.5 shows the Cost Level Diagrams for the three norms respectively. (The data shown is from the comparison of PI and ISMC design methods).

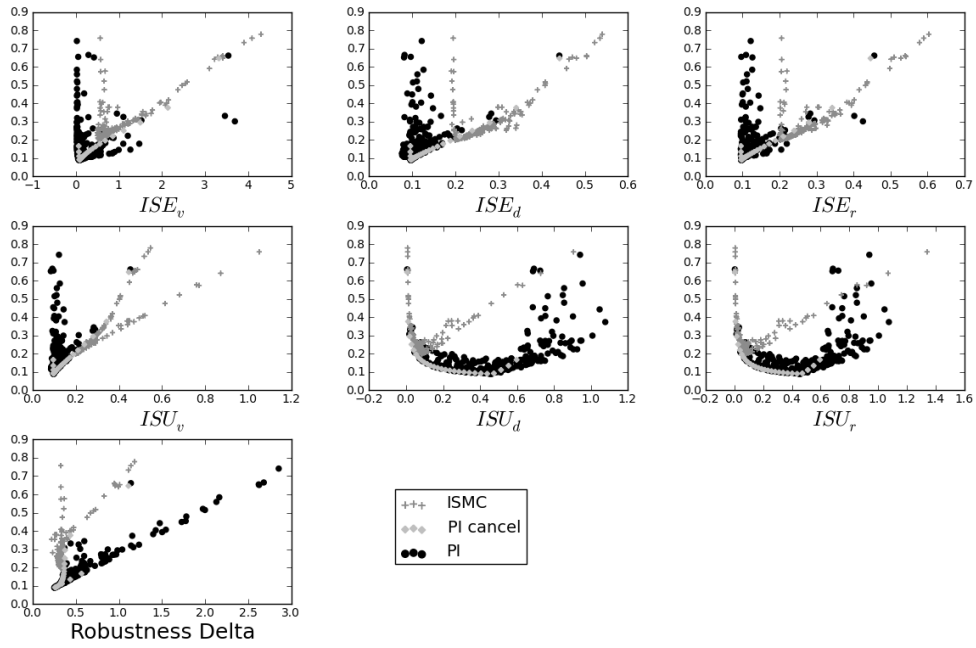


Figure 4.3: $\|\cdot\|_1$ Cost Level Diagrams for PI, PI Cancel and SMC

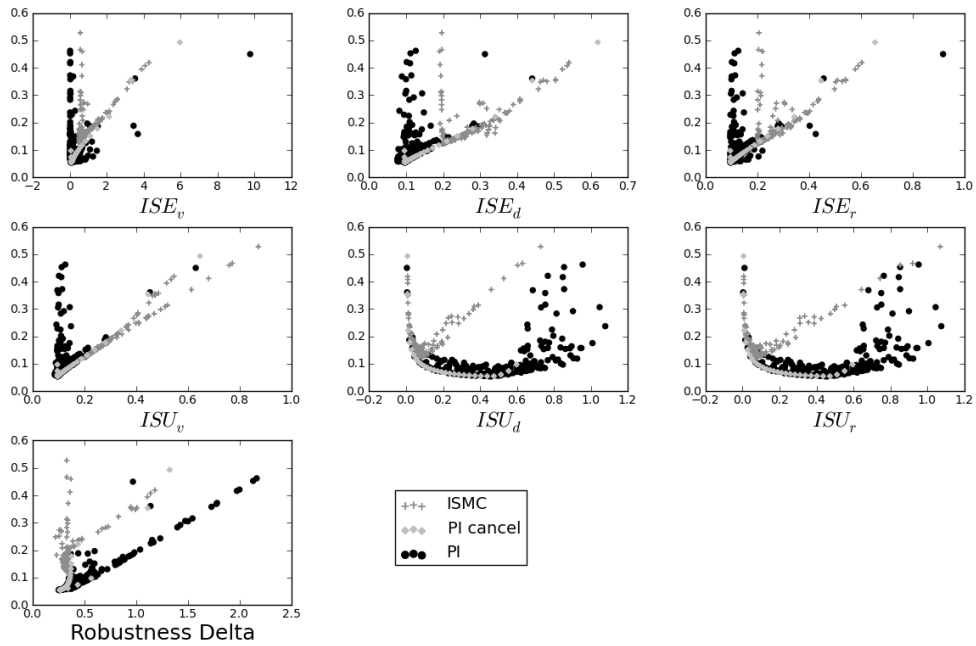


Figure 4.4: $\|\cdot\|_2$ Cost Level Diagrams for PI, PI Cancel and SMC

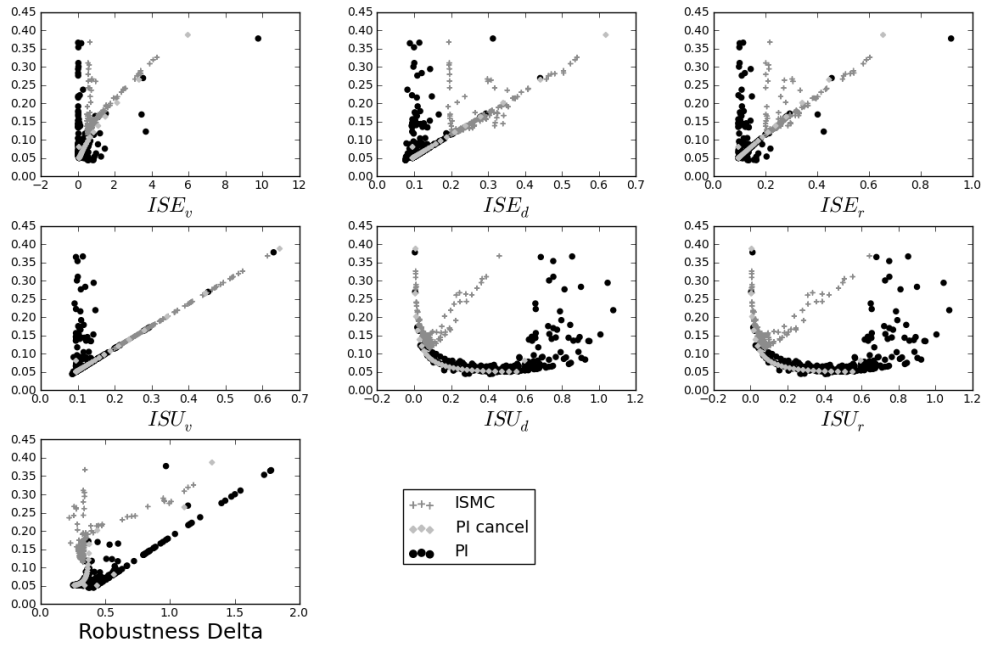


Figure 4.5: $\|\cdot\|_\infty$ Cost Level Diagrams for PI, PI Cancel and SMC

While the Level Diagrams do look slightly different, which could be useful for trade-off analysis within each Pareto front, for the purposes of comparison of Pareto fronts, the norms make no significant difference other than to the selection of the region of interest.

To demonstrate this, the ISU_d Level Diagrams are used as an example. The one norm, two norm and infinity norm are re-drawn in Figure 4.6.

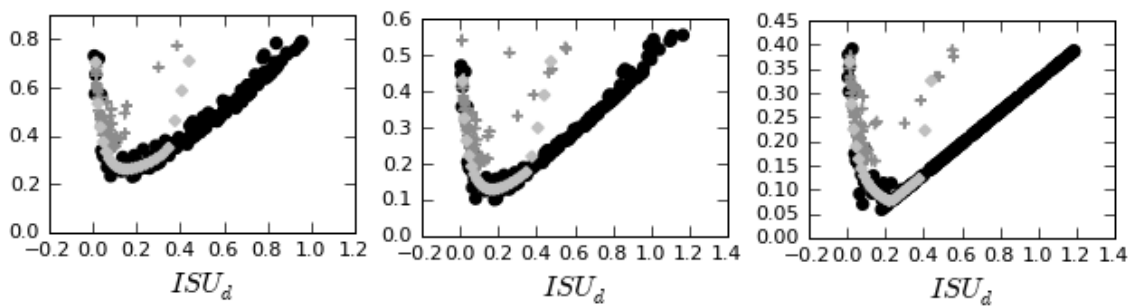


Figure 4.6: U_d Level Diagrams using the one, two and infinity norms

Each of the Level Diagrams has similar shape and features, indicating that the norms do not add any significant difference to the visual comparison of the Pareto fronts.

4.2.2 Number of Objectives

The specific objectives used affect the results of the optimisation process. This includes their definition, such as whether to normalise the ISE costs to the disturbances applied or not; or what the specific mix of objectives being used is, i.e. Different objectives being used: ISE_r, ISE_v, ISE_d , or EST_r, EST_v, EST_d for the plant error objectives.

In order to demonstrate the effect of the number of objectives and the effect of different objectives, the PI and ISMC design methods were optimised for two and four objectives using both the ISE and EST objectives.

Figure 4.7 shows the $\|\cdot\|_2$ Level Diagrams for PI versus PI Cancel design methods, optimising on ISE_r and ISU_r . The resulting PI controllers are optimised onto the pole-zero cancellation curve. The optimised PI controllers are shown in black and the PI pole-zero cancelling set is shown in grey. In this case focussing on the setpoint would be a typical result of loop tuning that is commonly applied in industry.

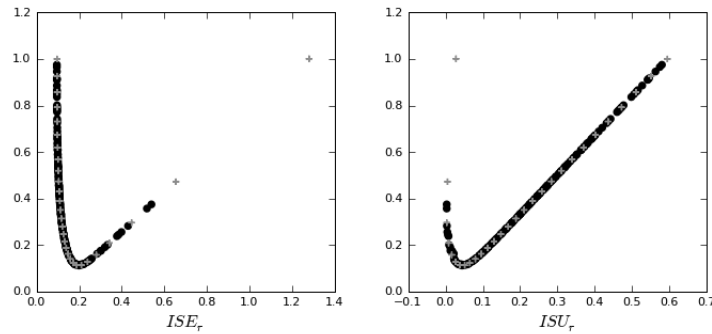


Figure 4.7: $\|\cdot\|_2$ ISE Cost Level Diagrams for PI and PI Cancel using two costs

Figure 4.8 shows the same comparison but optimising using ISE_r, ISU_r, ISE_v and ISU_v . Some of the resulting PI controllers are still being optimised onto the pole-zero

cancellation curve, but the inclusion of more costs has caused many to not cause pole-zero cancellation.

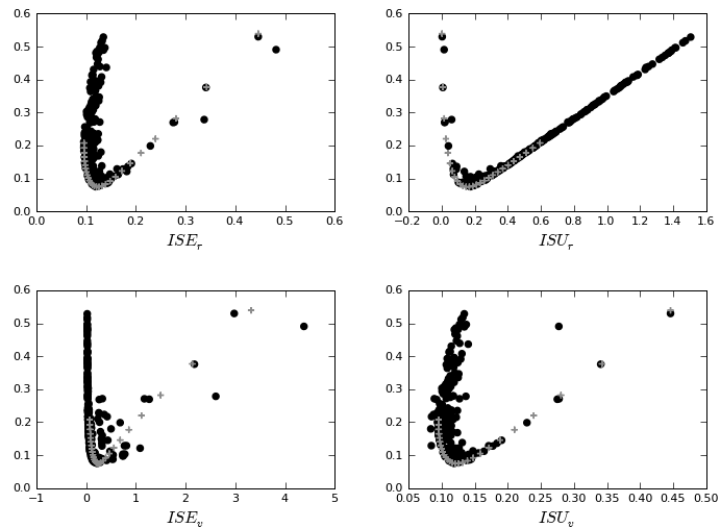


Figure 4.8: $\|\cdot\|_2$ ISE Cost Level Diagrams for PI and PI Cancel using four costs

Using the EST_r , ISU_r objectives yields the Level Diagrams shown in Figure 4.9, as can be seen, the EST costs do not optimise onto the pole-zero cancellation curve. The inclusion of two more objectives EST_v and ISU_v has the resulting PI controllers off of the pole-zero cancellation curve as well, as shown in Figure 4.10.

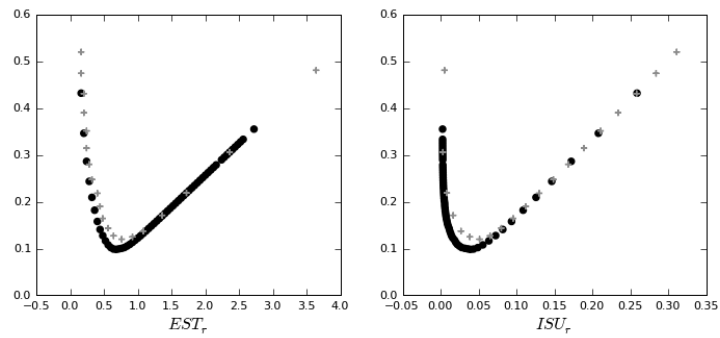


Figure 4.9: $\|\cdot\|_2$ EST Cost Level Diagrams for PI and PI Cancel using two costs

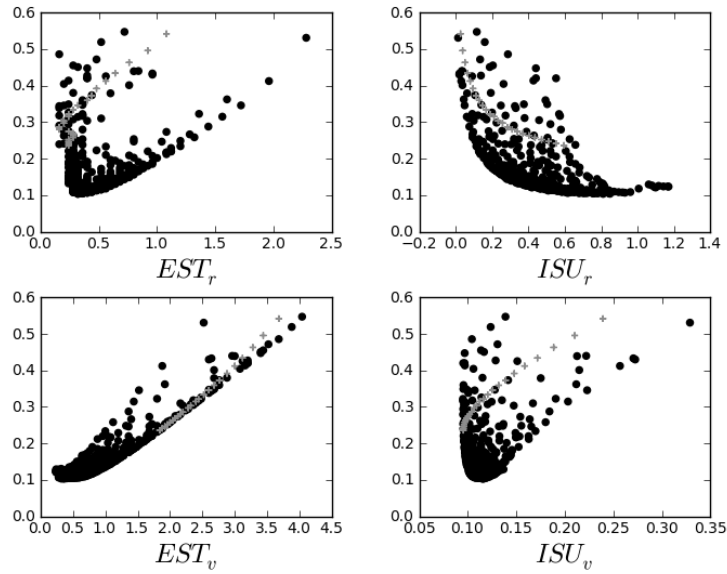


Figure 4.10: $\|\cdot\|_2$ *EST* Cost Level Diagrams for PI and PI Cancel using four costs

Thus, increasing the number of objectives should to a certain extent help avoid pole-zero cancellation, as well as provide a more robust analysis. The counter to this is that increasing the number of optimisation objectives increases the population size needed by the optimiser, as well as making analysis of the objectives more complex.

4.2.3 PI versus ISMC

Pareto fronts were generated for the PI and Integrated SMC controllers for both the *ISE* and *EST* objectives, using a first-order DC motor plant. The pole-zero cancellation curve is included as a known undesirable set. The resulting Level Diagrams are discussed in the following:

4.2.3.1 Integral Square Error Level Diagrams

Figure 4.11 shows the resulting Objective (or Cost) $\|\cdot\|_2$ Level Diagrams and Figure 4.12 show the resulting Decision (or Input) $\|\cdot\|_2$ Level Diagrams for the Pareto front for the PI (black), PI cancel (light grey) and ISMC (dark grey) methods using the *ISE* plant error objective.

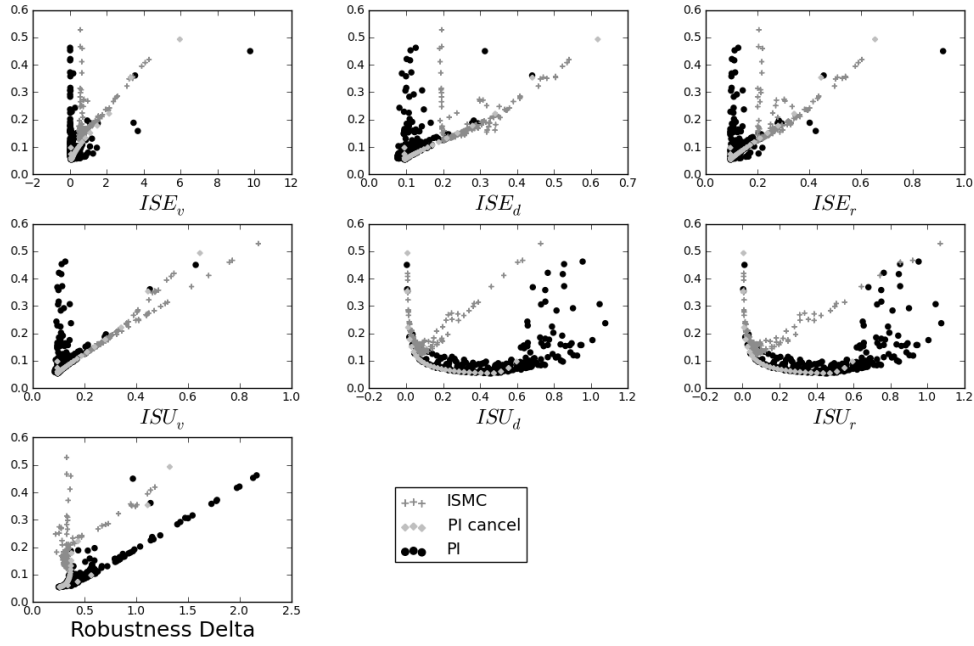


Figure 4.11: $\|\cdot\|_2$ *ISE* Cost Level Diagrams for PI, PI Cancel and ISMC

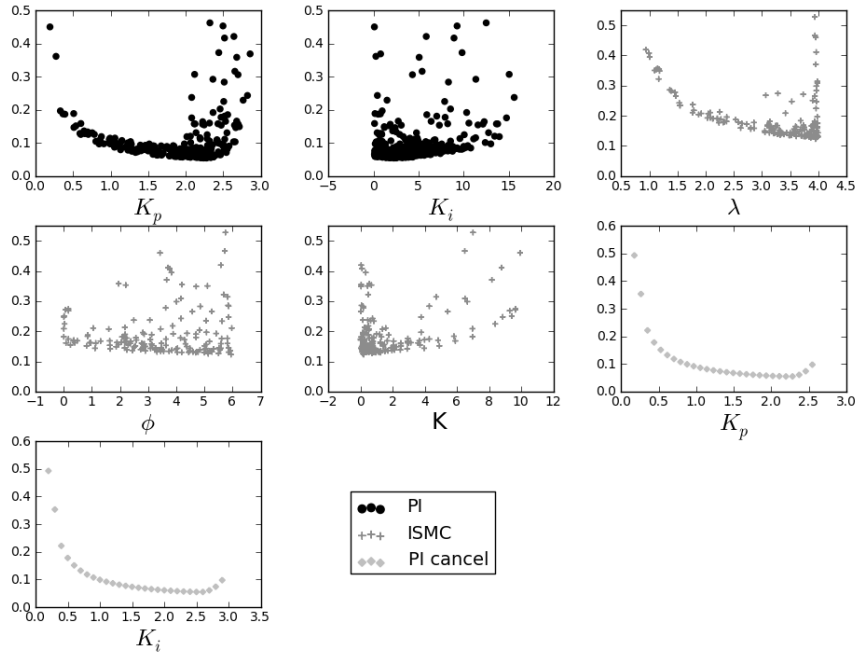


Figure 4.12: $\|\cdot\|_2$ ISE Input Level Diagrams for PI, PI Cancel and ISMC

A number of observations can be drawn from visual inspection of the Level Diagrams. From the *Robustness Delta* Level Diagram re-drawn in Figure 4.13, the ISMC controller is in general more robust to model changes than the PI controller; having a maximum *Robustness Delta* value of less than 1.5, while the PI solutions are, in general, more sensitive to model changes, as well as having some controllers that are far more sensitive.

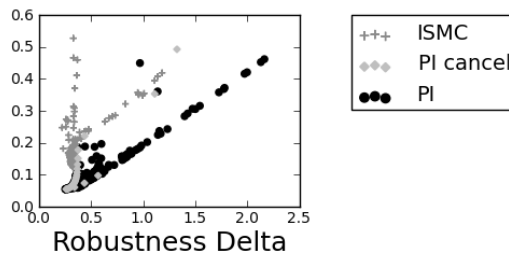


Figure 4.13: *RobustnessDelta* for PI, PI Cancel and ISMC

A number of observations can be drawn from the Level Diagrams for output disturbance and setpoint tracking (ISE_d, ISU_d, ISE_r and ISU_r) re-drawn in Figure 4.14.

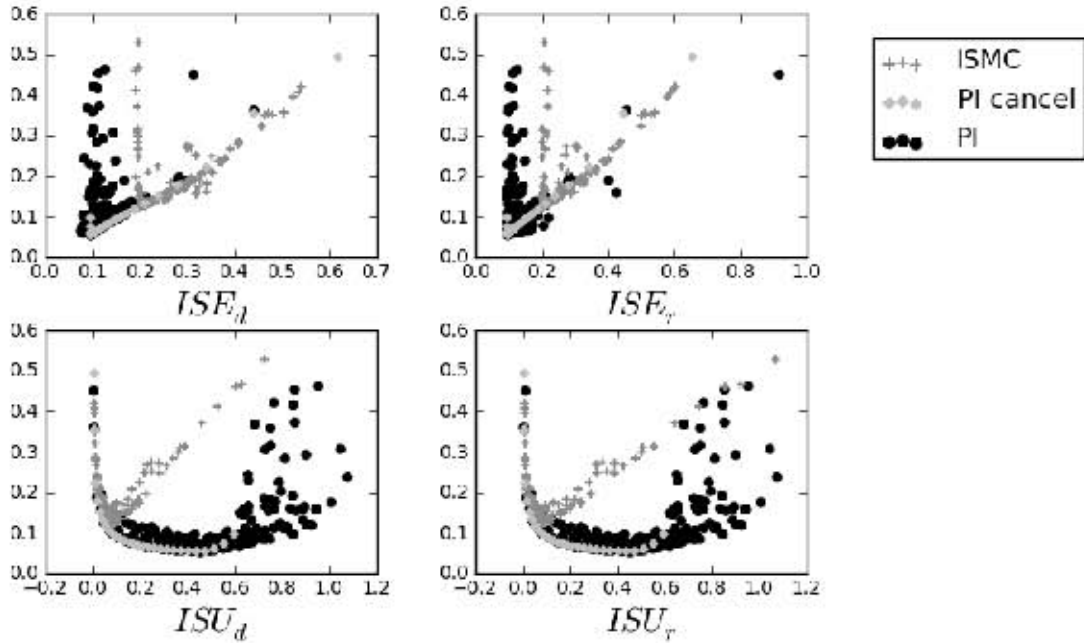


Figure 4.14: ISE_d, ISU_d, ISE_r and ISU_r for PI, PI Cancel and ISMC

The PI controller's response is the same for both setpoint and output disturbances due to it being a linear controller with unity feedback. While the ISMC controller's ISE for both the setpoint and output disturbance are different.

An obvious observation is that in general the PI controller achieves lower ISE_d and ISE_r for greater ISU , while the ISMC is the reverse. This is a fairly obvious result in that, if a controller puts more effort in, one expects it to respond faster and thus have a lower ISE . What is of interest is that each of these controllers result in a Pareto front that balances this trade-off differently.

For the given optimisation objective, the optimal PI control favours settling time over controller effort, while ISMC does the reverse.

This situation differs for the input disturbance (ISE_v and ISU_v) re-drawn in Figure 4.15. PI manages equal or better input disturbance rejection as shown in ISE_v , and generally requires less controller effort than the ISMC controller to achieve it, shown in ISU_v .

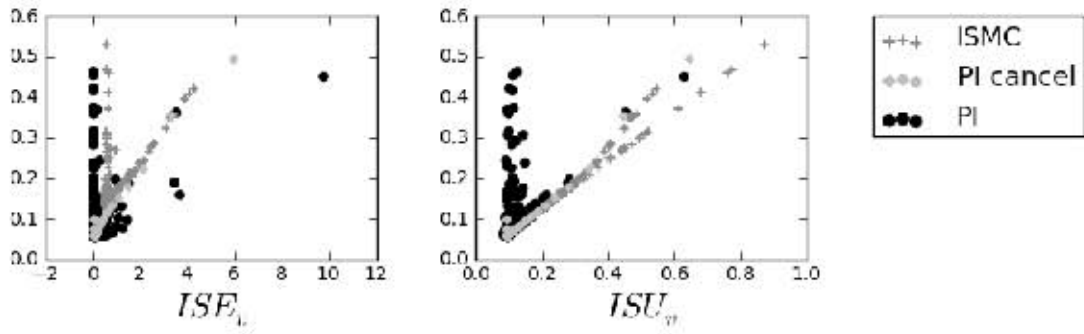


Figure 4.15: ISE_v and ISU_v for PI, PI Cancel and ISMC

A few other observations can be drawn from the Input Level Diagrams shown in Figure 4.12. Firstly, lower values of λ re-drawn in Figure 4.16 are undesirable as they limit the reaching speed on the ISMC and thus increase the various ISE measures: The value of λ sets one of the eigenvalues of the system dynamics, and thus directly affects the settling time. Higher values of λ tend to have lower norm values.

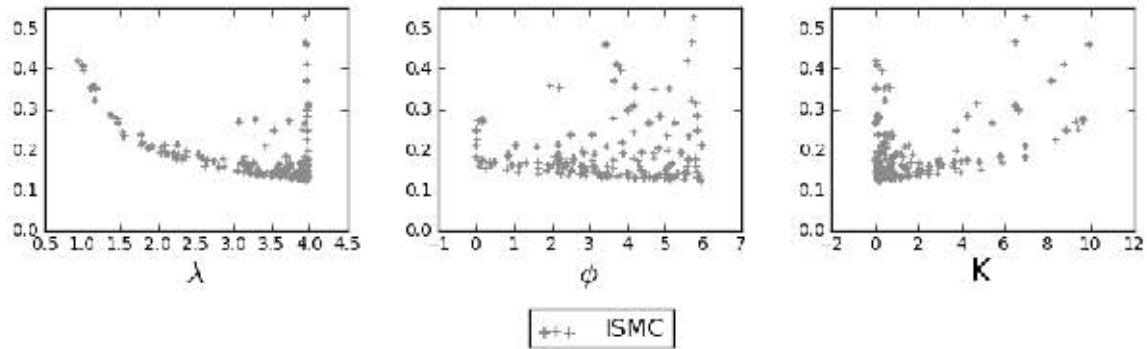


Figure 4.16: λ , ϕ and K for ISMC

Lower Values of K tend to result in lower norm values. If it were not for the integral action of ISMC, this could be attributed to the high frequency switching, i.e. smaller switching causes lower ISE and thus lower norms. But the integral action retards this such that there is only a minor ripple, but it would appear that this is enough to cause lower values of K to be favoured. Furthermore, the ϕ plot is troubling in that it is reasonably flat, indicating no significant trade-off relationship between the costs used and this decision variable. Though the higher ϕ appear to be achieving lower norm values, indicating that the reaching rate (the eigenvalue that ϕ sets) has a slight effect on the overall controller performance. Alternatively, this could indicate that the optimisation is over-parametrised; and that using fewer parameters could be necessary.

From the K_p Level Diagrams re-drawn in Figures 4.17 and 4.18, it is apparent that there is a band of optimal values, resulting in low norms, with a general minimum around $K_p = 2.25$. Also, from the K_i Level Diagram for PI, lower values of K_i are favoured, indicating that for this set of objectives, nearly proportional-only gain controllers do very well.

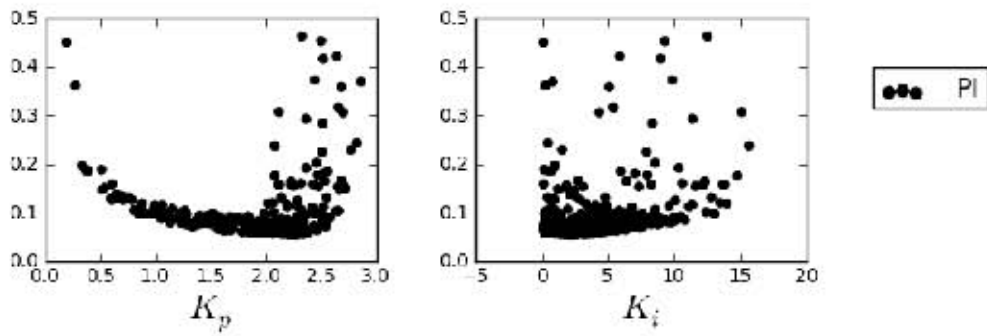


Figure 4.17: K_p and K_i for PI

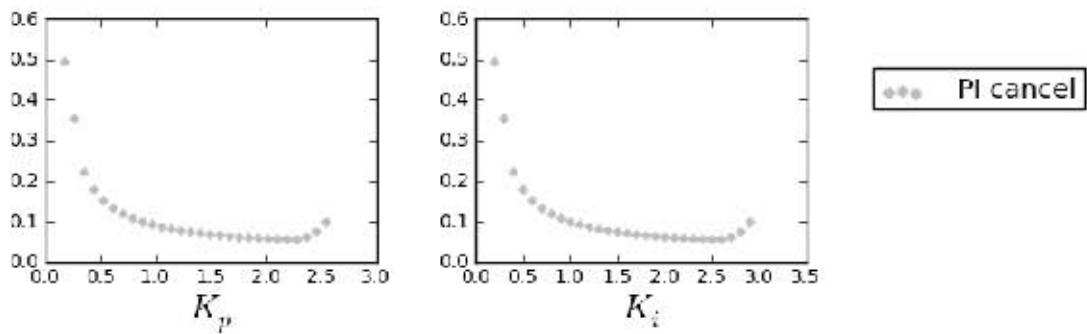


Figure 4.18: K_p and K_i for PI cancellation set

For the PI pole-zero cancellation set, values of K_p over 1.83 and K_i over 2.2 are actually non-dominant (they are included here to give a more complete view of the pole-zero cancellation set). Neither the root-locus, nor the Bode diagrams reveal any obvious causes for this sudden change in system behaviour. Further investigation revealed that while the higher gain K_p - K_i pairs resulted in very good disturbance rejection, the resulting control was very sensitive to model changes, causing the larger norms.

The low gain pole-zero cancelling pairs K_p and K_i resulted with worse (high) norm values.

Figure 4.19 shows an example step test of the lowest norm controllers from each of

the control design methods: PI and ISMC.

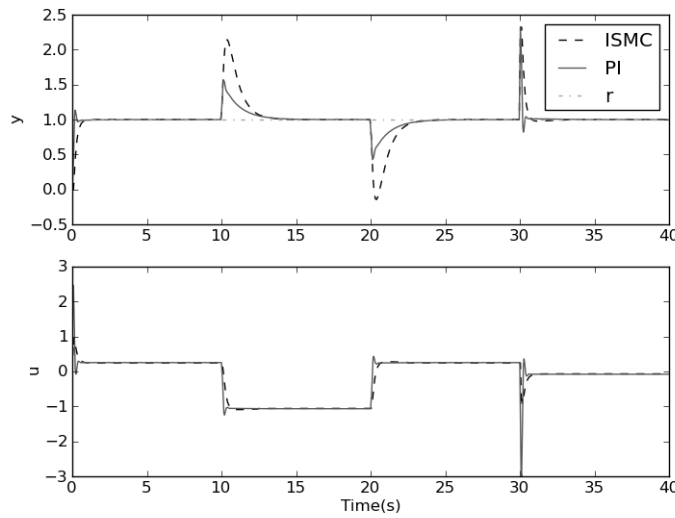


Figure 4.19: Lowest norm step tests for PI and ISMC for *ISE* objectives.

As with the Level Diagrams, PI achieves faster tracking for the setpoint. It is difficult to see the input signal, but neither takes long to reach their final stable values.

The input disturbance occurring at 10 seconds (and being removed at 20 seconds), is interesting in that while the PI controller has a lower *ISE* value than the ISMC, it in fact takes significantly longer to actually settle. The Level Diagrams show that PI controller outperforms ISMC for input disturbances, but it would appear that this is an artefact of the *ISE* objectives selected for optimisation.

4.2.3.2 Error Settling Time Level Diagrams

Figures 4.20 and 4.21 show the $\|\cdot\|_2$ Cost and Input Level Diagrams using the Error Settling Time Costs.

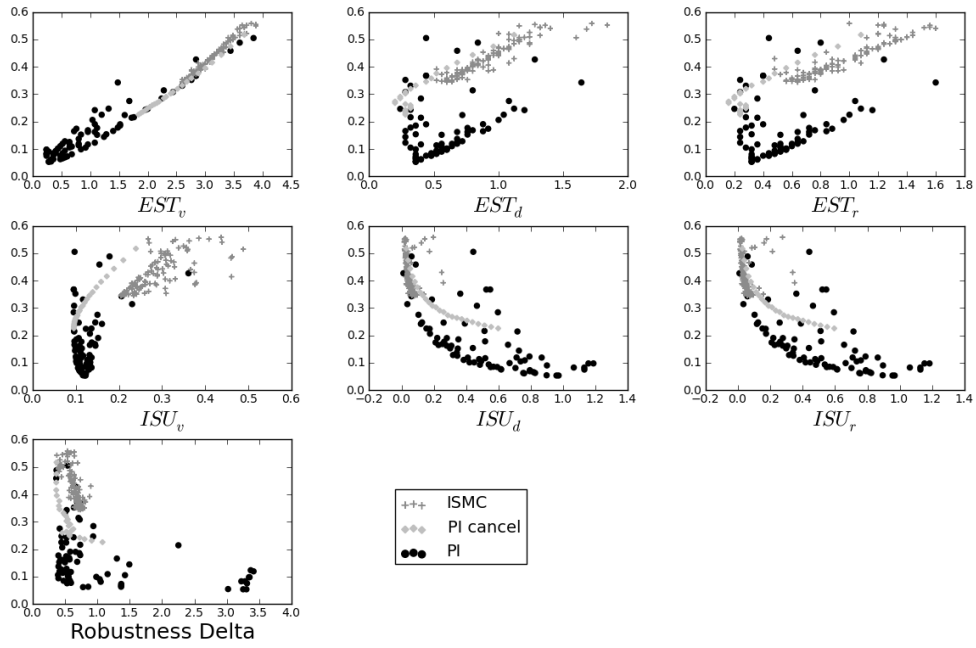


Figure 4.20: $\|\cdot\|_2$ *EST* Cost Level Diagrams for PI, PI Cancel and ISMC

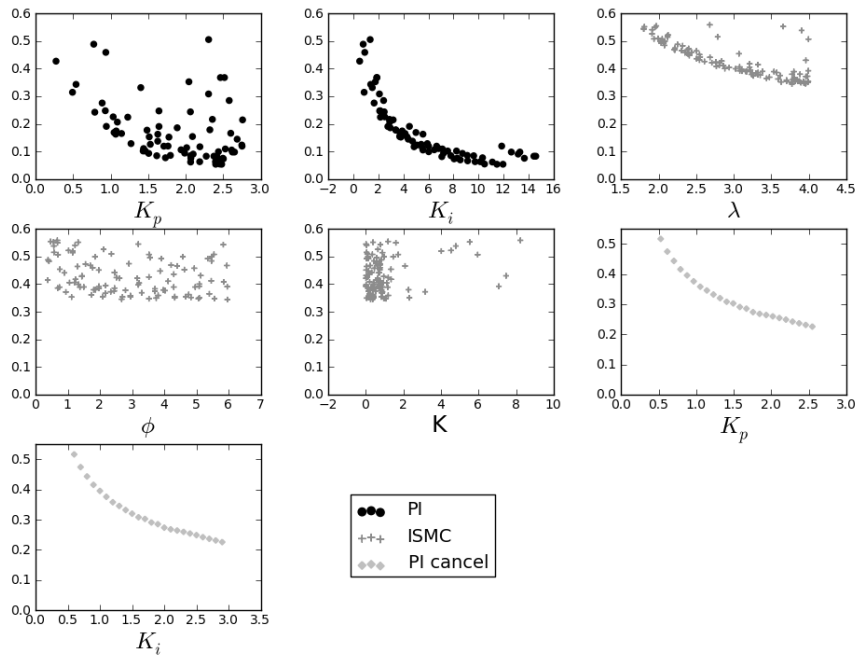


Figure 4.21: $\|\cdot\|_2$ *EST* Input Level Diagrams for PI, PI Cancel and ISMC

Figure 4.22 *RobustnessDelta* shows that the resulting PI controllers are in general far more robust in these objectives than they were when using the *ISE* objectives (see Figure 4.13), though the ISMC controllers are still far more robust.

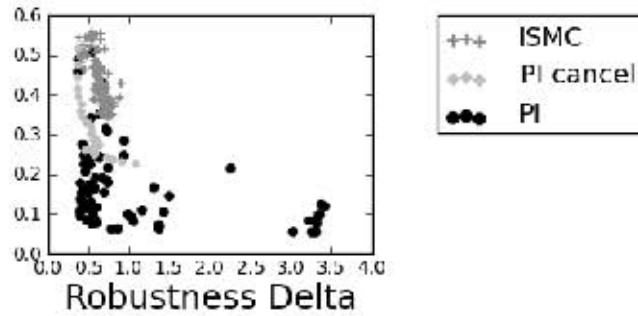


Figure 4.22: *RobustnessDelta* for PI, PI Cancel and ISMC

Similar to the *ISE* objectives, the ISMC controller as re-drawn in Figure 4.23 uses little ISU_r or ISU_d , but is unable to match the best results of the PI controllers in EST_r and EST_d .

Of note (though highlighted earlier) is the difference between the optimal controllers for PI compared to the PI cancellation set: for the *EST* objectives, they differ significantly, which is easily seen in these Figures.

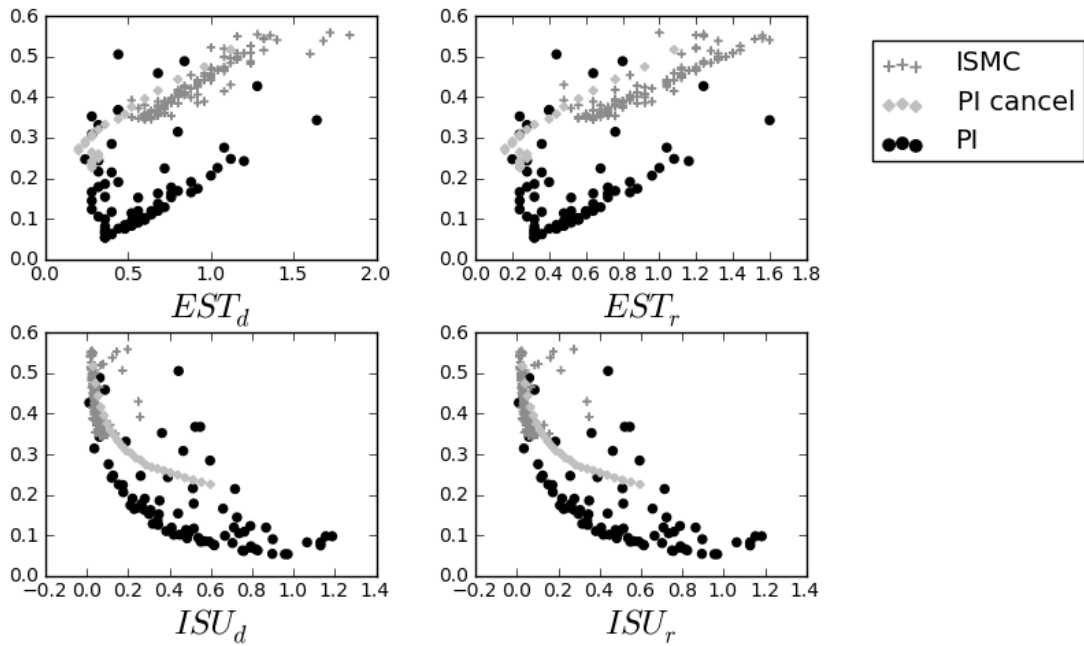


Figure 4.23: EST_d , ISU_d , EST_r and ISU_r for PI, PI Cancel and ISMC

EST_v and ISU_v re-drawn in Figure 4.24 shows that ISMC has significantly worse input-disturbance rejection than PI: in general using more controller effort ISU_v to produce slower error settling time EST_v than the PI controllers for input-disturbances.

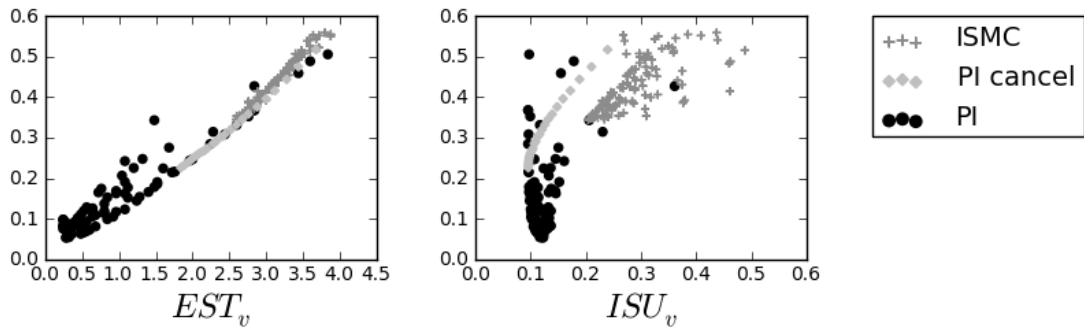


Figure 4.24: E_v and U_v for PI, PI Cancel and ISMC

The resulting K_p values in Figures 4.25 and 4.26 differ from the *ISE* objective results, they similarly favour K_p values around 2 - 2.5, while preferring much higher values of K_i .

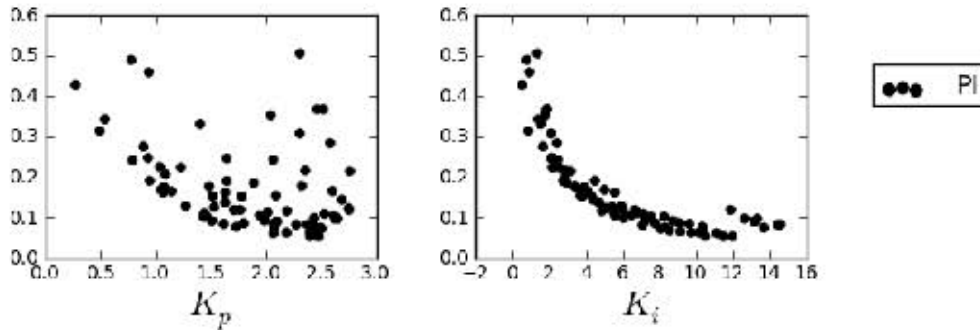


Figure 4.25: K_p and K_i for PI

The pole-zero cancellation now shows a definite preference for higher gain solutions.

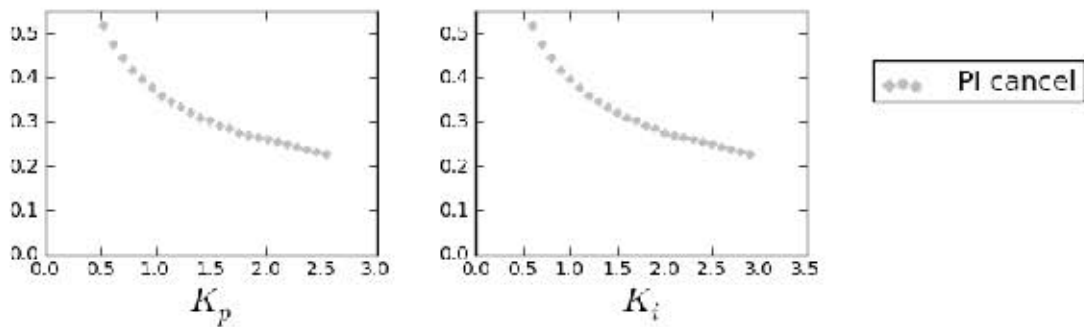


Figure 4.26: K_p and K_i for PI Cancel

Unlike the *ISE* objective Level Diagrams, a significant portion of the PI pole-zero cancellation set is dominated by the optimal PI controllers. Indicating that the *EST* objectives are better at avoiding (or outperforming) pole-zero cancellation.

Similarly to the Input Level Diagrams for the *ISE* objectives; The λ Level Diagram re-drawn in Figure 4.27 indicates a direct relationship between the value of λ and the

resulting norm. Smaller K values are heavily favoured for the ISMC designs. Furthermore, ϕ is similarly flat, indicating either a lack of trade-off relationship between this decision variable and any of the objectives, or over-parametrisation of the controller.

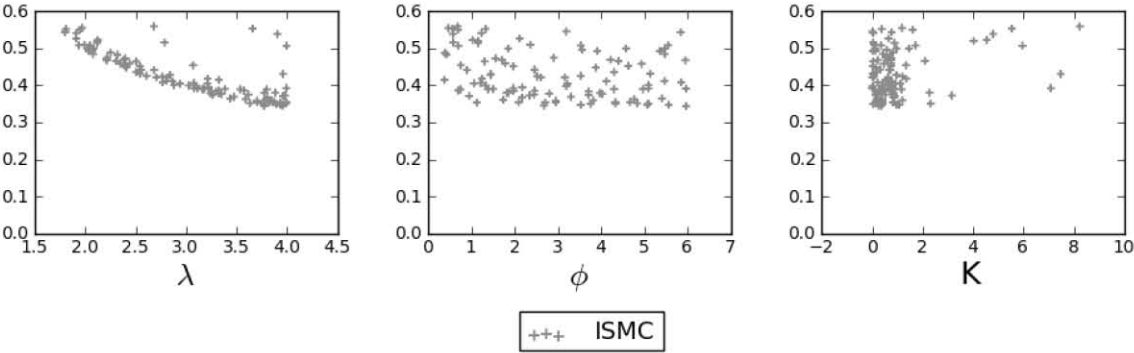


Figure 4.27: λ , ϕ and K for ISMC

Figure 4.28 shows an example step test of the lowest norm controllers from each of the control design methods using the *EST* objectives.

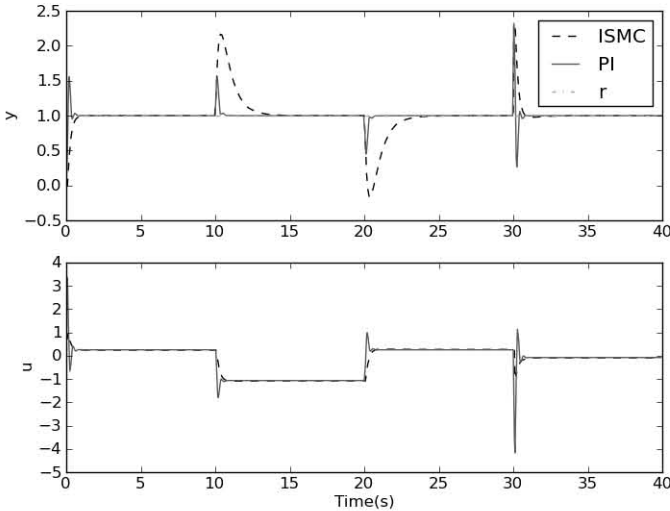


Figure 4.28: Lowest norm step tests for PI and ISMC for *EST* objectives.

Here we see a very different situation to the *ISE* case. The PI controller does appear to do better for each of the setpoint, input and output disturbances than the ISMC controller. This follows what the Level Diagrams indicate.

Similarly, for the controller effort objectives, the step tests agree with what the Level Diagrams were indicating: PI uses less controller effort for input disturbances (shown at 10 seconds and removed at 20 seconds), while ISMC performs equally well for setpoint and output disturbances.

While both controllers use similar *ISU* for the input disturbance, it is interesting that they do so differently. The ISMC controller has no overshoot on its control signal, settling to a final value and then having minor chattering. On the other hand, the PI controller has some overshoot before settling.

4.2.4 Integrated SMC versus Boundary Layer SMC

Pareto fronts were generated for the Integrated SMC (ISMC) and Boundary Layer SMC controllers (bSMC) for the *ISE* objectives. The *EST* objectives do not function well with the bSMC controllers as the finite offset error can cause the resulting controllers to never enter the $T_{\pm 5\%}$ band used by the *EST* objectives. The resulting Level Diagrams are discussed in the following:

4.2.4.1 Integral Square Error Level Diagrams

The resulting Cost and Input Level Diagrams for the *ISE* objective comparison of ISMC and bSMC are shown in Figures 4.29 and 4.30.

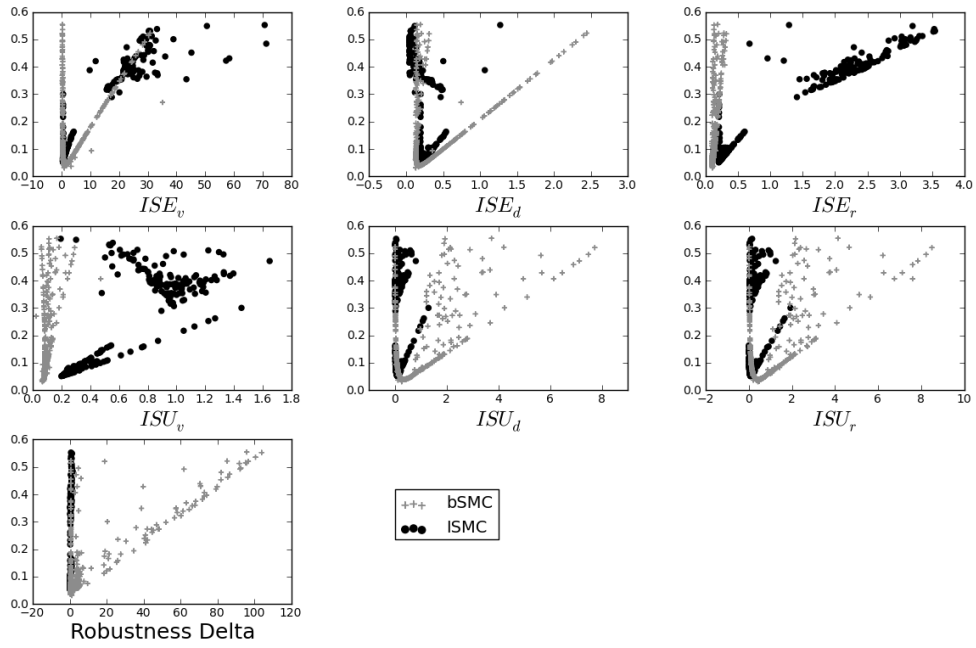


Figure 4.29: $\|\cdot\|_2$ ISE Cost Level Diagrams for ISMC and bSMC

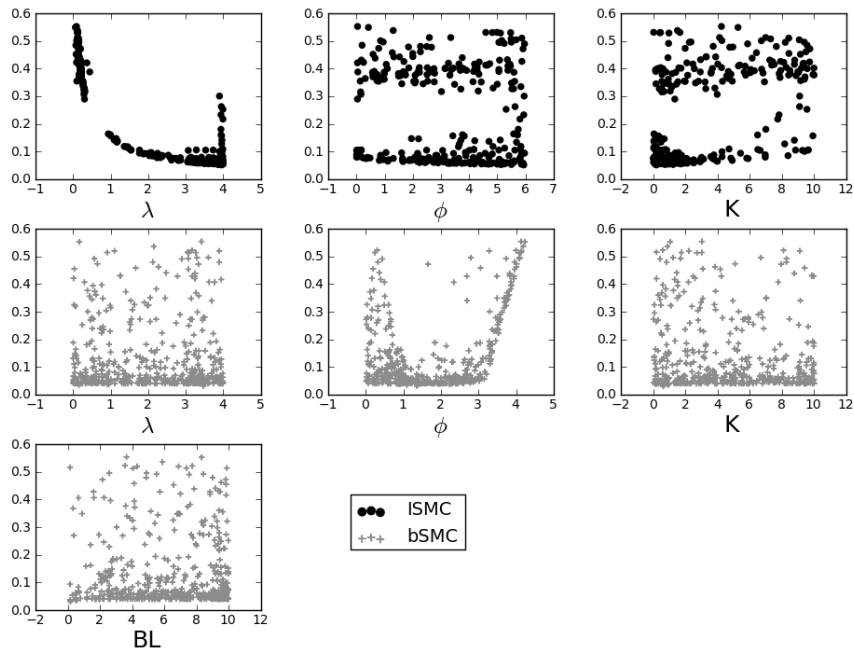


Figure 4.30: $\|\cdot\|_2$ ISE Input Level Diagrams for ISMC and bSMC

The ISMC controllers' Pareto front can be separated into two groups, a lower norm grouping (with a maximum of around 0.2), and an upper grouping. The lower group ISMC controllers are generally able to reject input disturbances equal to or better than the bSMC controller, this can be seen in the plot of ISE_v in Figure 4.31, while the upper group does significantly worse. This is caused by a combination of two factors: Firstly, the inclusion of the integrator in ISMC means that any input disturbance has to go through the plant dynamics before being "seen", as well as the integrator effectively limiting the rate at which the system can respond. This is also linked to the reaching rate, which affects how long it takes the system to get back onto the sliding mode (i.e. reject disturbances), which is set by λ , this is discussed further below. On the other hand, the bSMC controller uses very little controller effort ISU_v for input-disturbances, while both ISMC groupings use more controller effort.

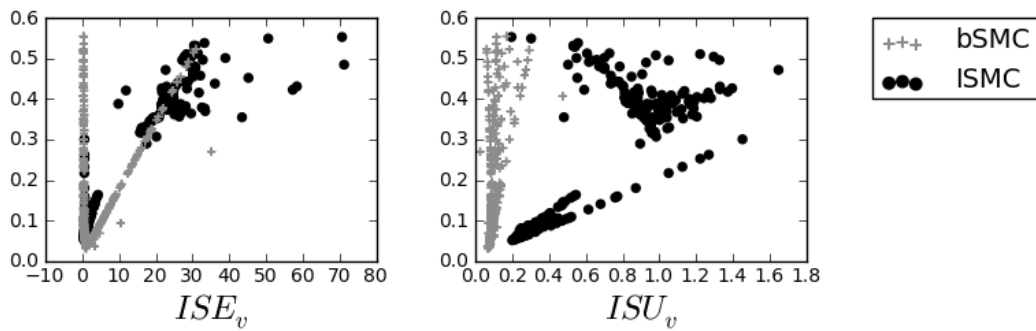


Figure 4.31: ISE_v and ISU_v for ISMC and bSMC

For output disturbances, ISMC generally tends to outperform bSMC in both plant error ISE_d and controller effort ISU_d .

bSMC achieves better setpoint tracking than ISMC, seen in the ISE_r Level Diagram. Though it does use more controller effort ISU_r to achieve these faster dynamics. The upper grouping does significantly worse at setpoint tracking.

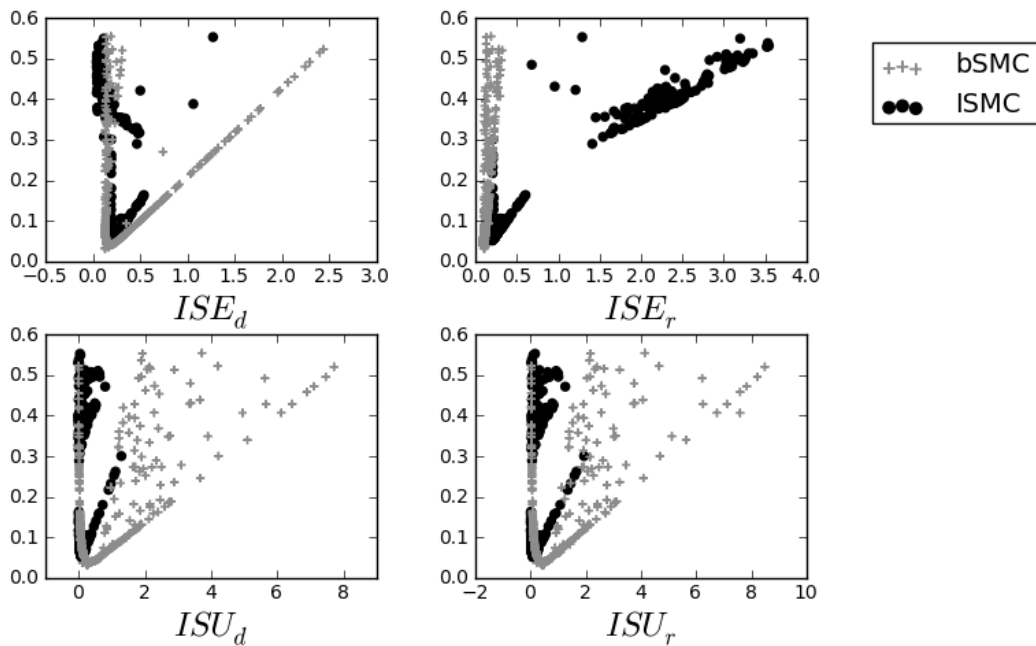


Figure 4.32: ISE_r , ISE_d , ISE_r and ISU_d for ISMC and bSMC

The *RobustnessDelta* Level Diagram indicates that the ISMC controllers are in general far more robust to model error than the bSMC controllers.

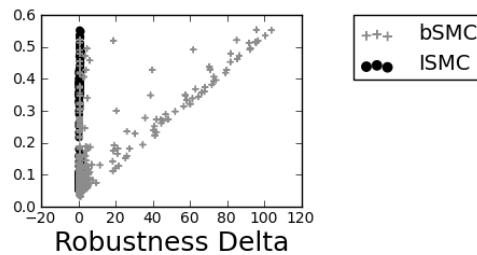


Figure 4.33: *RobustnessDelta* for ISMC and bSMC

The results from the Input Level Diagrams for the ISMC controller, re-drawn in Figure 4.34 are the same as those in the previous comparison (PI versus ISMC): Higher values

of λ are favoured as this achieves a faster sliding rate, Lower values of K are favoured, ϕ has no significant trade-off relationship with the ISE objectives or is over parametrised. One point of interest is the two groupings of controllers noted earlier: The cause of this can be seen in the λ Level Diagram. As λ sets the reaching rate of the SMC controllers, it directly affects the speed at which error is rejected; thus, lower values of λ will result in slower settling time and thus higher ISE .

What is of particular interest is the gap between the two groups in λ , which occurs as a result of the objectives used in the optimisation. The combined good ISE_r and ISE_d as well as very good $RobustnessDelta$ means that these controllers while slow are still on the Pareto front.

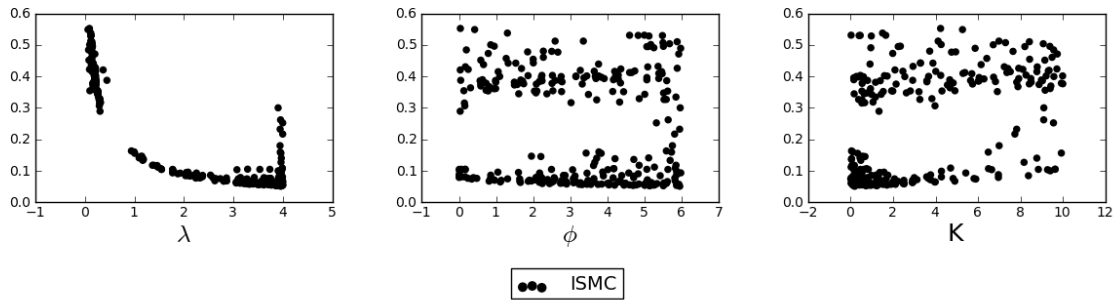


Figure 4.34: λ , ϕ and K for ISMC

The Input Level Diagrams for the bSMC controller re-drawn in Figure 4.35 are very troubling. Apart from a minor up curve of norm values for higher values of ϕ , there does not seem to be any trade-off relationship between any of the decision variables used in bSMC and the ISE costs. This means that the resulting Pareto front is unable to give any direction to designers as to what values should be preferred for any particular design. This could be as a result of two effects: Firstly that the input parameter has little effect on the chosen objectives, or secondly, that the controller is over-parametrised and the number of parameters used in the optimisation needs to be reduced, the obvious way

being to fix the K and/or BL parameters.

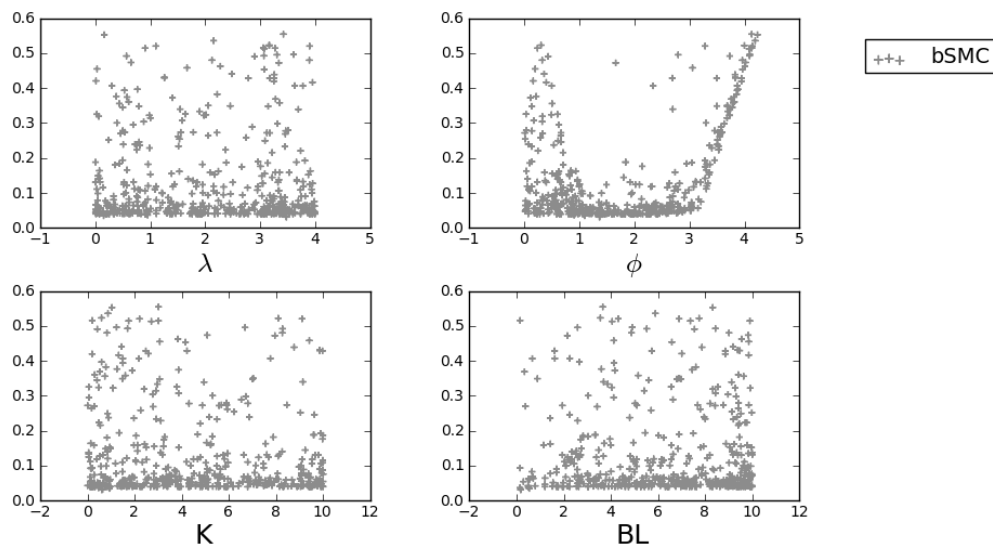


Figure 4.35: λ , ϕ , BL and K for bSMC

Figure 4.36 shows an example step test of the lowest norm controllers from each of the ISMC and bSMC control design methods using the ISE objectives.

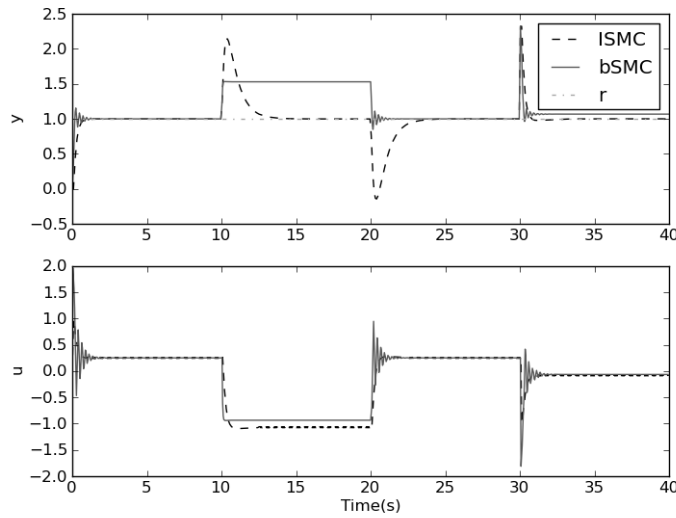


Figure 4.36: Lowest norm step tests for ISMC and bSMC for ISE objectives.

The step test reveals a significant problem with the bSMC method: it is incapable of rejecting input disturbances (at $t = 10$). This is of concern as the Level Diagrams do not reflect this in the ISE_v objective. According to the Level Diagrams, the lower norm grouping of the ISMC controller does comparably for input disturbances, but this does not agree with what is shown in the step tests. This indicates a weakness in the ISE objectives, that they cannot properly identify situations where the controller does not settle to the correct value, provided that value is reasonably close to the needed final value (such that the disturbance would be rejected).

When the input disturbance is removed (at $t = 20$), the bSMC controller settles back to the correct value much faster than the ISMC controller, showing the strength of the method when it comes to settling time.

For the output disturbance (at $t = 30$), both controllers settle comparably fast, but the bSMC controller does not settle to the correct value, there is still a finite offset error, though in this case it is far smaller than for the input disturbance. The larger ISU_d for bSMC is also clearly visible in the step test, resulting from a larger initial change

and some oscillation. Whereas the ISMC controller settles to its final value comparable quickly, with a smaller initial value and no oscillations.

4.3 Performance Measure Results

While the resulting Level Diagrams were presenting useful and informative analyses of the Pareto fronts, and thus their related controllers, it was found that it would be beneficial to quantify the results that were being seen in the Pareto fronts. Performance Measures were applied as a possible means to quantify the Pareto fronts and their results are discussed below

The three Quality Indicators I_H the unary hypervolume, I_{H2} the binary hypervolume and I_ϵ the binary ϵ indicators discussed in Section 2.5 were applied to the approximation sets (approximate Pareto fronts) resulting from the control design methods being compared.

Different setpoints r are used with the Performance Measures to validate the comparisons using Level Diagrams as well as to test for sensitivity based on the operating range.

The results are discussed in the following section based on the quality indicators and the comparison methods from Section 2.

Due to the statistical method by which I_H and I_{H2} are generated, some of the interpretations are dependent on the accuracy of their calculation and thus must be taken with some degree of caution.

4.3.1 PI versus ISMC

Table 4.1 shows the three performance measures comparing PI versus ISMC.

Using the interpretation functions described in Table 2.2 in Section 2, some observa-

<i>ISE</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.8425	0.3673	1.7483
ISM	0.4844	0.0092	20.6786
<i>r = 2</i>			
PI	0.8272	0.3650	2.1650
ISM	0.4711	0.0089	20.6438
<i>r = 5</i>			
PI	0.7953	0.3143	6.2444
ISM	0.4942	0.0131	20.7654
<i>r = 10</i>			
PI	0.8966	0.1230	8.5811
ISM	0.8023	0.0286	40.1438

<i>EST</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.6520	0.3200	inf
ISM	0.3460	0.0140	10.8333
<i>r = 2</i>			
PI	0.5130	0.3820	1.8196
ISM	0.1393	0.0083	9.1429
<i>r = 5</i>			
PI	0.4856	0.3601	3.0000
ISM	0.1406	0.0151	9.1429
<i>r = 10</i>			
PI	0.7556	0.2858	inf
ISM	0.5011	0.0312	10.6667

Table 4.1: Performance Measure Results for PI versus ISM

tions can be made.

According to dominance theory, the I_H results indicate that PI is not worse than ISM, $I_H(PI) > I_H(ISM)$. This holds true for both *ISE* and *EST* objectives. Considering the values from an engineering perspective, the I_H quality indicator is the hypervolume indicator: Thus, the results show that the PI controller dominates a larger portion of the possible objective space. This could be interpreted as having a larger chance of being able

to achieve a specific desired control scenario for this problem.

Both PI and ISMC cover a larger portion of the Region of Interest in the *ISE* costs than in the *EST* costs. Whether this indicates that it is harder to optimise the *EST* objectives or some other phenomena is unknown.

From the interpretation functions for I_{H2} in Table 2.2 the relationship \parallel applies. Meaning that the only result that can definitively be said is that the two sets are not comparable. This is due to $I_{H2}(PI) > 0, I_{H2}(ISMC) > 0 \rightarrow PI \parallel ISMC$. In other words, that neither controller is strictly better than the other.

The $I_{H2}(ISMC)$ is very near to zero. Considering that both the Pareto front and the method for determining the I_{H2} are approximate, it is possible that this value could really be zero, but due to systematic error it is not. This would allow far stronger dominance relations to be applied to the comparison.

From an engineering perspective, the difference between $I_{H2}(PI)$ and $I_{H2}(ISMC)$ is fairly significant. The PI control design method is able to produce a large portion of controllers that the ISMC control design method is not able to. While the ISMC control design method can only possibly produce a few solutions that are better than the PI control design method. The Level Diagram analysis would point towards these better solutions being because of their smaller *RobustnessDelta* (see Figure 4.13).

Finally, the I_ϵ quality indicator also indicates a situation where the control methods are incomparable. The ϵ -dominance indicator shows by how much each Pareto front would need to be scaled by in order for it to completely dominate the other front. Thus, it is still possible to infer some information from the measures.

The *ISE* performance measures the PI ϵ -indicator is smaller than the ISMC indicating that less scaling would be needed to dominate the ISMC Pareto front.

The *EST* performance measures are concerning in that the PI I_ϵ indicator is *inf* for two of the setpoint ranges. Investigation revealed that this is an inherent problem with

the way the indicator is formulated: It results for a situation when one of the objectives is zero, and cannot be scaled properly.

4.3.2 PI versus PI Pole-Zero Cancellation

Table 4.2 shows the three performance measures comparing PI versus PI pole-zero cancellation.

<i>ISE</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.7716	0.0712	4.8018
PI Cancel	0.7072	0.0067	2.7873
<i>r = 2</i>			
PI	0.7108	0.0820	2.4900
PI Cancel	0.6334	0.0046	2.7826
<i>r = 5</i>			
PI	0.7191	0.0878	1.5712
PI Cancel	0.6359	0.0045	2.7426
<i>r = 10</i>			
PI	0.9013	0.0571	3.3675
PI Cancel	0.8527	0.0084	5.3812
<i>EST</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.4224	0.1619	1.8511
PI Cancel	0.3143	0.0538	6.5714
<i>r = 2</i>			
PI	0.4697	0.2095	1.9317
PI Cancel	0.3041	0.0439	6.5714
<i>r = 5</i>			
PI	0.4352	0.1980	2.0838
PI Cancel	0.2763	0.0391	6.5714
<i>r = 10</i>			
PI	0.3692	0.1482	1.6888
PI Cancel	0.2377	0.0168	6.5714

Table 4.2: Performance Measure Results for PI versus PI Pole-Zero Cancellation

Using the interpretation functions described in Table 2.2 in Section 2, some observations can be made.

Firstly, looking at the *ISE* results: The $I_H(PI)$ is generally larger than the $I_H(PICancel)$, indicating that the PI set is not worse than the pole-zero cancellation set.

The I_{H2} results are interesting in that they again highlight the limitations caused by using approximate methods. The $I_{H2}(PI)$ is larger than the $I_{H2}(PICancel)$ for all setpoints. The $I_{H2}(PICancel)$ is close enough to zero that questions of statistical accuracy come into play. Assuming that the PI Cancel indicator is actually zero would mean that the dominance relationship \triangleright would hold, indicating that the optimal PI Pareto front is better than the PI pole-zero cancellation Pareto Front.

This result is for the most part supported by the I_ϵ results. The requirement $I_\epsilon(A, B) \leq 1$ is needed to infer any relationship greater than not comparable. While this is not true in any of the setpoint cases, the $I_\epsilon(PI)$ is smaller than the $I_\epsilon(PICancel)$, except in the case $r = 1$.

This first case being different is concerning, because it disagrees with all indications from the other quality indicators and Level Diagrams. A probable cause for this difference is that the region of interest being applied is causing parts of the front to be excluded that would allow for easier scaling. This does reveal another weakness in the I_ϵ quality indicator.

The *EST* objective results are more conclusive: Similar to the PI versus ISMC comparison, this I_H occupies a smaller portion of the region of interest.

The I_{H2} shows that a large portion of the generated PI hypervolume is unique to it, while the PI pole-zero cancellation has approximately zero I_{H2} .

Finally, the I_ϵ indicator shows a situation where in all cases, the PI Pareto front requires significantly less scaling than the PI Pole-Zero Cancellation set.

From this one may pragmatically infer that the PI controller is better than the PI

Pole-Zero Cancellation set.

4.3.3 Integrated SMC versus Boundary Layer SMC

Table 4.3 shows the three performance measures comparing ISMC versus bSMC.

There are no Performance Measures for the *EST* results. bSMC does not perform well according to these objectives: this is primarily due to bSMC's inability to reject input disturbances. This is as a result of the constant offset error inherent in the bSMC formulation (see Sections 2.1.2), and the way that the *EST* objective is calculated.

<i>ISE</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
ISMC	0.8186	0.0137	7.8524
bSMC	0.9336	0.1287	18.7868
<i>r = 2</i>			
ISMC	0.7183	0.0099	21.5277
bSMC	0.9165	0.2082	5.9874
<i>r = 5</i>			
ISMC	0.8557	0.0122	11.1256
bSMC	0.9489	0.1054	18.9131
<i>r = 10</i>			
ISMC	0.9611	0.0085	25290.7514
bSMC	0.9845	0.0319	20.6066

Table 4.3: Performance Measure Results for ISMC versus bSMC

Using the interpretation functions described in Table 2.2 in Section 2, some observations can be made.

The first result of note is that the results vary greatly depending on the setpoint. This is primarily due to the norm based region of interest being applied, but also due to variation in the two non-linear control methods caused by the different setpoints.

In general, the bSMC controller has a higher I_H than the ISMC, except at $r = 10$ where they have almost equal hypervolume. The engineering interpretation of which is

that bSMC design method is able to handle more of the possible desired control situations than the ISMC, except at extreme operating ranges.

The ISMC control design method has a much lower I_{H2} than the bSMC design method generally, indicating that bSMC can achieve most of the control situations that ISMC can, excepting at $r = 10$ where they are almost equal.

The I_ϵ quality indicator changes greatly between different operating ranges, and in all cases indicates that the two methods are not comparable.

University of Cape Town

CHAPTER 5

Other Plant Results

This chapter investigates how various other plants change the resulting Level Diagrams and Performance Metrics for the PI, ISMC and bSMC controllers.

Two other plants were investigated: a plant with input saturation, and a plant with parasitic dynamics.

5.1 Input Saturating Plant

The first non-linear plant has an input that saturates at ± 10 [Volts]. This is a realistic limitation based on the digital to analogue converter used to create the input signal.

5.1.1 PI versus ISMC

As with the first order results, the PI plant is used as a base comparison method; the pole-zero cancellation set is included for visual and performance metric comparisons.

5.1.1.1 Selection of Setpoint

The input saturation non-linearity created a problem in terms of selecting an appropriate setpoint for the simulation. The setpoint directly affects how much saturation will occur: a higher setpoint typically resulting in more input saturation.

Figures 5.1 and 5.2 show the resulting objective and decision *ISE* Level Diagrams for a setpoint of 1 ($r = 1$).

When compared to the Level Diagrams for the first order example in Section 4 (See Figures 4.11 and 4.12 on page 47), there is almost no difference in the resulting Level Diagrams.

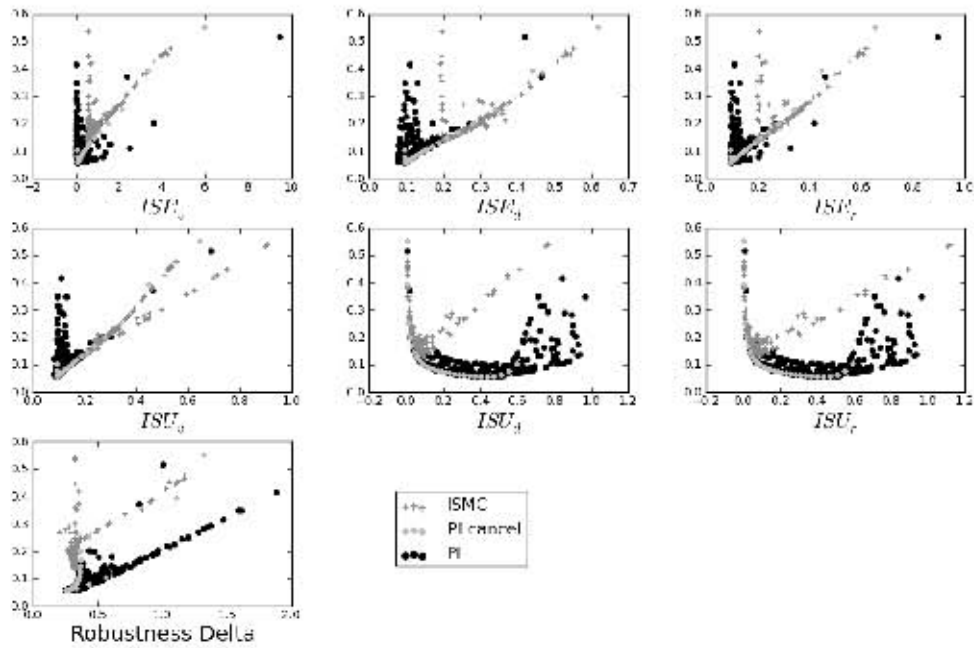


Figure 5.1: $\|\cdot\|_2$ *ISE* Cost Level Diagrams for PI, PI Cancel and ISMC

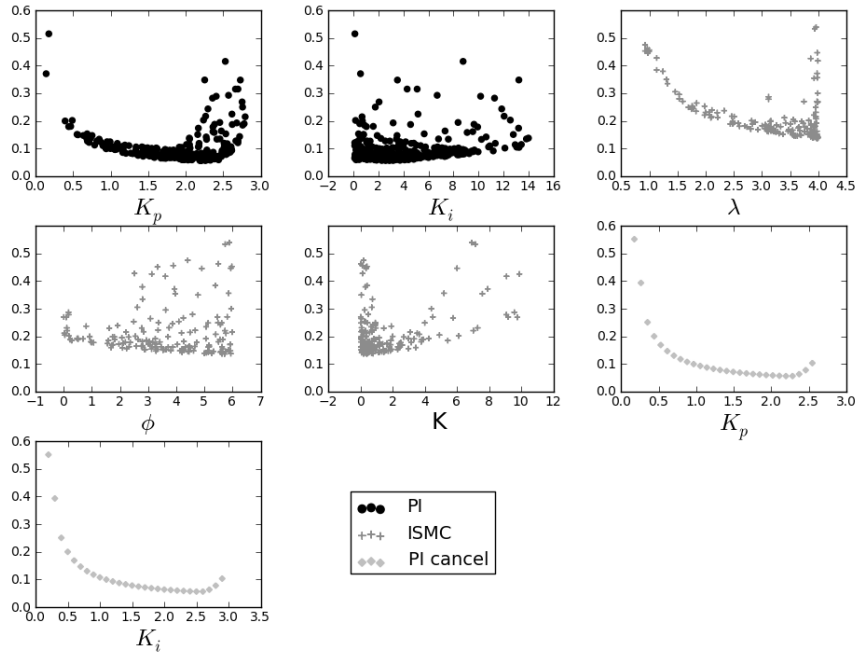


Figure 5.2: $\|\cdot\|_2$ ISE Input Level Diagrams for PI, PI Cancel and ISMC

This is primarily due to the fact that at this low setpoint, there is little significant saturation occurring. Thus for the following sections, a setpoint of $r = 10$ is used in order to force more saturation to occur. Furthermore, it was necessary to increase the magnitude of the disturbance (both input and output) such that they also experienced saturation effects.

5.1.1.2 Integral Square Error Level Diagrams

Figures 5.3 and 5.4 show the $\|\cdot\|_2$ Cost and Input Level Diagrams using the Integral Square Error costs.

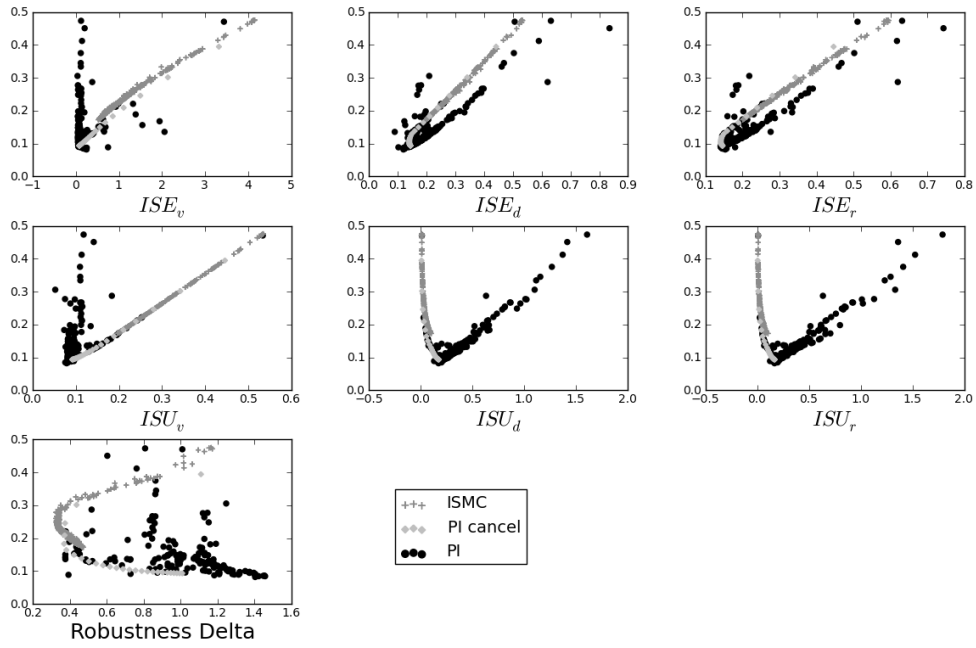


Figure 5.3: $\|\cdot\|_2$ ISE Cost Level Diagrams for PI, PI Cancel and ISMC

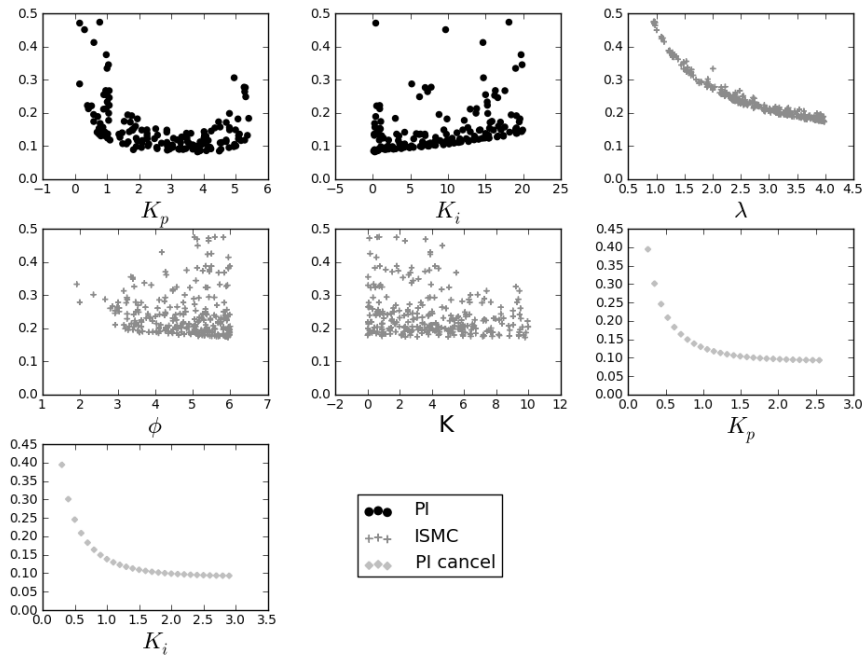


Figure 5.4: $\|\cdot\|_2$ ISE Input Level Diagrams for PI, PI Cancel and ISMC

The first thing to note is that the Pareto fronts in Figures 5.3 and 5.4 are far more sparse than those seen in Chapter 4. This shows that a far smaller portion of the total Pareto front falls below the region of interest cut off ($\text{norm} > 0.56$).

Also, it is interesting that the fronts appear to be less spread out in the various Cost Level Diagrams, one possible cause for this is that (due to the input saturation), the Pareto fronts are far more constrained than in the first-order example.

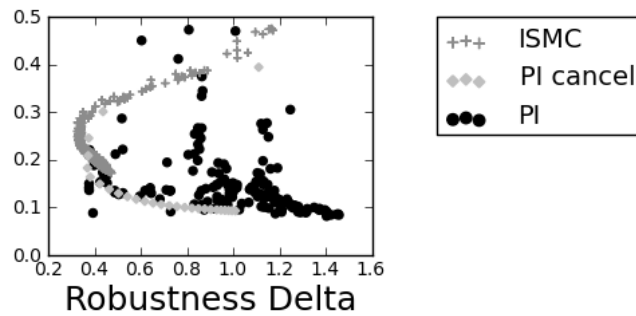


Figure 5.5: *RobustnessDelta* for PI, PI Cancel and ISMC

The *RobustnessDelta*, shown in Figure 5.5, is probably the most interesting of the *ISE* cost Level Diagrams for the input saturation example. The PI cancellation set does fairly well in terms of robustness compared to the PI set. While the ISMC controller manages, for the most part, to be the most robust. The ISMC has an odd shape, in that its most robust controllers are those in the middle of the norm range, while the low and high norm controllers, corresponding to the extremes on the other costs do worse.

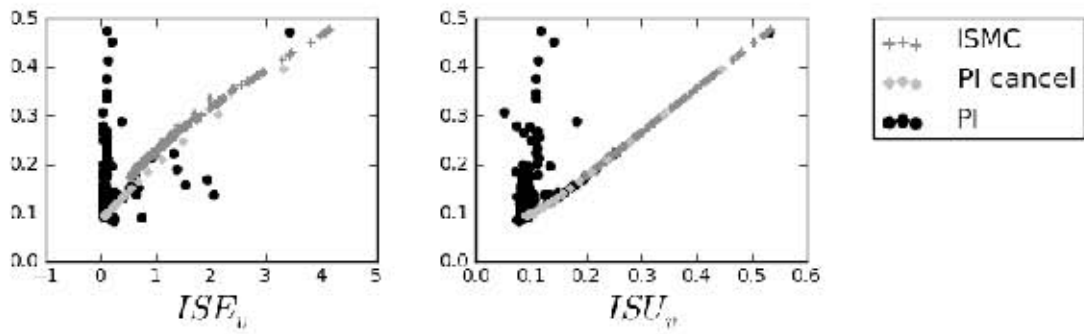


Figure 5.6: ISE_v and ISU_v for PI, PI Cancel and ISMC

The ISE_v and ISU_v Level Diagrams in Figure 5.6 show a familiar result to those of the first-order example: The PI controller achieves faster input disturbance rejection with lower ISE_v , with lower controller effort ISU_v than the ISMC controller. In this case, the PI cancellation set performs significantly worse than the PI controllers.

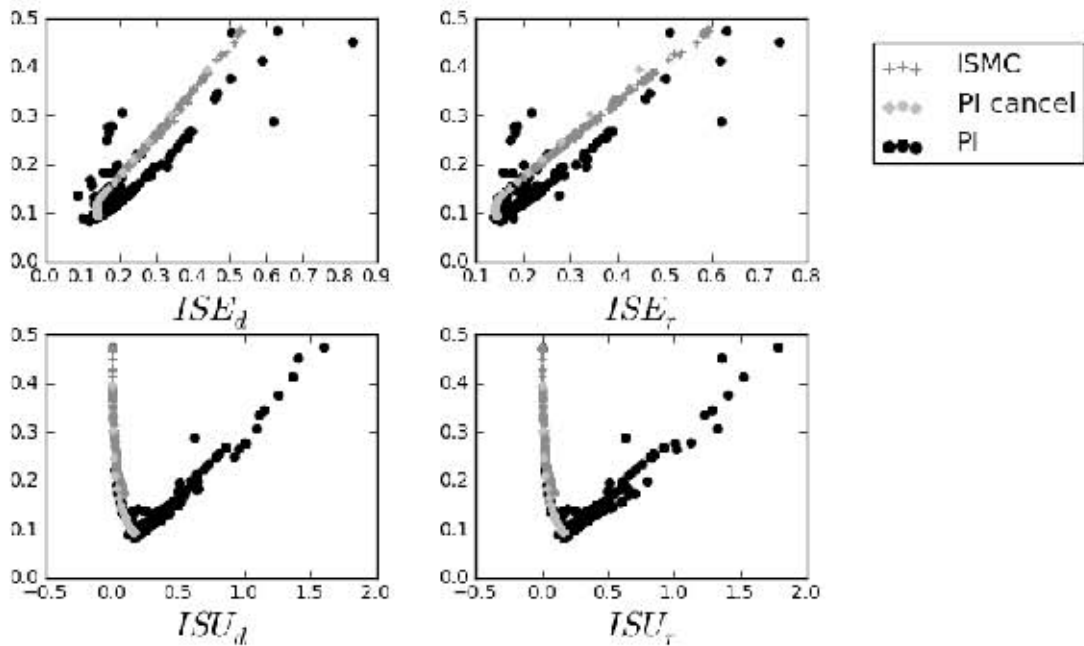


Figure 5.7: ISE_d , ISU_d , ISE_r and ISU_r for PI, PI Cancel and ISMC

The ISE_r , ISE_d , ISU_r , and ISE_d Level Diagrams, repeated in Figure 5.7, shows a similar result to those of the first-order example: In general, the ISMC controller uses significantly less controller effort to achieve its results than the PI controller, but achieves worse setpoint and output disturbance rejection. Though, in this case, the difference in ISE_r and ISE_d between PI and ISMC is smaller with many of the ISMC controllers achieving comparable results.

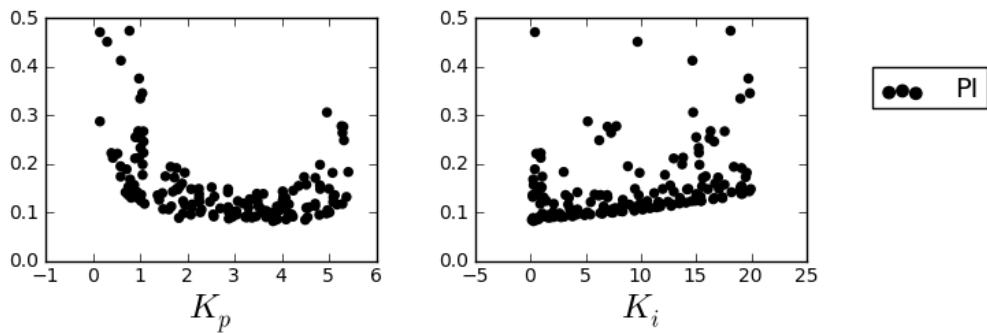


Figure 5.8: Separate Inputs: PI

The K_p and K_i Input Level Diagrams for the PI controller shown in Figure 5.8 show lower values of K_i being favoured. The K_p values are over a larger range than in the first-order example, with a minimum centred on $K_p = 4$.

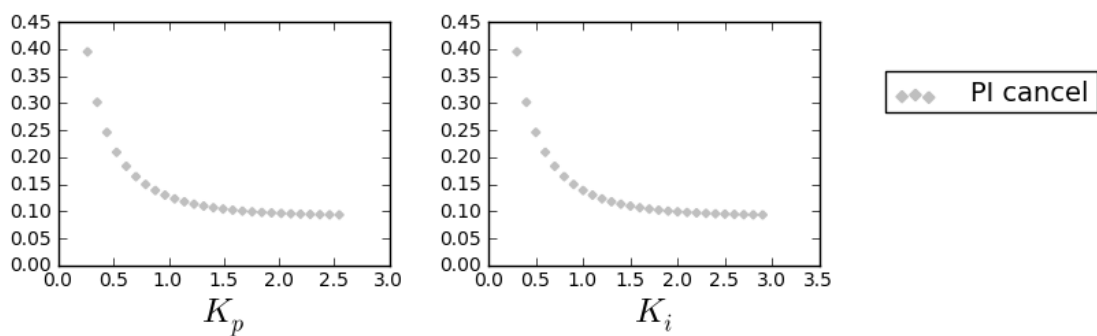


Figure 5.9: Separate Inputs: PI cancel

The PI cancellation set is interesting in that while lower gain controllers do badly, after a certain point the higher gain controllers perform equally well in terms of their norms, though it should be noted, they become less robust with higher gain.

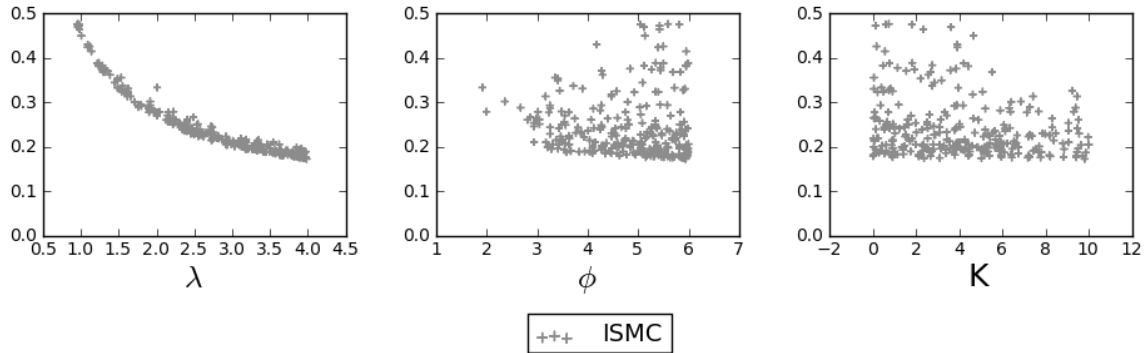


Figure 5.10: Separate Inputs: ISMC

Finally, the ISMC controller has a similar result for λ , high values being favoured due to the resulting faster dynamics, and thus lower ISE values. Interestingly, it is the ϕ values show a trend to higher values for input saturation, while the K values are now flat, indicating either a lack of effect on the chosen objectives or a over-parametrisation for the problem.

Figure 5.11 shows an example step test of the lowest norm controllers from each of the PI and ISMC control design methods using the ISE objectives.

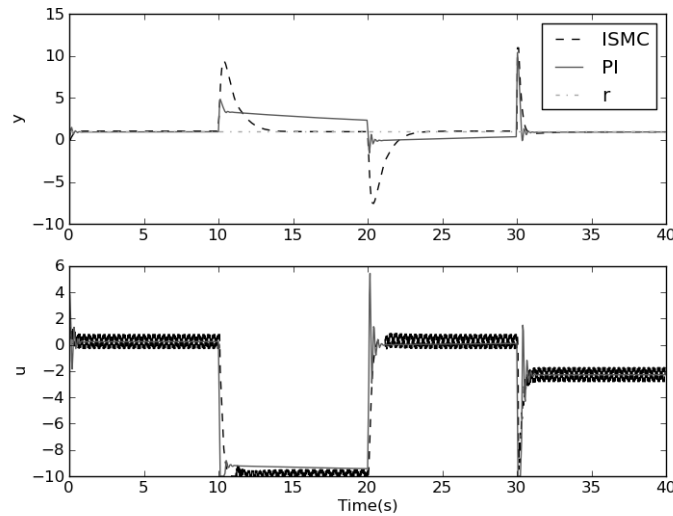


Figure 5.11: Lowest norm step tests for PI and ISMC for ISE objectives.

The step test reveals the same problem as with the ISE comparison of ISMC and bSMC in Chapter 4: The ISE objectives allow the system not to settle within the simulation time frame for input disturbances (as seen from $t = 10 \rightarrow 20$ and $t = 20 \rightarrow 30$). This is clearly seen here again where the lowest norm PI controller does not settle within the allowed 10 second time frame, but it still achieves a good ISE_v . In this case the step test reveals a situation where one could argue that the ISMC controller does significantly better than the PI controller.

For both the setpoint tracking (at $t = 0 \rightarrow 10$) and the output disturbance rejection (at $t = 30 \rightarrow 40$), both controllers perform equally quickly. The ISMC controller has significant chatter on its input, even with the integrator, this is as a result of the large K value of 9.8, yet a similar result could be achieved with far smaller K and thus far less or even no chatter as can be seen by the input Level Diagram for K .

5.1.1.3 Error Settling Time Level Diagrams

Figures 5.12 and 5.13 show the $\|\cdot\|_2$ Cost and Input Level Diagrams using the Error Settling Time costs.

Immediately it is apparent that the PI controller's Pareto front is extremely sparse, while the ISMC controller's is not. This is a result of the region of interest applied to the problem. While there are some PI controllers that fall below the 0.56 norm cut off, they are few in number compared to the ISMC.

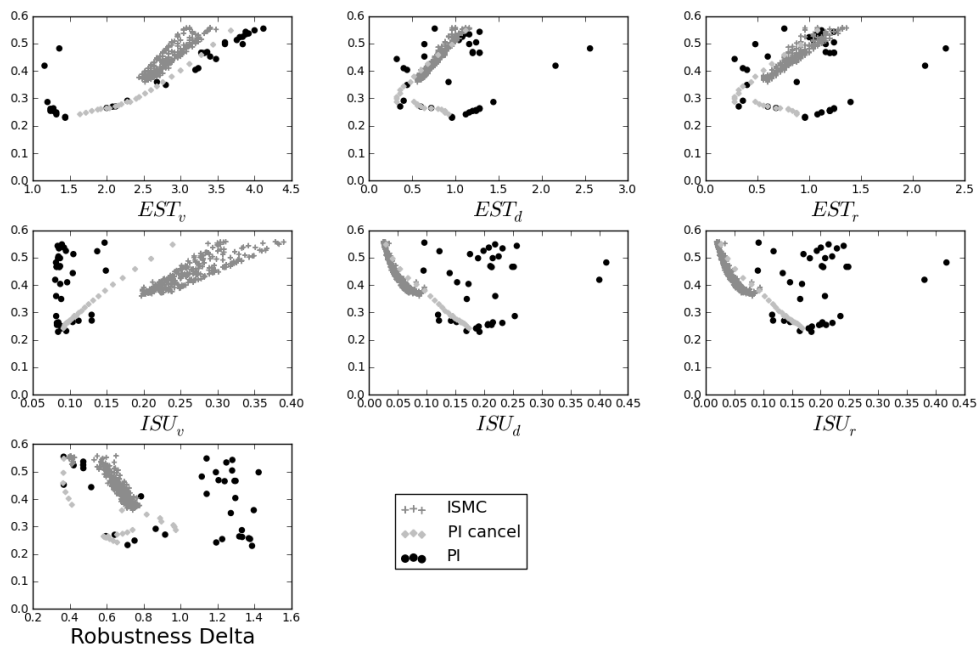


Figure 5.12: $\|\cdot\|_2$ *EST* Cost Level Diagrams for PI, PI Cancel and ISMC

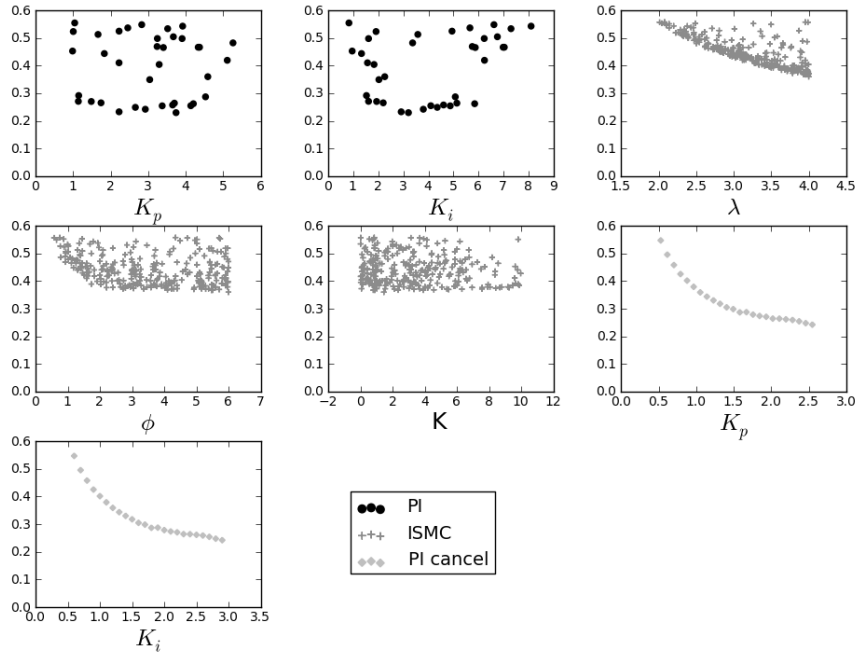


Figure 5.13: $\|\cdot\|_2$ EST Input Level Diagrams for PI, PI Cancel and ISMC

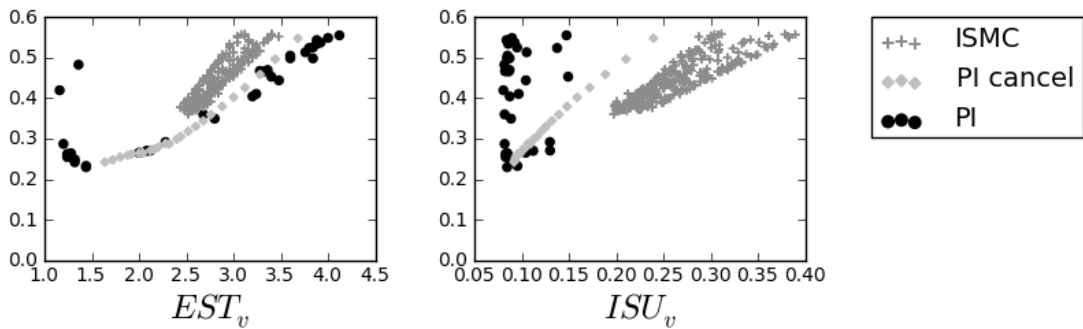


Figure 5.14: EST_v and ISU_v for PI, PI Cancel and ISMC

The EST_v and ISU_v Level Diagram, repeated in 5.14, show that the PI controller uses less controller effort than the ISMC or most of the PI cancel set, with some of the resulting controllers achieving better input disturbance rejection than the ISMC controllers.

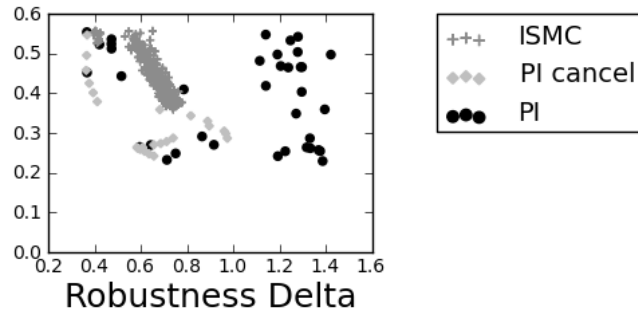


Figure 5.15: *RobustnessDelta* for PI, PI Cancel and ISMC

In terms of *RobustnessDelta*, all of the controllers do fairly well, with the highest value of 1.4, meaning that the controllers were able to deal fairly well with model inaccuracy.

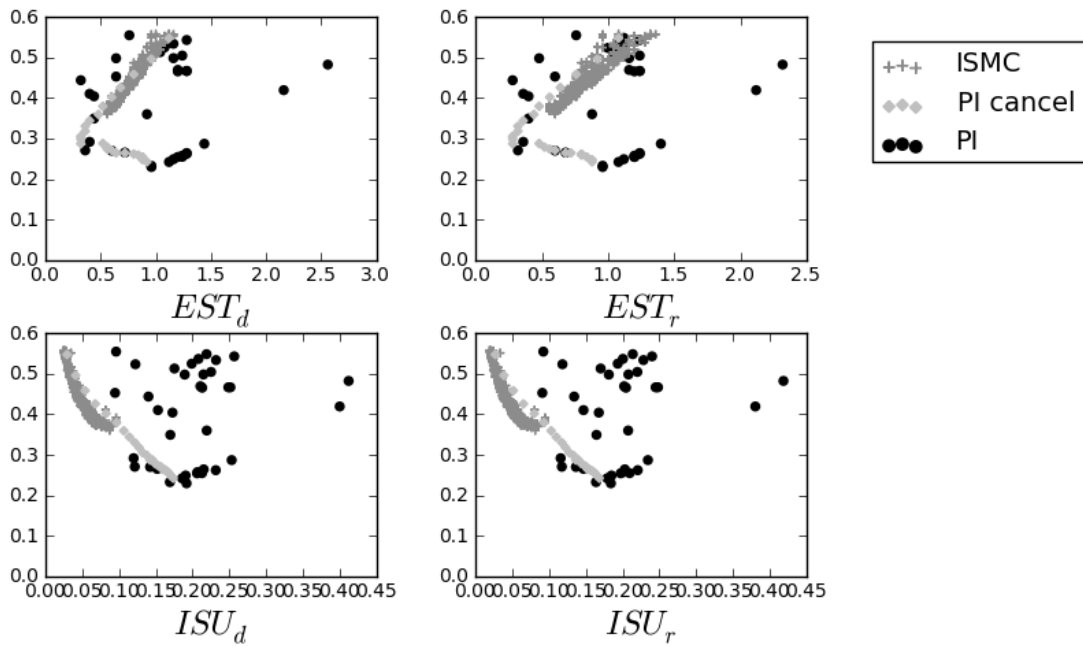


Figure 5.16: EST_d , ISU_d , EST_r and ISU_r for PI, PI Cancel and ISMC

The EST_r , ISU_r , EST_d and ISU_d Level Diagrams shown in Figure 5.16, have the ISMC controller using less controller effort for both setpoint tracking and output disturbance rejection than the PI controller, while doing comparably in both the EST_r and EST_d .

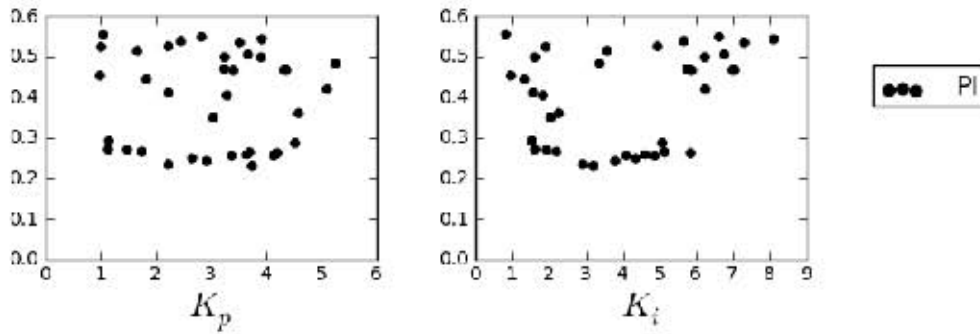


Figure 5.17: K_p and K_i for PI

Due to the sparsity of points in the K_p and K_i Level Diagrams for the PI controller it is not possible to make any comment on their results.

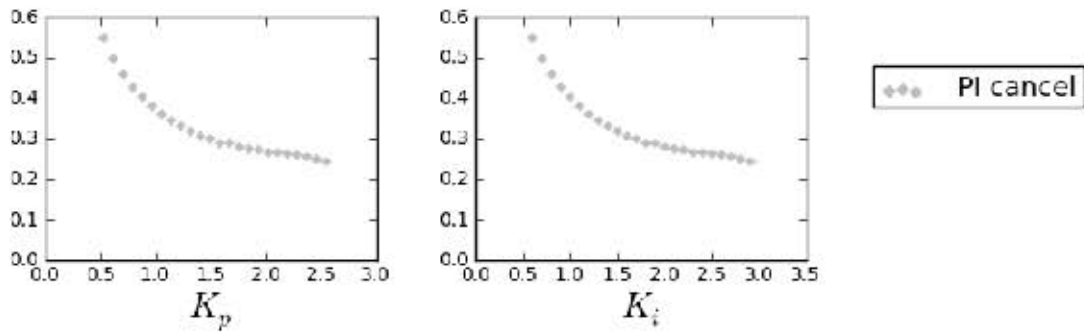


Figure 5.18: K_p and K_i for PI cancellation set.

The PI cancellation set's Input Level Diagrams, shown in Figure 5.18, show the same preference for higher gain controllers as in the ISE objectives.

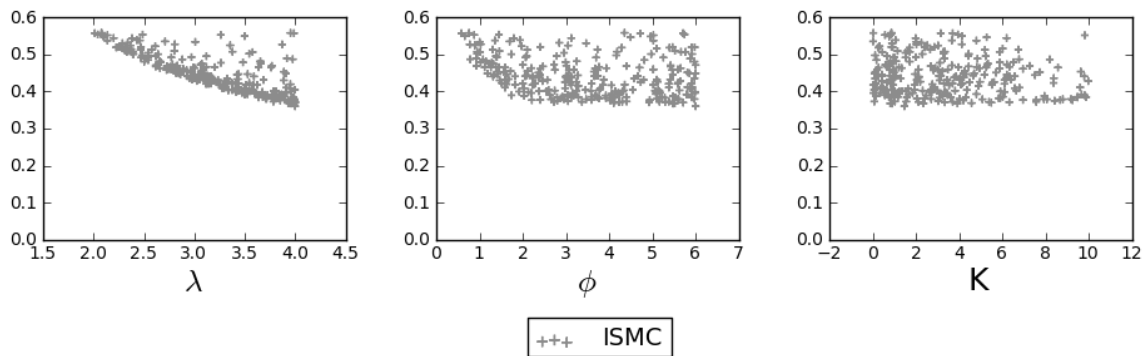


Figure 5.19: λ , ϕ and K for ISMC

The λ Level Diagram shows that higher values are favoured as expected. The K Level Diagram shows a slight trend to lower values, but is otherwise flat indicating either a lack of effect on the chosen objectives or over-parametrisation of the problem. The ϕ values are similarly flat, with the same implications as for K , except for a slight upturn of the synchronising norm for lower value of ϕ . This could be attributed to the fact that ϕ sets the sliding rate dynamic of the ISMC controller, which has not seemed to play a significant role in the previous examples, but could, due to the restriction placed upon it by the input saturation be causing slower settling time and thus higher norms.

Figure 5.20 shows an example step test of the lowest norm controllers from each of the PI and ISMC control design methods using the *EST* objectives.

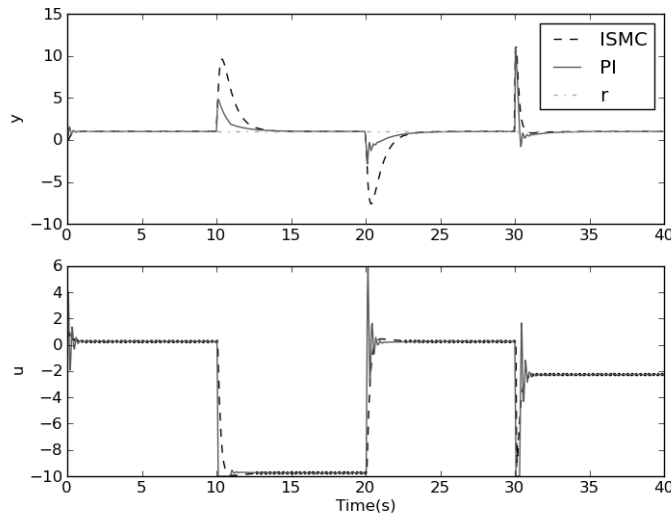


Figure 5.20: Lowest norm step tests for PI and ISMC for *EST* objectives.

As with the other example step tests, the ISMC controller performs almost exactly the same. This is reflected in the input variables, where the minimum norm almost always corresponds to a high λ and ϕ value. This makes sense, as both these parameters determine the system dynamics of the ISMC controller. It raises a question regarding the control setup used in the thesis, specifically the sampling time used; the sampling time sets the limit for λ and ϕ (as discussed in Section 2.1), so if it is too slow, the ISMC is effectively limited to a minimum possible response time, no matter how much the optimiser would like to make it go faster. This could then indicate that ISMC is unsuitable for use with this system, unless the sampling time could be improved.

The PI controller has far greater input overshoot than the ISMC controller, which considering the input limitation is a disadvantage at the extremes of the operating range. For the input disturbance, the effect of the saturation can be seen at $t = 10 \rightarrow 20$ where the PI's u saturates, meanwhile when the input disturbance is removed at $t = 20$, significant overshoot is present and the system settles marginally faster.

This step test for PI is especially interesting when compared to the previous step test

for the *ISE* objectives for the input disturbance. In this case the *EST* objectives have forced the optimiser to find solutions that actually settle in less than the maximum time (versus the *ISE* where they did not actually settle). While the optimiser seems to have found it far harder to find good solutions (seen from the sparsity of points on the PI Level Diagrams), the PI controllers resulting from the *EST* objectives are more in line with engineering desires (such as fast settling time), based on the two lowest norm step tests for the input saturation problem.

5.1.1.4 Performance Measures - PI vs ISMC

Table 5.1 shows the three performance measures comparing PI versus Integrated SMC for the input saturation example.

University of Cape Town

<i>ISE</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.8276	0.3580	2.5301
ISM	0.4792	0.0097	20.5799
<i>r = 5</i>			
PI	0.6829	0.4140	2.4198
ISM	0.2797	0.0107	21.7635
<i>r = 10</i>			
PI	0.8664	0.3353	2.0225
ISM	0.5283	0.0	inf
<i>r = 15</i>			
PI	0.8664	0.2147	3.5153
ISM	0.6552	0.0035	inf
<i>EST</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.6882	0.3245	inf
ISM	0.3777	0.0140	10.5000
<i>r = 5</i>			
PI	0.5492	0.2520	inf
ISM	0.3227	0.0255	10.5000
<i>r = 10</i>			
PI	0.7636	0.3186	nan
ISM	0.4538	0.0089	nan
<i>r = 15</i>			
PI	0.8737	0.0682	48.0000
ISM	0.8138	0.0083	inf

Table 5.1: Performance Measure Results for PI versus ISMC

Using the interpretation functions described in Table 2.2 in Section 2, some observations can be made.

The I_H quality indicator applied to both *ISE* and *EST* Pareto fronts, shows a similar trend to that of the first order example. In each case (excepting *EST* $r = 15$), $I_H(PI)$ is greater than $I_H(ISMC)$, indicating that the PI controller's Pareto front is at least,

not worse than the ISMC Pareto front. In engineering terms, the PI controller is able to design optimal controllers that dominate more of the possible objective space, or in other words, is more likely to be able to achieve any given control requirements. The final set, *EST* $r = 15$, shows the case at the extreme, when both controllers are running into significant input saturation. In this case, both fronts I_H are similar as there is little that either controller can do.

In terms of the I_{H2} quality indicator, the PI controller has significantly more unique Pareto hypervolume. In terms of the interpretation functions the only dominance relation that can strictly be applied is that $PI \parallel ISMC$, that is PI is incomparable to ISMC. In terms of the accuracy of the Pareto fronts, it could be possible that the $I_{H2}(ISMC, PI)$ is close enough to zero that it could be zero. If this is the case, then the interpretation function $I_{H2}(PI, ISMC) > 0$ and $I_{H2}(ISMC, PI) = 0$ could be applied, implying that PI is better than ISMC in a Pareto sense. If this were the case, from an engineering perspective, one could infer that for the chosen objectives and optimisation, the PI controller is able to match or better the optimal ISMC controllers. The *ISE* $r = 10$ set shows an interesting result in terms of the I_{H2} . The $I_{H2}(ISMC, PI) = 0$, which, combined with the $I_{H2}(PI, ISMC)$ and interpretation functions, allows us to imply strictly that the PI front is better than the ISMC front. The value 0.0 is a result of rounding, as the actual value was negative (-0.0007), resulting from the random nature of the calculation method. This highlights the accuracy issue when applying interpretation functions using a statistical I_H calculation method.

The I_ϵ quality indicator returns a mixed bag of results. One would expect for the I_H and I_{H2} results that the $I_\epsilon(PI, ISMC)$ would be close to or less than 1, and that the $I_\epsilon(ISMC, PI)$ would be greater than one.

The *ISE* Pareto fronts return almost what is expected, except for the $r = 10$ and $r = 15$, where the $I_\epsilon(ISMC, PI)$ is *inf* in both cases. This occurs as a result of the

quality indicator formulation when there is a zero objective in one of the fronts. While it is still the relative difference between the scaling required for either front to dominate the other, it is not a particularly useful value.

The *EST* Pareto fronts present a more troubling case. The other quality indicators indicate that the PI Pareto front is close to, if not better than the ISMC Pareto front. but in two of the *EST* sets ($r = 1$, $r = 5$), this is not the case. Furthermore, the $r = 10$ case shows what happens when there are zero value objectives in both fronts: No comparison can be made. The final set, $r = 15$, which is the closest matched of the sets still indicates a case where neither set is close to dominating the other.

5.1.1.5 Performance Measures - PI vs PI Pole-Zero Cancellation

Table 5.2 shows the three performance measures comparing PI versus the Pole-Zero Cancellation set for the input saturation example.

<i>ISE</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.7193	0.0909	1.9271
PI Cancel	0.6340	0.0057	2.7717
<i>r = 5</i>			
PI	0.6111	0.0796	1.5651
PI Cancel	0.5468	0.0153	2.9319
<i>r = 10</i>			
PI	0.8370	0.0829	7.7001
PI Cancel	0.7581	0.0041	inf
<i>r = 15</i>			
PI	0.6840	0.3840	nan
PI Cancel	0.2975	0.0	nan

<i>EST</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.4371	0.1639	1.4286
PI Cancel	0.3066	0.0334	6.5714
<i>r = 5</i>			
PI	0.3696	0.1323	1.5926
PI Cancel	0.2905	0.0531	6.5714
<i>r = 10</i>			
PI	0.7070	0.0688	1.7725
PI Cancel	0.6463	0.0081	inf
<i>r = 15</i>			
PI	0.8655	0.0166	1.7553
PI Cancel	0.8684	0.0195	inf

Table 5.2: Performance Measure Results for PI versus PI Pole-Zero Cancellation

Using the interpretation functions described in Table 2.2 in Section 2, some observations can be made.

Firstly, the obvious observation is that the magnitude of each set varies greatly depending on the setpoint. This is a phenomenon resulting from the application of the region of interest (norm cut off), and exactly where the Pareto front is cut off as a result.

Odd outlying points can cause the total volume of the bounding box used to calculate the I_H (see Appendix D) to vary, and thus the percentage volume can fluctuate greatly. Thus it is important to compare the difference keeping this fluctuation in mind.

For the I_H quality indicator, each set shows a similar result (expecting the ISE $r = 15$ set), $I_H(PI) > I_H(PICancel)$. This indicating that the PI Pareto fronts are not worse than the PI Cancel fronts, though the difference between these measures is rather small compared to other fronts comparisons (such as Pi vs ISMC). The ISE $r = 15$ set has a far larger difference between the $I_H(PI)$ and $I_H(PICancel)$, possibly indicating possibly that at this extreme operation point, the optimiser is able to do significantly better than the PI Pole-Zero Cancellation set.

The I_{H2} quality indicator is interestingly different for ISE and EST optimisations. For the ISE optimisation sets, $I_{H2}(PI, ISMC)$ is greater than $I_{H2}(ISMC, PI)$, with $I_{H2}(ISMC, PI)$ close to zero (and being zero for the $r = 15$ set). As with previous comparisons, the size of the $I_{H2}(ISMC, PI)$ raises the possibility that due to the accuracy PI could be better than ISMC. The $r = 15$ case again reveals a situation where the optimal PI controllers are better than the PI Pole-Zero Cancellation set, showing that at these extremes, pole-zero cancellation is not optimal.

The EST optimisation sets show similar results, except that for $r = 15$, the two fronts are almost matched. Further investigation revealed that this is caused by a limitation in the calculation of the EST measure. In order to make it possible to simulate all the step-responses used for the optimisation, the length of the simulation was limited. Thus, at this extreme operating range, the controllers are not settling within the maximum time allowed.

The last quality indicator I_ϵ has a similar result for each front, (excluding those with *inf* and *nan* conditions). The $I_\epsilon(PI, PICancel)$ is close to 1, while the $I_\epsilon(PICancel, PI)$ is larger. This indicates that the PI Pareto fronts are close to dominating the PI Pole-Zero

Cancellation set. While the reverse would require far more scaling to be true. The same *inf* and *nan* conditions occur when zero objectives occur in the various sets.

5.1.2 ISMC versus bSMC

5.1.2.1 Integral Square Error Level Diagrams

Figures 5.21 and 5.22 show the $\|\cdot\|_2$ Cost and Input Level Diagrams using the Integral Square Error costs.

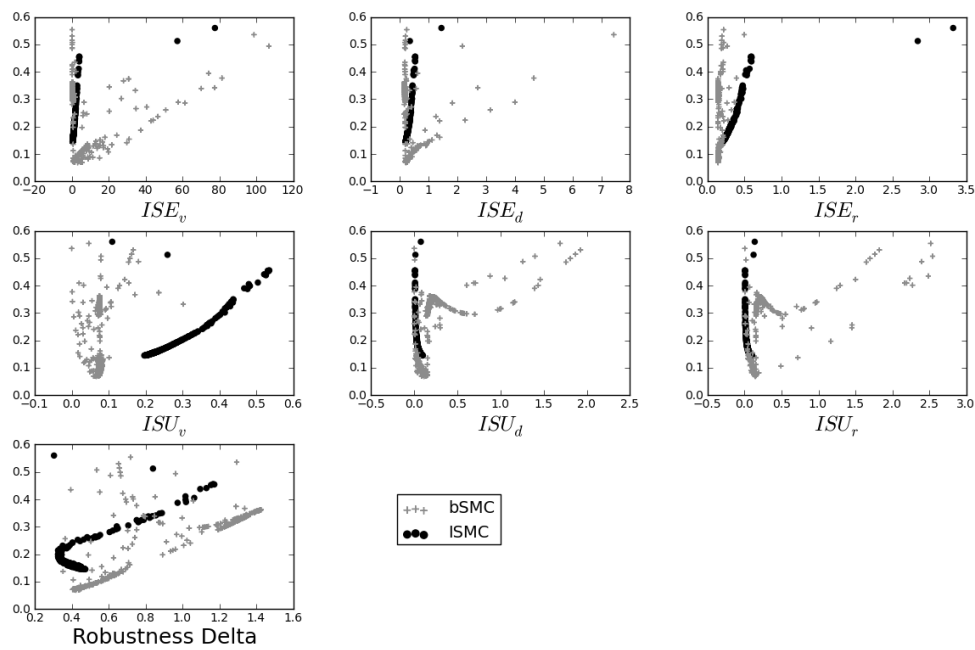


Figure 5.21: $\|\cdot\|_2$ ISE Cost Level Diagrams for ISMC and bSMC

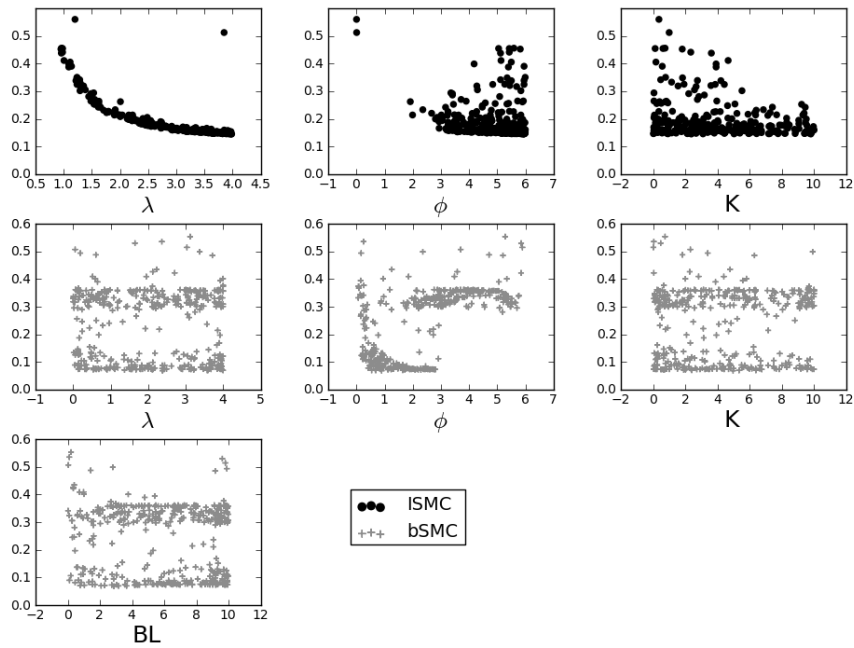


Figure 5.22: $\|\cdot\|_2$ ISE Input Level Diagrams for ISMC and bSMC

The first observation from the input saturation Level Diagrams is that visually they differ greatly from the first order example. But on close examination, there are only minor differences in the conclusions that can be drawn from them. The greatest difference being the region of interest cut off: Where in the first order example, the ISMC Pareto front is split into two groups, there is only one group and a couple of outlying points in the input saturation example. This is due to the upper grouping being cut off because their norms are, in this case, above the Region of Interest cut off point.

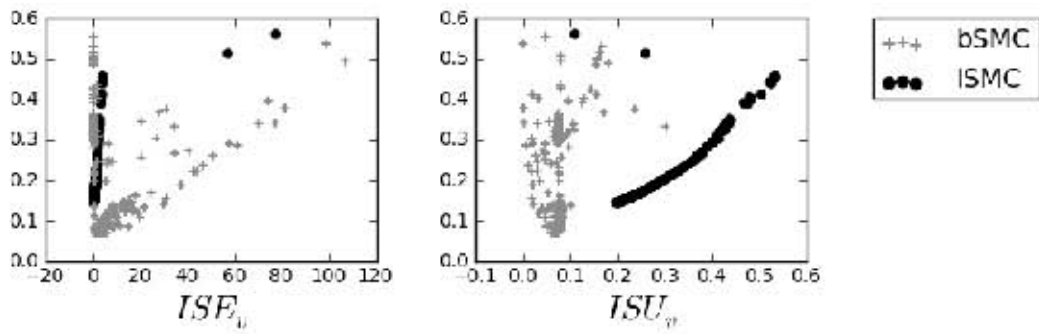


Figure 5.23: ISE_v and ISU_v for ISMC and bSMC

Figure 5.23 shows the ISE_v and ISU_v Level Diagrams. The ISMC controller is in general able to match or better the input disturbance rejection of bSMC, but does so at far higher controller effort. The two outlying points are interesting in that they seem to respond differently to the rest of the controllers, the reason for this is shown later on the input Level Diagrams.

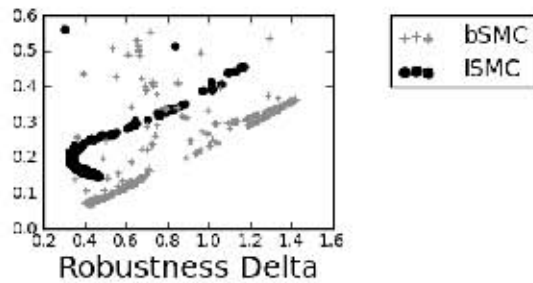


Figure 5.24: $RobustnessDelta$ for ISMC and bSMC

The $RobustnessDelta$ Level Diagram, in Figure 5.24, shows a different situation to the first order case. The two control design methods perform fairly similarly to one another. It is interesting to note, that while Level diagram for ISMC is a smooth increase curve, indicating a fairly consistent relationship between $RobustnessDelta$ and the synchronising

norm, the situation is very different for bSMC. Here we see that while there is a general trend to increasing norm with increasing *RobustnessDelta*, there are many of the points on the Pareto front that have norms that are significantly higher and off the trend line.

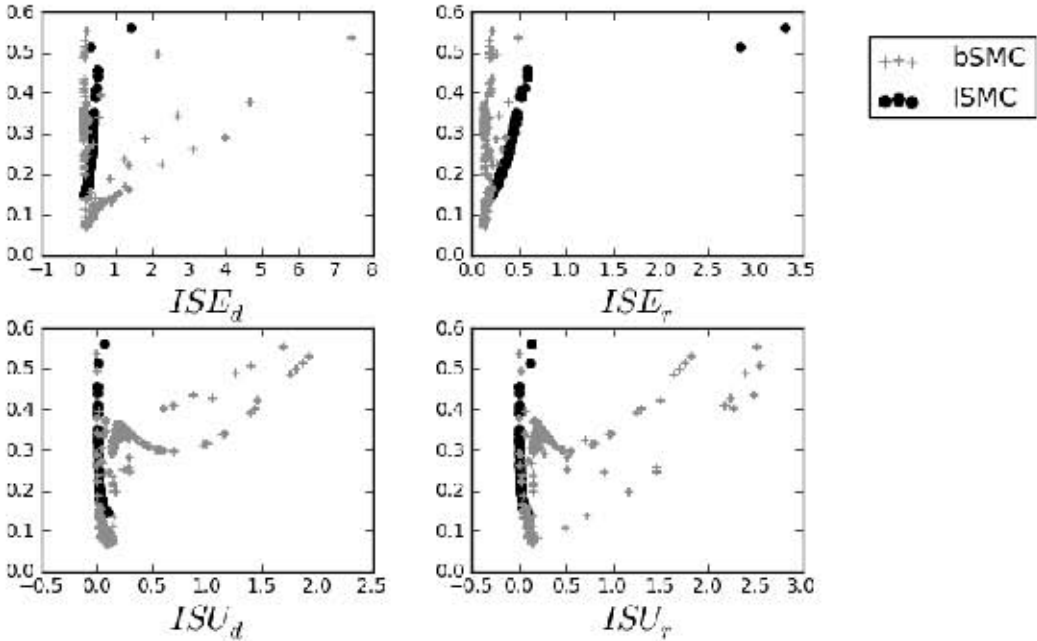


Figure 5.25: ISE_d , ISU_d , ISE_r and ISU_r for ISMC and bSMC

The ISE_d , ISU_d , ISE_r and ISU_r Level Diagrams shown in Figure 5.25, indicate the same relationships as in the first order example. ISMC in general has slower setpoint tracking than bSMC, but uses less controller effort to do so. Further it matches or betters bSMC for output disturbance rejection, again using far less controller effort.

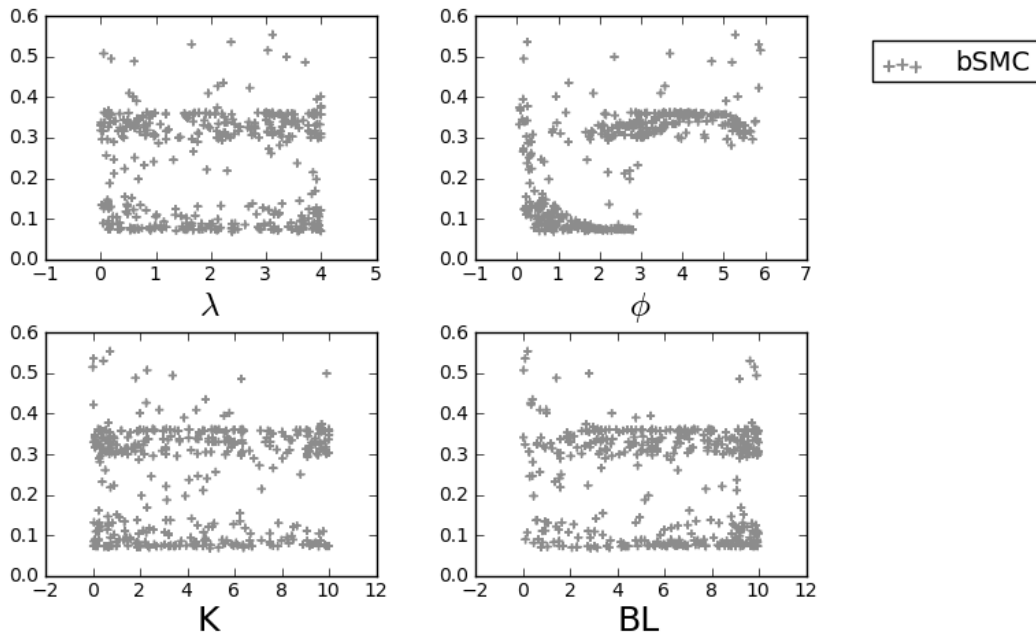


Figure 5.26: λ , ϕ , BL and K for bSMC

The Input Level Diagrams for bSMC, shown in Figure 5.26 show a significant change from the first order case. For the input saturation example, the optimisation appears to have resulted in two groupings, which appear to have resulted primarily due to the ϕ value: Lower values of ϕ resulting in low norm values, and higher values of ϕ resulting in high norm values. There is still the worrying flatness in the other inputs indicating a lack of interaction with the chosen objectives, or over-parametrisation of the controller.

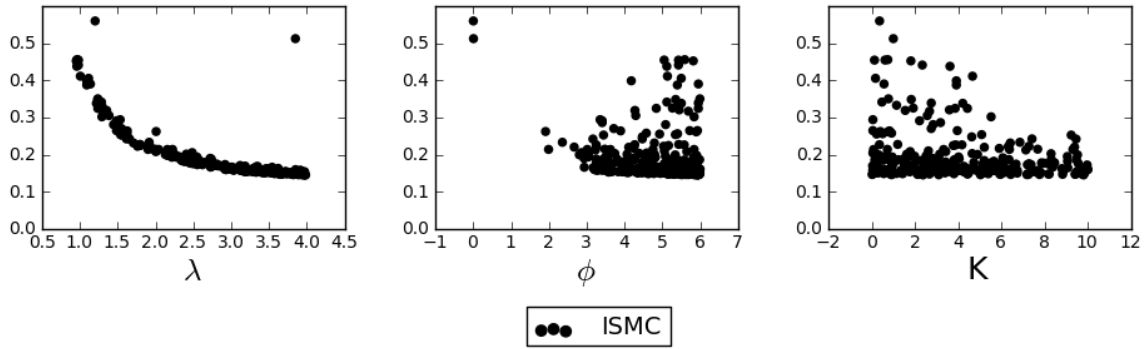


Figure 5.27: λ , ϕ and K for ISMC

The Input Level Diagrams for ISMC are shown again in Figure 5.27. These too are interesting; they continue the trend of higher λ values resulting in lower norms due to their setting the sliding rate of the SMC controller. K is similarly flat and troublesome in that its value does not appear to have a direct affect on the resulting objectives. Finally, ϕ shows a complete reverse of the bSMC situation: Values of $\phi > 3$ are favoured, with lower values resulting in significantly higher norms. The two outlying points noted early are seen here to be as a direct result of extremely low ϕ values.

The interesting part is that these two points don't even respond in the same fashion as one another. The upper point exists on the Pareto front because of its very low *RobustnessDelta*, while the slightly lower point uses comparatively little controller effort for input disturbances ISU_v . Yet, both appear to be as a result of their low ϕ value.

Figure 5.28 shows an example step test of the lowest norm controllers from each of the ISMC and bSMC control design methods using the *ISE* objectives.

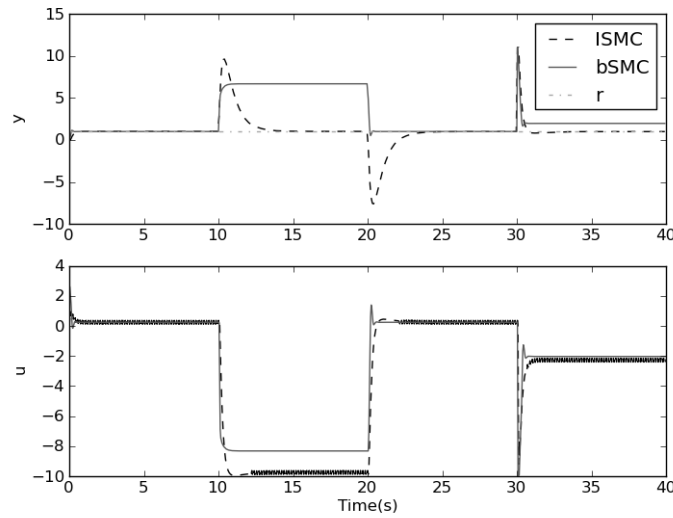


Figure 5.28: Lowest norm step tests for ISMC and bSMC for *ISE* objectives.

As with the first order example step test of the ISMC and bSMC controllers, the same problem with the bSMC controller is clearly visible: The bSMC controller is unable to reject input disturbances; it always has a large finite offset error. This was not unexpected, as was stated in Section 2.1, the controller is of Type 0. Yet it was hoped that this offset would be minimised as a result of the optimisation process. This step test confirms that this is not the case, and again highlights this weakness of the *ISE* objectives.

For the rest of the step test, the bSMC controller actually settles far faster than the ISMC controller, even if it is to the incorrect value. The ISMC controller shows small chattering on the input, resulting from a large K value.

The ISMC controller is, similarly to the previous step tests, limited in its response time by the sampling time of the system and its integrator.

5.1.2.2 Performance Measures - ISMC vs bSMC

Table 5.3 shows the three performance measures comparing Integrated SMC versus Boundary Layer SMC for the input saturation example.

<i>ISE</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
ISMC	0.7926	0.1110	10.4563
bSMC	0.7887	0.1070	31.1620
<i>r = 5</i>			
ISMC	0.8041	0.0399	61.2457
bSMC	0.9166	0.1524	15.2337
<i>r = 10</i>			
ISMC	0.5799	0.0279	170257.3391
bSMC	0.8635	0.3115	6.8434
<i>r = 15</i>			
ISMC	0.6254	0.0134	68.7961
bSMC	0.8414	0.2293	5.0786

Table 5.3: Performance Measure Results for ISMC versus bSMC

Using the interpretation functions described in Table 2.2 in Section 2, some observations can be made.

The ISMC versus bSMC Performance Measure results are the most interesting of the input saturation example as there is significant changes depending on the operating range.

The I_H quality indicator starts with I_H (ISMC) slightly greater than I_H (bSMC), but this situation reverses as the setpoint gets higher, with the difference becoming fairly sizeable at $r = 10$ and up. The interpretation that can be applied now depends on the setpoint, thus further complicating the matter.

The first of the binary indicators, I_{H2} , shows a similar result, with the I_{H2} (ISMC, bSMC) and I_{H2} (bSMC, ISMC) being fairly similar at $r = 1$, this meaning that each front has a similar portion of the hypervolume of optimal controllers that is unique to it. However, as the setpoint increases, the situation quickly changes to a situation similar to the PI vs

ISMC comparisons: That is, that the $I_{H2}(bSMC, ISMC) > 0$ and $I_{H2}(ISMC, bSMC)$ is close enough to zero that within reason one could infer that bSMC is better than ISMC.

The second binary indicator, I_ϵ , shows a similar result to the two previous results, though the actual values of the quality indicator are of interest. The I_ϵ indicator calculates the amount of scaling required for one front to strictly dominate the other. That being the case, a value of less than one infers that the front already dominates the other. A value of one or greater would infer that the front is either equal to, or would require some scaling to make it dominant; the larger the value, the more scaling that would be required.

For the $r = 1$ case, the $I_\epsilon(ISMC, bSMC)$ while being greater than 1, is smaller than $I_\epsilon(bSMC, ISMC)$, meaning that the ISMC Pareto front is closer to dominating the ISMC front than the other way around. The situation reverses for $r = 5$ and greater, with the size of scaling becoming fair sizeable for $r = 10$.

5.2 Parasitic Dynamic Plant

This plant includes a fast parasitic dynamic (that is simulated as a fast stable pole). It is meant to simulate parasitic sensor dynamics present in the system.

The controllers tested are the same controllers as used in the first order plant example. They are not redesigned to counteract the parasitic dynamics and thus are intended to investigate each method's sensitivity to non-dominant unmodelled dynamics.

5.2.1 PI versus ISMC

The first test for the plant with parasitic dynamics has the ISCM controller compared to the PI controller; the PI Pole-Zero Cancellation set is included as well.

5.2.1.1 Integral Square Error Level Diagrams

Figures 5.29 and 5.30 show the $\|\cdot\|_2$ Cost and Input Level Diagrams using the Integral Square Error costs.

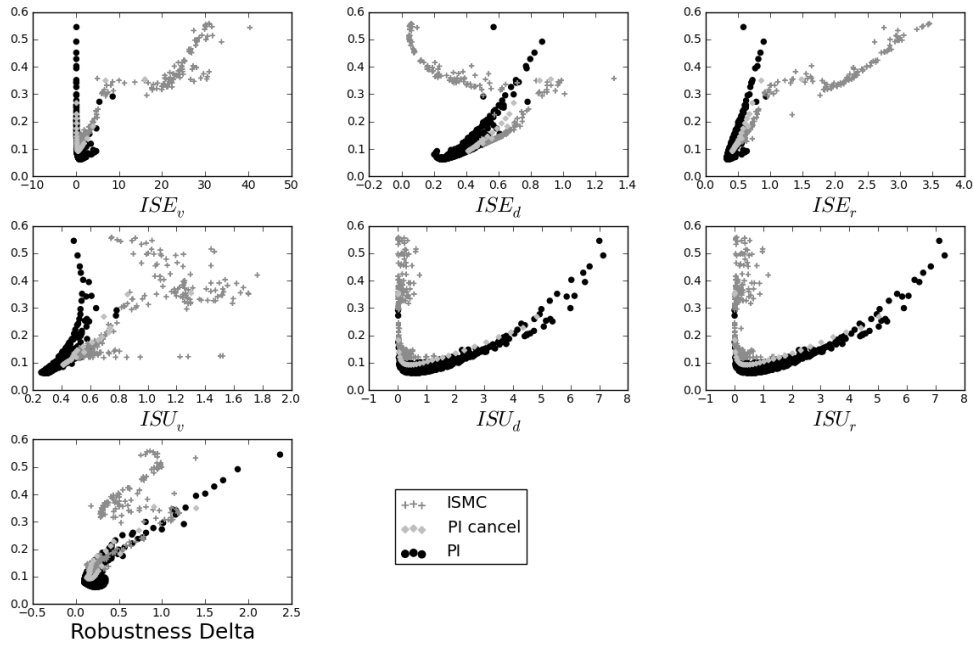


Figure 5.29: $\|\cdot\|_2$ ISE Cost Level Diagrams for PI, PI Cancel and ISMC

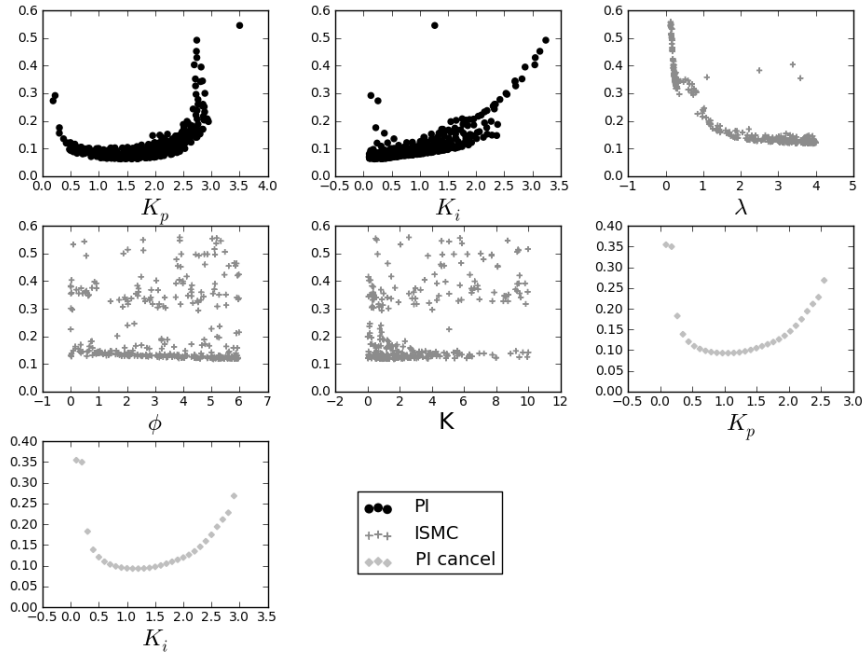


Figure 5.30: $\|\cdot\|_2$ ISE Input Level Diagrams for PI, PI Cancel and ISMC

Figure 5.31 repeats the Level Diagrams for ISE_v and ISU_v , from which can be seen that the PI controller manages better input disturbance rejection than the ISMC controller, at a lower control cost.

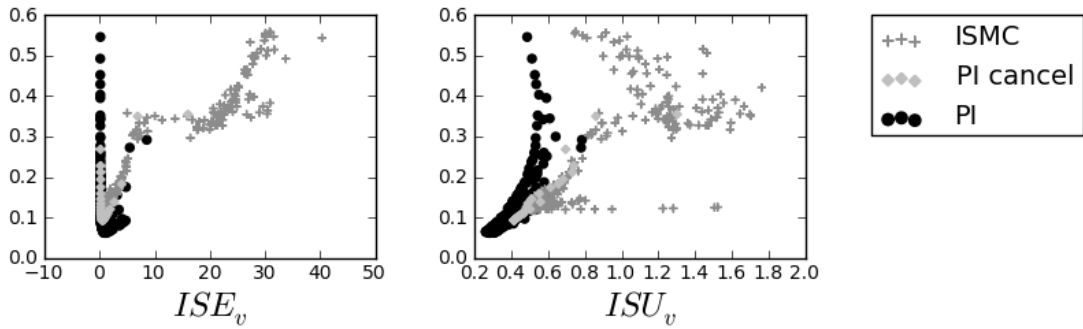


Figure 5.31: ISE_v and ISU_v for PI, PI Cancel and ISMC

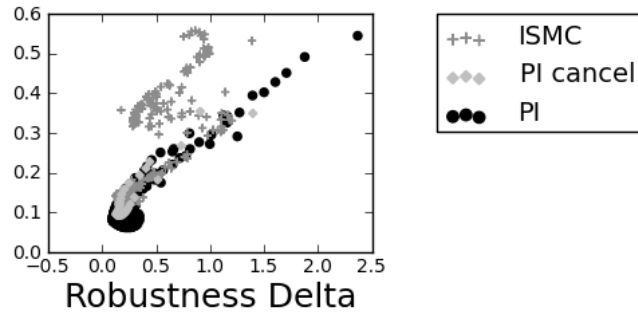


Figure 5.32: *RobustnessDelta* for PI, PI Cancel and ISMC

Both controllers' Pareto fronts indicate in Figure 5.32 that they are fairly robust to model inaccuracies. Although, the majority of the resulting PI controllers seem to outperform the ISMC controllers in terms of *RobustnessDelta*.

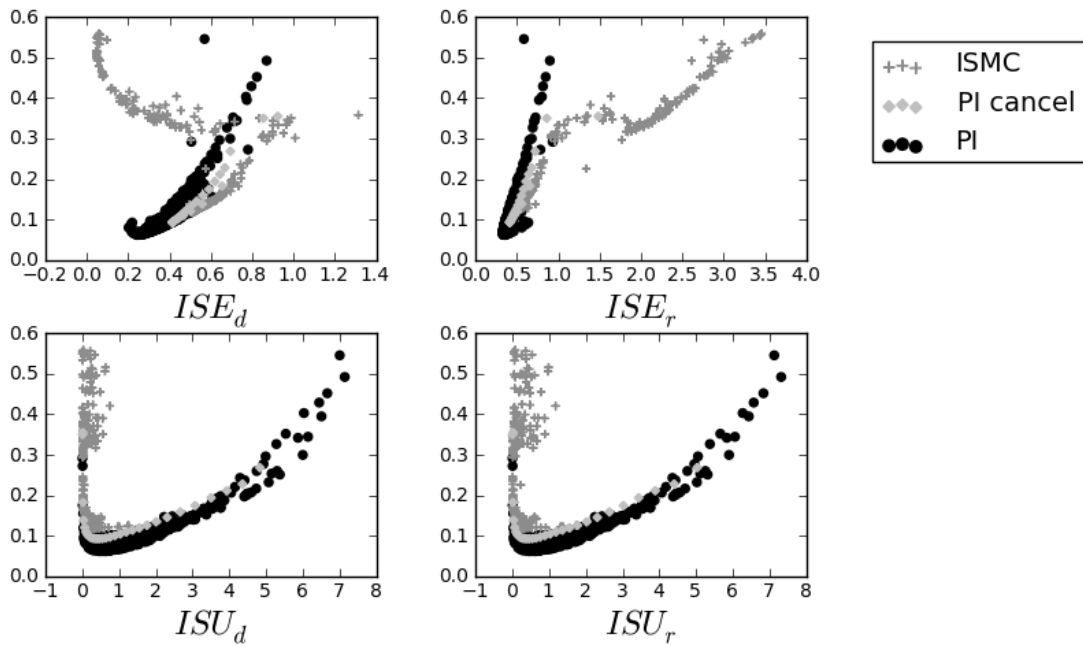


Figure 5.33: ISE_d , ISU_d , ISE_r and ISU_r for PI, PI Cancel and ISMC

The ISE_d , ISU_d , ISE_r and ISU_r Level Diagrams, shown in Figure 5.33, show an interesting difference to the first order: The ISE_r and ISU_r are fairly similar to the first order, with the two controllers managing the trade-off scenario differently; the PI controller favouring settling time over controller effort and the ISMC doing the reverse. What is interesting is that for output disturbances in the presence of unmodelled parasitic dynamics, the ISMC controller manages to equal and better the PI controller in terms of output disturbance rejection, as seen in the ISE_d plot, while using significantly less controller effort, in general, to do so.

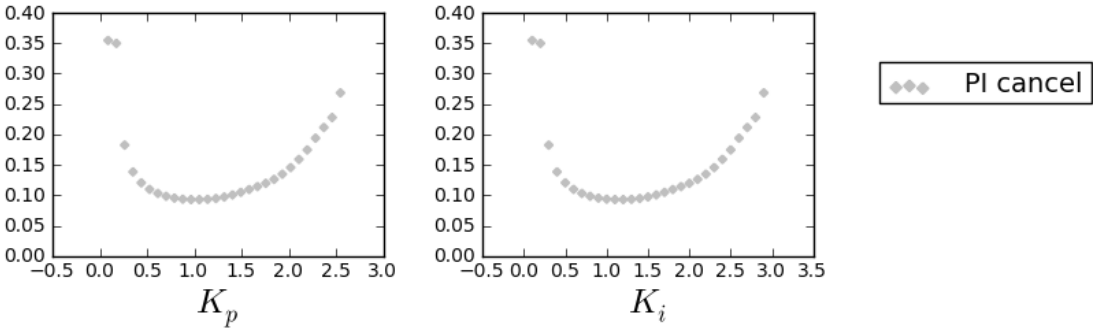


Figure 5.34: K_p and K_i for PI cancellation set.

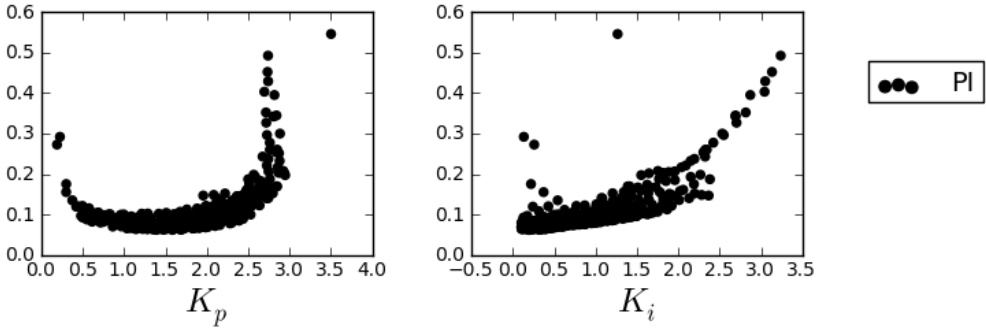


Figure 5.35: K_p and K_i for PI

The K_p and K_i Level Diagrams for the PI and PI Pole-Zero Cancellation sets are

shown in Figures 5.34 and 5.35. The Level Diagrams continue the trend from the previous example. Lower gain controllers are favoured, with a trend towards lower K_i values.

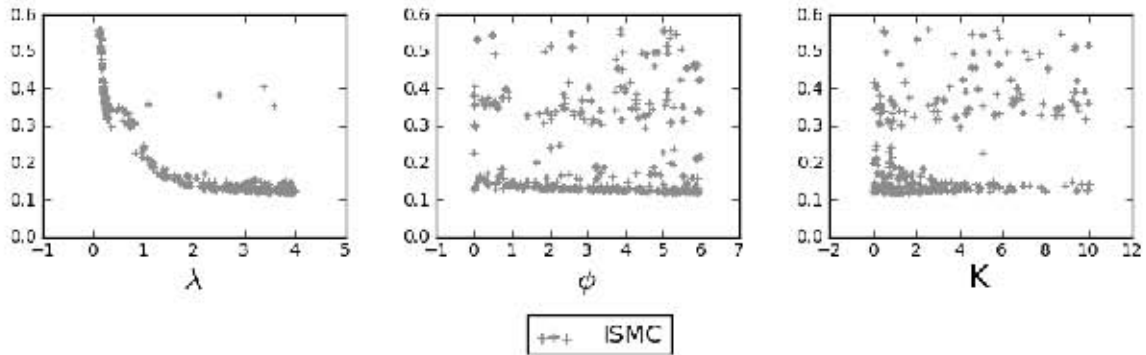


Figure 5.36: λ , ϕ and K for ISMC

The ISMC Input Level Diagrams are shown in Figure 5.36. These Level Diagrams are interesting in that they show two groupings based on the synchronising norm. This split is best seen in the λ Level Diagram, where there is a vertical discontinuity from 2.6 to 2.9. This grouping is less clear in the ϕ and K Level Diagrams. There is still the trend toward lower values of K , as well as the same flat scattered distribution seen in previous examples. As with the previous examples, this could indicate either a lack of effect on the optimisation objectives, or over-parametrisation of the controller.

Figure 5.37 shows an example step test of the lowest norm controllers from each of the PI and ISMC control design methods using the *ISE* objectives.

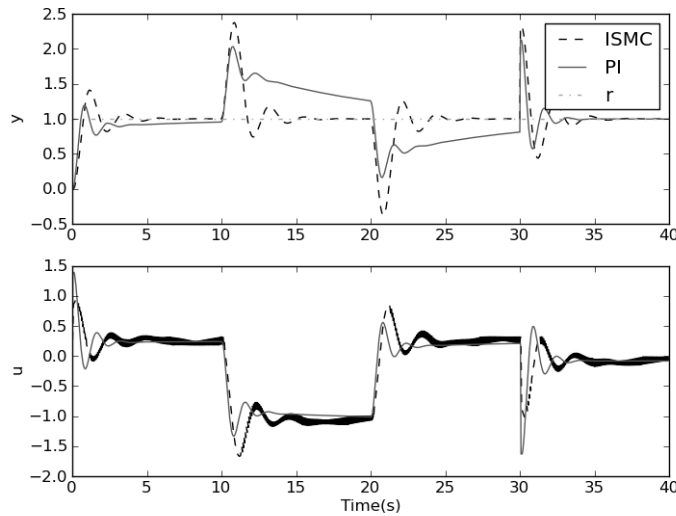


Figure 5.37: Lowest norm step tests for PI and ISMC for ISE objectives.

The first of the step tests for the parasitic dynamics plant is very interesting. Both of the resulting lowest norm controllers oscillate, which is undesirable. The PI controller suffers from the same problem with the ISE objective for the input disturbance seen previously; the objective allows for controllers that do not actually settle within the simulation time period, but still achieve reasonable ISE_v objective values.

The ISU objectives are interestingly reflected here in the step tests. The Level Diagrams show that ISMC controller uses more controller effort to reject the input disturbances than the PI controller, and this is clearly visible in the step test control signal u . Similarly the ISU_r and ISU_d Level Diagrams show the two controllers using for more closely matched controller effort for the setpoint and output disturbances. This is also fairly easily corroborated on the step tests; the PI controller has more overshoot, but the ISMC controller has chatter and takes slightly longer to settle.

5.2.1.2 Error Settling Time Level Diagrams

Figures 5.38 and 5.39 show the $\|\cdot\|_2$ Cost and Input Level Diagrams using the Error Settling Time costs.

The region of interest needed to be expanded for this example, as the previous 0.56 was no longer appropriate. For the following Level Diagrams, a ROI of 1.0 is used as the norm cut off. This has implications because it shows that none of the controllers are able to find particularly good solutions using the *EST* objectives. Consider that the maximum synchronising norm possible is \sqrt{n} , which in this case is $\sqrt{7} = 2.646$. A good solution would score significantly less than the maximum, as can be seen in the previous examples, which indicates the overall quality of the resulting solutions for this case.

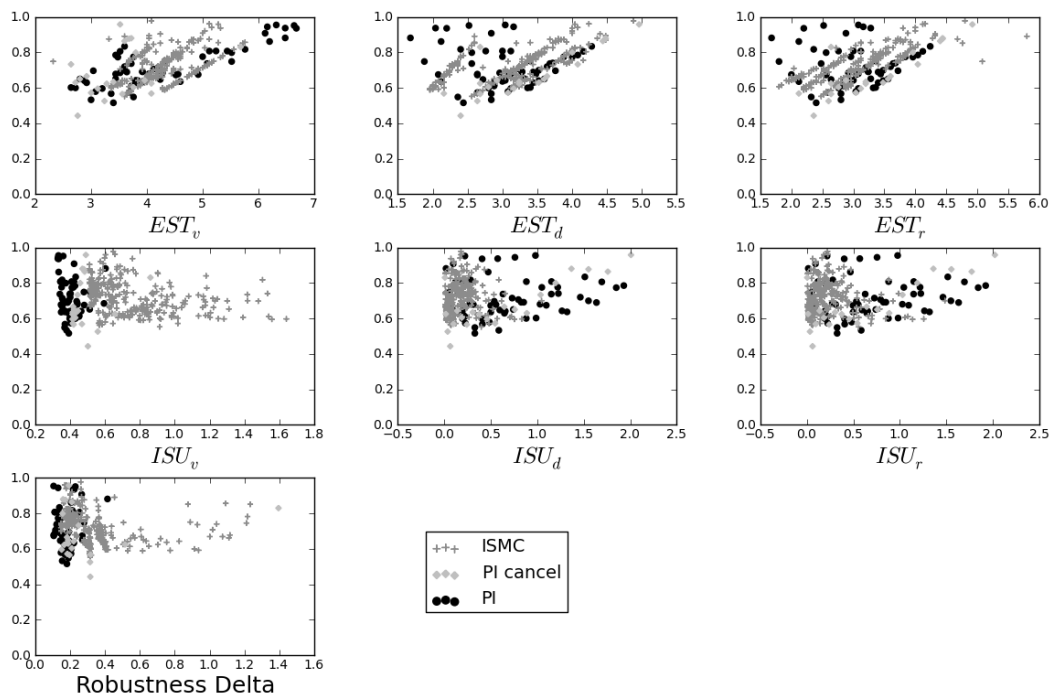


Figure 5.38: $\|\cdot\|_2$ *EST* Cost Level Diagrams for PI, PI Cancel and ISMC

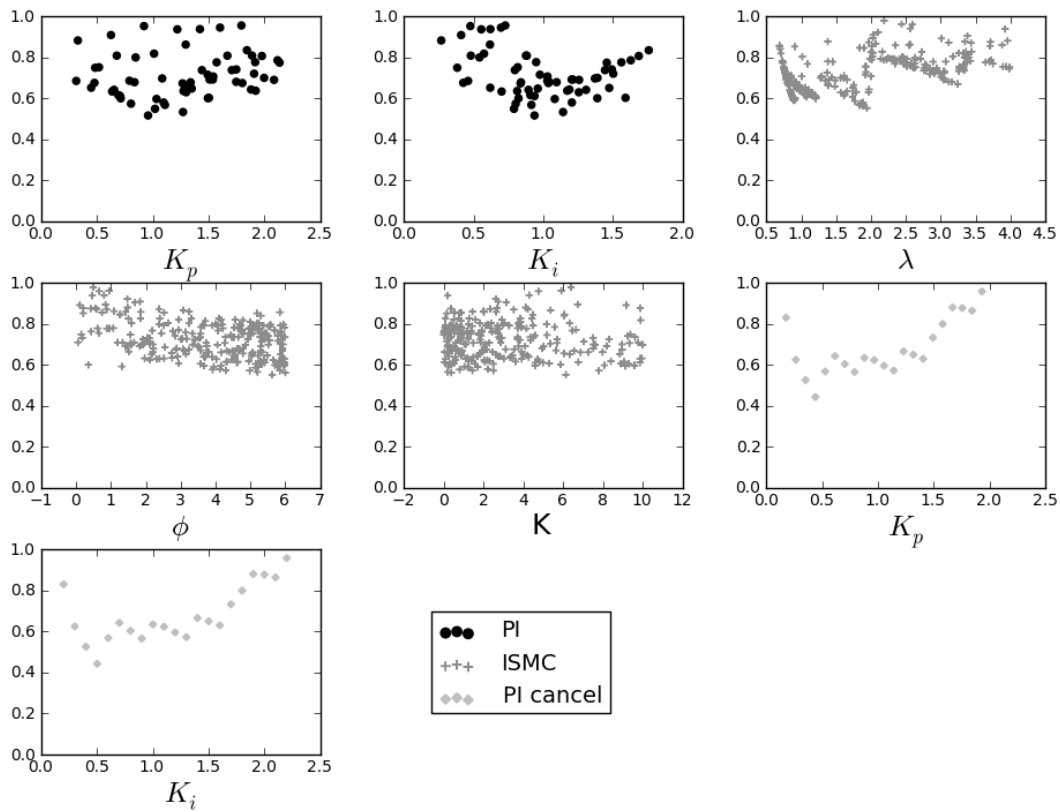


Figure 5.39: $\|\cdot\|_2$ *EST* Input Level Diagrams for PI, PI Cancel and ISMC

As can be seen from the Figures, none of the controllers perform particularly well and there are no major defining trends. This appears to be as a result of the optimiser struggling to minimise the objectives.

The various objectives are very segmented, this is clearly seen in the PI pole - zero cancellation set's Input Level Diagrams, where the Pareto front is clearly discontinuous across the various controller gains. Further analysis of this phenomenon points towards it being a result of the oscillatory nature of the plant and the approximate method with which the *EST* objectives are calculated.

Figure 5.40 shows an example step test of the lowest norm controllers from each of the

PI and ISMC control design methods using the EST objectives for the parasitic dynamics plant.

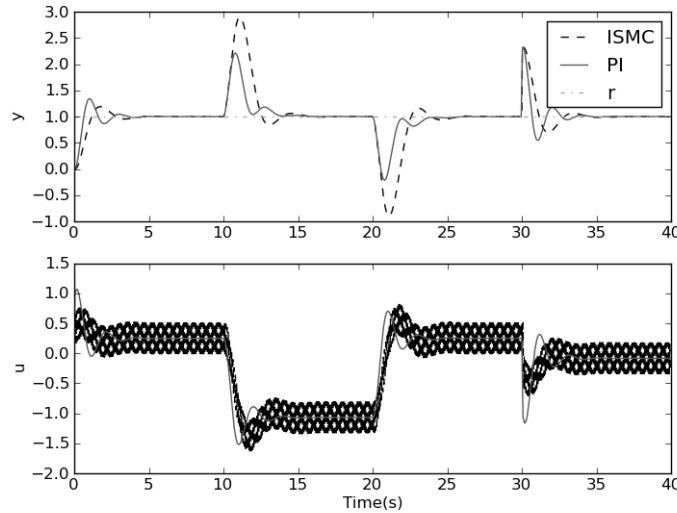


Figure 5.40: Lowest norm step tests for PI and ISMC for EST objectives.

The step tests show the two controllers performing comparably for each of the setpoint, input disturbance and output disturbance, with the PI controller settling slightly faster in each case.

The ISMC controller has significant chatter still on its input u , even with the integrated output. What is of interest is that even with this significant chatter the two achieve fairly similar ISU .

These Level Diagrams and step test have raised an interesting point. Previously, the ISE objectives were demonstrated to have a weakness when it comes to actually optimising onto controllers that settle within the desired time frame. Here we see the weakness of the EST objectives. The oscillatory nature of the plant being investigated and the approximate method used to calculate the EST objectives can cause the optimiser to struggle to successfully minimise the given objectives.

As the optimiser has struggled to minimise the front and the cause has been identified as resulting from the approximate method used to calculate the *EST* objectives, no further analysis is done on these Level Diagrams.

5.2.1.3 Performance Measures - PI vs ISMC

Table 5.4 shows the three performance measures comparing PI versus Integrate SMC for the parasitic dynamics example.

<i>ISE</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.8453	0.3407	10.4674
ISMC	0.5114	0.0068	7.0922
<i>r = 5</i>			
PI	0.8228	0.2882	4.1681
ISMC	0.5400	0.0054	12.6992
<i>r = 10</i>			
PI	0.7500	0.2825	10.0338
ISMC	0.4837	0.0163	21.9476

Table 5.4: Performance Measure Results for PI versus ISMC

Using the interpretation functions described in Table 2.2 in Section 2, some observations can be made.

The Performance Measure comparisons for PI versus ISMC for the parasitic dynamics plant are very similar to those of the first order example.

For I_H , PI is not worse than ISMC, as $I_H(PI) > I_H(ISMC)$ for each of the different setpoint.

I_{H2} shows a similar result with $I_{H2}(PI, ISMC)$ being far greater than $I_{H2}(ISMC, PI)$ as well as $I_{H2}(ISMC, PI)$ being close to zero. As with the other the first order example, as this value is close enough to be accounted for by inaccuracy in the calculation method, it is possible that $I_{H2}(ISMC, PI) = 0$, which would then allow us to infer that PI is

better than ISMC.

The I_ϵ shows strange behaviour. For the $r = 1$ set, one would expect, based on the I_H and I_{H2} indicators, that $I_\epsilon(PI, ISMC)$ would be less than $I_\epsilon(ISMC, PI)$. But, in this case we have the reverse. This was found to be caused by the combined effect of the selection of the region of interest and the strange shape of the Pareto fronts. For the other two setpoints, the expected situation of $I_\epsilon(PI, ISMC)$ smaller than $I_\epsilon(ISMC, PI)$ exists. Though, neither of I_ϵ values are close to one, so neither set is particularly close to dominating the other.

5.2.1.4 Performance Metrics - PI vs PI Pole-Zero Cancellation

Table 5.5 shows the three performance measures comparing PI versus the PI Pole-Zero Cancellation set for the parasitic dynamics example.

<i>ISE</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
PI	0.8622	0.4856	1.1939
PI Cancel	0.3761	-0.0005	2.2844
<i>r = 5</i>			
PI	0.8229	0.3755	1.1281
PI Cancel	0.4504	0.0030	3.0724
<i>r = 10</i>			
PI	0.7429	0.4358	1.4936
PI Cancel	0.3073	0.0002	3.4728

Table 5.5: Performance Measure Results for PI versus PI Pole-Zero Cancellation

Using the interpretation functions described in Table 2.2 in Section 2, some observations can be made.

This is one of the few cases where all of the different quality indicators agree, and their results are fairly definite.

Firstly for I_H , $I_H(PI)$ is significantly larger than $I_H(PI\ Cancel)$. While the only

dominance relationship we can infer is that PI is not worse than PI Cancel, in engineering terms the PI controller is able to produce a far broader range of controllers than the PI Pole-Zero Cancellation set represents.

The I_{H2} quality indicator has interesting results in that they actually can be said to infer a dominance relationship without having to take the inaccuracy of the Pareto fronts into account. In all three cases $I_{H2}(PI, PICancel)$ is large, while $I_{H2}(PICancel, PI)$ is close enough to zero, that if one takes the systematic error of the hypervolume calculation method into account, it can be assumed to be zero. As such in this case, PI is better than PI Cancel.

The I_ϵ results are similarly close to allowing a strict dominance relationship to be inferred. The $I_\epsilon(PI, PICancel)$ is very close to one in each case, while the $I_\epsilon(PICancel, PI)$ is greater than one. In terms of a pragmatic comparison, the PI Pareto front is far closer to being dominant than the PI Pole-Zero Cancellation set, and is itself very very close to dominant.

5.2.2 ISMC versus bSMC

5.2.2.1 Integral Square Error Level Diagrams

Figures 5.41 and 5.42 show the $\|\cdot\|_2$ Cost and Input Level Diagrams using the Integral Square Error costs.

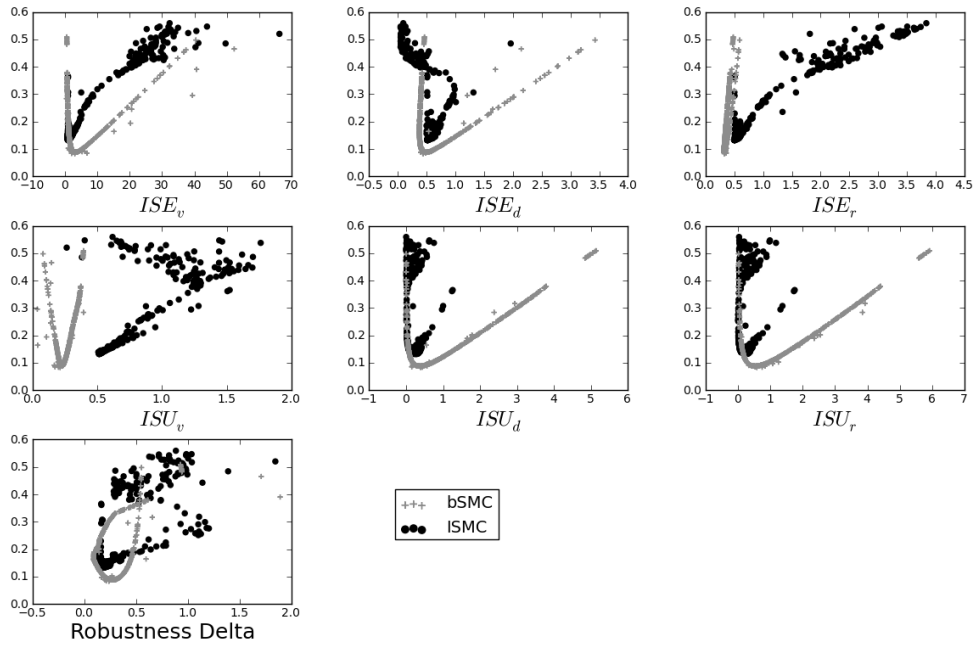


Figure 5.41: $\|\cdot\|_2$ ISE Cost Level Diagrams for ISMC and bSMC

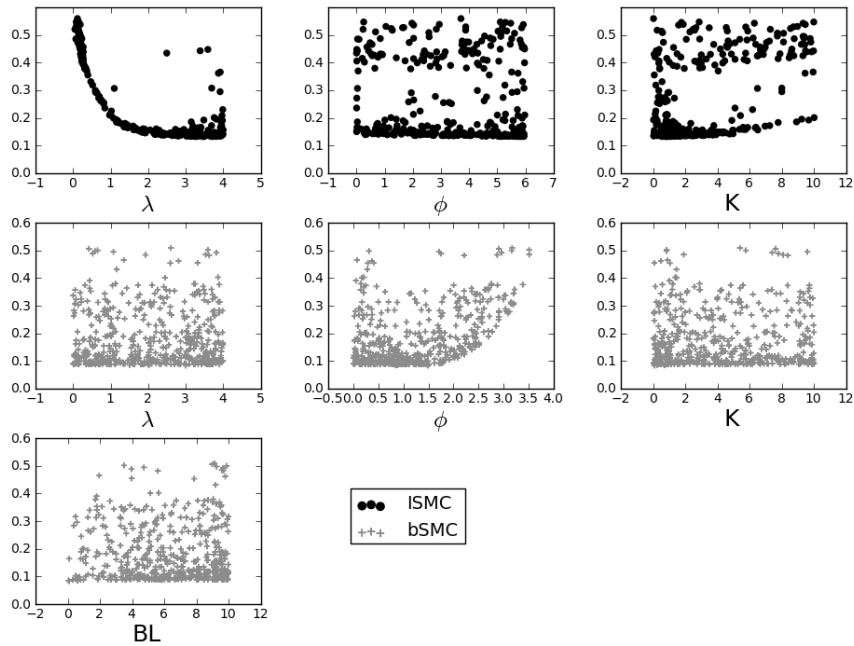


Figure 5.42: $\|\cdot\|_2$ ISE Input Level Diagrams for ISMC and bSMC

The Level Diagrams for ISMC and bSMC in Figures 5.41 and 5.42 are remarkably similar to the first order example's Level Diagrams, showing that the parasitic dynamics have not had a significant effect on the resulting control.

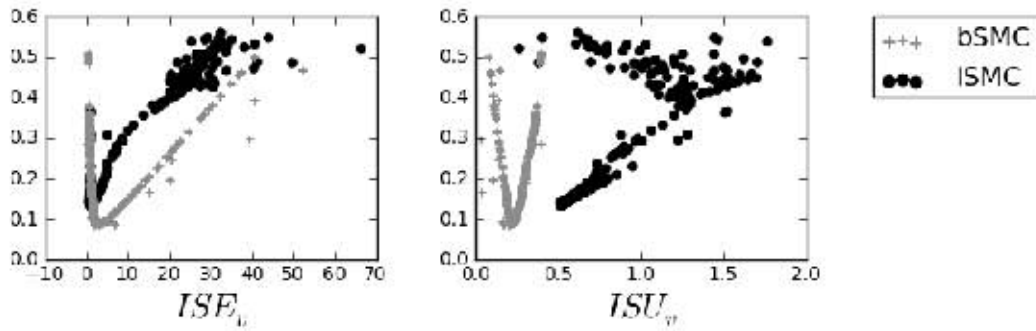


Figure 5.43: ISE_v and ISU_v for ISMC and bSMC

The ISE_v and ISU_v Level Diagrams show a familiar result: The bSMC controller using less controller effort than the ISMC controller to attain equal or better disturbance rejection for input disturbances.

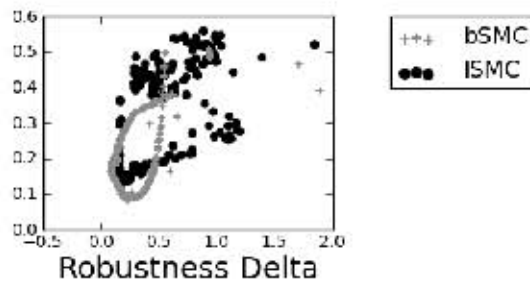


Figure 5.44: $RobustnessDelta$ for ISMC and bSMC

The $RobustnessDelta$ is the Level Diagram with the most significant differences. The bSMC controller appears far more robust in this case than for the first order example. This is a fairly strange result as one would expect the reverse to be true for the more complex

plant dynamics, but it is most likely a phenomenon of the method used to calculate the metric.

That being said, both controllers appear fairly robust to model inaccuracies, with the bSMC controller performing slightly better in general than the ISMC controller.

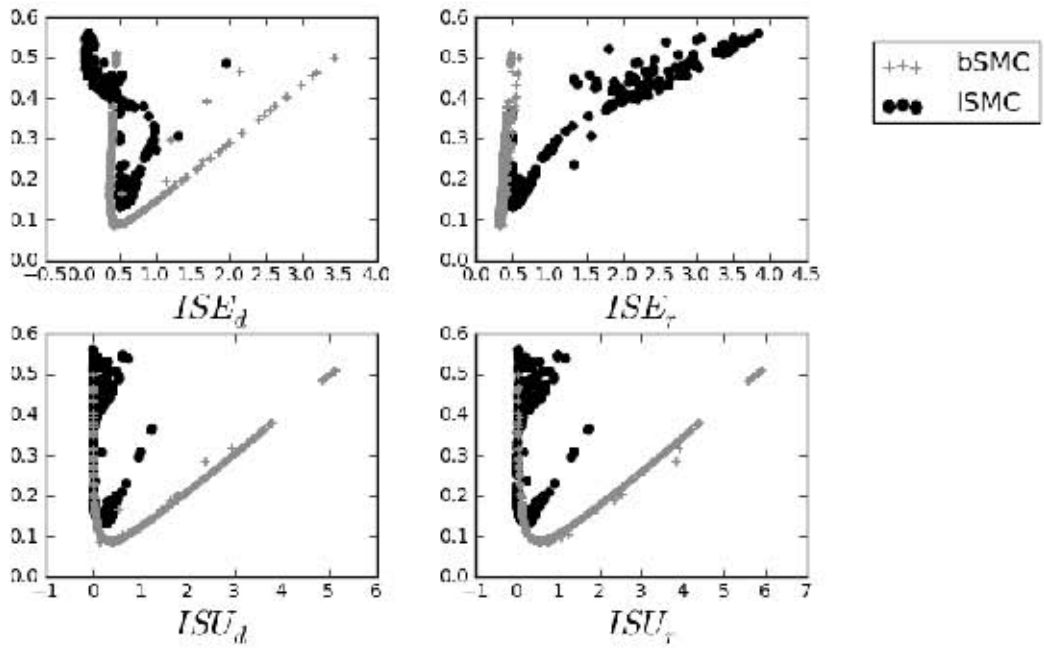


Figure 5.45: ISE_d , ISU_d , ISE_r and ISU_r for ISMC and bSMC

The ISE_d , ISU_d , ISE_r and ISU_r Level Diagrams, shown in Figure 5.45, has almost exactly the same shape and interpretation as the first order example: bSMC achieves much better setpoint tracking than ISMC, but in general uses more controller effort to do so. ISMC manages to equal or better bSMC in output disturbance rejection, and does so whilst using less controller effort.

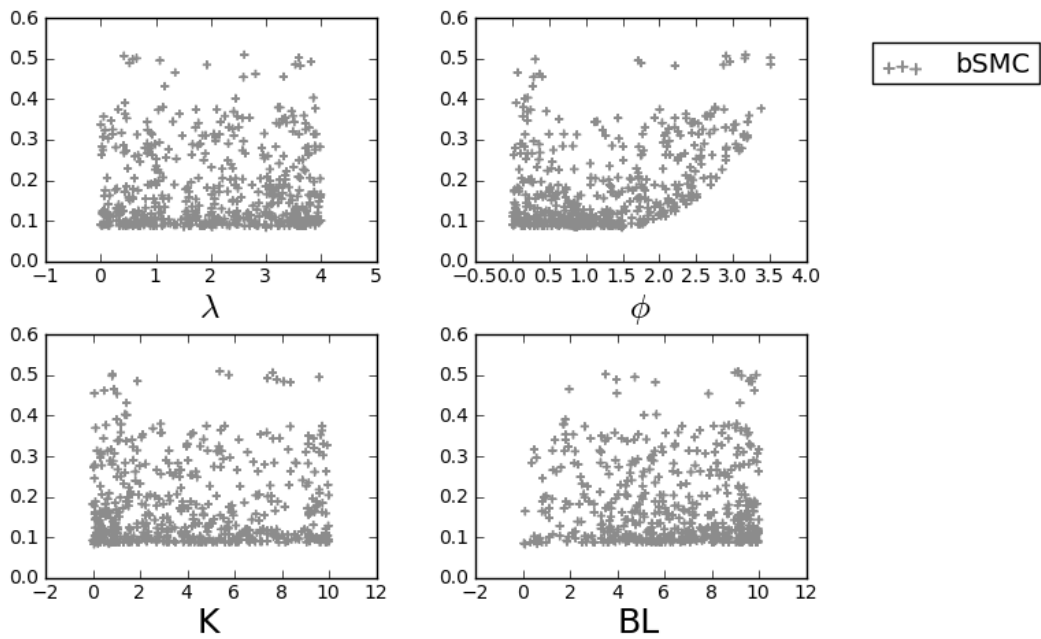


Figure 5.46: λ , ϕ , BL and K for bSMC

The bSMC Input Level Diagrams show the now familiar flat and trend-less features. The same slight upturn to the norms, resulting from higher values of ϕ exists as in the first order case.

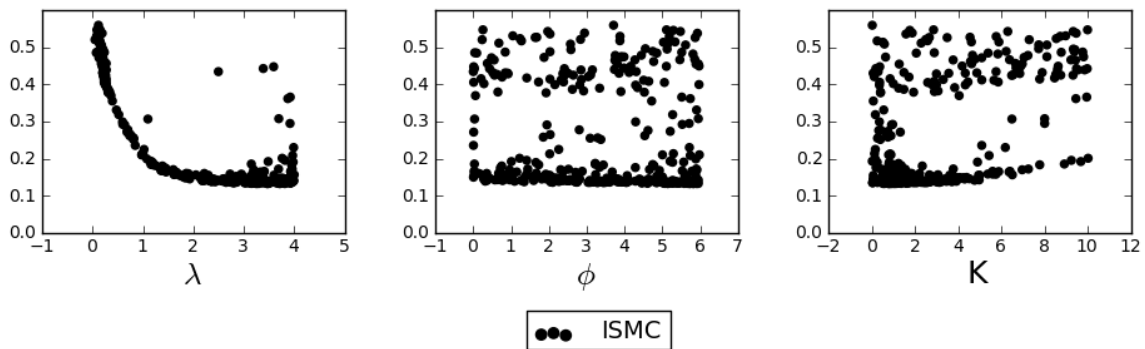


Figure 5.47: λ , ϕ and K for ISMC

The ISMC Input Level Diagrams are again similar to those of the first order example. The λ Level Diagram indicating a direct relationship between synchronising norm and the value of λ : higher λ resulting in lower norms. The ϕ is still flat indicating non interaction of the parameter with the objective, or over-parametrisation. K still favours lower values, but is still fairly flat.

The two group phenomenon is less defined than in the first order example, not having the vertical gap of the first order example. But, there is still a distinct separation visible, especially in the K Level diagram.

Figure 5.48 shows an example step test of the lowest norm controllers from each of the ISMC and bSMC control design methods using the *ISE* objectives.

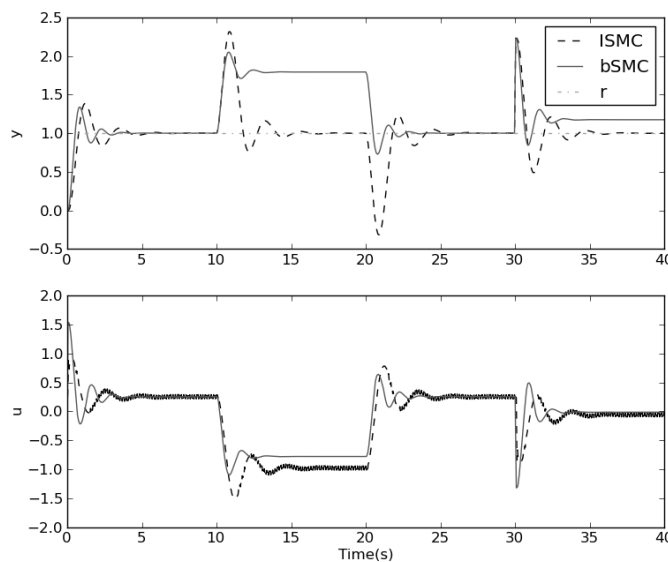


Figure 5.48: Lowest norm step tests for ISMC and bSMC for *ISE* objectives.

The example step test shows the same failing in the bSMC controllers: they are unable to reject input disturbances, or more accurately, they have a finite offset error that is very large. The same offset error exists for output disturbances and can be seen in the step test, but is much smaller.

The bSMC controller settles faster than the ISMC controller, even though it settles to the wrong value in the case of the input and output disturbances.

The ISU_v Level Diagram can now be seen in a very different light. While the bSMC controller uses significantly less controller effort than the ISMC controller, this is a false result as the bSMC controller does not actually reject the input disturbance, as would reasonably be required.

Otherwise the step test follows the Level Diagrams results. The ISMC and bSMC controllers do comparatively well at setpoint tracking and output disturbance rejection (with the noted offset error), while using relatively similar amounts of controller effort.

5.2.2.2 Performance Measures - ISMC vs bSMC

Table 5.6 shows the three performance measures comparing Integrate SMC versus Boundary Layer SMC for the parasitic dynamics example.

<i>ISE</i>			
Name	I_H	I_{H2}	I_ϵ
<i>r = 1</i>			
ISMC	0.5900	0.0360	7.3115
bSMC	0.7958	0.2419	24.9949
<i>r = 5</i>			
ISMC	0.6620	0.0362	77.6763
bSMC	0.8548	0.2289	18.8255
<i>r = 10</i>			
ISMC	0.9158	0.0538	268.7904
bSMC	0.9168	0.0548	27.8372

Table 5.6: Performance Measure Results for ISMC versus bSMC

Using the interpretation functions described in Table 2.2 in Section 2, some observations can be made.

The Performance Measure results for ISMC versus bSMC for the parasitic dynamics plant show mixed results.

The I_H unary hypervolume quality indicator shows similar results for $r = 1$ and $r = 5$ with $I_H(bSMC) > I_H(ISMC)$ indicating the relationship bSMC not worse than ISMC. For $r = 10$ the two Pareto fronts are fairly well matched, with their I_H both being around 0.91.

For I_{H2} we have a similar split result. For the $r = 1$ and $r = 5$ sets, bSMC does get a large binary hypervolume, while ISMC gets a binary hypervolume close to zero. As with the other examples, one can pragmatically assume $I_{H2}(ISMC, bSMC)$ is zero, and thus infer $bSMC \triangleright ISMC$, or in words that bSMC is better than ISMC. The $r = 10$ shows that the operating range matters for these non-linear controllers. Again having very similar I_{H2} values near zero, indicating no significant unique portions to either front.

Finally, the I_ϵ quality indicator has a different result for each setpoint range. At $r = 1$, $I_\epsilon(ISMC, bSMC)$ is less than $I_\epsilon(bSMC, ISMC)$, with neither being particularly close to one. For $r = 5$ and $r = 10$ the reverse is true, with $I_\epsilon(ISMC, bSMC)$ being greater than $I_\epsilon(bSMC, ISMC)$, with larger values for both. This is a troubling result as it only agrees with the hypervolume quality indicators at the $r = 5$ setpoint, and gives the opposite interpretation for the others.

CHAPTER 6

SMC Decision Space Investigation

In both Chapters 4 and 5 it became apparent from the Level Diagrams analysis of the SMC control design methods, that there was an issue with the design parameters being used. Specifically, the resulting Input Level Diagrams were flat and featureless.

This chapter deals with a short investigation into the cause of this phenomenon. Two possible causes for this occurring are that the decision variables do not interact with the controller in such a way as to affect the chosen optimisation objectives; or that the controllers are over-parametrised, meaning that there are too many input variables being optimised and as a result, multiple solutions, balancing different input variables, result in the same output scenario.

As the phenomenon was primarily noticed in the bSMC controllers, the investigation focuses on this controller.

6.1 Parameter Reduction

As stated above, one of the possible causes of the flat Level Diagrams could be that the controller designs are over-specified with too many input variables. Figures 6.1 and 6.2 shows the *ISE* Level Diagrams for various bSMC controllers, each with a different number of input parameters. The three controllers are: Standard bSMC, bSMC with the boundary layer BL fixed (fbSMC), and bSMC with both BL and K fixed (fKbSMC).

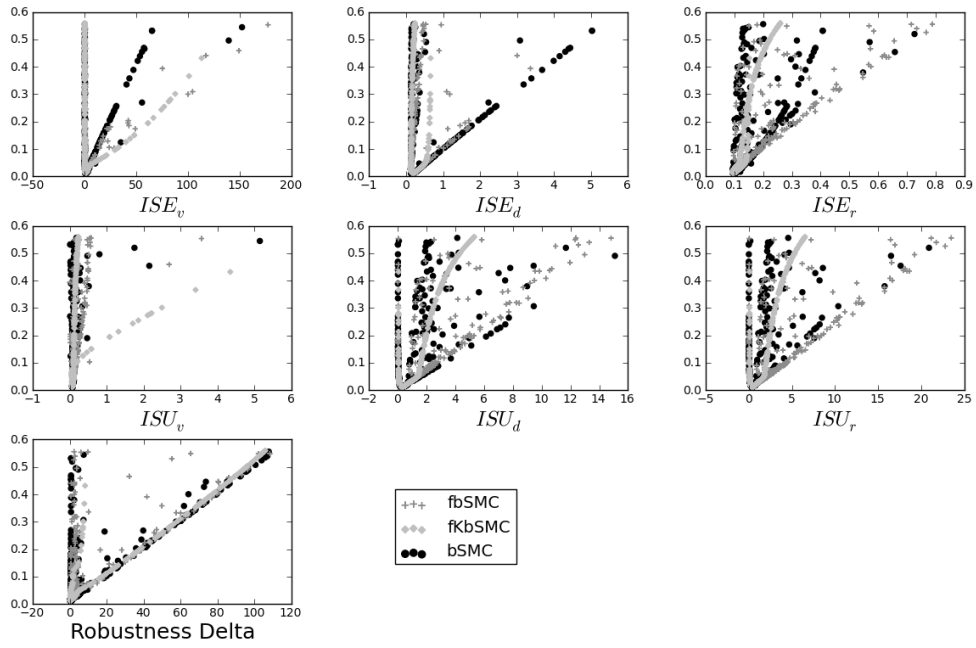


Figure 6.1: $\|\cdot\|_2$ ISE Cost Level Diagrams for bSMC, fbSMC and fKbSMC

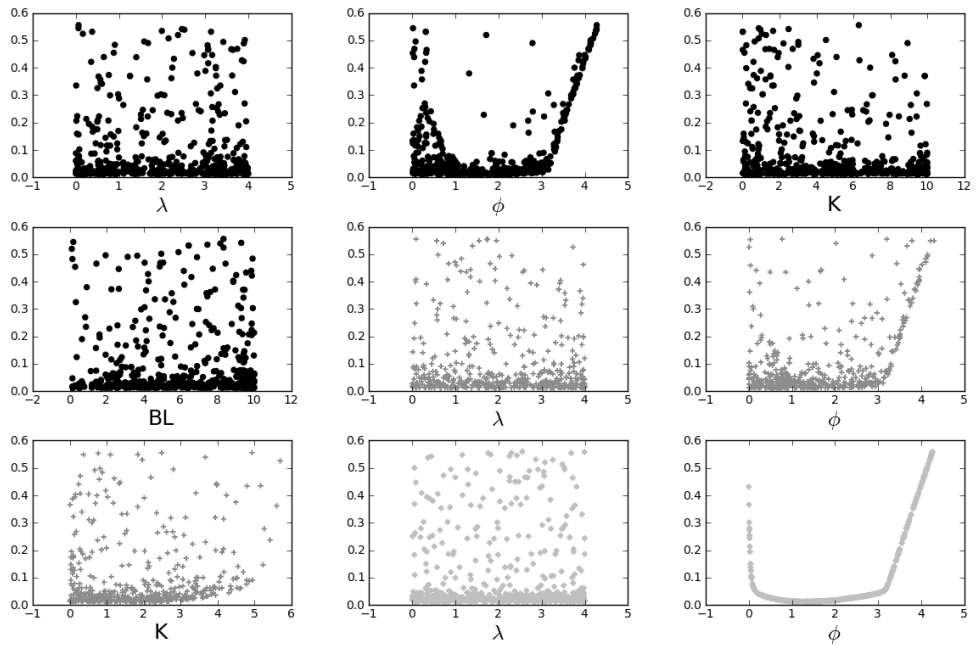


Figure 6.2: $\|\cdot\|_2$ ISE Input Level Diagrams for bSMC, fbSMC and fKbSMC

In three controllers, bSMC, fbSMC, fKbSMC the number of parameters are reduced. In each case this is done in a logical manner: The boundary layer BL is fixed to a value that should ensure stability while operating in the linear region, K is fixed to a value based on η -reachability condition (see Chapter 2.1), which depends on the model error introduced in the *RobustnessDelta* metric and the magnitude of the input and output disturbance signals.

The Level Diagrams of bSMC are as in Chapter 4. There are no obvious trends except for in ϕ where there is a slight upturn of the norm values as ϕ increases. As ϕ sets the sliding rate of SMC, one would expect an increase in error costs as a result.

fbSMC shows a similar trend in the norm values as ϕ increases. As well as slight upturn in norms as a result of high values of K .

The last reduced parameter controller, fKbSMC, is very interesting. The resulting norm values, are clearly dependant on the value of ϕ , very low and high values of ϕ resulting in high norm values, with a minimum around 1.3.

These results are particularly concerning. One would expect λ to be the primary parameter affecting the system performance, and thus the norm, but as is demonstrated here, this is not the case. The ϕ values seem to have the largest effect on each of the bSMC controllers; this value sets the reaching rate limit, which would seem to imply that the bSMC controllers are not managing to reach the sliding surface before settling.

The various Input Level Diagrams, shown in Figure 6.2, show a similarly concerning situation with λ values for each of the three controllers. In theory, λ should set the sliding rate of the various SMC controllers, but in this case it appears to have no effect on the resulting norm. This is further reinforced by the Input Level Diagrams for fKbSMC, where the λ plot is flat and featureless, while the ϕ value shows a direct effect on the resulting norms.

One possible reason for this non-interactivity of λ is due to the fact that bSMC does

not reject input disturbance significantly. Thus, for the input disturbance, the selection of λ does not obviously affect the ISE_v value, which could in part explain the non-interaction of λ .

This investigation has shown that reduction of parameters, which was done here in a logical design manner, does not fix this problem of over-parametrisation and non-interaction. Thus further investigation into this phenomenon is needed, but is deemed to be beyond the scope of this thesis.

University of Cape Town

CHAPTER 7

Conclusions

7.1 General Conclusions

The Level Diagrams investigated in this thesis have demonstrated their usefulness for comparison of control methods using high dimensional Pareto fronts. By dealing effectively with multi-dimensional space they allow clear representation of the Pareto fronts of various control design methods.

The choice of objectives has a great effect on the resulting Pareto fronts. The use of *ISE* and *EST* objectives has demonstrated this in that *ISE* was unable to avoid pole-zero cancellation for PI controllers, while using the *EST* error objectives was able to do so. Furthermore, the final conclusions able to be drawn from the Pareto front depends on the relationship between the different objectives.

A weakness with the *ISE* objectives was revealed from the step test examples for the PI controller, where the *ISE* objectives allowed PI controllers that did not actually settle. Similarly, the *EST* objectives as implemented, with their approximate method, struggled to optimise the oscillatory parasitic dynamics plant.

The Level Diagram analysis of the bSMC controller, highlighted the weakness of using optimisers in isolation. The step test examples showed that the optimisation process did not (or was not able to) minimise the expected offset error for input and output disturbances. Care needs to be taken when applying any optimisation scheme to verify that the results make sense.

Performance Measures provide a mathematically robust means to evaluate different Pareto fronts. The quality indicators used in this thesis, I_H , I_{H2} and I_ϵ , while having useful characteristics, were unable to provide definitive comparisons in terms of Pareto dominance for most of the Pareto front comparisons, though general comments could be inferred from them.

The I_ϵ measure was particularly troubling in that its results were spurious and not consistent with the other two indicators. From this thesis, it appears that I_ϵ is unsuitable for use with this sort of comparison, which is unfortunate as it is capable of inferring far stronger dominance relationships than the other quality indicators used. Further investigation is suggested.

Performance Measures were unable to definitively say that PI outperformed ISMC or visa versa, nor that ISMC was better or worse than bSMC. The Performance Measure results for certain sections at certain setpoints were close enough to infer relationships based on the quality indicators. Using these inferred results general comments and conclusions could be drawn, but should be treated cautiously as there are questions regarding their accuracy.

7.2 Comparison Conclusions

Using the two methods, Level Diagrams and Performance Measures, general conclusions and analysis of the various controllers was possible. For each comparison, the various strengths and weaknesses of each controller was easily deduced from the Level Diagrams, while an overall quantitative comparison of the various controllers was possible using the Performance Measures.

For the comparison of PI and ISMC, the results indicate that while ISMC is very robust, this design method suffers from poor rejection of input disturbances. Overall, PI control is competitive, being able to produce a larger hypervolume of the region of interest

as well a large portion of controllers that ISMC cannot match, for the given objectives. In general, the Performance Measures were able to infer that PI is better than ISMC.

Comparison of ISMC and bSMC was inconclusive, and the veracity of the results was called into question as the resulting bSMC controllers failed to reject input disturbances at all. It was similarly inconclusive from the Performance Measure perspective, though the bSMC was shown to be slightly better in terms of the the quality indicators applied. The Level Diagrams raised a valid concern regarding the bSMC control design method and also revealed a weakness in the Level Diagrams: the bSMC method is unable to reject input disturbances effectively. The Level Diagrams do not show this situation and it is only determined from the step tests. The resulting large offset error for input disturbances also highlights why the bSMC method does not work with the *EST* objectives. The ISMC controller is more robust to changes in the model, and is able to reject both input and output disturbances more effectively than bSMC.

In general, using the seven objectives in this thesis, the optimal PI controllers were able to outperformed the PI pole-zero cancellation set for all of the test problems.

For the parasitic dynamics plant, the resulting PI controllers are optimised such that they avoid high system gains, and as such avoid the unstable region on the Nyquist Diagram.

The performance of the ISMC and bSMC controllers was weaker than expected, possibly due to their simplistic designs. A more recent and complex SMC controller could probably outperform either of these.

ISMC's chatter reduction method was shown to be unsuccessful at times, though this was primarily due to extreme values of K . As K was shown to be non-interactive with the chosen optimisation objectives, it may be possible to reduce the value such that there is no resulting chatter.

It should be noted that a significant factor in the results was the selection of the

sampling time. This was done based on the real system from which the plant model was derived. However, it set a limit on the value of λ , which had the carry on effect of limiting the sliding rate and thus the settling time of both the ISMC and bSMC controllers. Thus, using a faster sampling time would affect the results and could significantly change the overall comparison.

7.3 Further Work

Further investigation is needed into the choice and effect of objective selection. The thesis demonstrated significant differences between two different objective sets, *ISE* and *EST*. Measures such as Integral Absolute Error should be investigated as possible alternative objectives.

The chosen SMC controllers were simplistic in conception and design and it would be interesting to see how one of the more modern and complex SMC methods would perform compared to PI using this comparison framework.

In terms of the PI controller, it became apparent that it may be easier in terms of analysis to use a different model for the PI controller and to optimise on different parameters. Specifically, it would be interesting to see what the Level Diagrams would look like if the gain (A) and time constant (T) model (as used for the plant models) were used instead. One major benefit would be that the zero position could be read more easily from the graphs.

The quality indicators used for Performance Measures were mostly unable to conclude conclusive results regarding the various comparisons. The accuracy of the Pareto fronts, combined with the systematic error of the I_H quality indicator made it difficult to draw accurate conclusions using the Performance Measures. Improving the accuracy of the Pareto fronts, through larger populations and stronger optimisation methods, as well as replacing the statistical hypervolume method with an analytical method would greatly

improve the accuracy and applicability of Performance Measures for control engineering comparisons.

The I_ϵ quality indicator needs further investigation as to the reasons for its inaccuracy and varied results in this thesis.

In addition, further investigation into various other quality indicators could provide more useful measures for controller comparison. Specifically, developing measures with specific engineering meaning would make these tools even more useful.

The concept of this thesis arose from various controller comparisons as put forward in [18], [25],[19],[15] and [16]. It would be interesting to test the controllers and plants from these papers using the framework from this thesis, and see how the results compare.

One of the major assumptions used in this thesis is that the Pareto fronts resulting from the Pareto Differential Evolution algorithm are fairly accurate. Further investigation would be useful with regard to testing its accuracy and finding ways to improve it if necessary.

REFERENCES

- [1] H. A. Abbass, R. Sarker, and C. Newton. Pde: A pareto-frontier differential evolution approach for multi-objective optimization problems. In *Proc. Conf. Evol. Comput.*, volume 2, pages 971–978, 2001.
- [2] H. A. Abbass and R. A. Sarker. The pareto differential evolution algorithm. *International Journal on Artificial Intelligence Tools*, 11(4):531–552, 2002.
- [3] J. Andersson. A survey of multiobjective optimization in engineering design, 2000.
- [4] J. Bader and E. Zitzler. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Tik report*, 286, 2008.
- [5] X. Blasco, J. M. Herrero, J. Sanchis, and M. Marnez. A new graphical visualisation of n-dimensional pareto front for decision-making in multiobjective optimization. *Information Sciences: an International Journal*, 178:3908–3924, October 2008.
- [6] D. Brockhoff and E. Zitzler. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. *Lec. notes in Comp. Sci*, 4193:533–542, 2006.
- [7] J. Calderwood. *Principles of Automatic Control*. The English Universities Press LTD, 1967.
- [8] C. A. C. Coello. Twenty years of evolutionary multiobjective optimization: A historical view of the field. *IEEE Comput. Intel. Magazine*, 1(1):28–36, 2006.
- [9] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. *Lecture Notes in Computer Science*, pages 849–858, 2000.
- [10] C. Edwards and S. K. Spurgeon. *Sliding Mode Control: Theory and Applications*. CRC Press, 1998.
- [11] P. Fleming and R. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10:1223–1241, 2002.
- [12] P. Fleming, R. Purshouse, and R. Lygoe. Many-objective optimization: An engineering design perspective. *Lec. Notes in Comp. Sci*, 3410:14–32, 2005.
- [13] C. M. Fonseca, J. D. Knowles, L. Thiele, and E. Zitzler. A tutorial on the performance assessment of stochastic multiobjective optimizers. Slides: Evolutionary Multi-Criterion Optimization Conference, 2005.

- [14] A. Gambier. MPC and PID control based on Multi-Objective Optimization. In *American Control Conference, 2008*, pages 4727–4732, 2008.
- [15] W. García-Gabín and E. F. Camacho. Sliding mode model based predictive control for non minimum phase systems.
- [16] S. Graebe. Robust and adaptive control of an unknown plant: A benchmark of new format. *Automatica*, 30(4):567–575, 1994.
- [17] D. C. J. Knowles. Approximating the non-dominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8:149–172, June 2000.
- [18] E. M. Jararav, M. N. A. Parlakçı, and Y. Istefanopulos. A new variable structure pid-controller design for robot manipulators. *IEEE Trans. Contr. Syst. Technol.*, 13(1):122–130, January 2005.
- [19] A. J. Koshkouei and K. J. Burnham. Control of dc motors using proportional integral sliding mode.
- [20] G. P. Liu, J. B. Yang, and J. F. Whildborne. *Multiobjective Optimization and Control*. Research Studies Press LTD., 2002.
- [21] J. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
- [22] T. Okabe, Y. Jin, and B. Sendhoff. A critical survey of performance indices for multi-objective optimisation. 2003.
- [23] P. A. L. Plante. *Comprehensive dictionary of electrical engineering*. CRC Press & IEEE Press, 1999.
- [24] M. Rapson. Pareto analysis of controller design methodologies for integrator plus dead time processes. *CISSE 2007*, 2007.
- [25] K. K. Shyu, F. J. Lin, H. J. Shieh, and B. Juang. Robust variable structure speed control for induction motor drive. *IEEE Trans. Aerosp. Electron. Syst.*, 35(1):215–224, January 1999.
- [26] J. Sikaundi. Obtaining initial controller parameters for approximate pole placement iterative feedback tuning. *IETA '2008*, 2008.
- [27] J. J. E. Slotine and W. Li. *Applied Nonlinear Control*, chapter 7, pages 276–309. Prentice-Hall International, 1991.
- [28] C. L. Smith. *Digital Computer Process Control*. Intext Education Publishers, 1972.

- [29] R. Storn and K. Price. Differential evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces, 1995.
- [30] L. While, P. Hingston, L. Barone, and S. Huband. A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, 10(1):29–38, 2006.
- [31] K. D. Young, V. I. Utkin, and U. Özgüner. A control engineer’s guide to sliding mode control. *IEEE Trans. Contr. Syst. Technol.*, 7(3):328–342, 1999.
- [32] E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology Zurich, 1999.
- [33] E. Zitzler, D. Brockhoff, and L. Thiele. The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In *Evolutionary Multi-Criterion Optimization*, pages 862–876. Springer.
- [34] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *EUROGEN*, 2001.
- [35] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.*, 7:117–132, April 2003.

APPENDIX A

Sliding Mode Control Design Method

The progression of a simple SMC design process for an SISO system follows. It follows closely the method and derivation in [27], with some modifications suggested in [10].

Consider a single input dynamic system:

$$x^{(n)} = f(\mathbf{x}) + b(\mathbf{x})u \quad (\text{A.1})$$

where x is the output (the n^{th} derivative of \mathbf{x}), u is the control input and $\mathbf{x} = [x, \dot{x}, \dots, x^{(n-1)}]^T$ is the state vector. The control problem is to get the state \mathbf{x} to track a desired state trajectory (time varying signal) $\mathbf{x}_d = [x_d, \dot{x}_d, \dots, x_d^{(n-1)}]^T$.

Letting $\tilde{x} = x - x_d$ define the tracking error for variable x , then:

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d = [\tilde{x}, \dot{\tilde{x}}, \dots, \tilde{x}^{(n-1)}]^T \quad (\text{A.2})$$

A sliding surface (sliding mode, manifold or hyper-surface) is then defined in state-space as shown in (A.3).

$$s(\mathbf{x}, t) = \left(\frac{d}{dt} + \lambda \right)^{n-1} \tilde{x} \quad (\text{A.3})$$

and λ is a strictly positive constant. Thus for a 2nd order system ($n = 2$)

$$s = \dot{\tilde{x}} + \lambda\tilde{x} \quad (\text{A.4})$$

Since s contains $\tilde{\mathbf{x}}^{(n-1)}$, differentiating s once allows the input u to appear. Continuing the 2nd order system above:

$$\dot{s} = \ddot{\tilde{x}} + \lambda\dot{\tilde{x}} = (\ddot{x} - \ddot{x}_d) - \lambda\dot{\tilde{x}} \quad (\text{A.5})$$

substituting \ddot{x} for (A.1)

$$\dot{s} = f(\mathbf{x}) + b(\mathbf{x})u - \ddot{x}_d + \lambda\dot{\tilde{x}} \quad (\text{A.6})$$

The simplified 1st-order problem of keeping the scalar s at zero can now be achieved by choosing u such that outside of $s(t)$,

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \quad (\text{A.7})$$

with η a strictly positive design constant. This is commonly referred to as the η -reachability condition [27].

SMC design requires that (A.3) and (A.7) are well-behaved functions of tracking error x_d . Then selecting the feedback control law u such that s^2 remains a Lyapunov candidate function of the closed loop system. The design procedure is then to select a suitable feedback control law u to verify condition (A.7) ensuring Lyapunov stability. However, to ensure this criteria in the presence of modelling error and disturbances, the control law is made discontinuous about $s = 0$.

The dynamics of the system while in sliding (i.e. $s = 0$) are given as

$$\dot{s} = 0 \quad (\text{A.8})$$

using the previous definition of s in (A.3) and solving for u obtains the equivalent continuous control law u_{eq} that would maintain $\dot{s} = 0$ if the dynamics were known exactly.

As stated above, in order to achieve this stability in the face of bounded disturbances and modelling errors, the system is made discontinuous across $s(t)$ ($s = 0$), this is typically achieved through the inclusion on a discontinuous term in the control law, in this case a sign function. A common concept used in SMC design is the concept of equivalent control u_{eq} , which can be interpreted as the continuous control law that would give $\dot{s} = 0$, if the dynamics were known exactly.

Using a second-order system example:

$$\ddot{x} = f + u \quad (\text{A.9})$$

the equivalent control u_{eq} needed would be

$$u_{eq} = -f + \ddot{x}_d - \lambda \dot{\tilde{x}} \quad (\text{A.10})$$

This still assumes that the dynamics of the system are known precisely. If for example, the dynamics f are not exactly known, but rather are estimated to \hat{f} . If we assume the estimation error on f is known and bounded to some function $F = F(x, \dot{x})$

$$|\hat{f} - f| \leq F \quad (\text{A.11})$$

The best approximate control \hat{u} that would achieve $\dot{s} = 0$ given the estimated \hat{f} is then

$$\hat{u} = -\hat{f} + \ddot{x}_d - \lambda \dot{\tilde{x}} \quad (\text{A.12})$$

In order to ensure that (A.7) is achieved we add a discontinuous term to \hat{u} , giving us the actual control law u .

$$u = \hat{u} - k \operatorname{sgn}(s) \quad (\text{A.13})$$

where sgn is the sign function:

$$\begin{aligned} \operatorname{sgn}(s) &= +1 \quad \text{if } s > 0 \\ \operatorname{sgn}(s) &= -1 \quad \text{if } s < 0 \end{aligned}$$

with a gain, k , chosen to be large enough to guarantee (A.7) and assuming that we desire $\ddot{x} = 0$.

$$\frac{1}{2} \frac{d}{dt} s^2 = \dot{s} \cdot s = \left[f - \hat{f} - k \operatorname{sgn}(s) \right] s = \left(f - \hat{f} \right) - k |s|$$

thus if we let

$$k = F + \eta \quad (\text{A.14})$$

then we get the desired stability:

$$\frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \quad (\text{A.15})$$

The size of k is related to the bound of any uncertainties, so it is also affected by any uncertainties in k from $b(x)$ as in:

$$\ddot{x} = f + bu \quad (\text{A.16})$$

Finally, a reasonable modification [10] is made by including $-\Phi s$. Φ is a strictly positive constant. This modification allows direct specification of both reaching and

sliding dynamics. So the control law becomes:

$$u = \hat{b}^{-1} [\hat{u} - k \operatorname{sgn}(s) - \Phi s] \quad (\text{A.17})$$

This must still satisfy

$$\frac{1}{2} \frac{d}{dt} s^2 = \dot{s} \cdot s \leq -\Phi s^2 - \eta |s| \quad (\text{A.18})$$

seeing as $\Phi s^2 \geq 0$, this sets up an η -reachability condition. Ignoring the non-linear components above, we get

$$\frac{d}{dt} |s(t)| \leq -\Phi |s(t)| \quad (\text{A.19})$$

which implies

$$|s(t)| \leq |s(0)| e^{-\Phi t} \quad (\text{A.20})$$

which shows that the inclusion of $-\Phi s$ gives the minimum rate at which sliding is achieved from some initial distance from the sliding mode $|s(0)|$.

APPENDIX B

Parallel Pareto Differential Evolution

During the course of research, it became apparent that the PDE algorithm could greatly benefit from parallelisation. Thus the author modified the PDE algorithm for use in this thesis such that it would allow the use of distributed computing in order to speed up the optimisation.

The Pareto Differential Evolution Algorithm [2] is shown in Algorithm 1.

In each generation, a number of new children are generated (read: bred using crossover and mutation operators). This step only requires the previous generation and a target parent (which is a member of the previous generation), and as such can be separated and treated as an embarrassingly parallel problem.

The proposed changes allow each child generation (breeding), in each generation, to be carried out in a separate thread, or in the case of multiprocessing in a separate process, conceivably in any sort of worker-node structure.

The current method generates each child consecutively to make a new population for that generation: Comparing each new candidate to its parent and placing the new child into the new population if it Pareto-dominates its parent. This is repeated until some maximum population is reached.

The key concept to understand how the modification works is that if there are more than α dominant members in a population (where α is less than the maximum population size), the neighbourhood rule is applied. This rule effectively reduces the population by

let G denote a generation, P a population of size M , and $\bar{x}_{G=k}^j$ the j^{th} individual of dimension N in population P in generation k , CR denotes the crossover rate, α the number of dominant solutions retained between each generation.

input: $N, M \geq 4, \alpha, F \in (0, 1+), CR \in [0, 1]$, and initial bounds: *lower* (x_i), *upper* (x_i), $i = 1, \dots, N$

initialise $P_{G=0} = \{\bar{x}_{G=0}^1, \dots, \bar{x}_{G=0}^M\}$ as

foreach individual $j \in P_{G=0}$ **do**

$x_{i,G=0}^j = \text{Gaussian}(0.5, 0.15), i = 1, \dots, N$

Repair $\bar{x}_{G=k}^j$ if any variable is outside its boundaries

end

evaluate $P_{G=0}$

$k = 1$

while the stopping criterion is not satisfied **do**

remove all dominated solutions from $P_{G=k-1}$

if the number of dominated solutions from $P_{G=k-1} > \alpha$ **then**

apply the neighbourhood rule

end

for $j = 0$ to number of non-dominated solution in $P_{G=k-1}$ **do**

$\bar{x}_{G=k}^j \leftarrow \bar{x}_{G=k-1}^j$

end

while $j \leq M$ **do**

randomly select $r_1, r_2, r_3 \in (1, \dots, \alpha)$, from the non-dominated solutions of $P_{G=k-1}$, where $r_1 \neq r_2 \neq r_3$

randomly select $i_{rand} \in (1, \dots, N)$

forall $i \leq N$, **do**

$$x'_i = \begin{cases} x_{i,G=k-1}^{r_3} + \text{Gaussian}(0, 1) \times (x_{i,G=k-1}^{r_1} - x_{i,G=k-1}^{r_2}) & \text{if } (\text{random}[0, 1] < CR \wedge i = i_{rand}) \\ x_{i,G=k-1}^j & \text{otherwise} \end{cases}$$

end

Repair $\bar{x}'_{G=k}$ if any variable is outside its boundaries

if \bar{x}' dominates $\bar{x}_{G=k-1}^{r_3}$ **then**

$\bar{x}_{G=k}^j \leftarrow \bar{x}'$

$j = j + 1$

end

end

$k = k + 1$

end

return the set of non-dominated solutions

Algorithm 1: The Pareto Differential Evolution Algorithm

iteratively removing members from the most densely populated portion of the Pareto Front until the population is α large. Thus, for every generation there are $M - \alpha$ new members that need to be generated.

The modification generates a new job (which can be processed in either a thread or process) for each member necessary to fill the new population to its maximum. Each of these jobs then needs to find a new child for the new population. They can be run concurrently, as no information (other than the previous population, which is fixed at this point) needs be shared between them.

One problem found during implementation was stall cases occurring in some of the jobs. That is, the optimisation algorithm is unable to find a solution where the child is "better" than its parent, which locks up the optimiser.

This was handled using a *best candidate* solution. For each job, a maximum number of iterations is allowed before the job is returned. For each iteration, the new child is compared to the parent, as well as to the previous best solution. If the child dominates the parent it is returned; if not, it is compared to the previous best solution. If it dominates the previous best solution, it replaces it. This is repeated either until a dominant child is found, or until the maximum number of iterations is reached, in which case the best candidate is returned.

The logic behind the best candidate method is as follows: Every parent is still in the population (parents are selected from the dominant population - after the neighbourhood rule is applied) when the next culling (removing of non-dominant members) occurs. The best candidate had to outperform 100 other candidates, so it is at least the 100th worst candidate (though in all probability a lot better). And finally, while the best candidate may not dominate its parent, it may be better than another member of the dominant population.

Using an intelligent job-scheduling method allows the optimiser to use all available

resources and greatly reduce the time taken for each optimisation run.

The Modified Pareto Differential Evolution Algorithm is shown in Algorithm 2.

let G denote a generation, P a population of size M , and $\bar{x}_{G=k}^j$ the j^{th} individual of dimension N in population P in generation k , CR denotes the crossover rate, MR denotes the mutation rate α the number of dominant solutions retained between each generation.

input: $N, M \geq 4, \alpha, F \in (0, 1+), CR \in [0, 1], MR \in [0, 1]$, and initial bounds: $lower(x_i), upper(x_i), i = 1, \dots, N$
 initialise $P_{G=0} = \{\bar{x}_{G=0}^1, \dots, \bar{x}_{G=0}^M\}$ as

foreach individual $j \in P_{G=0}$ **do**

$x_{i,G=0}^j = lower(x_i) + rand_i[0, 1] \times (upper(x_i) - lower(x_i)), i = 1, \dots, N$

Repair $\bar{x}_{G=k}^j$ if any variable is outside its boundaries

end

evaluate $P_{G=0}$

$k = 1$

while the stopping criterion is not satisfied **do**

remove all dominated solutions from $P_{G=k-1}$

if the number of dominated solutions from $P_{G=k-1} > \alpha$ **then**

apply the neighbourhood rule

end

for $j = 0$ to number of non-dominated solution in $P_{G=k-1}$ **do**

$\bar{x}_{G=k}^j \leftarrow \bar{x}_{G=k-1}^j$

end

while $j \leq M$ **do**

create new job:

job:

$count = 0$

repeat

$count = count + 1$

randomly select $r_1, r_2, r_3 \in (1, \dots, \alpha)$, from the non-dominated solutions of $P_{G=k-1}$, where $r_1 \neq r_2 \neq r_3$

randomly select $i_{rand} \in (1, \dots, N)$

forall $i \leq N$, **do**

$$x'_i = \begin{cases} x_{i,G=k-1}^{r_3} + Gaussian(0, 1) \times (x_{i,G=k-1}^{r_1} - x_{i,G=k-1}^{r_2}) & \text{if } (random[0, 1] < CR \wedge i = i_{rand}) \\ x_{i,G=k-1}^j & \text{otherwise} \end{cases}$$

end

forall $i \leq N$, **do**

$$x'_i = \begin{cases} x'_i + rand_i[0, 1] \times (upper(x_i) - lower(x_i)) & \text{if } (random[0, 1] < MR) \\ x_{i,G=k-1}^j & \text{otherwise} \end{cases}$$

end

Repair $\bar{x}_{G=k}^j$ if any variable is outside its boundaries

if \bar{x}' dominates $\bar{x}_{G=k-1}^{r_3}$ **then**

$\bar{x}_{G=k}^j \leftarrow \bar{x}'$
 break

end

else if \bar{x}' dominates \bar{x}^{best} **then**

$\bar{x}^{best} \leftarrow \bar{x}'$

end

until break or $count > 100$

if $count > 100$ **then**

$\bar{x}_{G=k}^j \leftarrow \bar{x}^{best}$

end

end

$j = j + 1$

end

$k = k + 1$

end

return the set of non-dominated solutions

Algorithm 2: The Modified Pareto Differential Evolution Algorithm

APPENDIX C

Plant Modelling and System Identification

DC motors are typically modelled as a two-pole model [7], where the output y is the speed and the input u is the control voltage of the motor:

$$\begin{aligned}G_{motor}(s) &= \frac{A}{(1 + sT_e)(1 + sT_m)} \\T_e &= \frac{L}{R} \\T_m &= \frac{J}{B}\end{aligned}\tag{C.1}$$

where T_e is the electrical circuit time constant and T_m is the mechanical (motor) time constant. Often the electrical time constant is far smaller than the mechanical time constant for motors $T_e \ll T_m$, as such the model is often approximated by the first order dynamic system:

$$G_{motor}(s) = \frac{A}{1 + Ts}\tag{C.2}$$

The values for A and T were regressed from step test data using the Nelder-Mead method [21], which is a commonly used non-linear optimisation technique.

A graphical example of the sampled versus modelled data is shown in Figure C.1.

A number of different step tests, operating on different ranges, were done. Using Nelder-Mead, the values of A and T were optimised on to minimise error between sampled data and the model data.

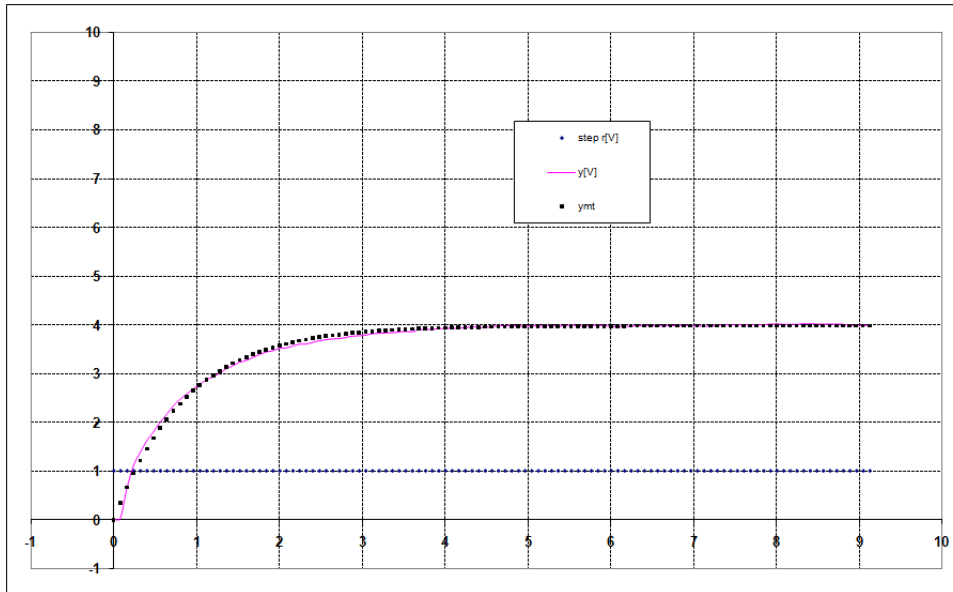


Figure C.1: Graph of Measured y [V] and Simulated DC motor ymt for a step input

Using these tests and Nelder-Mead, the final values of the plant gain and time constant were $A = 3.97$ and $T = 0.87$. Giving the final model used in the simulations in C.3.

$$G_{motor}(s) = \frac{(3.97 \pm 0.02)}{1 + s(0.87 \pm 0.02)} \frac{[V]}{[V]} \quad (C.3)$$

Pole-Zero Cancellation is a common phenomenon that occurs in Control engineering. It can be explained by examining the various transfer functions in a given control structure.

For each of the inputs: Setpoint, Input Disturbance and Output Disturbance, there exists a transfer function mapping it to the output. Using the control structure shown in Figure C.2, the derivation of these transfer functions is shown below.

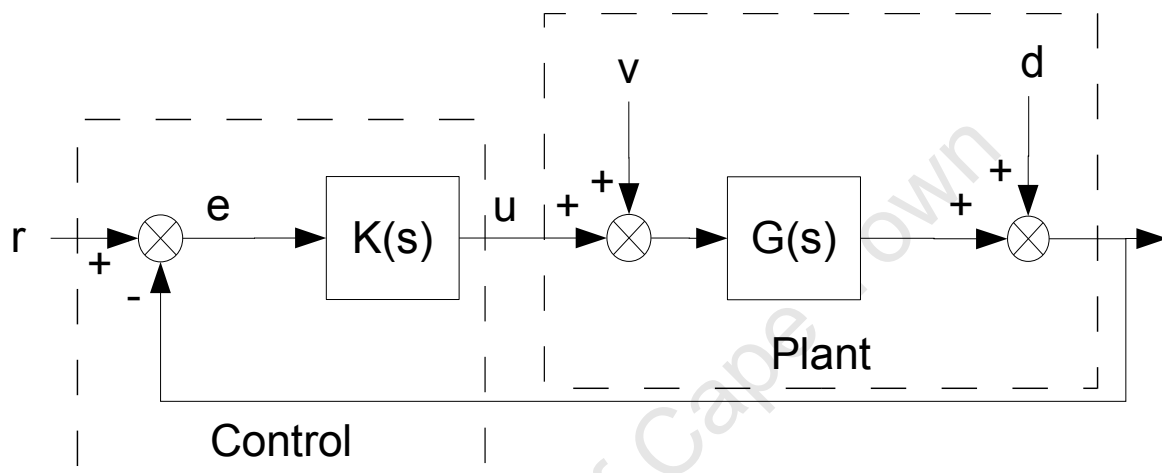


Figure C.2: General Control Setup

Mathematically these case be shown as follows:

For setpoint r to output y :

$$\begin{aligned}
 r &\rightarrow y \\
 y &= eKG \\
 e &= r - y \\
 y &= (r - y)KG \\
 y &= rKG - yKG \\
 y(1 + KG) &= rKG \\
 \frac{y}{r} &= \frac{KG}{1 + KG}
 \end{aligned} \tag{C.4}$$

Similarly for input disturbance v to output y it can be shown:

$$\frac{y}{v} = \frac{G}{1 + KG} \tag{C.5}$$

And for output disturbance d to output y :

$$\frac{y}{d} = \frac{1}{1 + KG} \tag{C.6}$$

Now considering, as an example, the case of PI control applied to a simple first order plant. Using the control loop shown in Figure 3.1 and the PI equations from 2.6. In order to simplify the equations a change of notation is used in C.7:

$$\mathbf{K} = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} = \frac{B(1 + sT_1)}{s} \tag{C.7}$$

The first order plant:

$$\mathbf{G} = \frac{a}{1 + sT_2} \tag{C.8}$$

In order for pole-zero cancellation to occur, T_1 must equal T_2 . Hence:

$$KG = \frac{B(1 + sT_1)}{s} \cdot \frac{A}{1 + sT_2} = \frac{AB}{s} \quad (\text{C.9})$$

Applying this result to the transfer functions above we get:

$$\frac{y}{r} = \frac{\frac{AB}{s}}{1 + \frac{AB}{s}} = \frac{A}{s + AB} \quad (\text{C.10})$$

$$\frac{y}{v} = \frac{\frac{A}{1 + sT_2}}{1 + \frac{AB}{s}} = \frac{As}{(1 + sT_2)(s + AB)} \quad (\text{C.11})$$

$$\frac{y}{d} = \frac{1}{1 + \frac{AB}{s}} = \frac{s}{s + AB} \quad (\text{C.12})$$

The transfer functions for r and d have their closed loop poles dependant only on the plant and control gains A and B , and thus can in theory be positioned wherever appropriate. On the other hand, the transfer function for v has two closed poles. The same pole as in r and d at $s = -AB$, and the open loop plant pole at $s = -\frac{1}{T_2}$. This pole does not depend on either the plant or control gain and as such is remains where it is. The issue is then that all input disturbances are controlled at a much slower rate than the setpoint and output disturbances, which are generally made fast relative to the open loop. This slow pole can be avoided by ensuring that pole-zero cancellation does not occur.

APPENDIX D

Performance Measure Calculation

D.1 Hypervolume

D.1.1 Unary Hypervolume

The hypervolume quality indicator I_H can be very difficult to calculate if the shape is unknown. It is even more complicated when the dimensionality of the problem is increased.

A number of different methods exist [33] [4] [30] for calculating the hypervolume metric. For this thesis, a simpler statistical method is used to calculate I_H . It relies on calculating what percentage of a random sampling of points is dominated by a given Pareto front, within a bounded hypercube. The number of random points used is N , and they are generated using a uniform random distribution.

The volume of an given hypercube (n-cube) is easily calculated, thus if one can calculate what percentage of that volume another shape occupies, then the volume of that shape can be easily calculated.

Consider an example in 2D. A Pareto front in 2D is a curve of some description, in this case a convex line. A hypercube in 2D is a square/rectangle.

This is shown in Figure D.1, where 3 in 10 of the random points fall within the hypervolume (in this case the shaded area). If the volume of the cube was known, then the shaded hypervolume can easily be calculated from the percentage of points falling

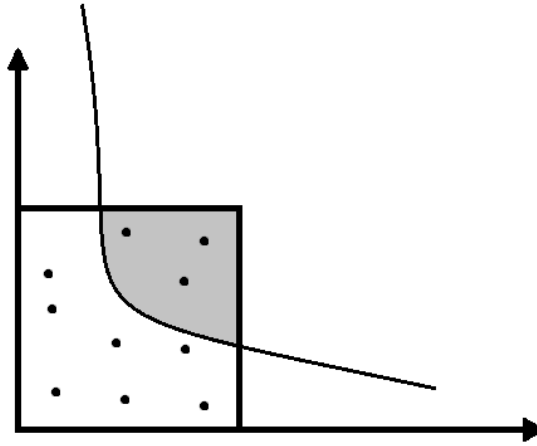


Figure D.1: Hypervolume Example

inside it.

While this is very inaccurate for lower values of N (where N is the total number of random points used), the higher N is, the more accurate it becomes.

The selection of the hypercubes bounds is done based on an ideal point and a worst (nadir) point. In the case of this thesis, these are the same ideal and worst points used in the normalisation process used for the Level Diagrams. That is, the Pareto front approximation objective minima and maxima respectively.

The hypervolume calculation can be summarised as follows:

1. Get Approximate Pareto front Ω_p^* .
2. Find ideal i and worst r points.
3. Generate field of N random uniformly distributed points within the hypervolume bounded by vertices i and r .
4. Determine what percentage of N is dominated by Ω_p^*

Empirical testing was used to select N , such that the accuracy was acceptable. This was done as shown in Table D.1:

Name	1	2	3	Avg	Stddev	Percentage
$N = 10000$						
PI	0.8491	0.8542	0.8507	0.851333	0.002608320	0.306380582%
SMC	0.5379	0.5315	0.5410	0.536800	0.004844585	0.902493406%
PIc	0.6866	0.6836	0.6816	0.683933	0.002516611	0.367961518%
$N = 100000$						
PI	0.85026	0.85067	0.85044	0.850456667	0.000205508	0.024164371%
SMC	0.54063	0.53727	0.54154	0.539813333	0.002249096	0.416643305%
PIc	0.68498	0.68614	0.68893	0.686683333	0.002030279	0.295664541%
$N = 1000000$						
PI	0.852271	0.851892	0.851982	0.852048333	0.000198016	0.023239995%
SMC	0.539544	0.539902	0.539056	0.539500667	0.000424661	0.078713792%
PIc	0.687212	0.686271	0.686449	0.686644000	0.000499889	0.072801770%

Table D.1: Empirical testing of I_H

$N = 100000$ was selected for use as it had a sample standard deviation of less than 0.5% for each of the different data sets. Further, each I_H was checked for all results for each problem to ensure that this accuracy held.

D.1.2 Binary Hypervolume

The binary hypervolume quality indicator $I_{H2}(A, B)$ is the hypervolume dominated by Pareto front A, not including any volume also dominated by Pareto front B.

$$I_{H2}(A, B) = I_H(A \cup B) - I_H(B) \quad (\text{D.1})$$

Thus the procedure for calculating the Binary hypervolume $I_{H2}(A, B)$ is:

1. Calculate the unary hypervolume for Pareto front $A \cup B$.
2. Calculate the unary hypervolume for Pareto front B
3. Calculate $I_H(A \cup B) - I_H(B)$

Considering that the I_H had a sample standard deviation of less than 0.5%, the additive

sample standard deviation of I_{H_2} is less than 1%.

APPENDIX E

PDE Optimiser Parameter Testing

The PDE Optimiser relies on a number of heuristic parameters which are not directly tied to the problem specification, but rather affect the performance and resulting accuracy of any Pareto front approximation. These include: the number of generations, the population size, the crossover and mutation rates.

Some of these parameters have a direct and obvious effect on the length of time it takes to complete any optimisation; the larger they are, the longer the simulation. However, they also affect the overall accuracy of the resulting Pareto front approximations.

Empirical testing was done on some of the parameters in order to select suitable values such that the Pareto fronts were deemed accurate enough, while making the optimisation not taking unreasonably long to complete.

E.1 Number of Generations

The final value for the number of generations was selected to be 30. This was done based on comparison of the resulting Pareto fronts using different numbers of generations.

An example of this is shown in the cost Level Diagrams in Figure E.1 where 3 Pareto front approximations are shown. (In this case using the resulting Pareto fronts from a PI controller). These resulted from 10 (light grey diamond), 20 (dark grey cross) and 30 (black dot) generations respectively.

As is apparent from cost Level Diagrams shown in Figure E.1. There is little variation

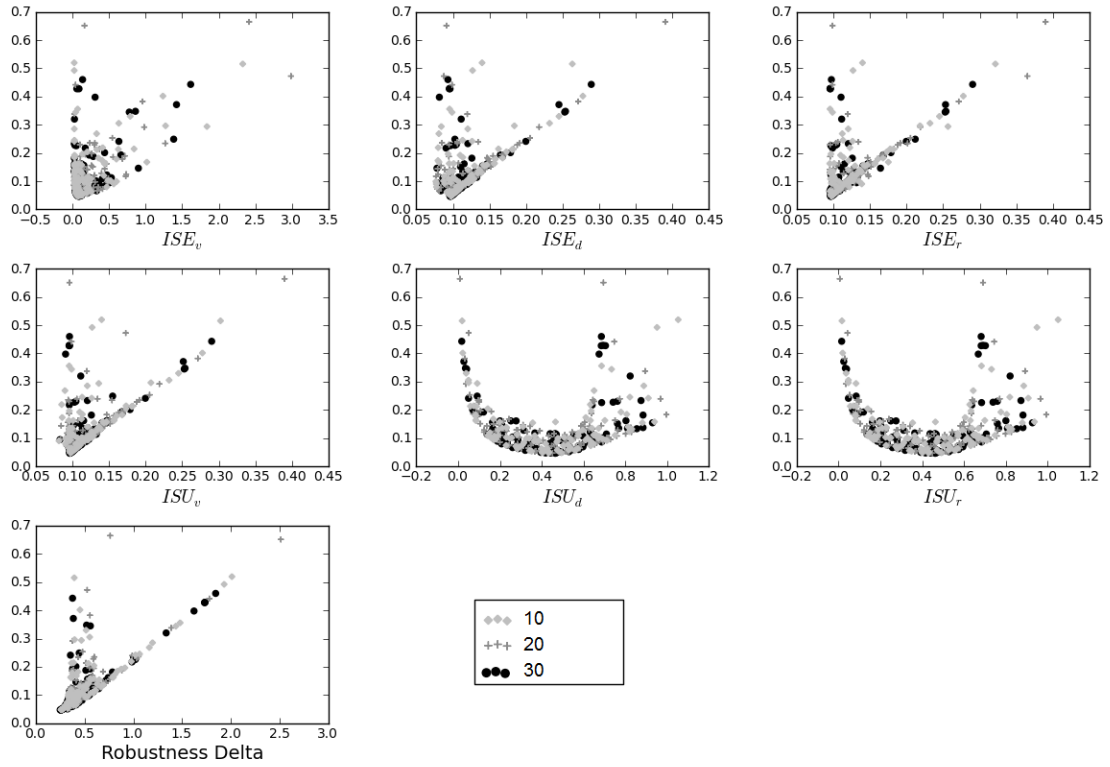


Figure E.1: Cost Level Diagrams of the Comparison of the Number of Generations.

in the overall shape of the resulting Pareto front beyond 10 Generations. The exact positions population members making up the approximation may vary, but the shape and general bounds do not.

Inspection of the input Level Diagrams in Figure E.2 reveals a similar situation, with only the ends of the Pareto front having minimal variation.

In terms of the Level Diagrams, the region of interest (i.e. the part of the Pareto front with the lowest synchronising norm (y-axis)) does not vary significantly in shape between the various fronts.

The performance measure results for comparing the three Pareto fronts are shown in Table E.1. From $I_H(G30) > I_H(G20) > I_H(G10)$ indicates G30 is not worse than G20,

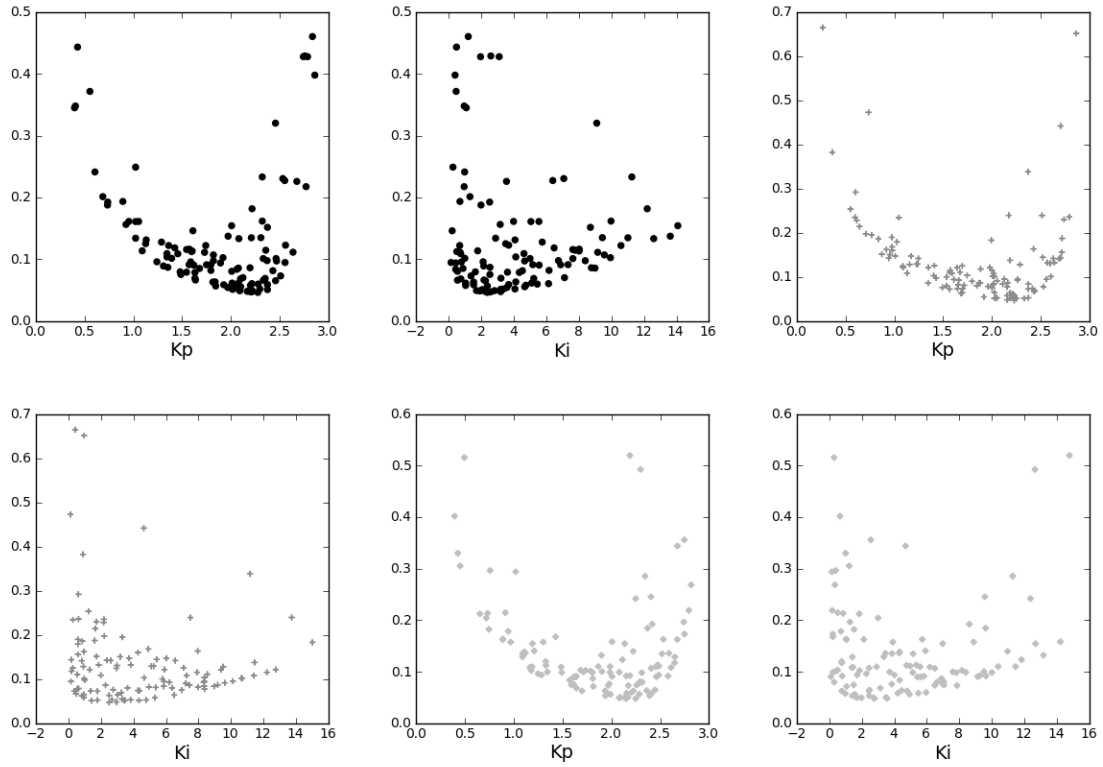


Figure E.2: Input Level Diagrams of the Comparison of the Number of Generations.

which is not worse than G10 as expected.

Name	I_H	I_{H2}	I_ϵ
<i>G30vsG20</i>			
G30	0.8554	0.0148	1.638
G20	0.8412	0.000	2.102
<i>G20vsG10</i>			
G20	0.8269	0.0146	2.347
G10	0.8126	0.000	3.435

Table E.1: Performance Measure Results for 30, 20 and 10 Generations

The interpretation function $I_{H2}(G30, G20) > 0$ and $I_{H2}(G20, G30) = 0$ indicates that $G30 \triangleright G20$, that is that using 30 generations is better than 20 generations. Similarly we get $G20 \triangleright G10$.

The I_ϵ quality indicator shows for all comparisons to be incomparable (all $I_\epsilon > 0$); but in each case the Pareto front with greater generations required less scaling in order to dominate the other front.

From this testing, 30 generations were selected, even though there was minimal variation between 10, 20 and 30 generations in most tests. It was decided that it would be better to err on the side of caution and give the optimiser longer to find the Pareto front, in case one of the test optimisations resulted in greater variation.

When using 30 generations, the Pareto front was found to have a reasonable degree of accuracy (with regard to generations), and could be completed in a reasonable time.

E.2 Population Size

In a similar fashion to the number of generations, the population size in each generation affects the accuracy of the Pareto front approximation and the optimisation length. The larger the population, the greater the accuracy of the Pareto front, but also the longer it takes. It is an application-specific variable

Another consideration that arose during testing was the issue of readability of the resulting Level Diagrams. While in theory larger populations mean better accuracy, they frequently made it more difficult to see the various Pareto front approximations being compared. It was found that in terms of comparisons, the general shape and position of the front was more important than definite hard curves which obscure the fronts behind them.

Figure E.3 shows an example of the resulting Pareto fronts for three different populations size optimisations: 200 (light grey diamond), 300 (dark grey cross) and 1000 (black dot).

The Performance Measure results for the comparison of population sizes are shown in

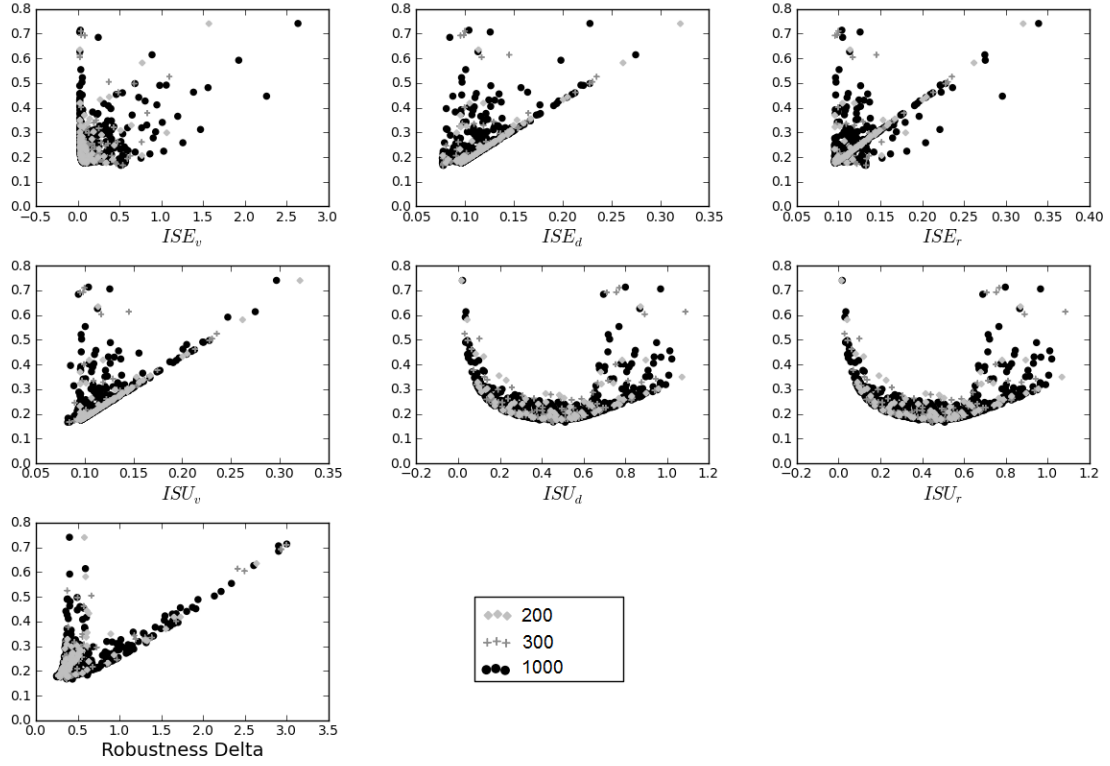


Figure E.3: Cost Level Diagrams of the Comparison of Population Sizes. Population = 200 (light grey diamond), 300 (dark grey cross), 1000 (black dot).

Table E.2. The I_H indicator shows that P1000 is not worse than P300, which is not worse than P200.

Name	I_H	I_{H2}	I_ϵ
$P_{1000} vs P_{300}$			
P1000	0.9741	0.0118	2.288
P300	0.9676	0.0053	3.152
$P_{300} vs P_{200}$			
P300	0.9847	0.0115	1.475
P200	0.9747	0.0015	1.663

Table E.2: Performance Measure Results for Population Size 1000, 300 and 200

The I_{H2} comparisons indicate that the sets are incomparable, but within experimental error we can apply the interpretation function $I_{H2}(P1000, P300) > 0$ and $I_{H2}(P300, P1000) =$

0, implying that the 1000 population Pareto front dominates the 300 population front. A similar relationship can be inferred with the comparison of the 300 and 200 population Pareto fronts.

The last quality indicator I_ϵ shows a similar situation as with the Generation comparisons above: All the fronts are incomparable, but if we look at the meaning of the indicator, the 1000 population front needs only a little scaling to dominate the 300 population fronts.

While the 1000 population size Pareto front is definitely more defined and covers a greater area of the front, the 300 population size Pareto front conveys the same general shape and position. Furthermore it does not completely obscure any fronts appearing behind it. Thus a population size of 300 was selected.

In some cases, where (on inspection of the Level Diagrams) the Pareto fronts appeared badly populated, the optimisations were rerun using high populations.

APPENDIX F

Performance Measure Relationships

F.1 Dominance Relationships

Table 2.1 in Section 2.5 shows the how dominance relationship terminology for comparison of individual vectors can be expanded for use with comparison of entire sets of vectors (in this case Pareto front approximation sets).

The Table 2.1 is repeated in Table F.1 for convenience.

relation	objective vectors		approximation sets	
strictly dominates	$\mathbf{J}^1 \succ \mathbf{J}^2$	\mathbf{J}^1 is better than \mathbf{J}^2 in all objectives	$A \succ \succ B$	every $\mathbf{J}^2 \in B$ is strictly dominated by at least one $\mathbf{J}^1 \in A$
dominates	$\mathbf{J}^1 \succ \mathbf{J}^2$	\mathbf{J}^1 is not worse than \mathbf{J}^2 in all objectives and better in at least one objective	$A \succ B$	every $\mathbf{J}^2 \in B$ is dominated by at least one $\mathbf{J}^1 \in A$
better			$A \triangleright B$	every $\mathbf{J}^2 \in B$ is weakly dominated by at least one $\mathbf{J}^1 \in A$ and $A \neq B$
weakly dominates	$\mathbf{J}^1 \succeq \mathbf{J}^2$	\mathbf{J}^1 is not worse than \mathbf{J}^2 in all objectives	$A \succeq B$	every $\mathbf{J}^2 \in B$ is weakly dominated by at least one $\mathbf{J}^1 \in A$
incomparable	$\mathbf{J}^1 \parallel \mathbf{J}^2$	neither \mathbf{J}^1 weakly dominates \mathbf{J}^2 nor \mathbf{J}^2 weakly dominates \mathbf{J}^1	$A \parallel B$	neither A weakly dominates B nor B weakly dominates A

Table F.1: cf.[35] Relations on Objective Vectors and Approximation Sets.

These can be understood by examining various situations. Figure F.1 shows the case where $A \succ \succ B$, every objective vector (\mathbf{J}^2) in the approximation set B is strictly dominated by at least one objective vector (\mathbf{J}^1) in approximation set A .

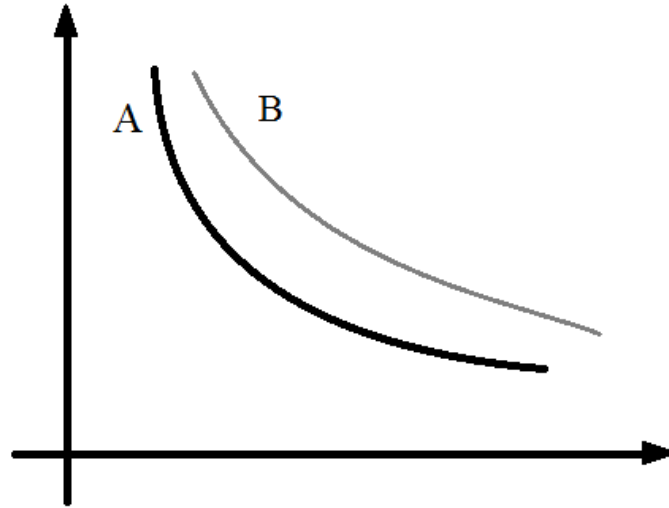


Figure F.1: Example of $A \succ\prec B$.

Further, the less strict relationships $A \succ B$, $A \triangleright B$ and $A \succeq B$ also apply.

Another example, shown in Figure F.2, give the case where $A \succ B$. This is true because while A strictly dominates B across most of the front, at the very right end of the sets A only dominates B . (i.e. at least one of the objective vectors \mathbf{J}^2 of approximation set B is only dominated by one of the objective vectors \mathbf{J}^1 of approximation set A .)

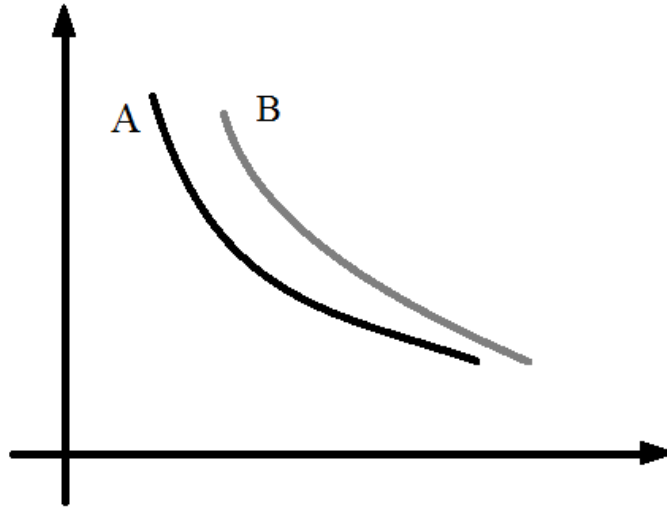


Figure F.2: Example of $A \succ B$.

As with the example above, less strict relationships of $A \triangleright B$ and $A \succeq B$ also apply.

A good example of the case where $A \triangleright B$ applies is when one approximation set is a subset of the other. An example of this is shown in Figure F.3. The less strict relationship $A \succeq B$ also applies.

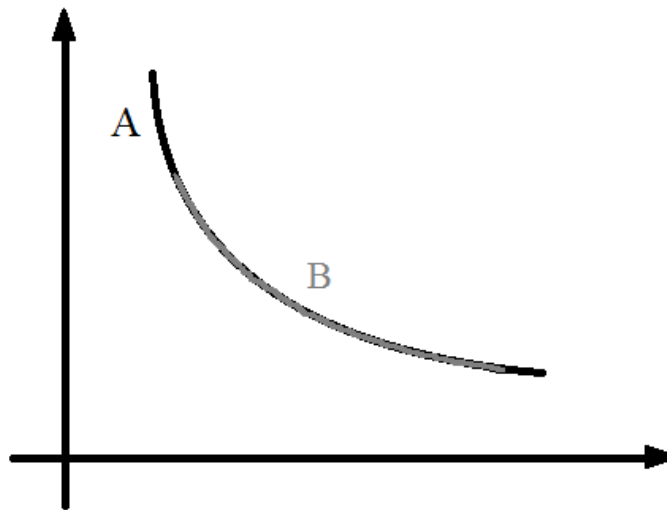


Figure F.3: Example of $A \triangleright B$.

Finally, when neither set exerts any dominance relationship on the other, or the relationship is mixed, the sets are said to be incomparable (in terms of dominance) $A \parallel B$. These situations are shown in Figures F.4 and F.5.

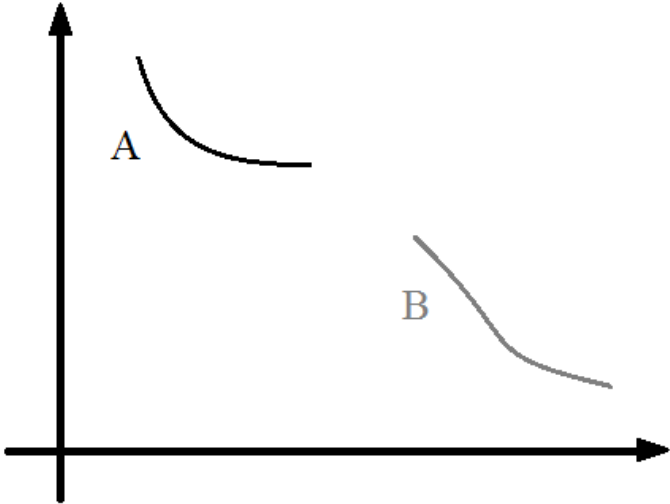


Figure F.4: Example of $A \parallel B$.

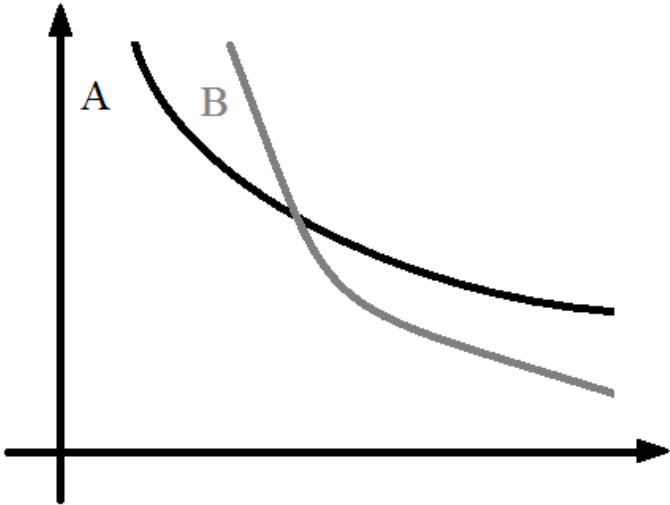


Figure F.5: Example of $A \parallel B$.

F.2 Quality Indicators and Interpretation Functions

Table 2.2 in Section 2.5 shows the various interpretation function relationships that can be applied using the three quality indicators in this thesis. Typically one would not test each interpretation function individually, but rather calculate the various quality indicators (I_H , I_{H2} and I_e) and see which (if any) of the relationships apply.

As an example, the binary hypervolume indicator interpretation functions are applied to the set dominance relationship examples shown above.

The first three examples demonstrate why only certain relationships can be inferred. The first example shown in Figure F.6, where A strictly dominates B , has $I_{H2}(A, B) > 0$ shown in light grey fill, and $I_{H2}(B, A) = 0$.

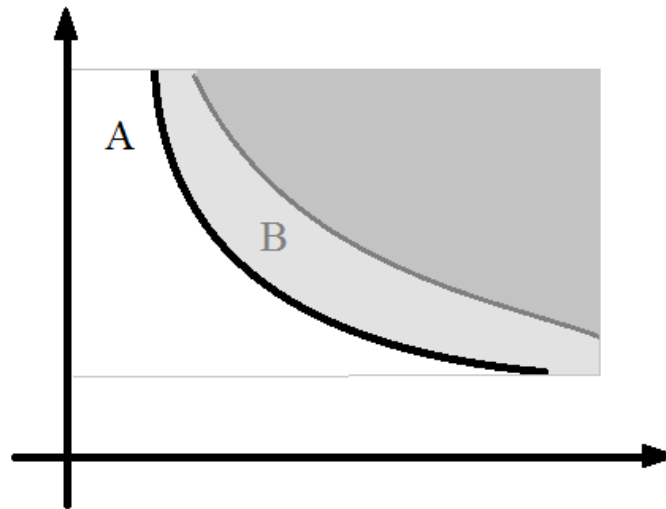


Figure F.6: $I_{H2}(A, B)$ Example: $A \succ \succ B$

In Figure F.7, A dominates B , but still has $I_{H2}(A, B) > 0$ and $I_{H2}(B, A) = 0$.

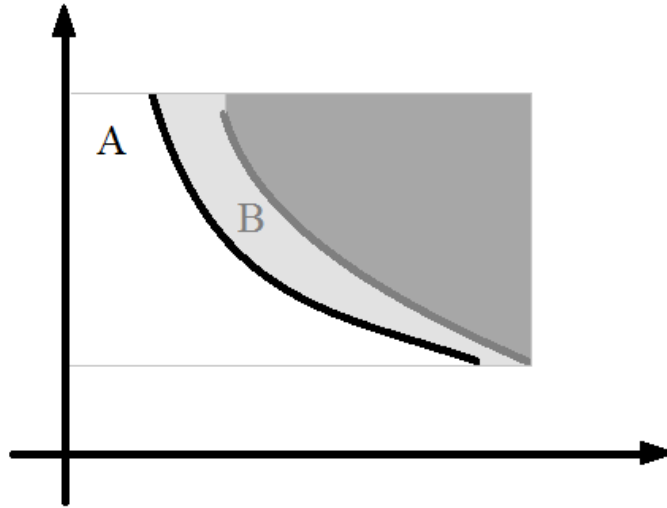


Figure F.7: $I_{H2}(A, B)$ Example: $A \succ B$

Next, Figure F.8 has A better than B . Also with $I_{H2}(A, B) > 0$ and $I_{H2}(B, A) = 0$. The light grey fill indicates a volume of the dominated objective space belonging exclusively to approximation set A

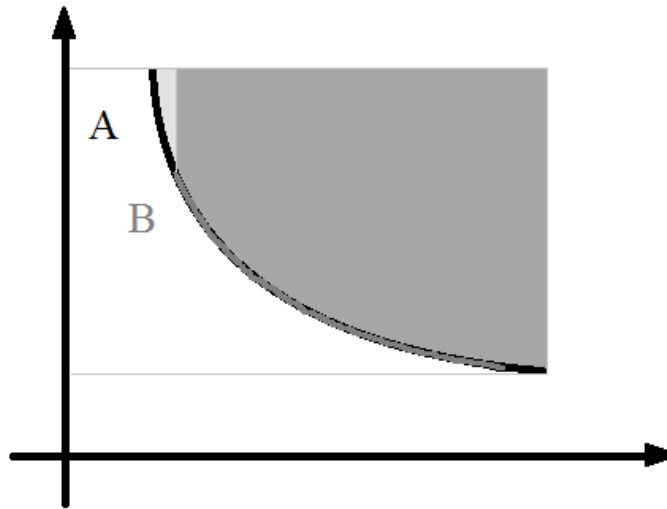


Figure F.8: $I_{H2}(A, B)$ Example: $A \triangleright B$

As was demonstrated above, these relationships vary in how strict their dominance

relationships are; with the stricter relationships also inferring the lesser relationships. Thus, I_{H2} is unable to say anything about strict dominance and dominance, but can indicate that A is better than B . That is to say that at best it can indicate $A \triangleright B$; a more strict dominance relationship may exist, but the I_{H2} quality indicator is unable to differentiate.

The relationship $A = B$ and the related interpretation function relationships $I_{H2}(A, B) = 0$ and $I_{H2}(B, A) = 0$ is obvious in its application. If approximation set A is the same as approximation set B , there will be no difference in their hypervolume, and as such will have $I_{H2} = 0$ in both cases.

Lastly, in the case where the sets are incomparable, two examples of which are shown in Figures F.9 and F.10, both $I_{H2}(A, B)$ and $I_{H2}(B, A)$ are non-zero (both shown in light grey in the figures).

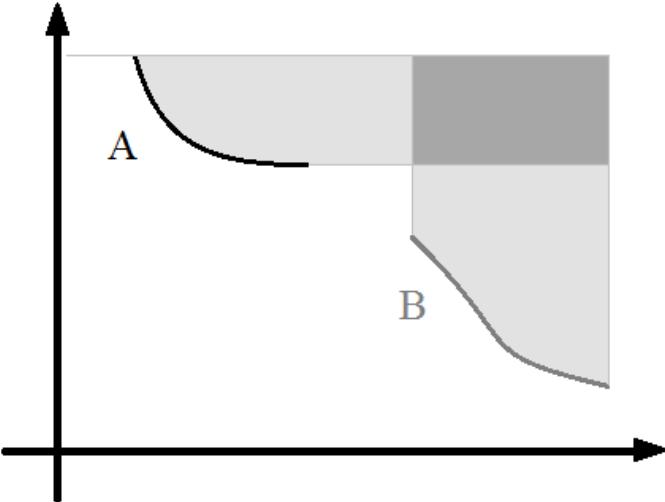


Figure F.9: $I_{H2}(A, B)$ Example: $A \parallel B$

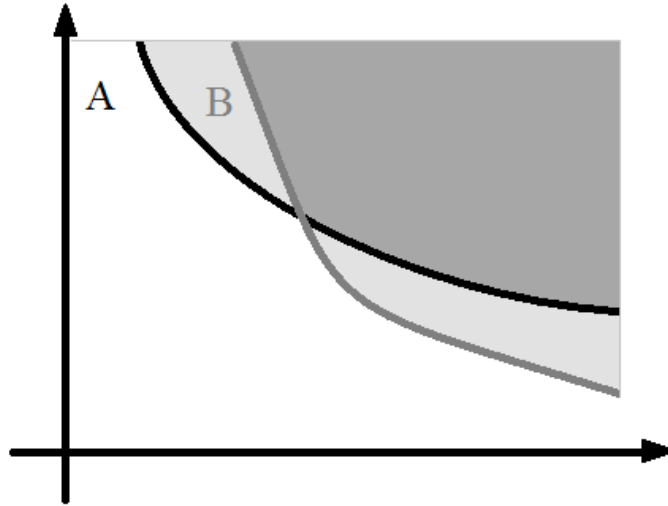


Figure F.10: $I_{H_2}(A, B)$ Example: $A \parallel B$

If both approximation sets have some volume of the dominated objective space then neither is better (in a Pareto dominance sense) than the other.

University of Cape Town