Controller tuning by means of evolutionary multiobjective optimization: a holistic multiobjective optimization design procedure

GILBERTO REYNOSO MEZA

EDITORIAL
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Collection doctoral thesis

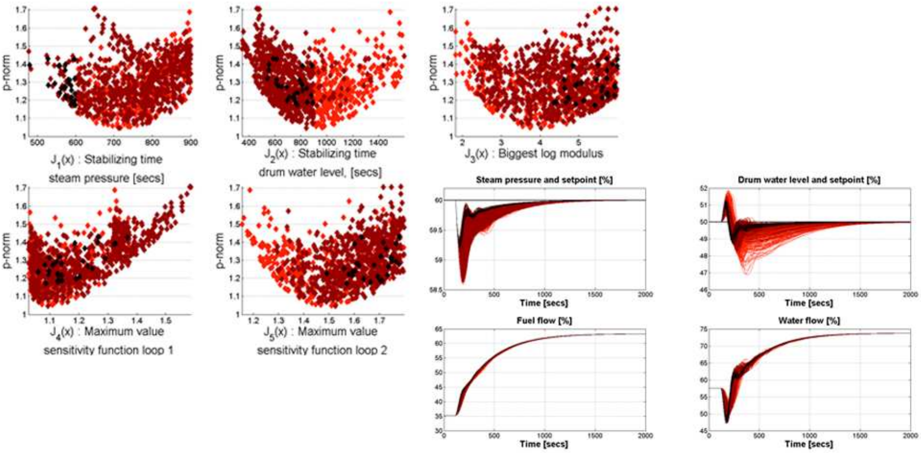
© 2014, of the present edition: Editorial Universitat Politècnica de València
Telf.: 963 877 012 / www.lalibreria.upv.es

ISBN: 978-84-9048-257-5 (printed version)

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Controller Tuning by Means of Evolutionary Multiobjective Optimization:

a Holistic Multiobjective Optimization Design Procedure



Gilberto Reynoso Meza

Supervisors:
Dr. Francesc Xavier Blasco Ferragud
Dr. Javier Sanchis Saez

June, 2014

UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Controller Tuning by Means of Evolutionary Multiobjective Optimization:

a Holistic Multiobjective Optimization Design Procedure

PhD dissertation by:
Gilberto Reynoso Meza

Supervisors:
Dr. F. Xavier Blasco Ferragud and **Dr. Javier Sanchis Saez**

cpoh
control y optimización

ai2
INSTITUTO DE
AUTOMÁTICA E
INFORMÁTICA
INDUSTRIAL

Instituto Universitario de Automática e Informática Industrial
Universitat Politècnica de València
June, 2014

This work was partially supported by:
Universitat Politécnica de Valéncia through the FPI fellowship FPI-2010/19.
The Spanish Ministry of Economy and Competitiveness through the projects ENE2011-25900 and TIN2011-28082.

PhD., MSc., Eng. Gilberto Reynoso Meza
`http://www.researchgate.net/profile/Gilberto_Reynoso-Meza/`
June, 2014

To the offspring and surviving individuals that shall evolve from this thesis.
Hoping to remain there for a couple more of iterations.

Agradecimientos

Antes que nada, quiero agradecer el gran apoyo que, a lo largo de estos años, me han brindado mis asesores Javi y Xavi. Gracias a su asesoramiento ésta investigación ha llegado a buen puerto; gracias a su amistad, he logrado encontrar un sitio dentro de la gran familia CPOH, a la que valoro y aprecio mucho. Dentro de esta familia, le doy las gracias a Miguel, que siempre ha estado al pendiente de mis avances; a Sergio por la oportunidad de trabajar con *cosas que se destruyen*; a JuanMa, por compartir su trabajo previo en optimización multiobjetivo, punto de partida de mi trabajo. De igual forma, agradezco a Manolo, Miguel M.L., Pedro y Santi por todos esos cafés en familia.

Agradezco a quienes me acogieron en las diversas estancias de investigación desarrolladas durante este tiempo. Specially, I would like to thank Peter F., for his valuable insights while I was reflecting about the *philosophy* behind the EMO process; A Max e Ed, per aprire la porta a nuovi problemi di ottimizzazione; A Leandro e Viviana, pelo apoio para o desenvolvimento de novos focos de pesquisa. Thank you Peter, grazie Max e Ed, obrigado Leandro e Viviana.

Le agradezco mucho a Patricia por arriesgarse conmigo como *compi de piso*; la convivencia durante este tiempo no tiene precio. Agradecido estoy con quienes tan resignada y pacientemente han soportado mi carácter, a cambio de alguna *guinness* en *plan abuelito*, durante estos años en la sala: Alejandro L., Alejandro V., Ana, Diego, Fátima, Gabi, José, José Luis, Manuel, Vanessa y Yadira. Les estoy agradecido también a Jesús V. y a Helem por ser el relevo generacional de este trabajo. Mucho éxito en sus enmiendas.

Gracias a tantos amigos que en diferentes etapas de este proceso me han apoyado en el extranjero: Angel, Cath, Han, Jazmín, Jessi, Jesús C., Laura, Mariana, Mario, Panchito, Saúl, Tere. Agradecido estoy con las diversas manadas de escaladores por las que vagué. Sin esta actividad, el último tramo de este proceso me hubiera consumido (*... o no*); en especial gracias a Eli, Manolo y Tito, quienes me motivan a *empujar el grado*.

Finalmente, pero jamás menos importantes, quiero agradecer a mis Padres y mi Familia. Sin la educación que recibí de ellos, sin las lecciones que aprendí, sin los valores inculcados, nada de esto podría haber sido posible. Le doy gracias a Dios por esta hermosa familia que me ha brindado.

Abstract

This thesis is devoted to Multiobjective Optimization Design (MOOD) procedures for controller tuning applications, by means of Evolutionary Multiobjective Optimization (EMO). With such purpose, developments on tools, procedures and guidelines to facilitate this process have been realized.

This thesis is divided in four parts. The first part, namely **Fundamentals**, is devoted on the one hand, to cover the theorical background required for this Thesis; on the other hand, it provides a state of the art review on current applications of MOOD for controller tuning.

The second part, **Preliminary contributions on controller tuning**, states early contributions using the MOOD procedure for controller tuning, identifying gaps on methodologies and tools used in this procedure. The contribution within this part is to identify the gaps between the three fundamental steps of the MOOD procedure: problem definition, search and decision making. These gaps are the basis for the developments presented in parts III and IV.

The third part, **Contributions on MOOD tools**, is devoted to improve the tools used in Part II. Although applications on the scope of this thesis are related to controller tuning, such improvements can also be used in other engineering fields. The first contribution regards the decision making process, where tools and guidelines for design concepts comparison in m-dimensional Pareto fronts are stated. The second contribution focuses on amending the gap between search process and decision making. With this in mind, a mechanism for preference inclusion within the evolutionary process is developed. With this it is possible to calculate pertinent approximations of the Pareto front; furthermore, it allows to deal efficiently with many-objective and constrained optimization instances.

Finally, in the fourth part, **Final contributions on controller tuning**, a stochastic sampling procedure for proportional-integral-derivative (PID) controllers is proposed, to guarantee that (1) any sampled controller will stabilize the closed loop and (2) any stabilizing controller could be sampled. Afterwards, two control engineering benchmarks are solved using this sampling strategy, the MOOD guidelines highlighted trough this thesis for multivariable controller tuning and the tools developed in Part III.

Resumen

Esta tesis está dedicada al proceso de diseño mediante optimización multiobjetivo (*MOOD*) para el ajuste de controladores, empleando algoritmos evolutivos multiobjetivo. Con dicho propósito, se han llevado a cabo desarrollos y mejoras en herramientas, metodologías y guías para facilitar este proceso.

Este documento está dividido en cuatro partes. La primera de ellas, llamada ***Fundamentals***, está dedicada por un lado a cubrir el marco conceptual requerido para ésta tesis; por otro lado, en ella se provee de un análisis en el estado del arte en el empleo del proceso *MOOD* para el ajuste de controladores.

En la segunda parte, ***Preliminary contributions on controller tuning***, se presentan contribuciones preliminares en el ajuste de controladores por medio del proceso *MOOD*, identificando áreas de oportunidad y mejora en las herramientas empleadas. La contribución de esta parte consiste en identificar lagunas, y proponer soluciones preliminares, entre los tres pasos básicos de este proceso: definición del problema, búsqueda de soluciones y selección. Dichas lagunas son la base sobre la que se desarrollan las contribuciones presentadas en las partes III y IV.

La tercera parte, ***Contributions on MOOD tools***, está dedicada a la mejora de las herramientas empleadas en la Parte II. Aunque las aplicaciones dentro del alcance de ésta tesis conciernen al ajuste de controladores, dichas herramientas pueden ser usadas en otras áreas de la ingeniería. La primera contribución es en la etapa de selección, donde se han elaborado herramientas para el análisis y comparación de conceptos de diseño en espacios m-dimensionales. La segunda contribución está relacionada con acercar el proceso de búsqueda de soluciones con la etapa de selección. Con ello en mente se ha desarrollado un nuevo mecanismo para la inclusión de preferencias en el proceso evolutivo de búsqueda; más allá, dicho mecanismo permite lidiar eficientemente con la optimización de muchos objetivos de diseño y con restricciones.

Por último, en la cuarta parte ***Final contributions on controller tuning***, se propone un muestreo estocástico para controladores PID estables orientado a los algoritmos evolutivos; con dicho muestreo se garantiza que cualquier controlador muestreado estabiliza el lazo de control y que todo controlador que estabiliza el lazo puede ser muestreado. Finalmente, se resuelven dos problemas de *benchmark* de ingeniería de control, empleando herramientas y guías de diseño para el proceso *MOOD* desarrolladas a lo largo de la Parte III para el ajuste de controladores multivariables.

Resum

Aquesta tesi està dedicada al procés de disseny mitjançant optimització multiobjectiu (*MOOD*) per a l'ajust de controladors, emprant algorismes evolutius multiobjectiu. Amb aquest propòsit, s'han dut a terme desenvolupaments i millores en eines, metodologies i guies per a facilitar aquest procés.

Aquest document està dividit en quatre parts. La primera d'elles, anomenada ***Fundamentals***, està dedicada d'una banda a cobrir el marc conceptual requerit per a aquest tesi; d'altra banda, en ella es proveeix d'un anàlisi de l'estat de l'art en aplicacions actuals del procés *MOOD* per a ajust de controladors.

En la segona part, ***Preliminary contributions on controller tuning***, es presenten contribucions preliminars en l'ajust de controladors per mitjà del procés *MOOD*, identificant àrees d'oportunitat i millora en les eines emprades. La contribució d'aquesta part consisteix en identificar llacunes, i proposar solucions preliminars, entre els tres passos bàsics d'aquest procés: definició del problema, cerca de solucions i selecció. Aquestes llacunes són la base sobre la qual es desenvolupen les contribucions presentades en les parts III i IV.

La tercera part, ***Contributions on MOOD tools***, és dedicada al desenvolupament i millora de les eines emprades en la Part II. Encara que les aplicacions dins de l'abast d'aquesta Tesi es centren en l'ajust de controladors, a-questes eines poden ser potencialment usades en altres àrees de l'enginyeria. La primera contribució és troba en la etapa de selecció, on són elaborades eines i guies per a l'anàlisi i comparació de diferents conceptes de disseny en espais m-dimensionals. La segona contribució està relacionada amb acostar el procés de cerca de solucions amb el del procés de selecció. Amb aquest propòsit un nou mecanisme per a la inclusió de preferències en el procés evolutiu de cerca és desenvolupat. Amb aquest mecanisme és possible calcular aproximacions al front de Pareto amb major pertinència; més enllà, aquest mecanisme permet fer front eficientment amb l'optimització amb restriccions i de molts objectius de disseny.

Finalment, en la quarta part ***Final contributions on controller tuning***, es proposa un mostreig estocàstic per a controladors PID estables orientat als algorismes evolutius; amb aquest mostreig es garanteix que qualsevol controlador mostrejat estabilitza el llaç de control i que tot controlador que estabilitza el llaç pot ser mostrejat. Finalment, dos problemes de *benchmark* d'enginyeria de control per a l'ajust de controladors multivariables són resolts, emprant les eines i guies de disseny per al procés *MOOD* desenvolupades al llarg de la Part III.

Contents

Main acronyms

AOF Aggregate objective function

BLT Biggest Log modulus tuning

D Desirable

DE Differential Evolution

DM Decision Maker

EA Evolutionary Algorithm

EMO Evolutionary Multiobjective Optimization

FEs Function evaluations

GFCL Generate First, Choose Later

GPP Global Physical Programming

HD Highly Desirable

HU Highly Undesirable

IAE Integral of the absolute value of error

ITAE Integral of the time weighted absolute value of error

LD Level Diagrams

LQC Linear Quadratic Controller

MCDM Multicriteria Decision Making

MIMO Multiple Input, Multiple Output

MODE Multiobjective Differential Evolution

MOEA Multiobjective evolutionary algorithm

MOOD Multiobjective Optimization Design Procedure

MOP Multiobjective Problem

PAc Parallel Coordinates

PI Proportional-integral

PID Proportional-integral-derivative

SCp Scatter plot

SISO Single Input, Single Output

sp-MODE Multiobjective Differential Evolution with Spherical Pruning

T Tolerable

TRMS Twin Rotor MIMO System

U Undesirable

Main symbols

$\boldsymbol{J_P^*}$ Approximated Pareto front

$\boldsymbol{J_P}$ Pareto front

$\boldsymbol{X_P^*}$ Approximated Pareto set

$\boldsymbol{X_P}$ Pareto set

$\boldsymbol{J}(\boldsymbol{\theta})$ Objective design vector

$J_i(\boldsymbol{\theta})$ *i*-th design objective

$\boldsymbol{\theta}$ Design alternative (decision) vector

$\boldsymbol{J}^{utopian}$ Utopian vector for design objectives

$\boldsymbol{J}^{nadir}$ The opposite design vector to $\boldsymbol{J}^{utopian}$

$\boldsymbol{J}^{T}$ The vector with the maximum value for each objective in the tolerable range.

$\boldsymbol{J}^{D}$ The vector with the maximum value for each objective in the desirable range.

$\boldsymbol{J}^{HD}$ The vector with the maximum value for each objective in the highly desirable range.

T_HypV: The hypervolume of the Pareto front approximation bounded by $\boldsymbol{J}^{T}$.

D_Hypv: The hypervolume of the Pareto front approximation bounded by $\boldsymbol{J}^{D}$.

HD_HypV: The hypervolume of the Pareto front approximation bounded by $\boldsymbol{J}^{HD}$.

T_$\boldsymbol{J}_P^*$: the Tolerable Pareto front approximation where all solutions dominate $\boldsymbol{J}^{T}$.

D_$\boldsymbol{J}_P^*$: the Desirable Pareto front approximation where all solutions dominate $\boldsymbol{J}^D$.

HD_$\boldsymbol{J}_P^*$: the Highly Desirable Pareto front approximation where all solutions dominate $\boldsymbol{J}^{HD}$.

List of Algorithms

List of Figures

List of Tables

Aims, Structure and Contributions of this Thesis

Aims, Structure and Contributions of this thesis

Oh, there were all sorts of things to wonder about, but the truth was simple: here stood this door alone on an endless stretch of beach, and it was for only one of two things: opening or leaving closed.

Stephen King, *The Dark Tower II.*

Justification & aim of this thesis

Satisfying a set of specifications and constraints required by real-control engineering problems is often a challenge. For parametric controller tuning, for example, these range from time-domain specifications (such as maximum overshoot, settling time, steady state error, and rise time) to frequency-domain requirements (noise rejection or stability margin, for example). Problems in which the designer must deal with the fulfillment of multiple objectives are known as multiobjective problems (MOPs).

It is common to define an optimization statement to deal with MOPs to calculate a solution with the desired balance among (usually conflictive) objectives. When dealing with an MOP, we usually seek an Pareto optimal solution [127] in which the objectives have been improved as much as possible without giving anything in exchange. Evolutionary Multiobjective Optimization (EMO) has been widely used to approximate the so-called Pareto set, where all solutions are Pareto optimal.

For a successful implementation of the EMO, a Multi-Criteria Decision-Making (MCDM) step needs to be carried out to select the most preferable solution from the set approximated in the EMO process. Hereafter, this holistic procedure (MOP definition, EMO process and MCDM step) will be known as the Multi-Objective Optimization Design (MOOD) procedure.

The aforementioned steps are important to guarantee the overall performance of a MOOD procedure. With a poor MOP definition, no matter how good the algorithms and MCDM methodology/tools are, the solutions obtained will not fulfill the decision maker's expectations. If the algorithm is inadequate for the problem at hand (regarding some desirable features to be commented), the decision maker (DM or designer) will not obtain a useful Pareto set to analyze; and therefore she (he) will not be able to select a solution that meets her (his) preferences. Finally, the incorrect use of MCDM tools and methodologies could imply a lower degree of embedment of the DM in the tradeoff analysis and solution selection. This last issue could easily discourage the DM from using a MOOD procedure.

Since the MOOD procedure provides the opportunity to obtain a set of solutions to describe the objective tradeoff for a given MOP, its use is worthwhile for controller tuning. Due to the fact that several specifications such as time and frequency requirements need to be fulfilled by the control engineer, a procedure to appreciate the tradeoff exchange for complex processes could be useful. Nevertheless, the EMO process and the MCDM step are usually handled separately. That is, there is an unbalanced interest in both processes. This thesis focuses on the MOOD procedure, addressing the specific problem of controller tuning, and bridging the gap between EMO and MCDM. For this purpose, several contributions have been made for each of the steps in this procedure. The major aim of this thesis is:

To provide a useful and decision making oriented framework for the MOOD procedure to improve its usability in controller tuning applications.

For this reason, the following specific aims are defined and covered through the chapters of this thesis:

1. Review the state of the art of MOOD procedures for controller tuning, to acquire an overall framework on the *work done* and the *work to be done*.

2. Identify gaps on developed tools and methodologies of the MOOD procedure for controller tuning, related with the expected quality of the Pareto front, the desirable features for EMO or in the MCDM step.

3. Develop proposals on tools and methodologies to improve the usability and performance of the MOOD procedure for controller tuning.

4. Test the aforementioned proposals in benchmark case studies.

5. Make available those developments by publishing results and by making code available for practitioners.

Structure of this dissertation

This thesis is divided into four parts, namely: Fundamentals (I); Preliminary contributions on controller tuning (II); Contributions on MOOD tools (III); and

Final contributions on controller tuning (IV). Part I is devoted to covering the theorical background required for this thesis and it also provides a state of the art review on current applications of MOOD for controller tuning. Part II states preliminary contributions using the MOOD approach for controller tuning and identifies gaps in the tools used. In Part III new tools for the MOOD procedure are presented to amend the aforementioned gaps, which are evaluated in Part IV with two control engineering benchmarks. Finally, some general conclusions and ideas for further research are discussed.

Contributions of this research

The thesis is contained within a research line in EMO. Its contribution focuses mainly on controller tuning applications; nevertheless, during its development, collaborations and developments in various engineering fields have been realized. The main contributions are the following:

- A new mechanism has been developed for the EMO process to improve the pertinency of solutions in the approximated Pareto front by means of preference inclusion using physical programming techniques. Furthermore, this mechanism enables constrained and many-objective optimization instances to be handled efficiently.
- Improvements for level diagram visualizations have been achieved for the MCDM step in order to provide a mechanism for m-dimensional design concept comparison. This mechanism enables the evaluation of various controller structures and MOEAs in a multidimensional space, so that strengths and drawbacks can be identified in each case.
- A new stochastic sampling procedure for PID-like controllers has been developed for controller tuning. This mechanism is useful for improving the convergence of EAs and MOEAs.
- Overall guidelines for the MOOD procedure for controller tuning are given through various benchmark case studies. These guidelines help define a controller tuning benchmark that is needed to evaluate the performance of algorithms and DM methodologies.

Contents of this thesis appear in the following publications:

Fundamentals, Chapter 1 :

- **G. Reynoso-Meza, J. Sanchis, X. Blasco, M. Martínez**. *Evolutionary Algorithms for PID controller tuning: Current Trends and Perspectives*. Revista Iberoamericana de Automática e Informática Industrial. 2013; 10: 251-268.

Fundamentals, Chapter 2 :

- **G. Reynoso-Meza, X. Blasco, J. Sanchis, M. Martínez**. *Controller tuning by means of evolutionary multiobjective optimisation: current trends and applications*. Control Engineering Practice. July 2014, Vol. 28, Pp. 58-73.

Preliminary contributions on controller tuning, Chapter 3 :

- **G. Reynoso-Meza, J. Sanchis, X. Blasco, J.M. Herrero**. *Multiobjective evolutionary algorithms for multivariable PI controller design*. Expert Systems with Applications. Vol. 39, Issue 9, July 2012. Pp. 7895-7907.

Preliminary contributions on controller tuning, Chapter 4 :

- **G. Reynoso-Meza, S. García-Nieto, J. Sanchis, X. Blasco**. *Controller tuning by means of multiobjective optimization algorithms: a global tuning framework*. IEEE Transactions on Control Systems. Vol. 21, Issue 2, March 2013. Pp. 445 - 458.

Contributions on MOOD Tools, Chapter 5 :

- **G. Reynoso-Meza, X. Blasco, J. Sanchis and J.M. Herrero**. *Comparison of design concepts in multi-criteria decision making using level diagrams*. Information Sciences, Vol. 221, Issue 1, February 2013. Pp. 124-141.

Contributions on MOOD Tools, Chapter 6 :

- **G. Reynoso-Meza, J. Sanchis, X. Blasco and S. García-Nieto**. *Physical Programming for preference driven Evolutionary Multiobjective Optimization*. Applied Soft Computing. Under review.

Final Contributions on Controller Tuning, Chapter 7 :

- **G. Reynoso-Meza, J. Sanchis, X. Blasco and J.M. Herrero**. *A stabilizing PID controller sampling procedure for stochastic optimizers*. The 19th World Congress of the International Federation of Automatic Control. Accepted on February 13th., 2014.

Also, a paper with content related with chapter 8 is under development.

Conference papers related with the aforementioned line of research are:

- **G. Reynoso-Meza, J. Sanchis, S. García-Nieto, J. Velasco**. *Ajuste de Controladores multivariables mediante algoritmos evolutivos multi-objetivo. Aplicación a un TRMS (Multivariable controller tuning by means of Evolutionary Multiobjective Optimization. A TRMS application).* IX Simposio CEA de Control Inteligente. Puerto de la Cruz, 26-28 de Junio, 2013.

- **G. Reynoso-Meza, X. Blasco, J.M. Herrero, J. Velasco**. *Ajuste de controladores monovariables mediante algoritmos evolutivos multiobjetivo. Aplicación en un robot manipulador (Controller tuning by means of Evolutionary Multiobjective Optimization. A Robotic manipulator example).* IX Simposio CEA de Control Inteligente. Puerto de la Cruz, 26-28 de Junio, 2013.

- **G. Reynoso-Meza, X. Blasco, J. Sanchis**. *Optimización Evolutiva Multiobjetivo y Selección Multicriterio para la Ingeniería de Control (Multi-objective evolutionary optimization and multicriteria decisión making for control engineering).* X Simposio Ingeniería de Control. Barcelona 1-2 Marzo 2012.

- **G. Reynoso-Meza, J. Sanchis, J.M. Herrero, C. Ramos**. *Evolutionary auto-tuning algorithm for PID controllers.* IFAC Conference on Advances in PID control. Pp. 1-6, 2012.

- **G. Reynoso-Meza, X. Blasco, J. Sanchis, S. García-Nieto**. *A multiobjective optimization design methodology for SISO PID controllers.* IFAC Conference on Advances in PID control. Pp.: 1-6, 2012.

- **G. Reynoso-Meza, M. Graff, J. Sanchis, X. Blasco**. *Análisis multi-objetivo y multi-criterio del Swedish Open Championship in robot control 2004 (Multiobjective and multicriteria analysis of the Swedish Open Championship in robot control 2004).* Asociación Mexicana de Mecatrónica. 10º Congreso Nacional de Mecatrónica, 257-262, 2011.

- **G. Reynoso-Meza, J. Sanchis, X. Blasco, and J.M. Herrero**. *Handling control engineering preferences: How to get the most of PI controllers.* In Proceedings of the IEEE congress on emerging technologies and Factory automation (ETFA2011). Toulouse (France). September 2011.

- **G. Reynoso-Meza, A. Montagud, J. Sanchis, J. Urchueguía**. *Simulation of the Synechocystis sp. PCC6803 metabolic behavior using stoichiometric representations and multiobjective evolutionary algorithms.* In Proceedings of the 12th international conference on systems biology (ICSB 2011). Heidelberg/Manheim (Germany). August 2011.

- **G. Reynoso-Meza, X. Blasco, J. Sanchis, Juan M. Herrero**. *An empirical study on parameter selection for multiobjective optimization algorithms using*

Differential Evolution. In IEEE Symposium Series on Computational Intelligence - SSCI 2011 Paris (France). April 2011

- **G. Reynoso-Meza, X. Blasco, J. Sanchis, M. Martínez**. *Multiobjective optimization algorithm for solving constrained single objective problems*. In IEEE World Congress on Computational Intelligence (WCCI 2010). Barcelona (Spain). July 2010.

- **G. Reynoso-Meza, J. Sanchis, X. Blasco**. *An Adaptive Parameter Control for the Differential Evolution Algorithm*. LNCS 5517. Pp. 375-382. 2009, Springer-Verlag.

- **G. Reynoso-Meza, J. Sanchis, X. Blasco**. 30 November - 2 December 2009. *Multiobjective design of a digital controller for the throttle control benchmark*. In: Memories of the IFAC Control Workshop on engine and powertrain control, simulation and modeling. IFP Rueil-Malmaison

Software development

Tools developed through this research are available at:

`www.mathworks.com/matlabcentral`

- Developed for this thesis:

MODE: Multiobjective Differential Evolution Algorithm with Spherical Pruning©. Software for MOEAs development at:
`.../fileexchange/38962`.

sp-MODE: Multiobjective Differential Evolution Algorithm with Spherical Pruning©. Software developed in [148] and basis of Chapter 6:
`.../fileexchange/39215`.

sp-MODE-II: Preference based Multiobjective Differential Evolution Algorithm with Spherical Pruning©. Software developed in Chapter 6:
`.../fileexchange/authors/289050`.

LD-Tool: Level Diagrams for multiobjective decision making and Design Concepts Comparison©. Software developed in [151] and presented in Chapter 5:
`.../fileexchange/39458`.

- Related with this thesis:

NNC: Normalized Normal Constraint (NNC) algorithm for multiobjective optimization
`.../fileexchange/38976`

T-DE: Hybrid Differential Evolution Algorithm With Adaptive Crossover Mechanism©. Software in Matlab Central: `.../fileexchange/39217`

Journal papers in collaboration[1] related with the aforementioned line of research are:

- **X. Blasco, G. Reynoso-Meza, S. García-Nieto**. *Resultados del Concurso de Ingeniería de Control 2012 y Convocatoria 2013 (Control Engineering Benchmark 2012 results and 2013 call for participation)*. Revista Iberoamericana de Automática e Informática industrial. Vol 10, Num 4 (2013) pp. 240-245.

- **J. A. Romero-Perez, O. Arrieta, F. Padula, G. Reynoso-Meza, S. García-Nieto, P. Balaguer**. *Estudio comparativo de algoritmos de auto-ajuste de controladores PID. Resultados del Benchmark 2010-2011 del Grupo de Ingeniería de Control de CEA [Comparative study of PID auto-tuning algorithms. Results of the 2011 Control Engineering Competition]*. Revista Iberoamericana de Automática e Informática industrial Vol. 9, Num. 2 (2012), pp. 182-193.

- **X. Blasco, S. García-Nieto, G. Reynoso-Meza**. *Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor. Simulación y evaluación de propuestas [Autonomous trajectory control of quadrotor vehicles. Simulation and proposals evaluation]*. Revista Iberoamericana de Automática e Informática industrial Vol. 9, Num. 2 (2012), pp. 194-199.

- **J.Sanchis, M. Martínez, X. Blasco, G. Reynoso-Meza**. *Modeling preferences in multiobjective engineering design*. Engineering Applications of Artificial Intelligence. Vol. 23, num. 8, pp. 1255-1264, 2010.

Conference papers in collaboration[2] related with the aforementioned line of research are:

- **A. Pajares, X. Blasco, G. Reynoso-Meza, J.M. Herrero Dura**. *Desarrollo de una herramienta para el análisis de datos multi-criterio. Aplicación en el ajuste de controladores del tipo PID (A Multi-criteria data analysis tool development. Application to PID controller tuning)*. XXXIV Jornadas de Automática, Terrasa, Barcelona, 4-6 Septiembre 2013.

- **H. Sabina, G. Reynoso-Meza, R. Vilanova, X. Blasco**. *Comparación de técnicas de optimización multi-objetivo clásicas y estocásticas para el ajuste de*

[1] While some tools and developments from this thesis were used on these works, their main results are not part of this dissertation. Such works are commented for informational purposes.

[2] *idem*

controladores PI [Comparison of classical and stochastic multiobjective optimization techniques for PI controller tuning]. XXXIV Jornadas de Automática, September 2013.

- **S. García-Nieto, G. Reynoso-Meza, J.F. Peñaranda-Foix, A. Borrell**. *Diseño óptimo multiobjetivo de PIDs para el control de temperatura en cavidades microondas [Multiobjective optimization design procedure of PIDs for microwave heating control]*. XXXIV Jornadas de Automática, September 2013.

- **J. Velasco, S. García-Nieto, G. Reynoso-Meza, J. Sanchis**. *Implementación de un sistema Hardware-In-the-Loop para la simulación en tiempo real de pilotos automáticos para UAVs (A Hardware in the loop implementarion for UAV's real time simulation)*. XXXIV Jornadas de Automática, Terrasa, Barcelona, 4-6 Septiembre 2013.

- **J. Velasco, S. García-Nieto, G. Reynoso-Meza, J. Sanchis**. *Desarrollo y evaluación de una estación de control de tierra para vehículos aéreos no tripulados [Development and evaluation of a ground control system por unmanned aerial vehicles]*. XXXIII Jornadas de Automática, September 2012.

- **A. Vignoni, G. Reynoso-Meza, X. Blasco, J. Picó**. *Marco de optimización multiobjetivo para el ajuste de parámetros en controladores con acondicionamiento de referencia por modos deslizantes: una primera aproximación (SMRC parameter tuning using a multiobjective optimization framework)*. In XXXII Jornadas de Automática. Sevilla (Spain). Septiembre 2011.

- **S. García-Nieto, J. V. Salcedo, M. Martínez, G. Reynoso-Meza**. *Iterative Discrete Forward-Backward Fuzzy Predictive Control*. In IEEE World Congress on Computational Intelligence (WCCI 2010). Barcelona (Spain). July 2010.

Final undergraduate project guidance:

- Pajares Ferrando, Alberto. Diseño multiobjetivo del sistema de gestión de la energía en vehículos híbridos. Universitat Politècnica de València, December 2013.

Part I

Fundamentals

PART I

Fundamentals

Firstly, this Part is devoted to present a theorical background on Multiobjective Optimization Design (MOOD) procedures by means of Evolutionary Multiobjective Optimization (EMO); secondly, it provides a state of the art review on current applications of MOOD procedures for controller tuning.

CHAPTER 1:

Tutorial on Multiobjective Optimization Design procedures

This chapter is devoted to present the theorical background on Multiobjective Optimization Design (MOOD) procedures by means of Evolutionary Multiobjective Optimization (EMO). Contents of this chapter appear in the following paper:

- **G. Reynoso-Meza, J. Sanchis, X. Blasco, M. Martínez**. *Evolutionary Algorithms for PID controller tuning: Current Trends and Perspectives*. Revista Iberoamericana de Automática e Informática Industrial. 2013; 10: 251-268.

1

Tutorial on Multiobjective Optimization Design procedures

Above all else, the mentat must be a generalist, not a specialist. It is wise to have decisions of great moment monitored by generalists. Experts and specialists lead you quickly into chaos. They are a source of useless nit-picking, the ferocious quibble over a comma. The mentat-generalist, on the other hand, should bring to decision-making a healthy common sense.

Frank Herbert, *The children of Dune*

1.1 Aim of this chapter

In this chapter a theorical background on Multiobjective Optimization Design (MOOD) procedures for engineering is given, specially oriented to controller tuning applications. The specific aims of this chapter are:

- To present the engineering design task as an optimization problem, to define the framework for the developments and contributions of this thesis.
- To define the MOOD procedure and its fundamental steps (measure, search, selection), to describe the overall design process used along this dissertation.

- To present the EMO in the MOOD procedure and its challenges, to validate its usage in such framework.
- To present elementary tools used in this work to state the basis of the developments in Section III.

The remainder of this chapter is as follows: in Section 1.2 an introduction on how to address an engineering design problem using an optimization statement is commented; in Section 1.3 a general background on multi-objective optimization is given, while in Section 1.4 the MOOD procedure is explained. In Section 1.5 basic tools and procedures (following the developments of [148]) are briefly commented. Finally, some concluding remarks are given.

1.2 Engineering design as a multiobjective optimization problem

In Figure 1.1, a general (maybe minimal) methodology to solve an engineering design problem by means of an optimization statement is presented. Firstly, the design problem has to be defined. This implies to identify decision variables $\boldsymbol{\theta}$ and design objectives $\boldsymbol{J} = [J_1(\boldsymbol{\theta}), \cdots, J_m(\boldsymbol{\theta})]$. If $m = 1$, it is said to deal with a singleobjective problem, while if $m \geq 2$ it is a multiobjective problem (MOP). A classical example of a single-objective optimization problem for controller tuning is the integral performance criteria IAE (integral of the absolute error) for a PI (proportional-integral) controller (see Figure 1.2). In such optimization statement, the sum of the absolute value of the error has to be minimized.

According to [119], there are two main approaches to solving an optimization statement for an MOP: the *aggregate objective function* (AOF) approach, and the *generate-first choose-later* (GFCL) approach. In the AOF context a single-index optimization statement that merges the design objectives is defined. A traditional example consist in using a weighting vector to indicate relative importance among objectives. In such cases, the decision maker (DM or simply the designer) needs to describe all the tradeoffs at once, at the beginning of the optimization process. A classical example (again), is the integral performance criteria ITAE (integral of the time weighted absolute error) for a PI controller (see Figure 1.2). In such optimization statement, the time weighted absolute value of the error is minimized. That is, two objectives are important to the designer: (minimize) the error and (minimize) the stabilizing time.

In the GFCL approach, the main goal is to generate many potentially desirable Pareto optimal solutions, and then select the most preferable solution. This is due to the impossibility of obtaining a solution that is *the best* for all objectives, and therefore several solutions with different tradeoff levels may appear. The selection takes place in a Multi-Criteria Decision-Making (MCDM)

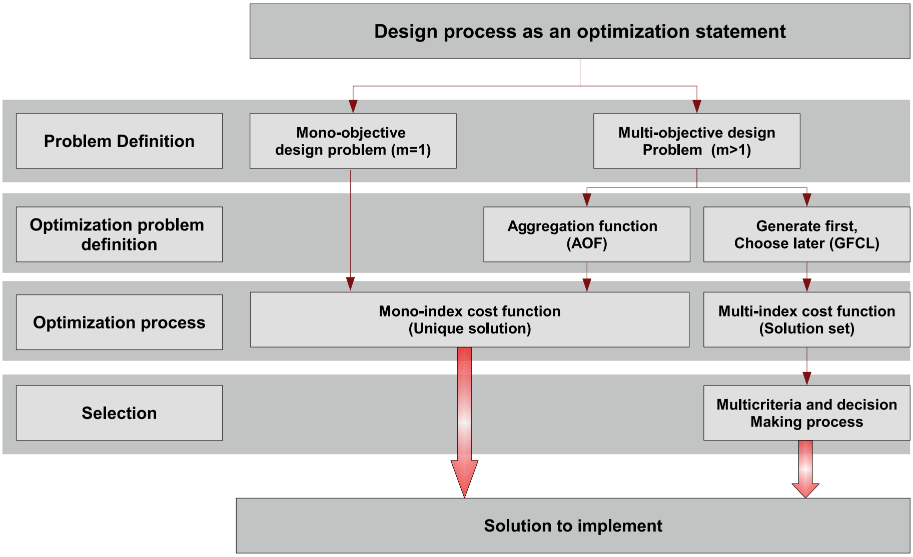

Design process as an optimization statement
Problem Definition
Mono-objective design problem (m=1)
Multi-objective design Problem (m>1)
Optimization problem definition
Aggregation function (AOF)
Generate first, Choose later (GFCL)
Optimization process
Mono-index cost function (Unique solution)
Multi-index cost function (Solution set)
Selection
Multicriteria and decision Making process
Solution to implement

Figure 1.1: Design methodology by means of optimization.

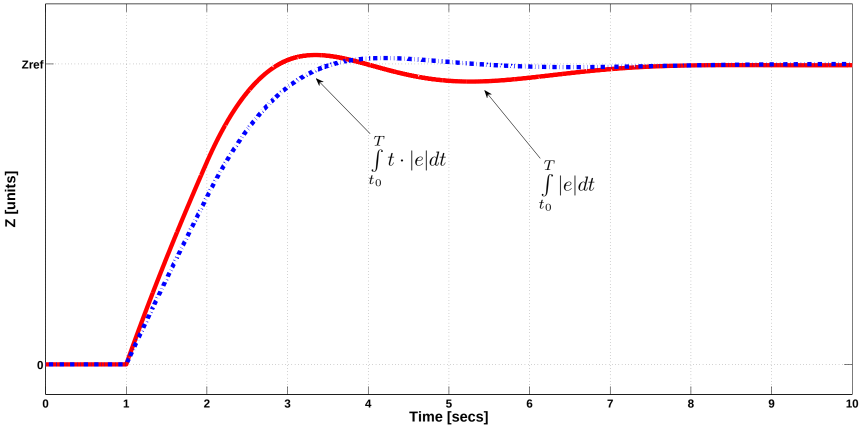


Figure 1.2: Integral performance criteria IAE(red full line) and ITAE (blue dotted line) as examples of mono-index and AOF optimization statements.

step, where the task of the DM is to analyze the tradeoff among objectives, and select the best solution according to his/her preferences. For the MCDM step, several techniques [60] and visualization approaches [106] have been used; this step may not be a trivial task since the analysis and visualization complexity increase with the number of objectives. Besides, this process could be more time consuming than the optimization process itself [32]. Continuing with the previous example, in the GFCL approach both objectives (error and stabilizing time) are minded separately, and the problem consists on finding other potential preferable solutions with different tradeoff. For example, in Figure 1.3, it can be noticed that the IAE criteria minimizes the error, at exchange of a larger stabilizing time; in counterpart, the ITAE minimizes the stabilizing time, at exchange of a larger error measure. The GFCL approach aims to find other solutions with different exchanges between those objectives.

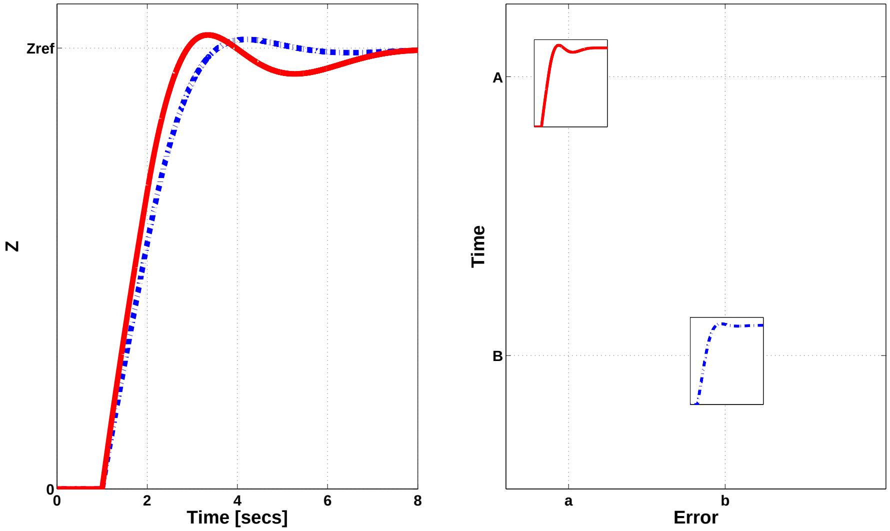


Figure 1.3: Integral performance criteria IAE(red full line) and ITAE (blue dotted line) as a multiobjective optimization statement.

One way to generate such sets of potential solutions in GFCL approach is by means of multiobjective optimization. This optimization approach seeks for a set of Pareto optimal solutions to approximate what is known as the Pareto set [127, 115]. A Pareto set approximation could help providing a preliminary idea of the objective space; and according to [21] it could be helpful when it is necessary to explain and justify the MCDM procedure. As drawbacks, more time and embedment from the DM in the overall process are required.

Classic techniques [127] to calculate this set of solutions have been used (such as varying weighting vectors, ϵ-constraint, and goal programming methods) as well as specialized algorithms (normal boundary intersection method [39], normal constraint method [121], and successive Pareto front optimization [162]). Nevertheless sometimes these problems could be complex, non-linear and highly constrained, situation which makes it difficult to find a useful Pareto set approximation. According to this, another way to approximate the Pareto set is by means of Evolutionary Multiobjective Optimization (EMO), which is useful due to the flexibility of MultiObjective Evolutionary Algorithms (MOEAs) in dealing with non-convex and highly constrained functions [32, 30]. Such algorithms have been successfully applied in several control engineering [61] and engineering design areas [167].

Regarding the GFCL framework, when the multiobjective optimization process is merged with the MCDM step for a given MOP statement, it is possible to define a multiobjective optimization design (MOOD) procedure [150]. This MOOD procedure can not substitute, in all instances, an AOF approach; nevertheless, it could be helpful for complex design problems, where a close embedment of the designer is needed. That is where an analysis on tradeoff could be helpful and necessary, before implementing a desired solution.

1.3 Background on Multiobjective Optimization

A MOP, without loss of generality,[3] can be stated as follows:

$$\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), \ldots, J_m(\boldsymbol{\theta})] \tag{1.1}$$

subject to:

$$\boldsymbol{g}(\boldsymbol{\theta}) \leq 0 \tag{1.2}$$

$$\boldsymbol{h}(\boldsymbol{\theta}) = 0 \tag{1.3}$$

$$\underline{\theta_i} \leq \theta_i \leq \overline{\theta_i}, i = [1, \ldots, n] \tag{1.4}$$

where $\boldsymbol{\theta}$ is defined as the decision vector with $\dim(\boldsymbol{\theta}) = n$; $\boldsymbol{J}(\boldsymbol{\theta})$ as the objective vector with $\dim(\boldsymbol{J}(\boldsymbol{\theta})) = m$; $\boldsymbol{g}(\boldsymbol{\theta})$, $\boldsymbol{h}(\boldsymbol{\theta})$ as the inequality and equality constraint vectors, respectively; and $\underline{\theta_i}, \overline{\theta_i}$ are the lower and upper bounds in the decision space for θ_i variable. As remarked previously, there is no single solution because in general there is no solution that is best for all objectives. Therefore, a set of solutions, the Pareto set, is defined. Each solution in the Pareto set defines an objective vector in the Pareto front. All solutions in the Pareto front are said to be a set of Pareto-optimal and non-dominated solutions:

[3] A maximization problem can be converted to a minimization problem. For each of the objectives that have to be maximized, the transformation: $\max J_i(\boldsymbol{\theta}) = -min(-J_i(\boldsymbol{\theta}))$ can be applied.

Definition 1. *(Pareto optimality [127]): An objective vector $\boldsymbol{J}(\boldsymbol{\theta}^1)$ is Pareto optimal if there is no other objective vector $\boldsymbol{J}(\boldsymbol{\theta}^2)$ such that $J_i(\boldsymbol{\theta}^2) \leq J_i(\boldsymbol{\theta}^1)$ for all $i \in [1, 2, \ldots, m]$ and $J_j(\boldsymbol{\theta}^2) < J_j(\boldsymbol{\theta}^1)$ for at least one j, $j \in [1, 2, \ldots, m]$.*

Definition 2. *(Strict Dominance [127]): An objective vector $\boldsymbol{J}(\boldsymbol{\theta}^1)$ is dominated by another objective vector $\boldsymbol{J}(\boldsymbol{\theta}^2)$ if $J_i(\boldsymbol{\theta}^2) < J_i(\boldsymbol{\theta}^1)$ for all $i \in [1, 2, \ldots, m]$.*

Definition 3. *(Dominance [127]): An objective vector $\boldsymbol{J}(\boldsymbol{\theta}^1)$ dominates another vector $\boldsymbol{J}(\boldsymbol{\theta}^2)$ if $\boldsymbol{J}(\boldsymbol{\theta}^1)$ is not worse than $\boldsymbol{J}(\boldsymbol{\theta}^2)$ in all objectives and is better in at least one objective; that is $\boldsymbol{J}(\boldsymbol{\theta}^1) \prec \boldsymbol{J}(\boldsymbol{\theta}^2)$*

Definition 4. *(Weak Dominance [127]): An objective vector $\boldsymbol{J}(\boldsymbol{\theta}^1)$ weakly dominates another vector $\boldsymbol{J}(\boldsymbol{\theta}^2)$ if $\boldsymbol{J}(\boldsymbol{\theta}^1)$ is not worse than $\boldsymbol{J}(\boldsymbol{\theta}^2)$ in all objectives.*

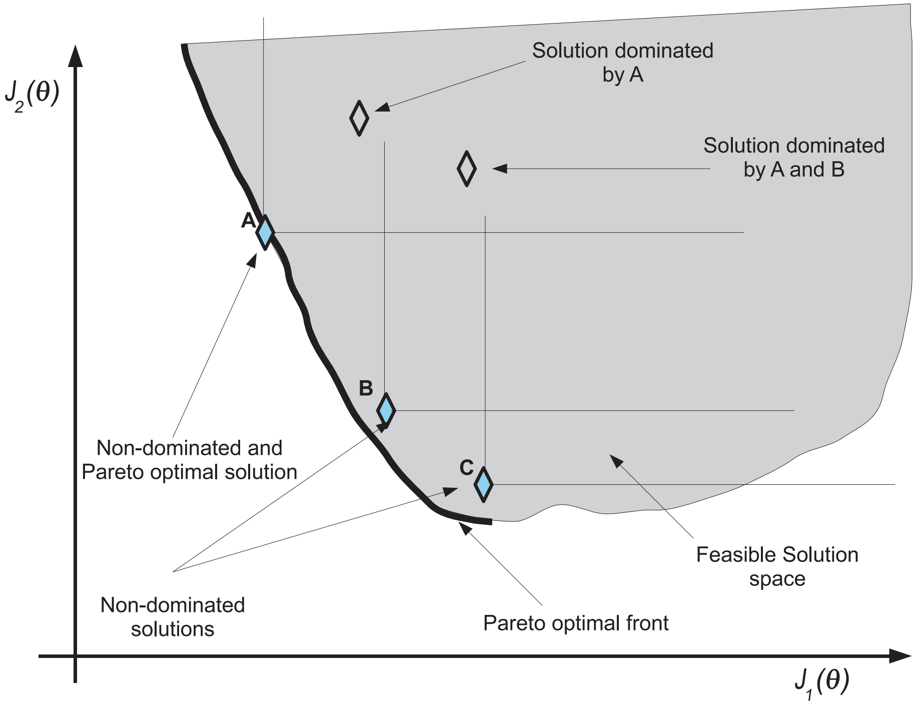


Figure 1.4: Pareto optimality and dominance definitions.

For example, in Figure 1.4, five different solutions (◊) are calculated to approximate a Pareto front (bold line). Solutions A, B, and C are non-dominated solutions, since there are no better solution vectors (in the calculated set) for all the objectives. Solutions B and C are not Pareto optimal, since some solutions (not found in this case) dominate them. Furthermore, solution A is also Pareto optimal, since it lies on the feasible Pareto front. The set of non-dominated solutions (A, B, and C) build the Pareto front $\boldsymbol{J}_P^*$ and Pareto set

$\boldsymbol{X}_P^*$ approximations. It is important to notice that most of the times the Pareto front is unknown and we shall only rely on approximations.

In [119], some refinement is incorporated into the Pareto front notion to differentiate design concepts. A Pareto front is defined given a design concept (or simply, a concept) that is an idea about how to solve a given MOP. This design concept is built with a family of design alternatives (Pareto-optimal solutions) that are specific solutions in the design concept. In the leading example, the PI controller is the design concept, whereas a specific pair of values of proportional and integral gains is a design alternative. For example, in Figure 1.5, a Pareto front approximation (bold line) for a particular design concept is calculated with a set of Pareto-optimal design alternatives (◊); we can state, for example, a PI controller for a given MOP as a design concept. But sometimes, there are different concepts, all of which are viable for solving an MOP (for example, we may use a PI *or* a fuzzy controller for a given process). Therefore, the DM can calculate a Pareto front approximation for each in order to make a comparison. Accordingly, in [119] the definition of a Pareto front and Pareto optimality were extended to a Pareto front for a set of concepts (s-Pareto front) where all solutions are s-Pareto optimal.

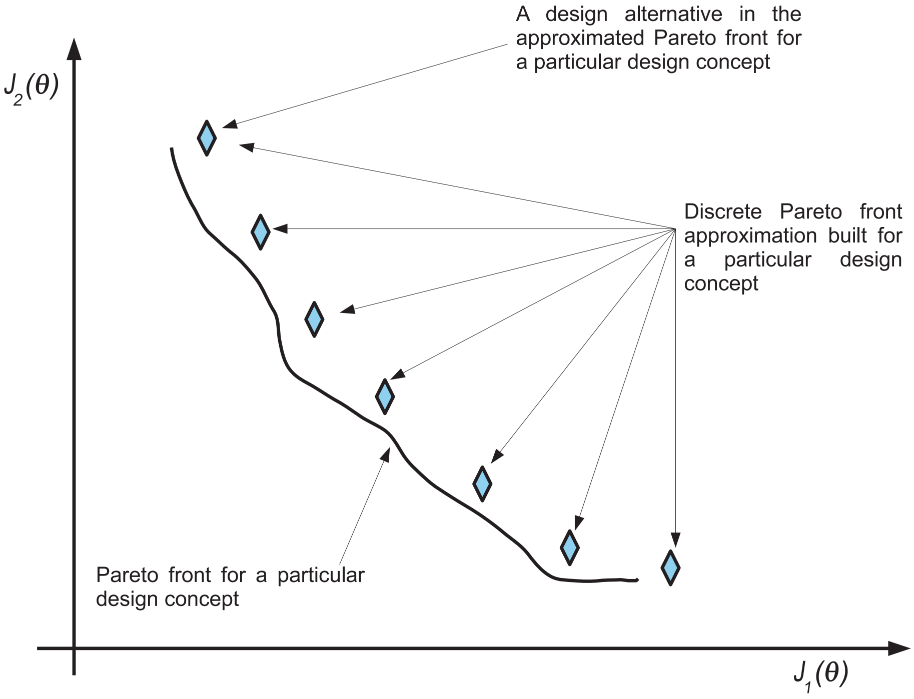

A design alternative in the approximated Pareto front for a particular design concept
$J_2(\theta)$
Discrete Pareto front approximation built for a particular design concept
Pareto front for a particular design concept
$J_1(\theta)$

Figure 1.5: Design concept and design alternative definitions.

Definition 5. *(s-Pareto optimality [119]): Given an MOP and K design concepts,* *an objective vector $\boldsymbol{J}(\boldsymbol{\theta}^1)$ is s-Pareto optimal if there is no other objective vector $\boldsymbol{J}(\boldsymbol{\theta}^2)$ in the design concept k such that $J_i(\boldsymbol{\theta}^2) \leq J_i(\boldsymbol{\theta}^1)$ for all $i \in [1, 2, \ldots, m]$ and all concepts k, $k \in [1, \ldots, K]$; and $J_j(\boldsymbol{\theta}^2) < J_j(\boldsymbol{\theta}^1)$ for at least one j, $j \in [1, 2, \ldots, m]$ for any concept k.*

Therefore, the s-Pareto front is built with the design alternatives of a set of K design concepts. In Figure 1.6 this notion is shown. Two different Pareto front approximations for two different concepts ($\triangle$ and $\Diamond$ respectively) are calculated (Figure 1.6a). In Figure 1.6b, an s-Pareto front with both design concepts is built.

As remarked in [119], a comparison between design concepts is useful for the designer, because he will be able to identify the concepts strengths, weaknesses, limitations and drawbacks. It is also important to visualize such comparisons, and to have a quantitative measure to evaluate strengths and weaknesses.

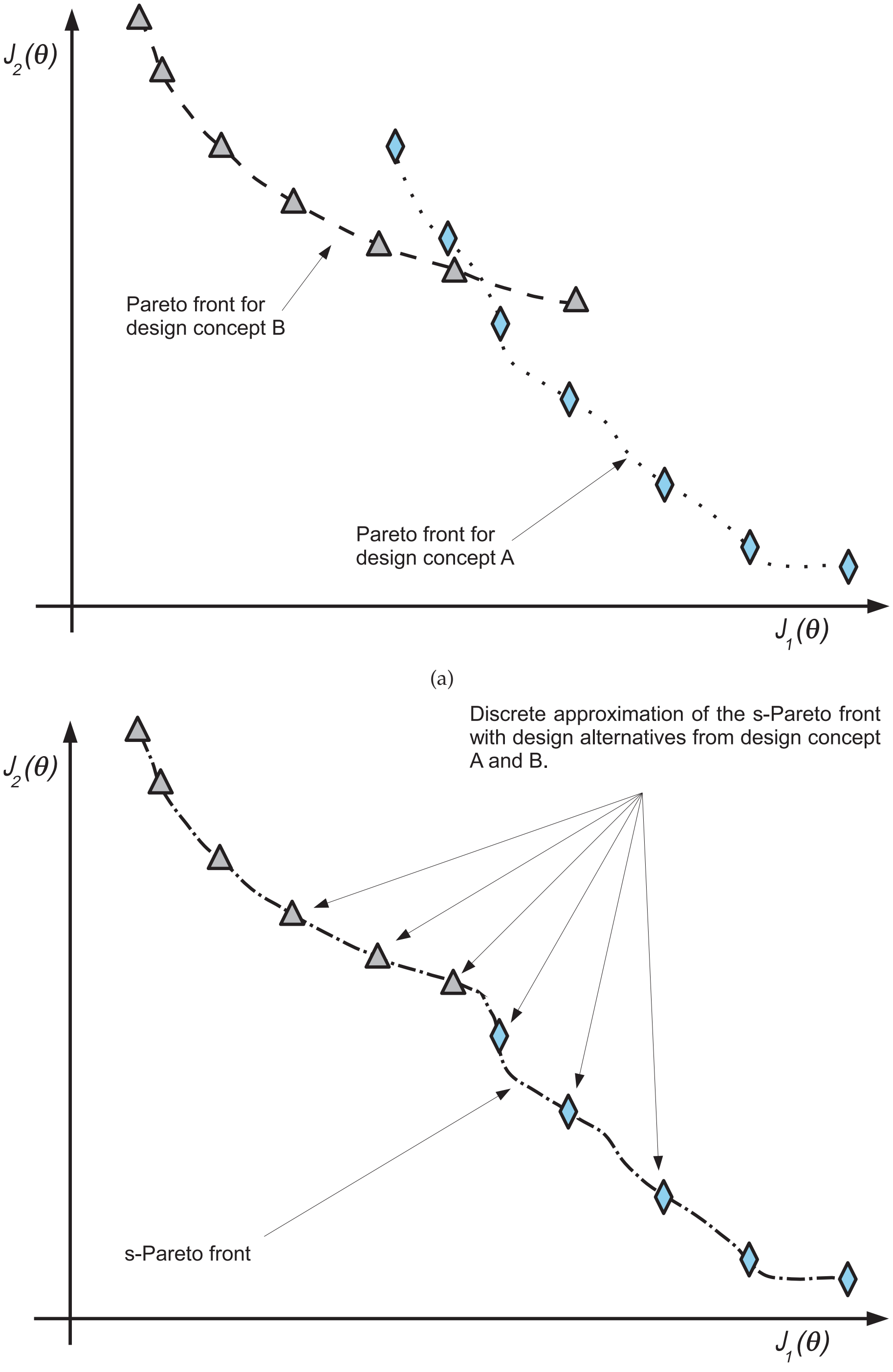
In the next section, it will be discussed how to incorporate such notions into a design procedure for multiobjective problems.

1.4 Multiobjective Optimization Design (MOOD) procedure

It is important to perform an integral procedure [21] minding equally the decision making and optimization steps [27]. Therefore, a general framework is required to successfully incorporate this approach into any engineering design process. A multiobjective optimization design (MOOD) procedure is shown in Figure 1.7. It consists of (at least) three main steps [31, 32]: the MOP definition (measurement); the multiobjective optimization process (search); and the MCDM stage (decision making).

1.4.1 Multiobjective Problem (MOP) definition

At this stage the design concept is defined (how to tackle the problem at hand); the engineering requirements (what is important to optimize); and the constraints (which solutions are not practical/allowed). In [119] it is noted that the design concept implies the existence of a parametric model that defines the parameter values (the decision space) that lead to a particular design alternative and its performance (objective space). This is not a trivial task, since the problem formulation from the point of view of the designer is not that of the optimizer [63]. Multi-objective problem definitions and their Pareto front approximation have been proposed in several fields as described in [30]; and reviews on rule mining [175], supply chains [4, 113], energy systems [52, 55], flow shop scheduling [185], pattern recognition [34], hydrological modeling



(a)

(b)

Figure 1.6: s-Pareto front definition.

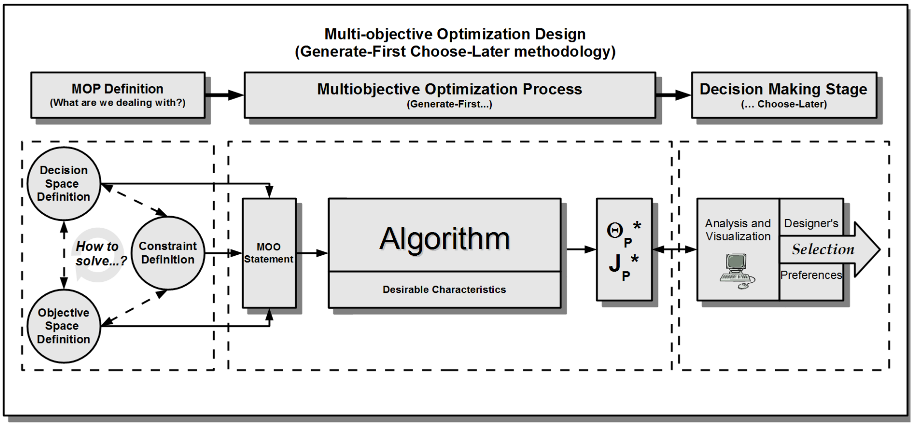

Multi-objective Optimization Design
(Generate-First Choose-Later methodology)
MOP Definition
(What are we dealing with?)
Multiobjective Optimization Process
(Generate-First...)
Decision Making Stage
(... Choose-Later)
Decision Space Definition
How to solve...?
Constraint Definition
Objective Space Definition
MOO Statement
Algorithm
Desirable Characteristics
Θ_P^*
J_P^*
Analysis and Visualization
Designer's
Selection
Preferences

Figure 1.7: A multiobjective optimization design (MOOD) procedure for control systems engineering.

[50], water resources [147], machining [197], and portfolio management [123] can be consulted by interested readers.

The designer will be searching for a preferable solution at the end of the optimization process. As this thesis is dedicated to control system engineering, the discussed design concepts will be entirely related to this field. As a controller must satisfy a set of specifications and design objectives, a MOOD procedure could provide a deep insight into controller's performance and capabilities. In exchange, more time is required for the optimization and decision making stages. Although several performance measurements are available, according to [5][4] the basic specifications cover:

- Load disturbance response
- Measurement noise response
- Set point response
- Robustness to model uncertainties.

It is worthwhile noting how the selection of the optimization objectives for measuring the desired performance can be achieved. A convenient feature of using MOEAs is the flexibility to select interpretable objectives for the designer. That is, the objective selection can be close to the point of view of the designer. Sometimes, with classical optimization approaches, a cost function is built to satisfy a set of requirements such as convexity and/or continuity; that is, it is built from the point of view of the optimizer, in spite of a possible

[4] Although specified in the context of PID control, they are applicable to all types of controllers.

loss of interpretability for the designer. Therefore, the multiobjective optimization statement is not a trivial task, since the problem formulation from the point of view of the designer is not that of the optimizer [63].

Given the MOP definition some characteristics for the MOEA to be used could be required. That is, according to the expected design alternatives, the MOEA would need to include certain mechanisms or techniques to deal with the optimization statement. Some examples are related to robust, multimodal, dynamic and/or computationally expensive optimization. Therefore, such instances could lead to certain desirable characteristics for the optimizer, which will be discussed below.

1.4.2 Evolutionary Multi-objective Optimization (EMO)

As noticed earlier, this thesis focuses on the MOOD procedure by means of EMO. Due to this, MOEAs will be discussed in this thesis. Some of the classical strategies to approximate the Pareto set include: Normal constraint method [121, 164], Normal boundary intersection (NBI) method [39], Epsilon constraint techniques [127] and Physical programming [122]. In [81], a Matlab© toolbox kit for automatic control[5] is developed that includes some of the aforementioned utilities for multiobjective optimization. For the interested reader, in [115, 127] reviews of general optimization statements for MOP in engineering are given.

MOEAs have been used to approximate a Pareto set [202], due to their flexibility when evolving an entire population towards the Pareto front. A comprehensive review of the early stages of MOEAs is contained in [33]. There are several popular evolutionary and nature-inspired techniques used by MOEAs. The former are based mainly on the laws of natural selection, where the fittest members (solutions) in a population (set of potential solutions) have the best chance of survival as the population evolves. The latter are based on the natural behavior of organisms. In both cases, they are used to evolve their populations towards the (unknown) Pareto front. Hereafter, it will be referred to both simply as evolutionary techniques.

The most popular techniques include Genetic Algorithms (GA) [174, 101], Particle Swarm Optimization (PSO) [96, 28], and Differential Evolution (DE) [183, 125, 41]. Nevertheless, evolutionary techniques as Artificial Bee Colony (ABC) [95] or Ant Colony Optimization (ACO) [49] algorithms are becoming popular. No evolutionary technique is better than the others, since each have drawbacks and advantages. These evolutionary/nature-inspired techniques require mechanisms to deal with EMO since they were originally used for singleobjective optimization. While the dominance criterion (Definition 3) could be used to evolve the population towards a Pareto front, it could be insufficient to achieve a minimum degree of satisfaction in other desirable character-

[5] Freely available at `http://www.acadotoolkit.org/`.

istics for a MOEA (diversity for instance). In Algorithm 1.1 a general structure for a MOEA is given. Its structure is very similar to that of almost any evolutionary technique ([61]): it builds and evaluates an initial population $P|_0$ (lines 1-2) and begins with the optimization (evolutionary) process (lines 5 to 10). Inside this optimization process, the evolutionary operators (depending on the evolutionary technique) will build and evaluate a new population (line 7-8), and the solutions with better cost function will be selected for the next generation (line 10). The main difference is regarding line 9, where the Pareto set approximation is built; according to the requirements of the designer, such process will incorporate (or not) some desirable features.

```
1  Build initial population $P|_0$ with $N_p$ individuals;
2  Evaluate $P|_0$;
3  Build initial Pareto set approximation $\boldsymbol{X}_P^*|_0$;
4  Set generation counter $G = 0$;
5  while convergence criteria not reached do
6      $G = G + 1$;
7      Build population $P^*|_G$ using $P|_G$ with an evolutionary technique;
8      Evaluate new population $P^*|_G$;
9      Build Pareto set approximation $\boldsymbol{X}_P^*|_G$ with $\boldsymbol{X}_P^*|_{G-1} \bigcup P^*|_G$ ;
10     Update population $P|_{G+1}$ with $P^*|_G \bigcup P|_G$;
11 end
12 RETURN Pareto set approximation $\boldsymbol{X}_P^*|_G$;
```

Algorithm 1.1: Basic MOEA

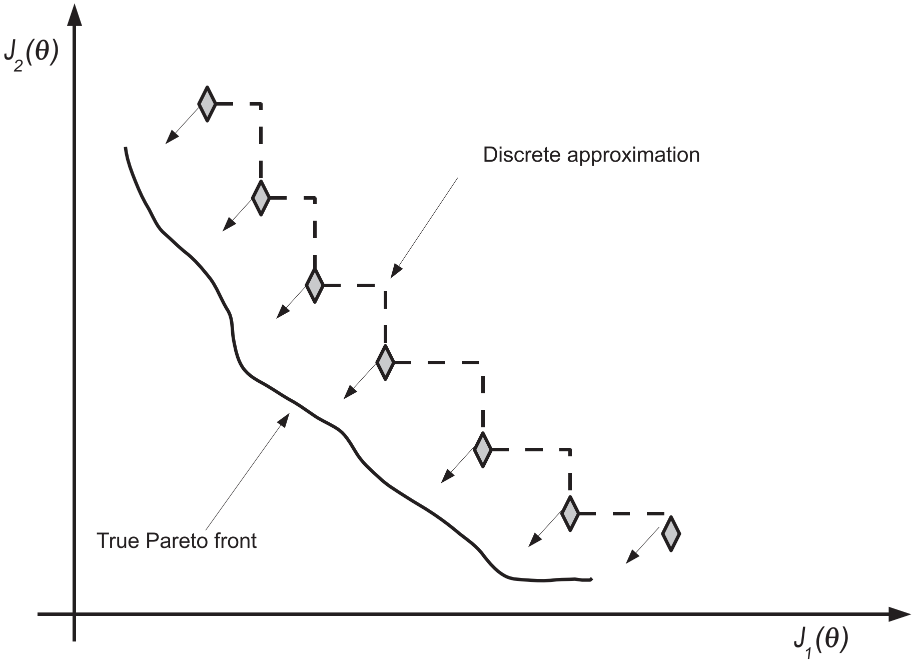
Desirable characteristics for an MOEA could be related to the set of (useful) solutions required by the DM or the MOP statement. Regarding the Pareto set sought, some desirable characteristics include (in no particular order) convergence, diversity and pertinency. Regarding the optimization statement, some features could be for handling constrained, many-objective, dynamic, multi-modal, robust, computationally expensive or large scale optimization instances. These desired characteristics are also a guide for appreciating current trends and on going research on EMO and MOEAs development [1], [202]. Some of which are explained below.

Feature 1. *Convergence*

Convergence refers to the algorithm's capacity to reach the real (usually unknown) Pareto front (Figure 1.8). It is known that convergence properties depend on the evolutionary parameters of an MOEA, modifying its exploitation and exploration capabilities [51]. In this sense, several adaptation mechanisms are provided, as well as several *ready to use* MOEAs with a default set of parameters, that are adjusted according several benchmarks. For example, the benchmarks of the CEC (congress on evolutionary computation) provide a good set of these algorithms, comprising evolutionary techniques as GA, PSO and DE (see for instance, [83, 200]). Another idea to improve the convergence properties of a MOEA is using local search routines through the evolutionary process. Such algorithms are known as memetic algorithms [134, 136].

Evaluating the convergence of a MOEA is not a trivial task, since it has to be evaluated the convergence of a Pareto front approximation. In two dimensions it could not be an issue, but in several dimensions it becomes more difficult. Several metrics have been used to evaluate the convergence properties (and other characteristics) for MOEAs [99, 206].

Convergence is a property common to all optimization algorithms; from the user's point of view, it can been said it is an expected characteristic. Nevertheless, in the case of MOEAs it could be insufficient, and another desired (expected) feature as diversity is required.



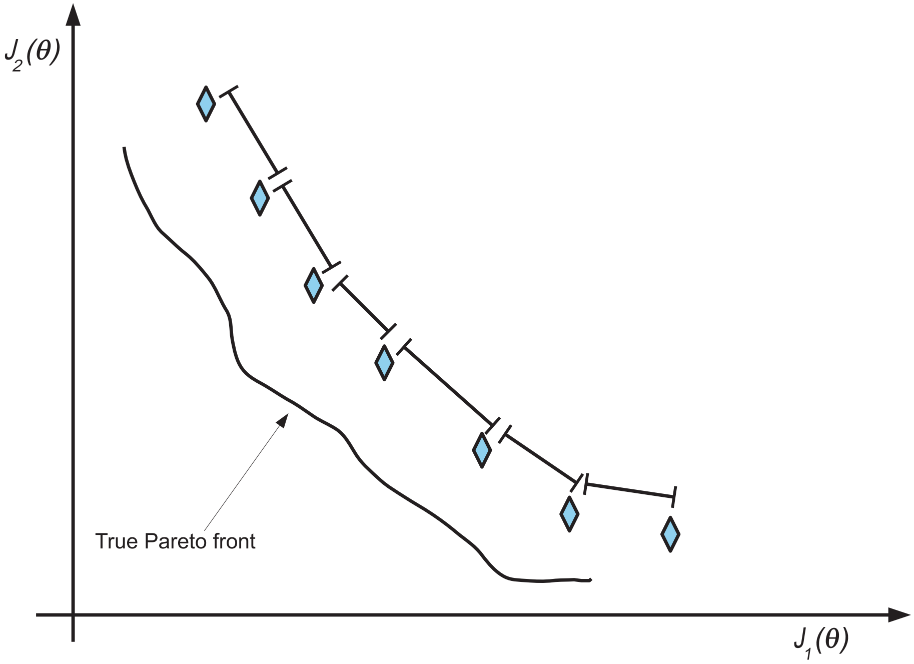
(a)

Figure 1.8: Convergence towards the Pareto front.

Feature 2. *Diversity Mechanism*

Diversity refers to the algorithm's capacity to obtain a set of distributed solutions that provide a useful description of objective tradeoff and decision variables (Figure 1.9). Popular ideas include pruning mechanisms, spreading measures or performance indicators of the approximated front.

Regarding pruning mechanisms, probably the first technique was the ϵ-dominance method [104] which defines a threshold where a solution dominates other solutions in their surroundings. That is, a solution dominates the solutions that are less fit for all the objectives, as well as the solutions inside a distance that is less than a given parameter ϵ. Such dominance relaxation has been shown to generate Pareto fronts with some desirable pertinency characteristics [117]. Algorithms based on this concept include ev-MOGA[6] [78], paϵ-MyDE [77], and paϵ-ODEMO [73]. Similar ideas have been developed using spherical coordinates (or similar statements) [22, 159, 12] in the objective or decision space.



(a)

Figure 1.9: Diversity notion in the Pareto front.

With regard to spreading measures, the crowding distance [43] is used to instigate an algorithm to migrate its population to less crowded areas. This approach is used in algorithms such as NSGA-II[7] [43], which is a very popular MOEA. Other algorithms such as MOEA/D [8] [199] decompose the problem in several scalar optimization subproblems, which are solved simultaneously

[6] Available for Matlab© at: `http://www.mathworks.com/matlabcentral/fileexchange/31080`

[7] Source code available at: `http://www.iitk.ac.in/kangal/codes.shtml`; also, a variant of this algorithm is available in the global optimization toolbox of Matlab©.

[8] Matlab© code available at `http://cswww.essex.ac.uk/staff/zhang/`

(as NBI algorithm) and thereby assure diversity as a consequence of space segmentation when defining the scalar subproblems.

In the case of indicator based MOEAs, instead of comparing members of the population, in each generation the members who best build a Pareto front are selected based on some performance indicator. An example is the IBEA algorithm [205] which is an indicator-based evolutionary algorithm. Most used performance indicators seem to be the hypervolume and the epsilon-indicator [206].

However a good diversity across the Pareto front must not be confused with solution pertinency (meaning interesting and valuable solutions from the DM point of view). Several techniques to accomplish a good diversity on the Pareto front seem to be based on (or compared with) uniform distributions. Nevertheless, a large set of solutions may not be of interest to the DM, owing to a strong degradation in one (or several) objectives [35]. Therefore, some mechanisms to incorporate design preferences could be desirable for the DM to improve solution pertinency.

Feature 3. *Pertinency*

Incorporating DM's preferences into the MOEA has been suggested to improve the pertinency of solutions (see for example [27, 38]). That is, the capacity to obtain a set of interesting solutions from the DM's point of view (Figure 1.10). The designer's preferences could be defined in the MOOD procedure in an *a priori*, *progressive*, or *a posteriori* fashion [135].

- A priori: the designer has some knowledge about his/her preferences in the design objective space. In such cases, he could be interested in using an algorithm that can incorporate such preferences in the optimization procedure.

- Progressive: the optimization algorithm embeds the designer into the optimization process to adjust or change his/her preferences *on the fly*. This could be a desirable characteristic for an algorithm when the designer has some knowledge of the objectives tradeoff in complex problems.

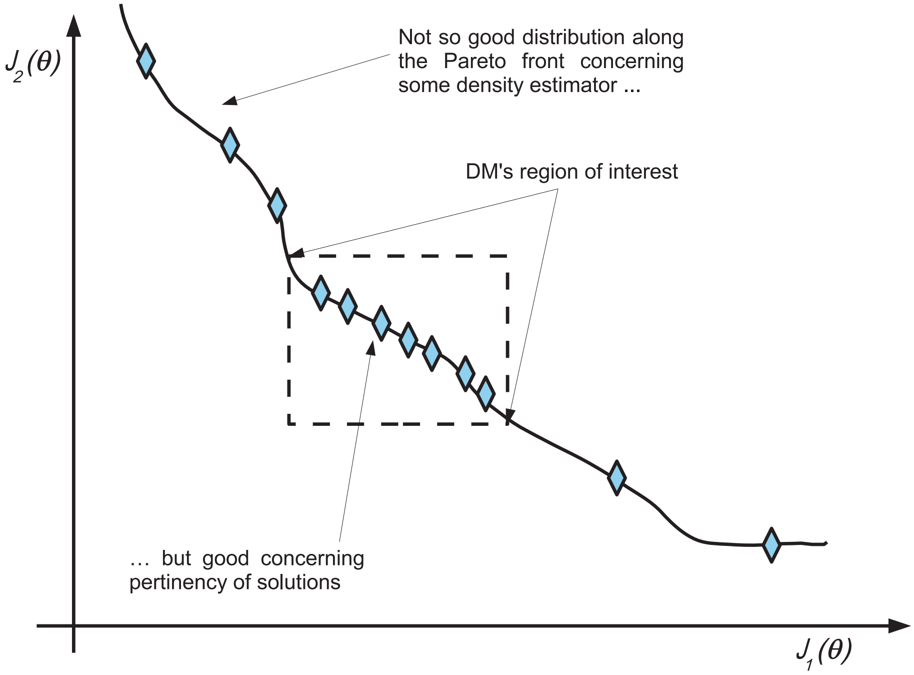
- A posteriori: the designer analyzes the Pareto front calculated by the algorithm, and according to the set of solutions, he/she defines his/her preferences in order to select the preferable solution.

Some popular techniques include ranking procedures, goal attainment, fuzzy relations, among others [27]. Possibly one of the first algorithms to include preference information is the MOGA [9] [62, 143] algorithm which uses a

IntrotoResearch/MOEAd.htm

[9] Toolbox for Matlab© is available at: http://www.sheffield.ac.uk/acse/research/ecrg/gat

goal vector scheme. In [58], the NSGA-II algorithm is improved using preferences in a fuzzy scheme. Other examples are presented in [189] where a preference information approach was merged with the IBEA proposal; or in [143] where a preference articulation technique is used in the MOGA framework. Examples where the ranking scheme has been used include [35, 192, 144, 11].


$J_2(\theta)$
Not so good distribution along the Pareto front concerning some density estimator ...
DM's region of interest
... but good concerning pertinency of solutions
$J_1(\theta)$

(a)

Figure 1.10: Pertinency notion.

Improving pertinency in multiobjective algorithms could have a direct and positive impact in the MCDM stage, since the DM could be provided with a more compact set of potential and interesting solutions. It has been suggested that the size of the Pareto front approximation must be kept to a manageable size for the DM. According to [122] it is usually impossible to retain information from more than 10 or 20 design alternatives.

A straightforward alternative to improve pertinency of solutions could be developed by means of optimization constraints (in addition to bound constraints on decision variables). This topic is exposed below.

Feature 4. *Constrained optimization*

Another desirable characteristic could be constraint handling because such mechanisms are always an interesting topic of research since most of the design optimization problems need to consider constraints. Various techniques have been developed over the years [62, 29, 124] for evolutionary optimization. The latter classifies current approaches as:

- Feasibility rules. An easy and basic manner to implement the approach is discussed in [42]. It consists in:
  - When comparing two feasible solutions, the one with the best objective function is selected.
  - When comparing a feasible and an infeasible solution, the feasible one is selected.
  - When comparing two infeasible solutions, the one with the lowest sum of constraint violation is selected.
- Stochastic ranking. This approach briefly consists in comparing two infeasible solutions by their fitness or by their constraint violations.
- ϵ-constrained method. This method uses a lexicographic ordering mechanism where the minimization of the constraint violation precedes the minimization of the objective function. This mechanism with an adaptive parameter scheme [10] won the CEC-2010 competition in a special session on constrained real-parameter optimization [111].
- Novel penalty functions and novel special operators.
- Multiobjective concepts. In the case of multiobjective optimization, it can be a straightforward approach where the constraint is treated as an additional objective to optimize to a desired value (goal vector).
- Ensemble of constraint-handling techniques. This approach involves taking advantage of all the mechanisms for constraint handling and using them on a single optimization run (for example [112]).

Regarding controller tuning, constrained optimization instances may appear in complex processes, where several constraints on settling time, overshoot and robustness must be fulfilled.

Feature 5. *Many-Objectives Optimization*

Algorithms with good diversity preservation mechanisms could face problems if solutions are dominance resistant in an m-dimensional objective space. This could lead to wasting time and CPU resources in non-optimal areas [142]. This is because of the self diverse nature and the large number of objectives (usually, $m \geq 5$). Furthermore, recent research has indicated that a random search approach can be competitive for generating a Pareto front approximation for a many-objective optimization [35]. In [90], a review of many-objective

[10] Code available at `http://www.ints.info.hiroshima-cu.ac.jp/~takahama/eng/index.html` for single objective optimization.

optimization techniques is given. According to this review, approaches to deal with many-objective optimization include:

- Modification of Pareto dominance to improve the selection pressure towards the Pareto front.
- Introduction of different ranks to define a metric based on the number of objectives for which a solution is better than the other.
- Use of indicator functions as performance indicators of the quality of the Pareto front approximation.
- Use of scalarizing functions (weighting vectors, for example).
- Use of preference information (see above), that is, information on the region of interest for the DM.
- Reduction in the number of objectives.

Some examples to deal with this instance include [110] where an objective reduction is used with principal component analysis, whereas in [171] a heuristic approach is used for dimensionality reduction. Besides, algorithms which incorporate preference information in the optimization approach could be used in many-objective instances [90].

In the specific case of controller tuning, a many-objective optimization instance would appear according with the complexity of a given control loop or process, and the number of requirements to fulfill.

Feature 6. *Dynamic Optimization*

Sometimes the static approach is not enough to find a preferable solution and therefore, a dynamic optimization statement needs to be solved where the cost function varies with time. The problem, besides tracking the optimal solution, is to select the desired solution at each sampling time. In [53, 37] there are extensive reviews on the topic.

As it can be noticed, this kind of capabilities would be useful for problems related to Predictive Control.

Feature 7. *Multi-modal optimization*

In multi-modal optimization, different decision variable vectors could have the same objective vector. In some instances, it could be desirable to ponder different solutions even if they have the same objective vector in the MCDM step. This could be important in instances where, for example, the decision variables have a physical meaning and it is convenient to analyze the impact of using one over another. In an EMO framework, this information could be added as additional objectives as noticed by [44]. For more details on multimodal optimization, the interested reader could refer to [40].

Multi-modal instances for controller tuning *per se* seem to be not usual; nevertheless they may appear in multi-disciplinary optimization [116] statements, where besides tuning parameters, other design variables (as mechanical or geometrical) are involved.

Feature 8. *Robust optimization*

In a general frame and according to [17], robust optimization could refer to the models used to measure the performance, or the sensitivity analysis of the calculated solutions. That is, how much the objective vector could be degraded under the presence of uncertainties. This sensibility analysis could be done by means of deterministic measures and/or with direct search (as Montecarlo methods). This kind of analysis could result in a different level of interpretability of the performance due to uncertainties in the model used in the optimization. This problem statement is related to reliability optimization, where a given performance must be assured for a certain process along different scenarios.

A practical application of this optimization statement is provided, in [176] an evaluation of the American Control Conference benchmark [194] based on Montecarlo methods is performed.

Feature 9. *Computationally Expensive optimization*

Computationally expensive optimization is related to line 8 in Algorithm I. Sometimes the cost function to evaluate requires a huge amount of computational resources. Therefore stochastic approaches could face a problem, given the complexity to evaluate the fitness (performance) of an individual (design alternative). Some tendencies to deal with this are mainly oriented to generate a surface *on-the-fly* of the objective space, with lower computational effort. One popular technique is to use Neural Networks, trained through the evolutionary process, but any kind of model or surface approximation could be used. A review on the topic can be consulted in [166]. In the field of control systems engineering, such type of instances would appear when expensive calculations in complex simulations are needed to compute the objective vector.

In other instances, such computational effort could be relative; that is, there are limited computational resources to evaluate a cost function. With this idea in the field of EA's compact evolutionary algorithms have been proposed, but such instance has not reached yet the EMO approach. Some examples are exposed in [76] and [128]. Instances where these capabilities could be desirable include embedded solvers for optimization.

Feature 10. *Large scale optimization*

This refers to the capabilities of a given MOEA to deal with MOP with any number of decision variables, with reasonable computational resources.

Sometimes a MOEA could have remarkable convergence properties for a relatively small number of decision variables, but it could be intractable (according to the computational resources available) to solve a problem with a larger number of decision variables. While in expensive optimization instances (Section 9) the complexity is due to the performance measurement (line 8 in Algorithm I), in a large scale instance it may be related to the algorithm mechanism used to approximate a new set of design alternatives (lines 7 and 9 in Algorithm I). In the former the complexity is added by the problem, in the latter complexity is added by the algorithm. A review on this topic can be consulted in [108].

The aforementioned features could be desirable characteristics for a given MOEA. It would depend on the designer's preferences and the MOP statement at hand. Afterwards, a MCDM step must be carried, in order to select the most preferable solution. This step is commented below.

1.4.3 Multi-Criteria Decision-Making (MCDM)

Once the DM has been provided with a Pareto front $\boldsymbol{J}_P^*$, he or she will need to analyze the tradeoff between objectives and select the best solution according to his/her preferences. A comprehensive compendium on MCDM techniques (and software) for multi-dimensional data and decision analysis can be consulted in [60]. Assuming that all preferences have been handled as much as possible in the optimization stage, a final selection step could be made with the approximated Pareto front. Here it will be emphasized the tradeoff visualization.

It is widely accepted that visualization tools are valuable and provide the DM with a meaningful method to analyze the Pareto front and take decisions [106]. Tools and/or methodologies are required for this final step to successfully embed the DM into the solution refinement and selection process. It is useful to the DM to understand and appreciate the impact that a given tradeoff in one sub-space could have on others [21]. Even if an EMO process has been applied to a reduced objective space, the DM sometimes needs to increase the space with additional metrics or measurements in order to have confidence in her/his own decision [21]. Usually, analysis on the Pareto front may be related to design alternatives comparison and design concepts comparison.

For two-dimensional problems (and sometimes for three-dimensional problems) it is usually straightforward to make an accurate graphical analysis of the Pareto front (see for example Figure 1.11), but the difficulty increases with the dimension of the problem. Tools such as VI$\underline{\text{D}}$EO [100] incorporate a color coding in three-dimensional graphs to analyze tradeoffs for 4-dimensional Pareto fronts. In [106], a review on visualization techniques includes techniques such as decision maps, star diagrams, value paths, GAIA, and heatmap graphs. Possibly the most common choices for Pareto front visualization and analysis in control systems applications are: scatter diagrams, parallel coordinates [86], and level diagrams [19, 151].

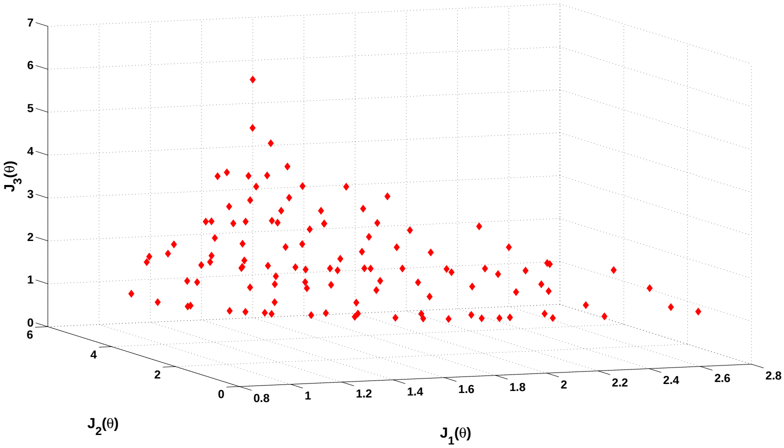


Figure 1.11: 3D Visualization of a 3-dimensional Pareto front

Scatter diagram plots (SCp) [11] are a straightforward visualization approach. They generate an array of 2-D graphs to visualize each combination of a pair of objectives (see Figure 1.12). This type of visualization is enough for two dimensional problems. To appreciate all the tradeoffs of an m-dimensional Pareto front, at least $\frac{m(m-1)}{2}$ combination plots are required. For example, the Pareto front of Figure 1.11 is visualized using SCp in Figure 1.12. If the DM would like to see the tradeoff for an objective and a decision variable from the n-dimensional decision space, she/he will need n times m additional plots.

The Parallel coordinate (PAc) visualization strategy [86] plots an m-dimensional objective vector in a two-dimensional graph [12]. For each objective vector $\boldsymbol{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), \dots, J_m(\boldsymbol{\theta})]$ the ordered pairs $(i, J_i(\boldsymbol{\theta}))$, $i \in [1, \dots, m]$ are plotted and linked with a line. This is a very compact way of depicting multidimensional information: just one 2-D plot is required. Nevertheless, to entirely represent the tradeoff surface some axis relocations may be necessary. For example, in Figure 1.13, it is possible to appreciate the PAc visualization of the Pareto front for Figure 1.11. To appreciate tendencies with the decision space variable, an extended plot with $n+m$ vertical axes is required. An independent graph could be plotted, but some strategy (such as color coding) will be needed to link an objective vector with its corresponding decision vector in order to appreciate tradeoff information from the objective space. This kind of feature is incorporated in visualization tools such as TULIP from INRIA [13],

[11] Tool available in Matlab©.

[12] Tool available in the statistics toolbox of Matlab©.

[13] Available at `http://tulip.labri.fr/TulipDrupal/`. Includes applications for multi-

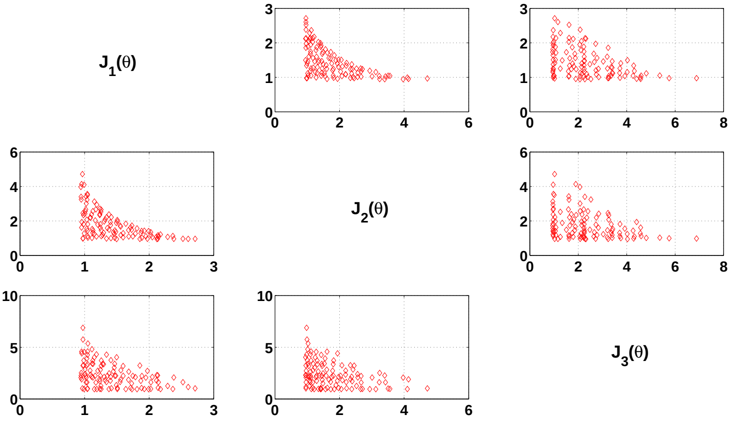


Figure 1.12: Scatter plot (SCp) visualization for Pareto front of Figure 1.11

which is also helpful for analyzing multidimensional data. Finally, a normalization or y-axis re-scaling can be easily incorporated, if required, to facilitate the analysis.

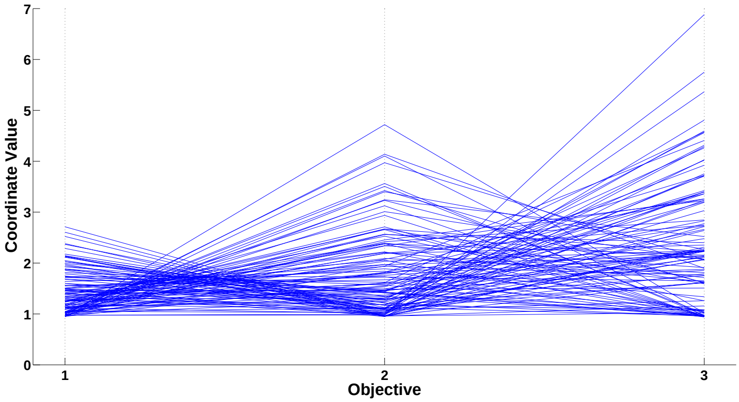


Figure 1.13: Parallel coordinates plot (PAc) visualization for Pareto front of Figure 1.11

The level diagram (LD) visualization [19] [14] is useful for analyzing m-objective Pareto fronts [204, 203], as it is based on the classification of the approximation $\boldsymbol{J}_P^*$ obtained. Each objective $J_q(\boldsymbol{\theta})$ is normalized $\hat{J}_q(\boldsymbol{\theta})$ with respect

dimensional analysis.

[14] GUI for Matlab© is available at: `http://www.mathworks.com/matlabcentral/fileexchange/24042`

to its minimum and maximum values. To each normalized objective vector $\hat{\boldsymbol{J}}(\boldsymbol{\theta})$ a norm is applied to evaluate the distance to an ideal solution $\boldsymbol{J}^{ideal}$[15]. The LD tool displays a two dimensional graph for each objective and decision variable. The ordered pairs $\left(J_q(\boldsymbol{\theta}), \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_p\right)$ in each objective sub-graph and $\left(\theta_l, \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_p\right)$ in each decision variable sub-graph are plotted (a total of $n+m$ plots). Therefore, a given solution will have the same y-value in all graphs (see Figure 1.14). This correspondence will help to evaluate general tendencies along the Pareto front and compare solutions according to the selected norm. Also, with this correspondence, information from the objective space is directly embedded in the decision space, since a decision vector inherits its $y - value$ from its corresponding objective vector.

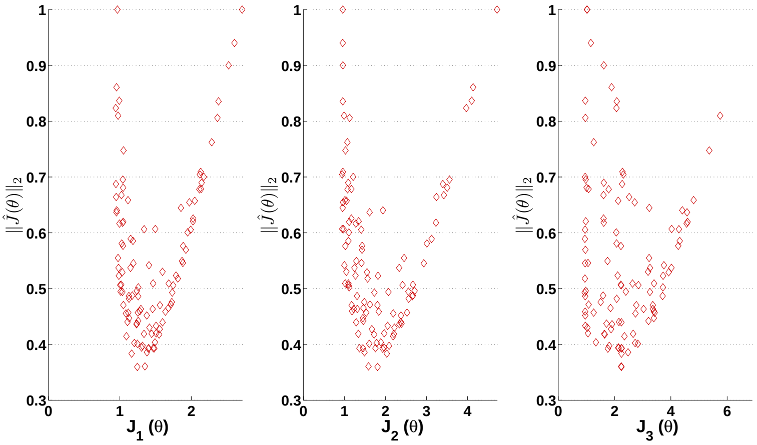


Figure 1.14: Level diagram (LD) visualization for Pareto front of Figure 1.11

In any case, the characteristics required for such a visualization were described in [106]: simplicity (must be understandable); persistence (information must be rememberable by the DM); and completeness (all relevant information must be depicted). Some degree of interactivity with the visualization tool is also desirable (during and/or before the optimization process) to successfully embed the DM into the selection process.

1.5 Background on MOOD Tools

In this section, some MOOD tools are explained. They are used as basis for the Preliminary Contributions on Controller Tuning (Part II) and they are basis of

[15] By default, the minimal values for each objective in the calculated Pareto front approximation could be used to build an ideal solution.

the further improvements presented on Contributions on MOOD Tools (Part III). They follow the initial research contained in [148].

Search process: the spherical pruning multiobjective differential evolution (sp-MODE) algorithm[16]. It is a MOEA resulting from a previous work [148] and validated as a suitable tool for the MOOD procedure in controller tuning.

Selection process: the level diagrams visualization toolbox (LD-Tool) for Matlab©[17]. This visualization tool has been also validated for controller tuning in a previous work [149].

While a detailed description of both tools for controller tuning is included in [148] (and the preliminary results which eventually led to this thesis), here just a brief description will be depicted for the sake of completeness.

1.5.1 EMO process: Multiobjective Differential Evolution with Spherical pruning (sp-MODE) algorithm.

A general pseudocode for MOEAs with pruning mechanism and external archive A is shown in Algorithm 1.2. The usage of an external archive A to store the best set of quality solutions found so far in an evolutionary process is common in MOEAs.

The basic idea of the spherical pruning is to analyze the proposed solutions in the current Pareto front approximation $\boldsymbol{J}_P^*$ by using normalized spherical coordinates from a reference solution (see Figure 1.15). With such approach, it is possible to attain a good distribution along the Pareto front [158, 148]. The algorithm selects one solution for each spherical sector, according to a given norm or measure. This process is explained in Algorithm 1.3, where the following definitions are required:

Definition 6. *(normalized spherical coordinates) given a solution $\boldsymbol{\theta^1}$ and $\boldsymbol{J}(\boldsymbol{\theta^1})$, let*

$$\boldsymbol{S}(\boldsymbol{J}(\boldsymbol{\theta^1})) = [\|\boldsymbol{J}(\boldsymbol{\theta^1})\|_2, \boldsymbol{\beta}(\boldsymbol{J}(\boldsymbol{\theta^1}))] \tag{1.5}$$

be the normalized spherical coordinates from a reference solution $\boldsymbol{J}^{ref}$ where $\boldsymbol{\beta}(\boldsymbol{J}(\boldsymbol{\theta^1}))$ $= [\beta_1(\boldsymbol{J}(\boldsymbol{\theta^1})), \ldots, \beta_{m-1}(\boldsymbol{J}(\boldsymbol{\theta^1}))]$ is the arc vector and $\|\boldsymbol{J}(\boldsymbol{\theta^1})\|_2$ the Euclidean distance to the reference solution.

It is important to guarantee that $\boldsymbol{J}^{ref}$ dominates all the solutions. An intuitive approach is to select:

$$\boldsymbol{J}^{ref} = \boldsymbol{J}^{ideal} = \left[\min J_1(\boldsymbol{\theta^i}), \ldots, \min J_m(\boldsymbol{\theta^i})\right] \forall \boldsymbol{J}(\boldsymbol{\theta^i}) \in \hat{A}|_G \tag{1.6}$$

[16] Available at: `www.mathworks.com/matlabcentral/fileexchange/38962`

[17] Available at: `www.mathworks.com/matlabcentral/fileexchange/24042`

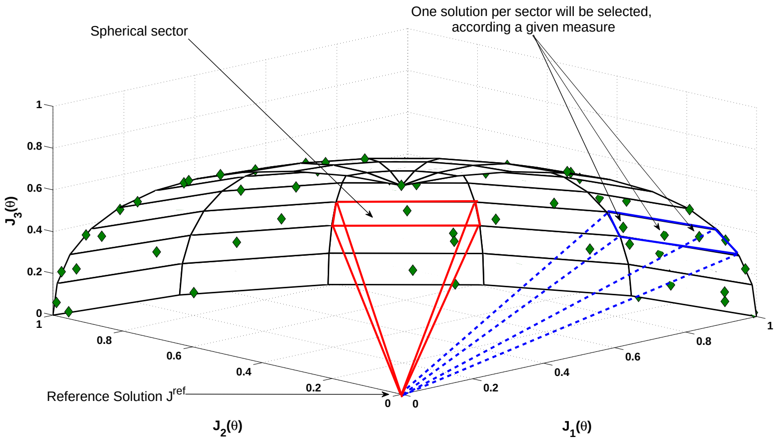


Figure 1.15: Spherical relations on $\boldsymbol{J}_P^* \subset \mathbb{R}^3$. For each spherical sector, just one solution, the solution with the lowest norm will be selected.

```
Generate initial population $P|_0$ with $N_p$ individuals;
Evaluate $P|_0$;
Apply dominance criterion on $P|_0$ to get archive $A|_0$;
while stopping criterion unsatisfied do
    Update generation counter G=G+1;
    Get subpopulation $S|_G$ with solutions in $P|_G$ and $A|_G$;
    Generate offspring $O|_G$ with $S|_G$;
    Evaluate offspring $O|_G$;
    Update population $P|_G$ with offspring $O|_G$;
    Apply dominance criterion on $O|_G \bigcup A|_G$ to get $\hat{A}|_G$;
    Apply pruning mechanism to prune $\hat{A}|_G$ to get $A|_{G+1}$;
    Update environment variables (if using a self-adaptive mechanism);
end
Algorithm terminates. $\boldsymbol{J_P}$ is approximated by $\boldsymbol{J}_P^* = A|_G$;
```

Algorithm 1.2: MOEA with pruning mechanism

Definition 7. *(sight range) The sight range from the reference solution $\boldsymbol{J}^{ref}$ to the Pareto front approximation $\boldsymbol{J_P^*}$ is bounded by $\boldsymbol{\beta^U}$ and $\boldsymbol{\beta^L}$:*

$$\boldsymbol{\beta^U} = \left[\max \beta_1(\boldsymbol{J}(\boldsymbol{\theta^i})), \ldots, \min \beta_{m-1}(\boldsymbol{J}(\boldsymbol{\theta^i}))\right] \quad \forall \quad \boldsymbol{J}(\boldsymbol{\theta^i}) \in \hat{A}|_G \qquad (1.7)$$

$$\boldsymbol{\beta^L} = \left[\min \beta_1(\boldsymbol{J}(\boldsymbol{\theta^i})), \ldots, \min \beta_{m-1}(\boldsymbol{J}(\boldsymbol{\theta^i}))\right] \quad \forall \quad \boldsymbol{J}(\boldsymbol{\theta^i}) \in \hat{A}|_G \qquad (1.8)$$

If $\boldsymbol{J}^{ref} = \boldsymbol{J}^{ideal}$, it is straightforward to prove that $\boldsymbol{\beta^U} = \left[\frac{\pi}{2}, \ldots, \frac{\pi}{2}\right]$ and $\boldsymbol{\beta^L} = [0, \ldots, 0]$.

Definition 8. *(spherical grid) Given a set of solutions in the objective space, the spherical grid on the m-dimensional space in arc increments $\boldsymbol{\beta_\epsilon} = [\beta_1^\epsilon, \dots, \beta_{m-1}^\epsilon]$ is defined as:*

$$\boldsymbol{\Lambda}^{\boldsymbol{J_P^*}} = \left[\frac{\beta_1^U - \beta_1^L}{\beta_1^\epsilon}, \dots, \frac{\beta_{m-1}^U - \beta_{m-1}^L}{\beta_{m-1}^\epsilon}\right] \tag{1.9}$$

Definition 9. *(spherical sector) The normalized spherical sector of a solution $\boldsymbol{\theta^1}$ is defined as:*

$$\boldsymbol{\Lambda_\epsilon}(\boldsymbol{\theta^1}) = \left[\left\lceil \frac{\beta_1(\boldsymbol{J}(\boldsymbol{\theta^1}))}{\Lambda_1^{J_P^*}} \right\rceil, \dots, \left\lceil \frac{\beta_{m-1}(\boldsymbol{J}(\boldsymbol{\theta^1}))}{\Lambda_{m-1}^{J_P^*}} \right\rceil\right] \tag{1.10}$$

Definition 10. *(spherical pruning) given two solutions $\boldsymbol{\theta^1}$ and $\boldsymbol{\theta^2}$ from a set, $\boldsymbol{\theta^1}$ has preference in the spherical sector over $\boldsymbol{\theta^2}$ iff:*

$$\left[\boldsymbol{\Lambda_\epsilon}(\boldsymbol{\theta^1}) = \boldsymbol{\Lambda_\epsilon}(\boldsymbol{\theta^2})\right] \wedge \left[\|\boldsymbol{J}(\boldsymbol{\theta^1})\|_p < \|\boldsymbol{J}(\boldsymbol{\theta^2})\|_p\right] \tag{1.11}$$

where $\|\boldsymbol{J}(\boldsymbol{\theta})\|_p = \left(\sum_{q=1}^{m} |J_q(\boldsymbol{\theta})|^p\right)^{1/p}$ *is a suitable p-norm.*

```
Generate initial population $P|_0$ with $N_p$ individuals;
Read archive $\hat{A}|_G$;
Read and update extreme values for $J_{ref}|_G$;
for each member in $\hat{A}|_G$ do
    calculate its normalized spherical coordinates (Definition 3);
end
Build the spherical grid (Definition 4 and 5);
for each member in $\hat{A}|_G$ do
    calculate its spherical sector (Definition 6);
end
for i=1:SolutionsInArchive do
    Compare with the remainder solutions in $\hat{A}|_G$;
    if no other solution has the same spherical sector then
        it goes to archive $A|_{G+1}$;
    end
    if other solutions are in the same spherical sector then
        it goes to archive $A|_{G+1}$ if it has the lowest norm (Definition 7);
    end
end
Return Archive $A|_{G+1}$;
```

Algorithm 1.3: Spherical pruning mechanism

In this implementation, the spherical pruning is merged with the DE algorithm [183, 41, 182]. Although any other evolutionary or nature inspired mechanism may be used, DE algorithm is selected because of its simplicity. The DE algorithm uses two operators: mutation and crossover (Equations (1.12) and (1.13) respectively) to generate its offspring (Algorithm 1.4).

Mutation: For each target (parent) vector $\boldsymbol{\theta}^i|_G$, a mutant vector $\boldsymbol{v}^i|_G$ is generated at generation G according to Equation (1.12):

$$\boldsymbol{v}^i|_G = \boldsymbol{\theta}^{r_1}|_G + F(\boldsymbol{\theta}^{r_2}|_G - \boldsymbol{\theta}^{r_3}|_G) \tag{1.12}$$

Where $r_1 \neq r_2 \neq r_3 \neq i$ and F is usually known as the scaling factor.

Crossover: For each target vector $\boldsymbol{\theta}^i|_G$ and its mutant vector $\boldsymbol{v}^i|_G$, a trial (child) vector $\boldsymbol{u}^i|_G = [u_1^i|_G, u_2^i|_G, \ldots, u_n^i|_G]$ is created as follows:

$$u_j^i|_G = \begin{cases} v_j^i|_G & if \quad rand(0,1) \leq Cr \\ \theta_j^i|_G & \quad\;\; otherwise \end{cases} \tag{1.13}$$

where $j \in 1, 2, 3 \ldots n$ and Cr is named the crossover probability rate.

```
for i=1:SolutionsInParentPopulation do
    Generate a Mutant Vector $\boldsymbol{v}^i$ (Equation (1.12)) ;
    Generate a Child Vector $\boldsymbol{u}^i$ (Equation (1.13)) ;
end
Offspring $O = U$;
```

Algorithm 1.4: DE offspring generation mechanism

The standard selection mechanisms are as follows:

- For single objective optimization, a child is selected over its parent (for the next generation) if it has a better cost value.
- In EMO, a simple selection based on dominance is used; a child is selected over his parent if the child strictly dominates his parent (Definition 2).

This solution, merging the MODE algorithm (Algorithm 1.5) and the spherical pruning mechanism (Algorithm 1.3) was named as sp-MODE algorithm (see Algorithm 1.6) and it is freely available at Matlab© Central[18]. Default parameters and guidelines for parameter tuning are given in table 1.1.

Other approaches have used spherical coordinates or similar ideas (as cone separation) in multiobjective optimization. For example, in [22] these ideas

[18] `http://www.mathworks.com/matlabcentral/fileexchange/39215`

```
Build initial population $P|_0$ with $N_p$ individuals;
Evaluate $P|_0$;
Build initial Pareto set approximation $\boldsymbol{X}_P^*|_0$;
Set generation counter $G = 0$;
while stopping criterion unsatisfied do
    $G = G + 1$;
    Build offspring $P^*|_G$ using $P|_G$ using DE algorithm operators (Algorithm 1.4).;
    Evaluate offspring $P^*|_G$;
    Update population $P|_{G+1}$ with $P^*|_G \bigcup P|_G$ using greedy selection mechanism of DE algorithm. ;
end
RETURN Pareto set approximation $\boldsymbol{X}_P^*|_G = P|_{G+1}$;
```

Algorithm 1.5: MODE

```
Generate initial population $P|_0$ with $N_p$ individuals;
Evaluate $P|_0$;
Apply dominance criterion on $P|_0$ to get $A|_0$;
while stopping criterion unsatisfied do
    Read generation count $G$;
    Get subpopulation $S|_G$ with solutions in $P|_G$ and $A|_G$;
    Generate offspring $O|_G$ with $S|_G$ using DE operators (Algorithm 1.4).;
    Evaluate offspring $O|_G$;
    Update population $P|_G$ with offspring $O|_G$ according to greedy selection mechanism.;
    Apply dominance criterion on $O|_G \bigcup A|_G$ to get $\hat{A}|_G$;
    Apply pruning mechanism (Algorithm 1.3) to prune $\hat{A}|_G$ to get $A|_{G+1}$;
    $G = G + 1$;
end
Algorithm terminates. $\boldsymbol{J_P}$ is approximated by $\boldsymbol{J}_P^* = A|_G$;
```

Algorithm 1.6: sp-MODE.

have been used for parallel computation, where cone separation is used in the decision space to split the search into several regions. In [12], a cone ϵ-dominance is defined merging cones with the ϵ-dominance mechanism. Nevertheless, with such approach a new parameter κ is introduced to adjust the capacity of the grid. If this parameter is not adequately adjusted, drawbacks of the ϵ-dominance mechanism could be inherited. Potential drawbacks are related with confusing regions with practically uninteresting tradeoff with those of interest for the designer (pertinent regions). As it will be shown in Chapter 6, such characteristic will be exploited to generate pertinent Pareto front approximations.

1.5.2 MCDM stage: Level Diagrams visualization

Level diagrams (LD) enables the DM to perform an analysis on the calculated Pareto front $\boldsymbol{J}_P^*$. LD visualization is based on the classification of the Pareto front $\boldsymbol{J}_P^*$. Each objective $J_q(\boldsymbol{\theta})$ is normalized with respect to its minimum and maximum values. That is:

$$\hat{J}_q(\boldsymbol{\theta}) = \frac{J_q(\boldsymbol{\theta}) - J_q^{min}}{J_q^{max} - J_q^{min}}, q \in [1, \ldots, m]. \tag{1.14}$$

where (with a little abuse of notation):

$$\boldsymbol{J}^{min} = \left[\min_{J_1(\boldsymbol{\theta}) \in J_P^*} J_1(\boldsymbol{\theta}), \ldots, \min_{J_m(\boldsymbol{\theta}) \in J_P^*} J_m(\boldsymbol{\theta}) \right] \tag{1.15}$$

$$\boldsymbol{J}^{max} = \left[\max_{J_1(\boldsymbol{\theta}) \in J_P^*} J_1(\boldsymbol{\theta}), \ldots, \max_{J_m(\boldsymbol{\theta}) \in J_P^*} J_m(\boldsymbol{\theta}) \right] \tag{1.16}$$

For each normalized objective vector $\hat{\boldsymbol{J}}(\boldsymbol{\theta}) = [\hat{J}_1(\boldsymbol{\theta}), \ldots, \hat{J}_m(\boldsymbol{\theta})]$ a p-norm $\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_p$ is applied to evaluate the distance to an ideal solution $\boldsymbol{J}^{ideal} = \boldsymbol{J}^{min}$; common norms are:

$$\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_1 = \sum_{q=1}^{m} \hat{J}_q(\boldsymbol{\theta}) \tag{1.17}$$

$$\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2 = \sum_{q=1}^{m} \hat{J}_q(\boldsymbol{\theta})^2 \tag{1.18}$$

$$\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_\infty = \max \hat{J}_q(\boldsymbol{\theta}) \tag{1.19}$$

The LD visualization displays a two-dimensional graph for every objective and every decision variable. The ordered pairs $\left(J_q(\boldsymbol{\theta}), \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_p\right)$ are plotted in each objective sub-graph and $\left(\theta_l, \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_p\right)$ in each decision variable sub-graph. Therefore, a given solution will have the same y-value in all graphs (see

Table 1.1: Guidelines for sp-MODE's parameters tuning.

Parameter	Value	Comments
	DE algorithm	
F (Scaling factor)	0.5	Recognized as good initial choice according to [183].
Cr (Crossover rate)	$[0.8, 1.0]$	Values recognized for non-separable problems according to [155, 41].
	$[0.1, 0.2]$	Values recognized for separable problems according to [155, 41].
	0.5	Trade-off value for separable and non-separable problems. Default value used (for example) by MOEA/D algorithm [199].
N_p (Population)	50	While a *five to ten times the number of decision variables* rule has been recognized as a thumb rule [183] for single objective optimization, here it is proposed a default size of 50 individuals.
	Pruning mechanism	
$\boldsymbol{\beta_\epsilon}$ (Arcs)	100	It is proposed for bi-objective problems, to bound the approximated Pareto front to 100 design alternatives.
	$[10, 10]$	It is proposed for 3-objective problems, to bound the approximated Pareto front to $10^2 = 100$ design alternatives.
	$\overbrace{[m, \ldots, m]}^{m-1}$	It is proposed for m-objective problems, to bound the approximated Pareto front to m^{m-1} design alternatives.
p (p-norm)	1	It is proposed as default value.

Figure 1.16). This correspondence helps to evaluate general tendencies along the Pareto front and compare solutions according to the selected norm. In all cases, the lower the norm, the closer the ideal solution[19]. For the remainder of this work and for the sake of simplicity, $\|\cdot\|_2$ norm will be used.

For example, an euclidian norm is helpful to evaluate the distance of a

[19] In this thesis, the minimal values for each objective in the calculated Pareto front approximation are used to build an ideal solution.

given solution with respect to the ideal solution, meanwhile a maximum norm will give information about the tradeoff achieved by this solution. Such norm, used to visualize tendencies in the Pareto front, does not deform the MOP essence, since this visualization process takes place after the optimization stage.

In all cases, the lower the norm, the closer to the ideal solution $\boldsymbol{J}^{min}$. For example, in Figure 1.16, point A is the closest solution to $\boldsymbol{J}^{min}$ with the $\|\cdot\|_1$ norm. This does not mean that point A must be selected by the DM. Selection will be performed according with the visual information from the LD visualization and the DM's preferences. In the same Figure, it is possible to visualize how the tradeoff rate changes in solution A. That is, it is possible to appreciate two different tendencies around solution A: on the one hand, the better $J_2(\boldsymbol{\theta})$ value, the worse $J_1(\boldsymbol{\theta})$ value (circles). On the other hand, the worse $J_2(\boldsymbol{\theta})$ value, the better $J_1(\boldsymbol{\theta})$ value (diamonds). It is difficult to appreciate such tendencies with classical visualizations with more than three objectives. For the remainder of this thesis, the $\|\cdot\|_2$ norm will be used.

Differences in LD from other visualization techniques rely mainly on the number of plots required to depict all the information (completeness). For example, scatter plots (SCp) require at least $\frac{m(m-1)}{2}$ graphs for tradeoff comparison of each pair of objectives; if correlations with the decision space are needed, $n \cdot m$ additional plots will be required. Another popular visualization strategy is Parallel Coordinates ($\|_p$), which is a very compact way to display multidimensional data. A single 2-dimensional plot is required, with as many vertical axes as objectives (plus decision variables, if required). Nevertheless, to fully appreciate all the tradeoff among objectives, some rearrangements of the vertical axes could be required. In the case of LD, a total of $n+m$ graphs are needed and no additional rearrangements for axes are required. Furthermore, the decision space variables inherit the norm value from the objective space, giving a straightforward correlation between objective and decision space. The LD visualization offers a tradeoff between SCp and $\|_p$ to visualize an m-dimensional Pareto front with the following assumption: the fewer the plots required to depict all the information, the greater the persistence.

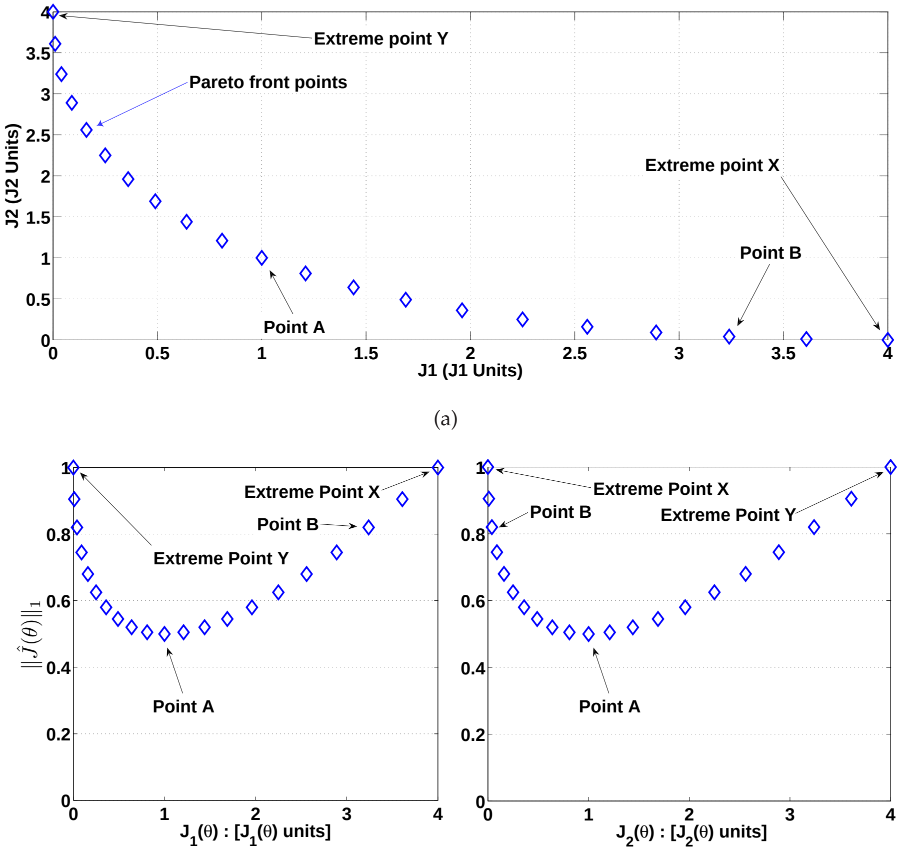
In any case, some degree of interactivity (with the data) would be desirable to facilitate the decision making procedure; this could improve the simplicity of any visualization technique. Moreover, elementary tools to identify preferable zones could be helpful (such as color coding for example); this could improve the persistence of the aforementioned visualization techniques. These types of features have been successfully included in software tools such as VI$\underline{\text{D}}$EO [100] (classical visualization) and TULIP (parallel coordinates) from INRIA [20].

The LD-ToolBox [21], a powerful tool to analyze m-objective Pareto fronts, [149, 156, 204] uses LD visualization and is employed at this point. It is a

[20] Available at `http://tulip.labri.fr/TulipDrupal/`

[21] available at `http://www.mathworks.com/matlabcentral/fileexchange/24042`

Matlab © toolbox that offers to the DM a degree of interactivity with multidimensional data. An in-depth explanation of the LD tool capabilities can be found in [19] [22].



(a)

(b)

Figure 1.16: Typical representation of the Pareto front for bi-objective problems using 2-D graph (a) and LD (b). Points at the same level in LD correspond on each graphic.

1.6 Conclusions on this chapter

In this chapter, some topics on MOP definitions, the EMO process and the MCDM step have been covered. The aforementioned steps are important to guarantee the overall performance of a MOOD procedure. With a poor MOP

[22] There are video tutorials available at `http://cpoh.upv.es/es/investigacion/software/item/52-ld-tool.html`.

definition, the solutions obtained will not fulfil the DM's expectations regardless of the quality of the algorithms and MCDM methodology/tools. If the algorithm is inadequate for the problem at hand (regarding the desirable characteristics from Section 1 to 10), the DM will not obtain a useful Pareto set to analyze and therefore he/she will not be able to select a solution that meets his/her preferences. Finally, the incorrect use of MCDM tools and methodologies could imply a lower degree of embedment of the DM in the tradeoff analysis and solution selection. The last issue could easily discourage the DM from using a MOOD procedure.

Regarding the MOP, some comments have been made regarding the capacity to reach a different level of interpretability on objective functions. In the MOOD approach there is no need to build a complicated aggregated function to merge the design objectives; therefore the objectives may be listed separately and optimized simultaneously. That is, the objective function statement could be made from the needs of the designer instead of the optimizer. This may facilitate the embedment of the designer into the overall procedure. In the case of EMO, it has been shown how MOEAs could be useful to face different optimization instances as well as bring some desirable characteristics to the approximated Pareto front. It is important to remember that the final purpose of any MOEA is to provide the DM with a useful set of solutions (Pareto front approximation) to perform the MCDM procedure [21]. With regard to the MCDM step, it can be seen that visualization of the Pareto front is a desirable tool for the DM to perform his/her selection at the MCDM stage. Finally, tools to be used as basis of this thesis has been presented.

Next, several design concepts and MOP statements used in control system engineering field will be discussed.

CHAPTER 2:

Controller tuning using evolutionary multiobjective optimization: current trends and applications

In this chapter a review on controller tuning applications using a MOOD procedure based on EMO is given. Contents of this chapter appear in the following paper:

- **G. Reynoso-Meza, X. Blasco, J. Sanchis, M. Martínez**. *Controller tuning by means of evolutionary multiobjective optimization: current trends and applications*. Control Engineering Practice. July 2014, Vol. 28, Pp. 58-73.

2

Controller tuning using evolutionary multiobjective optimization: current trends and applications

> *All collected data had come to a final end. Nothing was left to be collected. But all collected data had yet to be completely correlated and put together in all possible relationships. A timeless interval was spent in doing that.*
>
> Isaac Asimov, *The Last Question*.

2.1 Aim of this chapter

In this chapter a review on controller tuning applications using a MOOD procedure is given. The specific aims of this chapter are:

- To present a state-of-the art review of the last ten years (2002-2012) of MOOD procedures using EMO for controller tuning.
- To identify gaps between the EMO process and the MCDM step for those MOP statements.

Since the MOOD procedure provides the opportunity to obtain a set of solutions to describe the objectives tradeoff for a given MOP, it is worthwhile to use it for controller tuning. Due to the fact that several specifications such as time and frequency requirements need to be fulfilled by the control engineer, a procedure to appreciate the tradeoff exchange for complex processes could be useful. Controller design concepts such as PID, fuzzy, state space feedback and predictive controllers are covered, where by means of the MOOD procedure, the designer is seeking to improve their performance. At the end of this chapter, conclusions on potential directions for further research and applications development are commented.

2.2 Applications on controller tuning.

In [61], a review on the early stages of MOEAs in control systems is provided. To be in the scope of this work design applications may include parametric controller tuning for the 10 year period following the aforementioned review (from 2002-2012). Works on parametric model identification are not included since comprehensive reviews on modeling (in the broad sense) using EMO are available for fuzzy systems [56], neural networks, [71, 46], machine learning [93] and support vector machines [186].

Relevant applications are included, where desirable characteristics such as convergence and diversity are considered. This means that works where a proof of concept of an EMO for solving a given problem, or where a simple comparison between MOEAs is provided are intentionally omitted. Work with depth analysis that brings new desirable characteristics of MOEAs for controller tuning is included and discussed. This is because it is intended to offer to the interested reader a useful framework on work *to-be-done* and *already-done* on MOOD procedures for controller tuning.

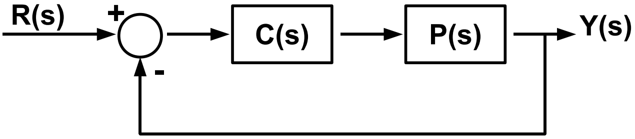


Figure 2.1: Basic control loop.

According to the basic control loop of Figure 2.1 and the review, some common choices for objectives in frequency domain are:

- Maximum value of the sensitivity function

$$J_{M_s}(\boldsymbol{\theta}) = \left\|(I + P(s)C(s))^{-1}\right\|_\infty \tag{2.1}$$

- Disturbance attenuation performance

$$J_{W_1}(\boldsymbol{\theta}) = \left\|W(s) \cdot (I + P(s)C(s))^{-1}\right\|_\infty < 1 \tag{2.2}$$

- Maximum value of the complementary sensitivity function

$$J_{M_p}(\boldsymbol{\theta}) = \left\| P(s)C(s)(I + P(s)C(s))^{-1} \right\|_{\infty} \tag{2.3}$$

- Robust stability performance.

$$J_{W_2}(\boldsymbol{\theta}) = \left\| W(s) \cdot (P(s)C(s)(I + P(s)C(s))^{-1}) \right\|_{\infty} < 1 \tag{2.4}$$

where $W(s)$ are weighting transfer functions commonly used in mixed sensitivity techniques. Meanwhile in time domain:

- Integral of the absolute error value

$$J_{IAE}(\boldsymbol{\theta}) = \int\limits_{t=t_0}^{T_f} |r(t) - y(t)| \, dt \tag{2.5}$$

- Integral of the time weighted absolute error value

$$J_{ITAE}(\boldsymbol{\theta}) = \int\limits_{t=t_0}^{T_f} t \, |r(t) - y(t)| \, dt \tag{2.6}$$

- Integral of the squared error value

$$J_{ISE}(\boldsymbol{\theta}) = \int\limits_{t=t_0}^{T_f} (r(t) - y(t))^2 \, dt \tag{2.7}$$

- Integral of the time weighted squared error value

$$J_{ITSE}(\boldsymbol{\theta}) = \int\limits_{t=t_0}^{T_f} t \, (r(t) - y(t))^2 \, dt \tag{2.8}$$

- Integral of the control action value

$$J_{ISU}(\boldsymbol{\theta}) = \int\limits_{t=t_0}^{T_f} (u(t))^2 \, dt \tag{2.9}$$

- Total variation of control action

$$J_{TV}(\boldsymbol{\theta}) = \int\limits_{t=t_0}^{T_f} \left| \frac{du}{dt} \right| \tag{2.10}$$

- Maximum value of control action

$$J_{maxU}(\boldsymbol{\theta}) = \max(u(t)), t \in [t_0, T_f] \tag{2.11}$$

where $r(t)$, $y(t)$, $u(t)$ are the references, measured variable and control actions in time t. Such objectives, for the sake of simplicity, have been stated in a general sense; details regarding specific implementation issues can be consulted in the references by the interested reader in each case.

2.2.1 PI-PID controller design concept

PID controllers are reliable digital control solutions thanks to their simplicity and efficacy [5, 8]. They represent a common solution for industrial applications and therefore, there is still ongoing research on new techniques for robust PID controller tuning [191]. Any improvement in PID tuning is worthwhile, owing to the minimum number of changes required for their incorporation into already operational control loops [187, 177]. As expected, several works have focused on the PID performance improvement.

Given a process $P(s)$, the following general description for a PID controller is used:

$$\begin{aligned} C(s) &= k_p\left(a + \frac{1}{T_i s} + b\frac{T_d}{\frac{T_d}{N}s+1}\right)R(s) \\ &- k_p\left(1 + \frac{1}{T_i s} + \frac{T_d}{\frac{T_d}{N}s+1}\right)Y(s) \end{aligned} \tag{2.12}$$

where k_p is the controller proportional gain, T_i the integral time, T_d the derivative time, N the derivative filter, a, b the setpoint weighting for proportional and derivative actions, $R(s)$ the reference and $Y(s)$ the measured signal. Therefore, the following design concepts (controllers) with their decision variables could be stated:

PI: $\boldsymbol{\theta}_{PI} = [k_p, T_i]$

PD: $\boldsymbol{\theta}_{PD} = [k_p, T_d]$

PID: $\boldsymbol{\theta}_{PID} = [k_p, T_i, T_d]$

PID/N: $\boldsymbol{\theta}_{PID/N} = [k_p, T_i, T_d, N]$

PI^1: $\boldsymbol{\theta}_{PI^1} = [k_p, T_i, a]$

PID^2: $\boldsymbol{\theta}_{PID^2} = [k_p, T_i, T_d, a, b]$

PID^2/N: $\boldsymbol{\theta}_{PID^2/N} = [k_p, T_i, T_d, N, a, b]$

Applications on PI-PID controller design concept

In [79], authors focus on defining a general MOP statement for a wide variety of applications and it is one of the few works that uses as decision variables all the flexibility degrees of a PID controller (gains, weighting factors, and filters). It proposes an MOP using four objectives for a PID^2/N controller. The aforementioned objectives are related with different norms of $J_{IAE}(\boldsymbol{\theta}_{PID^2/N})$, $J_{Ms}(\boldsymbol{\theta}_{PID^2/N})$, $J_{ITAE}(\boldsymbol{\theta}_{PID^2/N})$ (Equations (2.5), (2.1) and (2.6) respectively) and $J(\boldsymbol{\theta}_{PID^2/N}) = \left\|W(s) \cdot (I + P(s)C(s))^{-1}\right\|_2$ for noise reduction. Constraints to avoid saturation effects are also included. The PI^1 and PID^2/N concepts are compared (using SCp visualization) in order to appreciate the tradeoff differences in the MOP given for single input single output and multiple input multiple output processes (SISO and MIMO respectively).

[188] seems to be the first work that provides tuning rules for an MOP statement; that is, defining tuning rules with a certain degree of flexibility for the designer to select a desirable controller according to his/her preferences. It uses an integral MOOD tuning procedure for PI^1 controllers to build a multiobjective-based tuning rule procedure. The methodology begins with an identified first order plus dead time model to perform the optimization and MCDM step procedure. In the first step, a 3D Pareto front is analyzed to select a solution with the desired tradeoff among $J_{IAE}(\boldsymbol{\theta}_{PI^1})$ (for load disturbance), $J_{Ms}(\boldsymbol{\theta}_{PI^1})$ and $J_{TV}(\boldsymbol{\theta}_{PI^1})$ (Equations (2.5), (2.1) and (2.10) respectively). A second analysis in a 2D Pareto front is then performed to select the proper value of the setpoint weighting for setpoint response. The Pareto fronts are built according to a given value of the normalized time delay. The approach was validated with several SISO processes with different characteristics.

In [82], an EMO constrained statement for a PID/N controller is proposed for SISO processes. Three main objectives are defined: a low frequency performance objective based on $J(\boldsymbol{\theta}_{PID/N}) = P(s)(I+P(s)C(s))^{-1}$, a pass-band robust objective using an aggregate function of $J_{M_s}(\boldsymbol{\theta}_{PID/N})$ and $J_{M_p}(\boldsymbol{\theta}_{PID/N})$ (Equations (2.1) and (2.3) respectively), and a control activity objective based on $J(\boldsymbol{\theta}_{PID/N}) = C(s)(I + P(s)C(s))^{-1}$. Constraints are also incorporated to improve the pertinency of solutions. The MOEA proposed in the paper uses a single-objective optimization procedure in a subset of its population to improve convergence and speed. A SISO example is provided and after several comparisons (using SCp visualization), the authors noticed that using only frequency domain objectives does not give enough information to the designer regarding pertinency in time domain specifications. Therefore, the authors suggest that mixing objectives from both domains would significantly improve the interpretability and pertinency of the design alternative.

In [85], an MOEA using a simulated annealing (SA) based algorithm [184] is proposed for PID^2/N controller tuning for square MIMO processes. The algorithm uses a generation mechanism based on orthogonal experiment designs; this mechanism is used in order to incorporate a type of *systematic rea-*

soning method to generate promising Pareto optimal solutions. The design objectives used are the sensitivity and complementary sensitivity functions $J_{W_1}(\boldsymbol{\theta}_{PID^2/N})$, $J_{W_2}(\boldsymbol{\theta}_{PID^2/N})$ and squared error $J_{ISE}(\boldsymbol{\theta}_{PID^2/N})$ (Equations (2.2), (2.4) and (2.7) respectively). The algorithm is compared with a state-of-the-art algorithm and is evaluated in an aircraft simulated model. In this proposal, the constraint-handling and pertinency mechanisms are supported by the weighting function approach employed in the mixed sensitivity technique. Comparison with other techniques is shown, where a mixed 2-norm and 1-norm criterion is used to select a preferable solution.

In [149], a PID^2/N controller was tuned using a MOOD procedure. The controller employed a hybrid constraint approach using mixed sensitivity constraints as objectives. The process under consideration was a black-box nonlinear model with three different dynamics (underdamped, overdamped, and unstable) according to the operational zone. A set of nominal models was identified using a prior EMO statement and a total of 15 objectives (5 for each operational zone) based on $J_{IAE}(\boldsymbol{\theta}_{PID^2/N})$, $J_{ITAE}(\boldsymbol{\theta}_{PID^2/N})$, $J_{TV}(\boldsymbol{\theta}_{PID^2/N})$, $J_{W_1}(\boldsymbol{\theta}_{PID^2/N})$ and $J_{W_2}(\boldsymbol{\theta}_{PID^2/N})$ (Equations (2.5), (2.6), (2.10), (2.2) and (2.4) respectively) were stated. A subset of design alternatives was selected, according to the 2-norm from the normalized ideal solution in an LD visualization framework.

In [156] the optimization statement developed by [6] for PI^1 controllers is stated as an EMO problem for (square) MIMO processes. The purpose was to analyze the tradeoff between integral gain $J_{k_i}(\boldsymbol{\theta}_{PI^1}) = -\frac{k_p}{T_i}$, maximum value of the sensitivity function $J_{M_s}(\boldsymbol{\theta}_{PI^1})$ (Equation (2.1)) and maximum value of the complementary sensitivity function $J_{M_p}(\boldsymbol{\theta}_{PI^1})$ (Equation (2.3)) for setpoint changes (one triplet for each loop); additionally as overall robust performance the largest log modulus defined in [109] is stated. The tradeoff is discussed, and the design alternatives in the Pareto front are compared with other AOF and EMO approaches using an LD visualization.

In [168], a PID controller is tuned for a flexible alternating current transmission system. The overall goal was to find a controller capable of improving the load disturbance rejection performance with the minimum control effort. Two objectives were defined for this purpose: error measurement $J_{ISE}(\boldsymbol{\theta}_{PID})$ and control effort $J_{ISU}(\boldsymbol{\theta}_{PID})$ (Equations (2.7) and (2.9) respectively). The NSGA-II algorithm was used, and the selection procedure was performed using preference articulation by means of a fuzzy inference system. In this inference system, the minimum and maximum values of each objective in the calculated Pareto front are used to calculate the best solution.

It has been noticed that the MOOD procedure can be more time consuming than an AOF approach. Nevertheless, sometimes it is posible to run the optimization for a given nominal case, and use it as the basis for a particular (different) statement. A good example is provided in [3] where two design concepts, a PID and an I-PD controller (a PID^2 controller with $a = 0, b = 0$) are tuned using an MOEA for an optimal chemotherapy control model for cancer treatment. Both controllers are used to supply drug doses for a given period of time. Three objectives are defined: (maximize) tumour cell killing, (minimize) toxicity, and (minimize) tolerable drug concentration. The MOOD approach is used to observe and evaluate the effects of treatment. The purpose is to develop a *reference case* on the performance of the controllers to be used by the medical staff to determine the best treatment for a particular patient (since it is impractical to generate a Pareto front for *each* patient). To improve the pertinency of solutions, a goal vector approach is used. The visualization and selection procedure were performed using SCp when comparing both design concepts. The visualization process was useful, since it performed an in-depth analysis on the characteristics and physical meaning of the solutions.

In [196] a set of four PI controllers is proposed for the ALSTOM gasifier problem [48]. This gasifier is a popular MIMO popular benchmark, [23] and a wide variety of control structures have been developed and evaluated on it. It is a non-linear model of a coal gasifier, with several constraints. The defined MOP consisted of six objectives, each related to the integral of the absolute error $J_{IAE}(\boldsymbol{\theta}_{PI})$ (Equation (2.5)) for different load and disturbance scenarios. An SCp visualization in the decision space is used to analyse the performance of the calculated PI controllers; the MCDM selection was performed using a filtering process on the Pareto front approximation and a coal quality indicator (*i.e.* a new measurement is included in the MCDM process). This shows the flexibility that can be incorporated in the MCDM step, with new indicators included to evaluate the potential sets of solutions, before taking a final selection.

In [146], a double-acting hybrid magnetic thrust bearings and its controller were designed and optimized using a unified MOOD framework. A GA-based MOEA was used to optimize an MOP statement with 5 objectives, 14 constraints and 16 decision variables. Regarding the decision space, two decision variables correspond to a PD controller design concept; regarding objective space, 3 objectives are related to mechanical properties (exerted force, power loss and weight) and two with control performance: $J_{ISE}(\boldsymbol{\theta}_{PD})$ and $J_{ISU}(\boldsymbol{\theta}_{PD})$ (equations 2.7 and 2.9 respectively). Scatter diagram visualization is used in the MCDM stage and as a selection criteria, the closest solutions to the utopia point were selected and compared. In this case, this application could be an example of *holistic mechanical-and-control design*. That is, a multi-disciplinary design approach, where control and geometric decision variables are optimized on the same level.

In [201] a multivariable PI controller structure is tuned for two MIMO processes: a distillation column and an aircraft model. A bi-objective optimization statement is defined, using an AOF for robust stability and disturbance attenuation by means of $J_{W_1}(\boldsymbol{\theta}_{PI})$ and $J_{W_2}(\boldsymbol{\theta}_{PI})$ (Equations (2.2) and (2.4) re-

[23] Simulink models available at `http://www-staff.lboro.ac.uk/~elrd2/`

spectively). As a second objective, the integral of the squared error $J_{ISE}(\boldsymbol{\theta}_{PI})$ (Equation (2.7)) is stated; a PSO-based MOEA (lb-MOPSO) is proposed and compared with NSGA-II. Norm 1 criteria are used to compare controllers from the Pareto front with other design concepts.

In [179, 178] a controllability analysis of electronic valves using MOEAs is performed. A given valve is evaluated with respect to seven different performance objectives, based on a measure of transition time from release to landing, applied forces, landing speed, and armature velocity. The main goal is to determine the Pareto-optimal set of candidate actuators that fulfil a set of control requirements. Further analysis on the Pareto front is performed using PAc visualization. In the light of this visualization, several remarks are made about the tradeoff between objectives. This is a good example of how MOOD methodology can be used to design components with a guaranteed degree of controllability, *i.e.*, a design for controllability.

Conclusions on PI-PID controller design concept

In Table 2.1 a summary of these applications is provided. Brief remarks on MOP, EMO and MCDM for each work are given. From the above it is possible to notice the following:

Regarding the MOP statement, although works as [6], [140] and [89] remarked the importance of considering decision variables as N, a, b as integral parts of PID tuning procedure, few works focus on using those decision variables within an integral PID tuning methodology.

Moreover, several works focus mainly on SISO plants and tuned a single PID controller. Due to the fact that several and well established tuning procedures exist, there is a missing justification on when a MOOD approach could produce better solutions for the designer. Such justification is not related with the optimization problem *per se*; that is, it is not about the difficulty in finding a solution, but the difficulty in finding a solution with a reasonable tradeoff between conflicting objectives. In this framework, the MOOD approach for SISO loops can be justifiable.

MIMO processes seem to be a promising area where the capabilities of the MOOD procedure could be evaluated. This is due to the fact that MIMO processes can be more complex, and a reasonable tradeoff between conflicting objectives can be hard to find. In this case, dealing with MIMO processes, mechanisms for many-objective and expensive optimization could be valuable.

It is also interesting to notice those works where multidisciplinary optimization is performed, merging for example control and mechanical decision variables. It could lead to interesting optimization instances, since the shape of the decision space will differ to that of the control parameters. Multimodal optimization instances could be interesting to apply in such problems, since physical variables need to be minded.

Regarding the optimizer, MOEAs based on GA seem to be more popular for PID controller tuning than their counterparts. This leads to the following question: *Are the operators used in GA best suited for PID controller tuning? Is the stochastic search employed by MOEAs that is based on GA capable of finding more accurate Pareto front approximations in the case of PID tuning?* There are no works addressing this question, which could be an interesting issue for practitioners and designers when selecting an algorithm.

A less considered topic about the stochastic search used in MOEAs for PID controller tuning, is about the proper selection of decision variable bounds. That is, *how to select the* $[\underline{\theta}_i, \overline{\theta}_i]$ *for each* $\theta_i \in \boldsymbol{\theta}$ *with respect to internal closed-loop stability?* While commonly uncommented, this is a primary constraint in controller tuning. In this sense, given the stochastic nature of MOEAs, it could be important to fulfill the following two requirements for stochastic sampling:

1. Any sampled controller must stabilize the closed loop.
2. Any stabilizing controller $C(s)$ of the process $P(s)$ must be contained in $[\underline{\theta}_i, \overline{\theta}_i]$, $\theta_i \in \boldsymbol{\theta}$.

A common approach for feature 1 is to define bounds on the parameter that avoid all non-stable (but also some stable) PID parameters; therefore, feature 2 is not fulfilled. A second alternative, is to bound the search space with all stable PID parameters, but including non-stable parameters that are verified while the algorithm is running; this obviously does not fulfil feature 1, and could misspend computational (CPU time) resources. Therefore, techniques for stochastic sampling that address both requirements could be interesting.

It seems that the MOOD procedure brings a suitable framework that includes constraints in the optimization problem, in comparison with evolutionary approaches for AOF statements for PID controllers [152]. It seems that the simplicity of the PID and algorithms available have enabled many-objective optimization problems to be stated. While it seems that the optimizers incorporate some constraint handling mechanism quite well, it is not usual to find applications with preference handling incorporated in the algorithm to improve pertinency (in an *a-priori* or *interactive* sense).

Finally, in the MCDM, classical approaches for visualization based on SCp and 3D representation are the most used, despite the number of objectives defined for the MOP. However, guidelines on this procedure, although valuable, are not well covered.

2.2.2 Fuzzy controller design concept

Fuzzy systems have been widely and successfully used in control system applications as referenced in [57]. As in the PID design concept, the MOOD is

Table 2.1: Summary of MOOD procedure for PID design concept. $\boldsymbol{J}(\boldsymbol{\theta})$ *refers to the number of objectives;* $\boldsymbol{\theta}$ *to the number of decision variables and* $\boldsymbol{g}(\boldsymbol{\theta})$, $\boldsymbol{h}(\boldsymbol{\theta})$ *to the number of inequality and equality constraints, respectively (according with examples provided in each paper). In some instances, constraints are also stated as objectives.*

			MOP			EMO		MCDM	
Concept(s)	Process(es)	Ref	$\boldsymbol{J}(\boldsymbol{\theta})$	$\boldsymbol{\theta}$	$(\boldsymbol{g}(\boldsymbol{\theta}), \boldsymbol{h}(\boldsymbol{\theta}))$	Algorithm	Related features from Chapter 2	Plot	Selection insights
PID^2/N, PI^1	SISO, MIMO	[79]	4	7	(2,0)	GA	4	3D, SCp	Concepts comparison.
PI^1	FOPDT	[188]	4	3	(0,0)	GA		3D, 2D	Tuning rule methodology.
PID	Electromagnetic	[178]	7	27	(2,0)	GA	5; 4; 3	PAc	Iterative controllability analysis for a given design.
PID/N	SISO	[82]	3	4	(4,0)	Ad hoc	4; 3	SCp	Incorporate analysis of time domain objectives.
PID	Aeronautical	[85]	3	27	(2,0)	Ad hoc	4	SCp	Analysis with other techniques.
PD	Mechatronical	[146]	5	14	(14,0)	GA	4; 5	SCp	Design alternatives comparison.
PID^2/N	SISO	[149]	15	7	(6,0)	GA	4; 5	LD	Selection according to preferences.
P + PI (3)	Chemical	[196]	6	8	(1,0)	NSGA-II	4; 5	SCp	New indicator included for selection.
PID	Electrical	[168]	2	3	(0,0)	NSGA-II		None	Fuzzy based selection.
PID, I-PD	Bio-medical	[3]	2	3	(3,0)	GA	4	SCp	Concepts comparison; intended to support specific treatment.
PID	Chemical, Aeronautical	[201]	2	27	(1,0)	PSO	4	SCp	Norm 1 selection.
PI	Chemical	[156]	7	4	(3,0)	DE	4; 5	LD	Trade-off analysis.

useful for analyzing the tradeoff between conflicting objectives. In this case, the fuzzy controller is more complex to tune, given its non-linearity. A comprehensive compendium on the synergy between fuzzy tools and MOEAs is given in [56]. This thesis will focus on controller implementations. In general, decision variables regard $\boldsymbol{\theta} = [\Lambda, \Upsilon, \overline{\overline{\Lambda}}, \overline{\overline{\Upsilon}}, \mu]$ where:

Λ**:** is the membership function shape.

$\overline{\overline{\Lambda}}$**:** is the number of membership functions.

Υ**:** is the fuzzy rule structure.

$\overline{\overline{\Upsilon}}$**:** is the number of fuzzy rules.

μ**:** are the weights of the fuzzy inference system.

Applications on fuzzy controller design concept

In [97] a fuzzy logic controller was adjusted using NSGA-II for a base-isolation system using a magnetorheological damper. A total of four objectives were stated, based on the maximum displacement, maximum acceleration, and their root mean squared values for different earthquake profiles using as decision variables $\boldsymbol{\theta} = (\Lambda, \Upsilon, \mu)$. A design concept comparison with a skyhook controller was then performed using an SCp visualization. It was remarked that with regard to a pair of objectives, there was practically no difference among concepts (using as a basis the number of dominated solutions). Nevertheless when four objectives were considered, the MOOD tuning procedure using NSGA-II outperformed the other design concept. This is a good example of how it is possible to ask more from an existing tuning rule by adding new objectives and indices to discriminate solutions; nevertheless, in this work it was remarked that the more the objectives, the greater the ratio of non-dominated solutions. This issue is typical in many-objective optimization problems, as discussed above.

In [25], a fuzzy scheduling controller is adjusted using an MOEA for a gas turbine engine model. This is a complex aerospace system where several factors (such as safety, reliability, and maintainability) need to be considered as control requirements. A MOGA algorithm is used to tune the membership functions Λ, and the scaling factors μ of the fuzzy controller. A total of nine different objectives were stated: rise time and settling time of the compressor, engine thrust and its rise time, maximum nozzle and turbine temperatures, the fan pressure ratio, low pressure surge margin and thermodynamic stress. A PAc visualization is used to analyze the tradeoff among design alternatives. In this case, no controller was capable of fulfilling all the requirements, and an extensive analysis of the conflicting objectives was made using a tradeoff matrix to find suitable controllers. This is a good example of the designer interacting with the optimizer to improve the pertinency of the solutions (progressive preference articulation).

In [114], a MOOD is used for parameter tuning of a fuzzy controller for vibration suppression on smart structures. Objectives based on $J_{ISE}(\Lambda, \mu, \Upsilon)$ (Equation (2.7)) for nodal displacements, velocities and accelerations were stated. Several tests on the behavior of the design alternatives were made to guide the final controller selection by means of a multi-objective PSO algorithm. The main objective of this work is to focus on the EMO stage by evaluating the best PSO strategy based on a number of non-dominated solution criteria. Several design alternatives are evaluated to analyze the tradeoff achieved by each MOEA alternative. A work like this could be used as a reference to select the most adequate evolutionary strategy (regarding its exploitation and exploration capabilities) for a given MOP.

In [65] a fuzzy controller is tuned by optimizing a quality function and the number of rules in the fuzzy logic controller. The example provided is a heat, ventilation, and air conditioning (HVAC) system with a fuzzy controller with nine inputs and three outputs. It is a very complex process given the considerable time requirements and computation time needed to run a simulation to measure the performance. The quality function used comprises an aggregate objective function $J_{AOF}(\Lambda, \overline{\overline{\Upsilon}})$ of thermal comfort, air quality requirements, energy consumption, and system stability. The aim of this work was to obtain a controller to accomplish the quality function cost with a reduced (tractable) number of rules; and therefore $J(\Lambda, \overline{\overline{\Upsilon}}) = \overline{\overline{\Upsilon}}$ is considered as a second objective. A design concept analysis (2D visualization) is presented at two different levels: among various MOEAs and controllers. The advantages of considering the number of rules was demonstrated since it was possible to reduce them and still fulfil the requirements with better results in the cost function.

In [54], a fuzzy logic controller for drug infusion for anesthesia was proposed. Traditionally, this administration is performed by an expert; therefore, a fuzzy controller is a good candidate to incorporate such expertise into an automatic control scheme. Nevertheless, when trying to optimize the performance using membership functions and rule structures as decision variables, the interpretability of the controller may be lost. This is a undesirable situation given that for the medical staff it is fundamental to understand the logic behind the controller for the correct anesthesia infusion. For this reason, two objectives were stated: control action quality by means of an aggregate objective function $J_{AOF}(\Lambda, \Upsilon)$ based on $J_{ISE}(\Lambda, \Upsilon)$, $J_{ISU}(\Lambda, \Upsilon)$; and an interpretability index $J_{II}(\Lambda, \Upsilon)$. In the MCDM step, several design concepts are compared with an alternative solution that is closer to the ideal solution in the calculated Pareto front.

Finally, in [180], fuzzy logic controllers are adjusted on-line using MOEAs. The main purpose of this work was to investigate the applicability of MOEA for control design in a hardware-in-the-loop context. The selected process was a sealed pump running on magnetic bearings. Four objectives were minimized: rise time, steady state error, power utilization, and control complexity, using as decision variables $\boldsymbol{\theta} = (\overline{\overline{\Upsilon}}, \overline{\overline{\Lambda}}, \Upsilon)$. Whereas several controllers fulfil the required specifications, those with the least complexity were selected. As observed by the authors, lower controller complexity offers several computational advantages when working on-line.

Conclusions on fuzzy controller design concept

In Table 2.2 a summary on these applications is provided. The difference in the quantity of the works dedicated to fuzzy controllers and PID controllers is noticeable.

With regard to MOP definition, it seems that EMO has been popular to simultaneously optimize objectives related with performance and interpretability of the fuzzy inference system. It can also be observed that few works incorporate constraints in the MOP statement; in the same way, it is unusual to find instances of many-objective optimizations. This could be due to the fact that such types of design concepts need to be handled with large scale optimization instances (several decision variables). This is justifiable if it is considered that the first issue to handle are the parameters of the controller, before dealing with additional objectives and/or constraints. The reason constraints are not

Table 2.2: Summary of MOOD methodology for Fuzzy Controller design concept. $\boldsymbol{J}(\boldsymbol{\theta})$ refers to the number of objectives; $\boldsymbol{\theta}$ to the number of decision variables and $\boldsymbol{g}(\boldsymbol{\theta})$, $\boldsymbol{h}(\boldsymbol{\theta})$ to the number of inequality and equality constraints, respectively (according with examples provided in each paper). In some instances, constraints are also stated as objectives.

			MOP			EMO		MCDM	
Concept(s)	Process(es)	Ref	$\boldsymbol{J}(\boldsymbol{\theta})$	$\boldsymbol{\theta}$	$(\boldsymbol{g}(\boldsymbol{\theta}),$ $\boldsymbol{h}(\boldsymbol{\theta}))$	Algorithm	Related features from Chapter 2	Plot	Selection insights
Fuzzy scheduling	Aeronautical	[25]	9	100	(9, 0)	GA	3,5	PAc	Constraint violation analysis; fine tuning.
PID, Fuzzy Controller	DC motor (HiL)	[180]	4	34	(0,0)	GA	6	None	According performance.
Fuzzy controller	Geological	[97]	4	160	(0,0)	NSGA-II	10	SCp	Design alternatives comparison.
Fuzzy controller	Bio-medical	[54]	2	40	(0,0)	SPEA based	10	2D	Design alternatives/concepts comparison with other controllers. Selection by norm-2 criteria.
Fuzzy controller	Mechanical	[114]	3	30	(0, 0)	PSO		3D	Design alternatives comparison.
Fuzzy controller	HVAC system	[65]	2	3096	(0, 0)	SPEA based	10,9	2D	Design alternatives comparison at two levels: different controllers and different MOEAs.

fully incorporated, could be because of incompatibilities with large scale optimization and the constraint handling; therefore the question here is: *Is there a gap in MOEAs to satisfy such a need?* That this scalability issue is an open problem was also noticed in [56]. Finally, in the MCDM step, SCp tools have been sufficient for Pareto front visualization and analysis, due to the low number of objectives stated in the MOP. Nevertheless, as more objectives are included in the MOP, some mechanism to deal with scalability and many-objective optimization will be required, with integration of decision maker preferences, as stated in the same review.

2.2.3 State space feedback controller design concept

The state space representation has shown to be a remarkable tool for controller design. Several advanced control techniques use this representation to calculate a controller (in the same representation) with a desired performance. The decision variables stated are the gains of the matrix $\boldsymbol{G}$. Classical optimization approaches in a MOOD framework have been used in [120] with good results. In several instances, it seems that the MOOD procedure has been used to compare classical approaches with the EMO approach, as presented below.

Applications on state space feedback controller design concept

In [91], a new algorithm is developed for state space feedback controller tuning. The algorithm is based on an ϵ-elimination procedure, which eliminates similar individuals in the objective space sense, and in the decision variable space. This could be helpful to improve the diversity of solutions when the decision vector could be crucial in discriminating or accepting solutions. The algorithm is evaluated in a single inverted pendulum model using two objectives in two different instances; in the first instance, an aggregate function objective of the settling time and overshoot of the cart $J_{AFO_1}(\boldsymbol{G})$ and pendulum $J_{AFO_2}(\boldsymbol{G})$ were stated. In the second, the probabilities of failure $J_{Pr_1}(\boldsymbol{G})$, $J_{Pr_2}(\boldsymbol{G})$ measured in a probabilistic set of models were used as objectives. As two objectives were stated for each instance, a classical SCp visualization and decision analysis were performed.

In [80], the algorithm developed in [79] (covered in the PID design concept) is used for n-order controller tuning for the SISO and MIMO processes. Initially, the algorithm's capacity to calculate controllers under an H_2/H_∞ framework was compared with LMI techniques using a 2D representation. One of the advantages of the MOOD approach is flexibility in fixing the controller order size. Several controllers with different order sizes (to define different design concepts) were proposed for MIMO controller tuning. This last statement enabled the performance improvement of the controller to be analyzed as the controller order size variate. Similarly, in [129], the MOOD approach is compared with the LMI concept for an LQR and H_2/H_∞ controllers using as objectives noise sensitivities for output and actuator.

Conclusions on state space feedback controller design concept

In Table 2.3 a summary on these applications is provided. There are still few works focusing on this design concept and therefore, it is difficult to extrapolate conclusions as in the PID and fuzzy cases. It can be observed that the MOOD procedure has been compared with modern tuning techniques such as H_2/H_∞ and LQC techniques. The stochastic sampling used by evolutionary techniques to search the $\boldsymbol{G}$ gains on the control matrix could lead to the same issues noted in PID control (section 2.2.1). Given that H_2/H_∞ and LQC techniques are also based on optimization, it could be interesting to hybridize both approaches for this design concept.

2.2.4 Predictive control design concept

Online applications for MOOD are not straightforward, since the MCDM stage must be carried out, in some instances, automatically. As a result, analysis that relies on the DM must be codified to become an automatic process. Predictive control techniques have been incorporating the MOOD framework and the MCDM procedure in their optimization stages. Good examples using deterministic approaches are presented in [15, 198, 145]. Approaches using EMO in the MOOD procedure are presented below; decision variables regard to $\boldsymbol{\theta} = \boldsymbol{U}$ (control action through the control horizon) or $\boldsymbol{\theta} = \boldsymbol{R}$ (references given to the controllers through the control horizon).

Table 2.3: Summary of MOOD methodology for state space feedback controller design concept. $\boldsymbol{J}(\boldsymbol{\theta})$ refers to the number of objectives; $\boldsymbol{\theta}$ to the number of decision variables and $\boldsymbol{g}(\boldsymbol{\theta})$, $\boldsymbol{h}(\boldsymbol{\theta})$ to the number of inequality and equality constraints, respectively (according with the examples provided in the papers). In some instances, constraints are also counted as objectives.

			MOP			EMO		MCDM	
Concept(s)	Process(es)	Ref	$\boldsymbol{J}(\boldsymbol{\theta})$	$\boldsymbol{\theta}$	$(\boldsymbol{g}(\boldsymbol{\theta}),$ $\boldsymbol{h}(\boldsymbol{\theta}))$	Algorithm	Related features from Chapter 2	Plot	Selection insights
H_2/H_∞	SISO, MIMO	[80]	3	13	(1,0)	GA	4	SCp	Concepts comparison with LMI design
LQG, H_2/H_∞	SISO	[129]	3	6	(1,0)	GA	4, 3	2D	Concepts comparison with LMI.
State space controller	Mechanical	[91]	4	4	(0,0)	GA	7, 8	SCp	Design alternatives comparison

Applications on predictive control design concept

Recently, [66] proposed a MOOD procedure, where an evolutionary approach and an inference system are used for non-linear systems. Firstly, a NARX dynamic neural network is used to build a model of the plant. Then, a multiobjective genetic algorithm is used to approximate a Pareto front on the selected prediction horizon $J_{ISE}(\boldsymbol{U})$; afterwards a fuzzy inference system is used to select the control action to be applied. This is an interesting example where several computational intelligence tools are used for modeling, optimizing and selecting the most suitable control action.

In [26] NSGA-II is used to generate the Pareto set of optimal trajectories $\theta = \boldsymbol{R}$ for setpoint changes for supervisory control of flotation columns. The MOOD is used to meet the engineering and market requirements that are specifically stated for the control problem. Stated objectives include reducing the change of the references upon the control horizon, economic profit, hydraulic stability (subject to constraints related with froth overloading), quality of the concentrate, and cleaning efficiency. The optimization, an automatic decision making procedure based on normalized distances to the ideal solution and constraint fulfillment, is then defined to select the best trajectory.

Conclusions on predictive control design concept

In Table 2.4 a summary on these applications is provided. Predictive control seems to be an opportunity to apply the MOOD approach, due to the few works dedicated to this control design alternative. Nevertheless, it can also be seen that the problem relies on tracking the Pareto front each sampling time, as well as performing the selection procedure *on the fly* (interactive approach). An alternative, not exploited here, is to use the MOOD approach in an *upper* layer where the DM may analyze the Pareto set approximation and change the control objectives accordingly (as employed in [145] with multiobjective deterministic algorithms).

Table 2.4: Summary of MOOD methodology for predictive control concept. $\boldsymbol{J}(\boldsymbol{\theta})$ *refers to the number of objectives;* $\boldsymbol{\theta}$ *to the number of decision variables and* $\boldsymbol{g}(\boldsymbol{\theta})$*,* $\boldsymbol{h}(\boldsymbol{\theta})$ *to the number of inequality and equality constraints, respectively (according with the examples provided in the papers). In some instances, constraints are also counted as objectives.*

			MOP			EMO		MCDM	
Concept(s)	Process(es)	Ref	$\boldsymbol{J}(\boldsymbol{\theta})$	$\boldsymbol{\theta}$	$(\boldsymbol{g}(\boldsymbol{\theta}),$ $\boldsymbol{h}(\boldsymbol{\theta}))$	Algorithm	Related features from Chapter 2	Plot	Selection insights
Predictive control	Mechanical	[66]	2	8	(0,0)	GA	6	None	Fuzzy inference system is used.
Predictive control	Chemical	[26]	8	-	(4,0)	NSGA-II	6; 4	None	Successive ordering according to feasibility.

2.2.5 Other design concept approaches

The MOOD methodology has also been used with *ad-hoc* controller structures to address the performance of particular and complex processes. That is, specific controllers structures for specific processes. They are commented below.

Applications

In [107] a disk drive servo controller was tuned using an MOEA. An *ad-hoc* algorithm was proposed that assigns greater priority to constraint objectives than optimization objectives. These constraints are bounds on certain design objectives that avoid non-pertinent solutions. A goal objective vector is defined with a prioritized ranking approach, for two different tracks that seek values where constraints (3) are included as objectives (for a total of 10 objectives). Objectives are based on $J_{ISU}(\boldsymbol{\theta})$ and $J_{TV}(\boldsymbol{\theta})$ constrained to desired values of steady state error, standard deviation of following error and seek time. To deal with the many objective optimization instance, a preference articulation is included in the algorithm to focus the search towards a pertinent Pareto front. In this work, the designer identifies the importance of having a solution reference (goal vector) and uses it in the evolution process to improve the pertinency of solutions. Pareto front visualization and analysis were performed using a PAc visualization. After a validation in a simulated model, the design alternatives selected are evaluated in the physical process.

In [181] a pole placement controller for an electronically operated throttle system is adjusted using a MOOD methodology. The process is a highly non-linear system, with several time domain specifications and requirements to fulfil. Firstly, a low-order model is used to identify the tractable number of poles in the controller (nine in this case). A high-order non-linear model (experimentally verified) is then employed to select the pole location using a MOEA. A total of five objectives are defined: rise time, overshoot, settling time, steady-state error, and system delay. The final selection was used by the designer in accordance with his expertise. It was remarked that the extra design work was worthwhile since a controller with a better response lag was calculated.

In [2] a MOGA algorithm is used to generate *command shapers* for the control of a flexible manipulator; this kind of manipulator is light and fast but vibration control is difficult to achieve. A command shaper control is a good alternative for control, but several design objectives could be in conflict. For this reason, the MOOD approach is used to generate a set of potentially preferable controllers. Six objectives are defined and a PAc visualization is used to compare design alternatives. Objectives are based on settling time (for hub and end-point), rise time, peak to peak oscillation, infinity norm and root mean squared value of end-point acceleration. The main advantage identified by the authors is that the MOEA does not require a priori knowledge of the natural frequency to calculate the command shaper parameters, as required by the original design methodology. The methodology process includes a preliminary analysis of three different command shapers (design concepts) in 2D; the most promising concept is then used to deal with a six objective space.

In [36], a design concept comparison for different controllers was performed using the MOOD approach. A MOEA was used to calculate the best control action that cancels vibrations in a magnetorheological damper (MRD). Since this is not possible (in the practical sense) because it requires a perfect knowledge of the current state of the damper, such a control action and its performance were used to bound the performance for a set of controllers. A Skyhook control, a feedback linearization control, and a sliding mode control were compared using this framework and two objectives were used: mean dissipated power and absorbed power. A design concept comparison was commented on that lead to the selection of the best control solution according to the current state of the MRD. This is a good example of how a design concept comparison can assist the DM in selecting a desired controller according to his/her preferences.

Finally, in [74], a MOEA is used to tune the parameters of a fractional PID controller (PI$^\lambda$D$^\nu$) [130] using a reliability based optimization approach with stochastic sampling. This type of controller has some advantages over the classical PID controllers, but tuning techniques are more difficult. The authors chose a MOOD approach to select the parameters of this type of controller in order to fully appreciate its performance. A design concept comparison with a classical PID controller was made using an LD visualization framework with five design objectives. Objectives stated include the probabilities of failure of $J_S(\theta_{PI^\lambda D^\nu})$, $J_T(\theta_{PI^\lambda D^\nu})$, $J_{ITSE}(\theta_{PI^\lambda D^\nu})$ and $J_{U_{max}}(\theta_{PI^\lambda D^\nu})$ (Equations (2.1), (2.3), (2.8) and (2.11)) mixing time domain and frequency domain objectives. Using such an approach, it was possible to appreciate the drawbacks and advantages of using a complex PID controller. As mentioned previously, this could be useful for the designer in justifying his/her selection.

Table 2.5: Summary of MOOD methodology for control systems engineering. $\boldsymbol{J}(\boldsymbol{\theta})$ *refers to the number of objectives;* $\boldsymbol{\theta}$ *to the number of decision variables and* $\boldsymbol{g}(\boldsymbol{\theta})$, $\boldsymbol{h}(\boldsymbol{\theta})$ *to the number of inequality and equality constraints, respectively (according with the examples provided in the papers). In some instances, constraints are also counted as objectives.*

			MOP			EMO		MCDM	
Concept(s)	Process(es)	Ref	$\boldsymbol{J}(\boldsymbol{\theta})$	$\boldsymbol{\theta}$	$(\boldsymbol{g}(\boldsymbol{\theta}), \boldsymbol{h}(\boldsymbol{\theta}))$	Algorithm	Related features from Chapter 2	Plot	Selection insights
State space controller	Electronic	[107]	10	7	(6,0)	GA	4, 5,3	PAc	Design alternatives comparison
RST controller	Mechanical	[181]	5	9	(0,0)	GA	5	None	Design alternative evaluation
Command Shapers	Robotics	[2]	2,6	8	(0,0)	GA	5	PAc	Concepts comparison in 2D. Selection of a flexible controller for 6D objectives.
Several	Mechanical	[36]	2	–	(0,0)	GA		2D	Concepts comparison.
PID, PI$^{\lambda}$D$^{\mu}$	SISO	[74]	5	5	(0,0)	NSGA-II	5; 8	LD	Design alternatives comparison.

Conclusions

In Table 2.5 a summary on these applications is provided. An interesting fact is that GA seems to be the first option as a evolutionary technique for *ad-hoc* controllers. Also it seems that the option with specific controllers is to evaluate their performance in many-objective optimization instances.

2.3 Conclusions on this chapter

A review of MOOD methodologies in control systems engineering field was presented. The MOOD procedure is a GFCL approach that includes a MOP statement, a multiobjective optimization process, and an MCDM step. All of them are important for a successful implementation of the MOOD approach and embedment of the DM into the design process. This review has focused on EMO techniques for the optimization stages for MOPs dealing with controller tuning.

The MOOD procedure has been shown to be a useful tool for parametric controller tuning. Such approach allows the designer to have a different insight into design alternatives and their tradeoff, in order to select the most convenient or preferable solution for the DM. The MOOD procedure requires a closest embedment of the designer and it is more time consuming than an AOF approach, due to the multiobjective optimization stage and the MCDM step. For this reason, this approach could be reserved for complex MOP instances, where it is worthwhile to expend more time in the design phase to analyze the objective exchange among design alternatives.

Several applications on controller tuning for different design concepts have been presented and discussed. The MOOD procedure has been used for different kinds of controllers, from simple to complex architectures in a wide variety of applications. Several trends will be discussed next with regard to

Table 2.6: Summary of EMO features in controller tuning applications.

Concept(s)	Ref	Pertinency	Constrained	Many Objectives	Dynamic	Multi-modal	Robust	Expensive	Large Scale
PID Control	[79]		X						
	[188]								
	[178]	X	X	X					
	[82]	X	X						
	[85]		X						
	[146]		X	X					
	[149]		X	X					
	[196]		X	X					
	[168]								
	[3]		X						
	[201]		X						
	[156]		X	X					
Fuzzy Control	[25]	X	X	X					
	[180]				X				
	[97]								X
	[54]								X
	[114]								
	[65]							X	X
Space	[80]		X						X
	[129]	X	X						
	[91]					X	X		
Predictive Control	[66]				X				
	[26]		X		X				
State	[107]	X	X	X					
Others	[2]	X		X					
	[181]			X					
	[36]								
	[74]			X			X		

each of the steps in the MOOD procedure.

2.3.1 The multiobjective problem statement

Perhaps the first question to answer regarding the MOP is: *What kind of problems are worth to address with MOOD?* As noticed in [126], more activity should be focused on identifying which problems are *real world problems* that should be handled by MOEAs. There is no doubt that controller tuning is a real problem with practical applications. Nevertheless, for the same reason there are several techniques for controller tuning. Therefore, some research should be focus on identifying what type of MOP is best solved by the MOOD procedure in controller tuning. The following two questions could be helpful:

- *Is it difficult to find a controller with a reasonable balance among design objectives?*
- *Is it worthwhile analyzing the tradeoff among controllers (design alternatives)?*

If the answer is yes to both questions, then the MOOD procedure could be an appropriate tool for the problem at hand. Otherwise, tuning techniques or AOF approaches could be enough.

For complex controllers (Fuzzy controllers for instance), the MOOD could be a valuable alternative. But in the specific case of PID controller, it has been noticed that several works have focused on tuning a single loop. In such case, the MOOD has to compete with several and well established tuning procedures and techniques. Perhaps efforts should be focused on MIMO processes, that could be more complex to tune and therefore, to find a solution with a reasonable tradeoff.

An alternative to the MOOD procedure is to be used at the beginning of the design phase. The multi-disciplinary optimization approach, where mechanical and control design could be merged, is recognized as a mathematical challenge in optimization [161] and a promising optimization approach for design [116]. This integration is exploited by commercial products as modeFRONTIER[24] which may introduce interesting possibilities for holistic analyzes in multi-disciplinary design.

According to Table 2.6, some gaps among controller design concepts and MOEAs are detected. Such gaps could be due to:

- an MOP where such desirable characteristics could be required has not been proposed;
- the available algorithms do not reach the specifications required by the designer to provide useful Pareto front approximations.

Regarding the former possibility, if no MOPs are defined that require such features, is this because such MOPs: *are not interesting from the point of view of the designer?* or because *the designer is unaware of the possibility of dealing with them using an EMO approach?* With regard to the latter possibility, if MOEAs are not good enough, *is it due to a lack of understanding of the MOP with respect to controller tuning?* In any case, these questions could provide a starting point for new MOP statements.

Regarding the objectives definition, frequency and time domain performance measures have been used to identify preferable solutions. Nevertheless, few works use both kind of objectives in the same MOP statement. Therefore, merging both to improve the pertinency of solutions could be a possible direction.

[24] `http://www.esteco.com/modefrontier`

2.3.2 The evolutionary multiobjective process

With regard to the optimizer, its selection should be made according to the MOP statement in order to guarantee the designer's desired performance. That is, it should be made according to the different features that could be desired for the algorithm. Those characteristics could be related to the quality of the Pareto front (convergence, diversity, pertinency) or to the MOP stated (constrained, multi-modal, many-objective, computationally expensive, dynamic or robust optimization for example).

As remarked in [173], there are several works focusing on *new* evolutionary or bio-inspired techniques instead of focusing in other more *practical* aspects of the optimization problem. The question is *do we need so many MOEAs to address the controller tuning problem?*

Several MOEAs are available for the designer offering such capabilities, and research on new algorithms for controller tuning should be oriented to amend any existing gap (according Table 2.6) identified by the designer's requirements. Research should be focused on mechanisms to support (if required) the different optimization instances from section 1.4.2, instead of defining new evolutionary techniques. Given that convergence and diversity are expected properties, and almost every MOEA includes mechanisms for those purposes, the efforts could be oriented to preferences inclusion. As it was commented before, the MOOD in the controller tuning framework could be used for those problems where it is difficult to find a reasonable tradeoff. That means that the designer would have some ideas concerning his or her needs, and this information could be merged into the evolutionary process. This could be a valuable mechanism to lead the evolutionary search efficiently towards a pertinent Pareto front approximation. This could facilitate the optimization in many-objective and large-scale optimization instances.

Also, the following questions should be answered: *which evolutionary techniques fit better for a given design concept?* That is, *which kind of exploitation and exploration capabilities are better for approximating a Pareto front? Which operators fit better on the objective space, to approximate a Pareto front given a controller? Which problems are best addressed by a particular evolutionary technique? Is there any difference? How can problems that are separable, non-separable or a mixture of both, be addressed?* One possibility for answering those questions is to define a proper control benchmark, suitable to be solved by using the MOOD procedure. Although there are several works using this procedure for controller tuning, a common benchmark problem has not been stated to compare performance on MOEAs and decision making procedures. The Alstom gasifier [48] or the Boiler process [133] could represent excellent benchmark platforms for this purpose.

Finally, more basic research would be helpful, to ensure the two characteristics for stochastic sampling referred in Section 2.2.1 as well as work focusing on optimization hybrid techniques.

2.3.3 The multicriteria decision making step

Concerning the MCDM step several methodologies and visualizations have been presented that could help the DM take his/her final resolution. Nevertheless, more insights into the MCDM procedure need to be documented, since it is not always an easy task to perform. This gap could be amended by bringing more tools from the multi-dimensional data and decision analysis field.

In the visualization case, *is it possible to define different visualization approaches for analyzing multidimensional data and multidimensional Pareto front that fit better for controller tuning?* There are several GUIs developed for controller tuning, but merging such capabilities with an analysis of Pareto front (with regard to simplicity, persistence and completeness) could be useful for the designer. Also an interesting feature to develop would be a design concepts comparison of different controller in multidimensional Pareto fronts. Such an analysis could bring conclusions about when it is worthwhile using a complicated control technique rather than a simpler one.

Finally, there is an uncommented issue regarding the quantity of Pareto optimal solutions that a MOEA needs to approximate for the MCDM stage. Several algorithms evaluate their performance by generating a very dense Pareto front approximation with several solutions. Nevertheless, it is necessary to remember that the designer will analyze the tradeoff of those solutions, and a large number of solutions could be more confusing than helpful. Therefore, MOEAs should focus on producing the most pertinent solutions.

Next, in Part II some preliminary contributions of this MOOD procedure for controller tuning for MIMO processes will be presented. Such contributions focus on multivariable processes (identified as a potential research area) and in many-objective optimization instances with the tools presented in Section 1.5. In those applications, opportunities for further improvements on tools for the MOOD procedure will be detected, becoming the basis for contributions in Part III.

Part II

Preliminary Contributions on Controller Tuning

PART II:

Preliminary Contributions on Controller Tuning

This part is devoted to present early contributions using the MOOD procedure for controller tuning. Such contributions focus on multivariable processes (identified as a potential research area in Chapter 2) and on many-objective optimization instances. Basic ideas developed through this part will show their usefulness for such optimization statements; nevertheless, some gaps (regarding the tools from Section 1.5) will be likewise identified. These gaps will become the basis for the contributions to be presented in Part III (Contributions on MOOD Tools).

CHAPTER 3:

Multiobjective optimization design procedure for multivariable PI controller tuning

In this chapter, preliminary contributions on PI controller tuning for multivariable processes using the MOOD procedure are presented. Contents of this chapter appear in the following paper:

- **G. Reynoso-Meza, J. Sanchis, X. Blasco, J.M. Herrero**. *Multiobjective evolutionary algorithms for multivariable PI controller design*. Expert Systems with Applications. Vol. 39, Issue 9, July 2012. Pp. 7895-7907.

3

Multiobjective optimization design procedure for multivariable PI controller tuning

> *Humankind cannot gain something without first giving something in return. To obtain, something of equal value must be lost. This is Alchemy's First Law of Equivalent Exchange.*
>
> Hiromu Arakawa, *Fullmetal Alchemist*

3.1 Aim of this chapter

In this chapter, a preliminary contribution on multivariable PI controller tuning using the MOOD procedure is presented. It follows the previous development reported in [148] for controller tuning applications. The aims of this chapter are:

1. To evaluate current tools for Evolutionary Multiobjective Optimization and to find its limitations, concerning desirable Features 1-10 for PI controllers in MIMO processes.

2. To evaluate current visualization tools for m-dimensional Pareto fronts and design alternatives analysis, in order to find their limitations.

The remainder of this chapter is as follows: in section 4.2 a MOOD procedure for multivariable PI controller tuning is explained. In section 4.3 this procedure is evaluated on a multivariable benchmark process. Finally, some concluding remarks are given.

3.2 Optimization statement

MIMO systems are common in industry. Their complexity is due to their coupling effects between inputs and outputs. Consider a NxN multivariable process modeled by the following transfer matrix:

$$P(s) = \begin{bmatrix} P_{11}(s) & \dots & P_{1N}(s) \\ \vdots & \ddots & \vdots \\ P_{N1}(s) & \dots & P_{NN}(s) \end{bmatrix} \tag{3.1}$$

The selected controller design concept must fulfill a set of requirements, in accordance with the given process. Common choices for controlling MIMO systems are: decoupled PI-PID controllers [87]; centralized PI-PID controllers [88]; state space feedback techniques [68, 153]; or predictive control [67, 172, 105]. The selection of one technique over another depends on the desired balance between complexity and tradeoff between design specifications.

In this chapter, a set of decoupled PI controllers is proposed to tackle the control problem in a MIMO system. PI controllers are simple but successful solutions, and they can be improved with complementary techniques (see [8]). Equation (3.2) shows the structure of the chosen PI controller:

$$C(s) = k_p \left(1 + \frac{1}{T_i s}\right) E(s) \tag{3.2}$$

where k_c is the proportional gain, T_i the integral time (secs), and $E(s)$ the error signal. The decoupled PI controller $C(s)$ design has N SISO controllers:

$$C(s) = \begin{bmatrix} C_1(s) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_N(s) \end{bmatrix} \tag{3.3}$$

Therefore, the decision space is defined as:

$$\boldsymbol{\theta} = [k_{p_1}, T_{i_1}, \dots, k_{p_N}, T_{i_N}] \in \mathbb{R}^{2N} \tag{3.4}$$

The non-convex optimization developed by [6] will be used as guideline for the SISO PI controllers. This optimization procedure is numerical and model oriented and does not require any time domain function computations (simulations). It defines a given value of the maximum sensitivity function as a design constraint $J_{M_s}(\boldsymbol{\theta}_{PI^1})$ (Equation (2.1)) and/or the maximum complementary sensitivity function $J_{M_p}(\boldsymbol{\theta}_{PI^1})$ (Equation (2.3)). A numerical nonconvex optimization is then solved, by increasing as much as possible the integral gain $J_{k_i}(\boldsymbol{\theta}_{PI^1}) = -\frac{k_p}{T_i}$ subject to the values of M_s and M_p, in order to obtain a desired tradeoff between load rejection and robustness.

The previous tuning procedure can be adapted for MOEAs by defining as engineering control objectives k_i, M_s and M_p. Such objectives give the DM some insight regarding the tradeoff for robustness, load rejection, and set point response as in [6]. To apply this tuning procedure in a multivariable process, an index to measure the overall MIMO system stability is required. Here, the closed loop log modulus (L_{cm}) will be used as a robustness indicator. This index leads to the well-known Biggest (*sic*) Log Modulus Tuning (BLT) criterion for diagonal PID controllers in MIMO processes [109]. The criterion is defined as:

$$L_{cm} = 20\log\left|\frac{W(s)}{1+W(s)}\right| \leq L_{cm}^{max} \tag{3.5}$$

where $W(s) = -1 + \det\left(I + P(s)C(s)\right)$. This criterion proposes a *de-tuning* of the proportional gains of each controller, in order to fulfill a maximum value of the closed loop log modulus L_{cm}^{max}. Therefore, the MOP at hand is to find a tradeoff solution $\boldsymbol{\theta}$, that is:

$$\min \boldsymbol{J}(\boldsymbol{\theta}) = [-k_{i_1}, M_{s_1}, M_{p_1}, \ldots, -k_{i_N}, M_{s_N}, M_{p_N}, L_{cm}] \in \mathbb{R}^{3N+1} \tag{3.6}$$

The objective vector as defined by Equation (3.6) does not guarantee to give the DM a useful Pareto front with a good degree of flexibility to select a reliable and practical solution. It is well-known that certain practical limits to M_s, M_p and L_{cm} values are needed to guarantee a minimum of stability margin. Therefore, the MOP statement must consider the following practical limits:

$$\begin{aligned}
k_{p_1} + \nu_1 \cdot k_{p_1}/T_{i_1} &\leq K_{u_1} \\
&\vdots \\
k_{p_N} + \nu_N \cdot k_{p_N}/T_{i_N} &\leq K_{u_N} \\
1.2 \leq M_{s_{1,\ldots,N}} &\leq 2.0 \\
1 \leq M_{p_{1,\ldots,N}} &\leq 1.5 \\
0 \leq L_{cm} &\leq 2N
\end{aligned} \tag{3.7}$$

Where ν is the maximum value between the time delay process and 1. Constraint $k_p + \nu \cdot k_c/T_i \leq K_u$ is used to bound the maximum allowed control action effort to the ultimate gain K_u. Constraints $1.2 \leq M_s$ and $1 \leq M_p$ are used to avoid controllers with a sluggish performance, while constraints $M_s \leq 2.0$ and $M_p \leq 1.5$ guarantee a minimum of stability margin [6]. The empirical rule of keeping $L_{cm} \leq 2N$ [109] is adopted.

As constraints are considered in the MOP, a constraint handling mechanism is used. According to the practical and empirical limits defined for $\boldsymbol{J}_P^*$ by Equation (3.7), any unfeasible solution is punished. In [42], a penalty function without penalty parameter is proposed. Such penalty function enforces the following criteria:

1. Any feasible solution is preferred to any infeasible solution.
2. Between two feasible solutions, the solution with the better objective function value is preferred.
3. Between two infeasible solutions, the solution with the smaller constraint violation is preferred.

Following these ideas, the objective vector takes the form:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^{2N}} \mathfrak{J}(\boldsymbol{\theta}) = \begin{cases} \boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^{3N+1} & \boldsymbol{if} \sum\limits_{k=1}^{7} \phi_k(\boldsymbol{\theta}) = 0 \\ \boldsymbol{offset} + \left(\sum\limits_{k=1}^{7} \phi_k(\boldsymbol{\theta})\right) \cdot \boldsymbol{R} \in \mathbb{R}^{3N+1} & \boldsymbol{otherwise} \end{cases} \tag{3.8}$$

where:

$$\begin{aligned} \boldsymbol{offset} &= \max\left(\boldsymbol{J}^{max}\right) \cdot \boldsymbol{R} \\ \phi_1(\boldsymbol{\theta}) &= \max\{0, k_{p_1} + \frac{\nu_1 k_{p_1}}{T_{i_1}} - K_{u_1}, \ldots, k_{p_N} + \frac{\nu_N k_{p_N}}{T_{i_N}} - K_{u_N}\} \\ \phi_2(\boldsymbol{\theta}) &= \max\{0, 1.2 - M_{s_1} \ \ldots, 1.2 - M_{s_N}\} \\ \phi_3(\boldsymbol{\theta}) &= \max\{0, 1.0 - M_{p_1} \ \ldots, 1.0 - M_{p_N}\} \\ \phi_4(\boldsymbol{\theta}) &= \max\{0, M_{s_1} - 2.0, \ldots, M_{s_N} - 2.0\} \\ \phi_5(\boldsymbol{\theta}) &= \max\{0, M_{p_1} - 1.5, \ldots, M_{p_N} - 1.5\} \\ \phi_6(\boldsymbol{\theta}) &= \max\{0, L_{cm} - 2N\} \\ \phi_7(\boldsymbol{\theta}) &= \max\{0, -L_{cm}\} \\ \boldsymbol{J}^{max} &= \boldsymbol{J}^{nadir} = \left[\max_{J(\boldsymbol{\theta})\in J_P^*} J_1(\boldsymbol{\theta}), \ldots, \max_{J(\boldsymbol{\theta})\in J_P^*} J_m(\boldsymbol{\theta})\right] \end{aligned} \tag{3.9}$$

and $\boldsymbol{R}$ is a vector with the following structure: $\boldsymbol{R} = [\overbrace{1, \ldots, 1}^{m}]$.

3.3 Benchmark setup: the Wood & Berry distillation column process

To show the applicability of the MOOD procedure for multivariable PI tuning, the well-known distillation column model defined by Wood and Berry will be used [16, 195]. It represents the control of the composition of two products (a mixture of methanol and water for this specific case), where the manipulated variables are the reflux and steam flows:

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \tag{3.10}$$

As mentioned earlier, any kind of parametric controller can be tuned with the MOOD procedure, but for comparison purposes two PI controllers will be used:

$$C(s) = \begin{bmatrix} k_{p1}\left(1+\frac{1}{T_{i1}s}\right) & 0 \\ 0 & k_{p2}\left(1+\frac{1}{T_{i2}s}\right) \end{bmatrix} \tag{3.11}$$

3.3.1 Multiobjective Problem Definition

Given equations (3.10) and (3.11), the MOP at hand is to find a tradeoff solution $\boldsymbol{\theta} = [k_{p_1}, T_{i_1}\ k_{p_2}, T_{p_2}]$ for the design objectives:

$$\boldsymbol{J}(\boldsymbol{\theta}) = [-k_{p_1}/T_{i_1}, M_{s_1}, M_{p_1}, -k_{p_2}/T_{i_2}, M_{s_2}, M_{p_2}, L_{cm}] \tag{3.12}$$

subject to:

$$\begin{aligned} k_{p_1} + k_{p_1}/T_{i_1} &\leq K_{u_1} \approx 2.0 \\ |k_{p_2} + 3k_{p_2}/T_{i_2}| &\leq |K_{u_2}| \approx |-0.42| \\ 1.2 \leq M_{s_{1,2}} &\leq 2.0 \\ 1 \leq M_{p_{1,2}} &\leq 1.5 \\ 0 \leq L_{cm} &\leq 4 \end{aligned} \tag{3.13}$$

3.3.2 Evolutionary Multiobjective Optimization process

The objective vector shown in Table 4.1 is in accordance with Equation (3.8). The optimization process is performed with three different MOEAs:

- DE algorithm without archiving strategy (NA); namely, a child will be selected over his parent if *child* $\prec$ *parent* (MODE algorithm, see Algorithm 1.5). Parameter values $F = 0.5$, $Cr = 0.8$ are used (which are standard parameters in accordance with [182]) and an initial population of 50 random decision vectors.

- A DE algorithm with spherical pruning [158, 148] (sp-MODE algorithm, see Algorithm 1.6). Parameter values $F = 0.5$, $Cr = 0.8$, a population of 50 solutions, and a spherical grid resolution of 5 are used (see Table 1.1).
- The *gamultiobj* algorithm provided by MatLab© is used to calculate a Pareto front for reference. This algorithm uses a controlled elitist genetic algorithm (a variant of NSGA-II [43]). Diversity is maintained by controlling the elite members of the population as the algorithm progresses by using a crowding distance index. Default parameters are used and the BLT solution [109] is used in its initial population.

The maximum allowable function evaluations (FEs) for each method is bound to 6000, and 25 independent runs will be evaluated to analyze their performance. Each execution from the sp-MODE and the NA strategy will be compared with the Pareto front $J_P^*|_{GA}$ built with the executions of the *gamultiobj* algorithm.

To evaluate the performance of each MOEA, the I_ϵ binary indicator [206, 99] is used. The indicator indicates the factor $I_\epsilon(A, B)$ by which an approximation set A is worse than another set B with respect to all the objectives. Using a comparison method (see Table 5.1) $C_{I_\epsilon,E}(A, B) = E(I_\epsilon(A, B), I_\epsilon(B, A))$ $= \{false, true\}$ the Eps binary indicator is a compatible and complete operator [25] and this is useful to determine if two Pareto fronts are incomparable, equal, or if one is better than the other [206].

The optimization experiments were carried on an a standard PC, with a Pentium(R) processor running at 3.40 GHz and 2 GB RAM. The results after 25 independent trials with each proposal are shown in Table 3.2 (performance indicators) and Table 3.3 (non-dominated solutions attained).

As evidenced by the given results, the sp-MODE algorithm represents a viable approach for generating the Pareto front. The sp-MODE algorithm outperforms the gamultiobj algorithm, since $I_\epsilon(sp - MODE, GA) < 1$. Besides, the sp-MODE algorithm has a better improvement over $J_P^*|_{GA}$ than the NA-strategy ($I_\epsilon(sp - MODE, GA) < I_\epsilon(NA, GA)$).

3.3.3 Decision making stage

To validate the MOOD procedure as a competitive and practical solution for controller tuning, the Pareto front with 153 solutions (median value in Table 3.3) is selected and used for controller evaluation.[26] In Figure 3.1 the Pareto

[25] Given a binary relation on approximation sets $(\cdot)$, the comparison method is compatible if $C_{I,E}(A, B) \rightarrow A \cdot B \vee B \cdot A$. However, the comparison method is complete if $A \cdot B \vee A \cdot B \rightarrow C_{I,E}(A, B)$.

[26] The best solution attained could be used for this analysis, but this will not be entirely realistic, since it is not always possible to run an optimization algorithm several times.

Table 3.1: MOP statement for optimization of the multivariable PID controller case.

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^4} \mathfrak{J}(\boldsymbol{\theta}) = \begin{cases} \boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^7 & \textbf{\textit{if}} \sum\limits_{k=1}^{7} \phi_k(\boldsymbol{\theta}) = 0 \\ \textbf{\textit{offset}} + \sum\limits_{k=1}^{7} \phi_k(\boldsymbol{\theta}) \cdot \boldsymbol{R} \in \mathbb{R}^7 & \textbf{\textit{otherwise}} \end{cases}$$

$\boldsymbol{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), J_2(\boldsymbol{\theta}), J_3(\boldsymbol{\theta}), J_4(\boldsymbol{\theta}), J_5(\boldsymbol{\theta}), J_6(\boldsymbol{\theta}), J_7(\boldsymbol{\theta})]$	$\boldsymbol{\theta} = [k_{c1}, T_{i1}, k_{c2}, T_{i2}]$	$\textbf{\textit{offset}} = \max(\boldsymbol{J}^{nadir}) * \boldsymbol{R}$
Objectives	Decision variables	Constraints
$J_1(\boldsymbol{\theta}) = -k_{i1} = -k_{p1}/T_{i1}$	$K_{p1} \in [0.001, K_{u1}]$	$\phi_1(\boldsymbol{\theta}) = \max\left(0, k_{p1} + k_{p1}/T_{i1} - 2.1, \lvert k_{p2} + 3k_{p2}/T_{i2}\rvert - 0.42\right)$
$J_2(\boldsymbol{\theta}) = M_{s1} = \max\left\lvert \frac{1}{1+C_1(\jmath\omega)P_{11}(\jmath\omega)} \right\rvert$	$K_{p2} \in [K_{u2}, -0.001]$	$\phi_2(\boldsymbol{\theta}) = \max\left(0, 1.2 - M_{s1}, 1.2 - M_{s2}\right)$
$J_3(\boldsymbol{\theta}) = M_{p1} = \max\left\lvert \frac{C_1(\jmath\omega)P_{11}(\jmath\omega)}{1+C_1(\jmath\omega)P_{11}(\jmath\omega)} \right\rvert$	$T_{i1,i2} \in [0.001, 40]$	$\phi_3(\boldsymbol{\theta}) = \max\left(0, 1.0 - M_{p1}, 1.0 - M_{p2}\right)$
$J_4(\boldsymbol{\theta}) = -k_{i2} = -k_{p2}/T_{i2}$		$\phi_4(\boldsymbol{\theta}) = \max\left(0, M_{s1} - 2.0, M_{s2} - 2.0\right)$
$J_5(\boldsymbol{\theta}) = M_{s2} = \max\left\lvert \frac{1}{1+C_2(\jmath\omega)P_{22}(\jmath\omega)} \right\rvert$		$\phi_5(\boldsymbol{\theta}) = \max\left(0, M_{p1} - 1.5, M_{p2} - 1.5\right)$
$J_6(\boldsymbol{\theta}) = M_{p2} = \max\left\lvert \frac{C_2(\jmath\omega)P_{22}(\jmath\omega)}{1+C_2(\jmath\omega)P_{22}(\jmath\omega)} \right\rvert$		$\phi_6(\boldsymbol{\theta}) = \max\{0, L_{cm} - 4.0\}$
$J_7(\boldsymbol{\theta}) = L_{cm} = 20\log\left\lvert \frac{W(S)}{1+W(S)} \right\rvert, W(s) = -1\det\left(I + P(s)C(s)\right)$		$\phi_7(\boldsymbol{\theta}) = \max\{0, -L_{cm}\}$

Table 3.2: Performance achieved by MOEAs.

		Eps Indicator	
		$I_\epsilon(NA, GA)$	$I_\epsilon(sp-MODE, GA)$
Pareto	Worst	2.34E-001	1.01E-001
Set for	Best	7.74E-002	4.69E-002
Reference	Median	1.07E-001	7.55E-003
$J_P^*\vert_{GA}$	Mean	1.19E-001	7.53E-002
(676)	Std	3.32E-002	1.44E-002

set and Pareto front using the LD tool are shown respectively.

The controller selection procedure lies on the DM's preferences and desired specifications. To illustrate the tradeoff achieved by different solutions, six controllers $G_c(s)$ were selected from the Pareto front for further evaluation (see Table 3.4). The controllers with the lowest $\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_1$, $\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ and $\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_\infty$ norm are selected. An overall tradeoff between objectives is expected using these controllers.

The remaining controllers are selected according to DM's preferences. It is assumed, for example, that the DM is interested in controllers over the A-line in objective $J_7(\boldsymbol{\theta})$ (see Figure 3.1b) and decides to perform a further analysis on three controllers from such a geometric locus.

The controller resulting from the BLT tuning [109] (oriented to MIMO-stability using the Ziegler-Nichols procedure), as well as the controller proposed in [87] (WIB) that minimizes the integral of the absolute error for a specific test, are finally included.

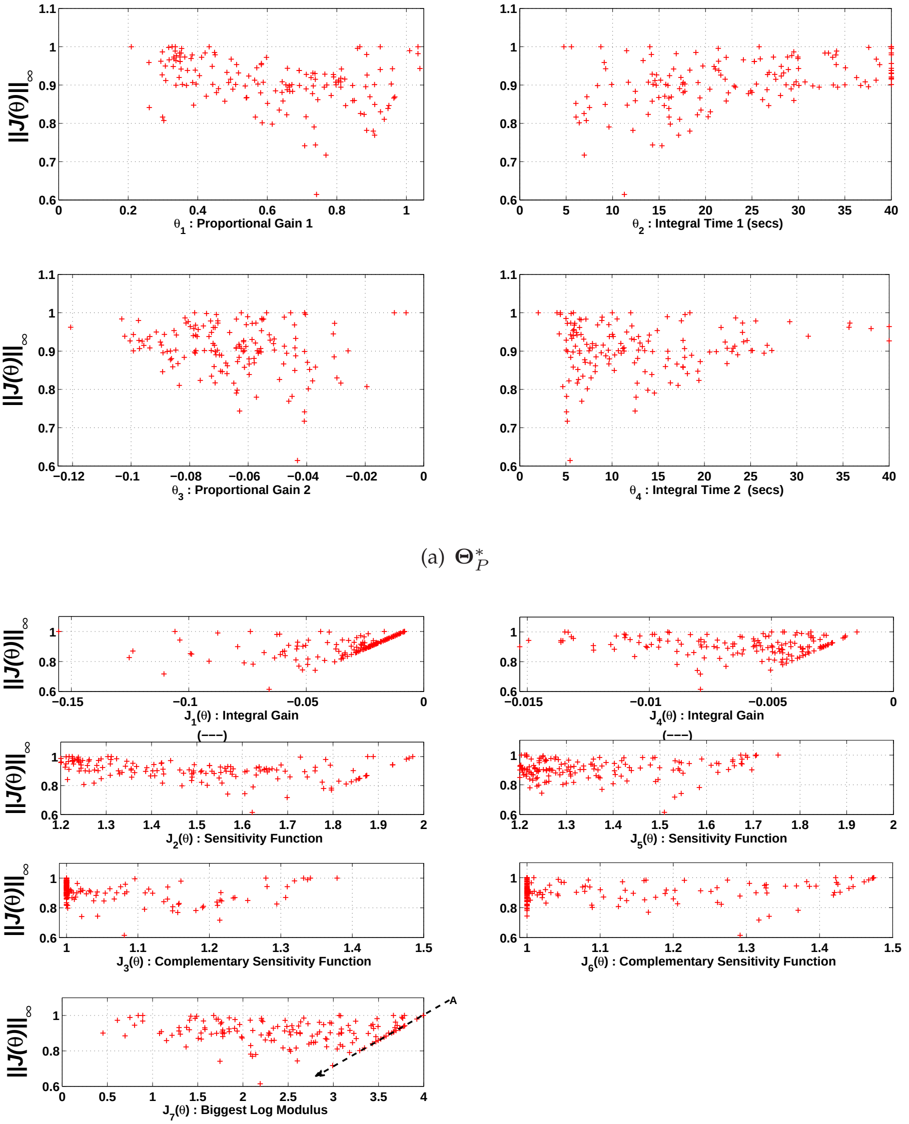
These controllers will be tested in the multivariable model in three different instances:

1. Set point change in Controlled Variable 1; consequently, the performance to reject the disturbance in Controlled Variable 2 is evaluated.

2. Set point change in Controlled Variable 2; consequently, the performance to reject the disturbance in Controlled Variable 1 is evaluated.

3. Simultaneous set point change in both controlled variables.

In all cases, the integral of the absolute error (IAE), the integral of the absolute derivative of the control action (IADU), the settling time (ST) at $\pm 2\%$, the rise time (RT) from 10% to 90%, the maximum deviation (MD), and the overshoot (OS) will be evaluated. In Tables 3.5, 3.6, 3.7 and in Figures 3.2,



(a) $\mathbf{\Theta}_P^*$

(b) $\boldsymbol{J}_P^*$

Figure 3.1: Pareto set and Pareto front approximations for experimental setup of section 3.3. $\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_\infty$ norm is used. A-line indicates controllers that match a hypothetical preference of the DM.

Table 3.3: Number of solutions achieved by MOEAs.

	NA	**sp-MODE**
Worst	0	109
Best	48	243
Median	43	153
Mean	4.03E+001	1.56E+002
Std	9.15E+000	3.09E+001

Table 3.4: Controllers selected for further evaluation.

	k_{p1}	T_{i1}	k_{p2}	T_{i2}	$J_1(\boldsymbol{\theta})$	$J_2(\boldsymbol{\theta})$	$J_3(\boldsymbol{\theta})$	$J_4(\boldsymbol{\theta})$	$J_5(\boldsymbol{\theta})$	$J_6(\boldsymbol{\theta})$	$J_7(\boldsymbol{\theta})$
BLT [109]	0.3750	8.2900	-0.0750	23.6000	-0.0452	1.2953	1.1081	-0.0032	1.2513	0.9998	3.8599
WIB [87]	0.8485	326.3462	-0.0132	1.9130	-0.0026	1.6663	1.0178	-0.0069	2.0569	1.7259	0.6024
$\min \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$	0.4245	15.6135	-0.0397	7.0977	-0.0272	1.3090	1.0014	-0.0056	1.3090	1.1047	1.5054
$\min \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_1$	0.3351	34.1079	-0.0476	8.9239	-0.0098	1.2263	1.0000	-0.0053	1.2496	1.0427	0.7488
$\min \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_\infty$	0.7415	11.2697	-0.0431	5.4571	-0.0657	1.6220	1.0809	-0.0079	1.5097	1.2914	2.1913
$J_7(\boldsymbol{\theta}) = 2.9922$	0.7687	6.9516	-0.0408	5.1598	-0.1106	1.6989	1.2144	-0.0079	1.5316	1.3170	2.9922
$J_7(\boldsymbol{\theta}) = 3.4956$	0.8458	12.4453	-0.0858	17.6735	-0.0680	1.7434	1.1414	-0.0049	1.3092	1.0000	3.4956
$J_7(\boldsymbol{\theta}) = 3.995$	0.92489	8.7357	-0.0783	5.8147	-0.1059	1.8880	1.2790	-0.0135	1.6731	1.4436	3.9950

3.3 and 3.4 the obtained results for each controller are shown. Some expected behaviors are noted:

- For controllers in the A-line (see J_7 at Figure 3.1b) the greater the L_{cm}, the greater the control action and the worse the tradeoffs. That is evident since such controllers are incapable of performing well in all the experiments. Notice how these controllers become more oscillating as $J_7(\boldsymbol{\theta})$ increases (Figures 3.2b, 3.3b, and 3.4b).

- Controller WIB obtains the best value in IAE for Experiment 3; this was expected since this controller was tuned to minimize IAE for the same instance. Notice how this outstanding performance has a lower tradeoff when single setpoint change in controlled variable 1 is applied (Figures 3.2a and 3.3a).

- Controllers with $\min \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$, $\min \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_1$ and $\min \|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_\infty$ have a balanced tradeoff between objectives and achieving a good overall performance (Figures 3.2b, 3.3b, and 3.4b).

It is important to remark that there are no controllers with poor performance but controllers with different tradeoffs between objectives. As it can be

Table 3.5: Controller performance in benchmark setup for a step change in reference Y1. In ***bold*** *appears the best value and in italics the worst value in each experiment.*

		IAE	IADU	ST	RT	MD	OS
	BLT	4.54E+002	8.71E-001	2.29E+001	3.68E+000	–	10.38%
Unit Step	WIB	*2.48E+003*	2.02E+000	*+2.00E+002*	1.39E+000	–	7.24%
Reference	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_2$	5.78E+002	8.99E-001	5.81E+001	4.02E+000	–	0.24%
Y1	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_1$	1.57E+003	**6.86E-001**	1.40E+002	*5.53E+001*	–	**0.00%**
	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_\infty$	3.34E+002	1.76E+000	3.27E+001	1.50E+000	–	13.78%
Y1	$J_7(\boldsymbol{\theta}) = 2.9922$	3.32E+002	1.97E+000	2.18E+001	1.35E+000	–	24.43%
	$J_7(\boldsymbol{\theta}) = 3.4956$	3.18E+002	2.28E+000	2.90E+001	1.27E+000	–	29.11%
	$J_7(\boldsymbol{\theta}) = 3.995$	**3.16E+002**	*2.74E+000*	**1.99E+001**	**1.11E+000**	–	*30.04%*
	BLT	*1.65E+003*	1.02E-001	*1.38E+002*	—	6.70E-001	—
Unit Step	WIB	1.01E+003	7.02E-002	6.78E+001	—	8.07E-001	—
Reference	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_2$	9.54E+002	**5.34E-002**	**5.12E+001**	—	6.48E-001	—
Y1	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_1$	9.92E+002	5.43E-002	7.00E+001	—	**5.36E-001**	—
	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_\infty$	8.20E+002	6.97E-002	5.93E+001	—	8.47E-001	—
Y2	$J_7(\boldsymbol{\theta}) = 2.9922$	8.59E+002	7.35E-002	6.35E+001	—	9.27E-001	—
	$J_7(\boldsymbol{\theta}) = 3.4956$	1.10E+003	1.63E-001	8.40E+001	—	8.94E-001	—
	$J_7(\boldsymbol{\theta}) = 3.995$	**6.44E+002**	*1.72E-001*	5.27E+001	—	*9.79E-001*	—

Table 3.6: Controller performance in benchmark setup for a step change in reference Y2. In ***bold*** *appears the best value and in italics the worst value in each experiment.*

		IAE	IADU	ST	RT	MD	OS
	BLT	3.38E+002	1.92E-001	4.58E+001	—	1.82E-001	—
Unit Step	WIB	*4.22E+003*	1.58E-001	*+2.00E+002*	—	1.49E-001	—
Reference	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_2$	5.63E+002	**1.53E-001**	7.27E+001	—	1.36E-001	—
Y2	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_1$	1.56E+003	**1.53E-001**	1.57E+002	—	1.89E-001	—
	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_\infty$	2.32E+002	1.54E-001	4.03E+001	—	9.63E-002	—
	$J_7(\boldsymbol{\theta}) = 2.9922$	**1.40E+002**	1.57E-001	2.86E+001	—	**8.68E-002**	—
Y1	$J_7(\boldsymbol{\theta}) = 3.4956$	2.25E+002	*2.62E-001*	3.85E+001	—	1.39E-001	—
	$J_7(\boldsymbol{\theta}) = 3.995$	1.47E+002	2.42E-001	**2.61E+001**	—	*1.44E-001*	—
	BLT	*3.26E+003*	1.63E-001	*1.73E+002*	*8.58E+001*	—	**0.00%**
Unit Step	WIB	1.80E+003	1.24E-001	6.85E+001	2.05E+001	—	*6.19%*
Reference	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_2$	1.85E+003	**1.04E-001**	6.78E+001	3.59E+001	—	**0.00%**
Y2	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_1$	1.94E+003	**1.04E-001**	9.48E+001	3.72E+001	—	**0.00%**
	$\min \lVert\hat{\boldsymbol{J}}(\boldsymbol{\theta})\rVert_\infty$	1.38E+003	1.10E-001	**3.84E+001**	2.34E+001	—	1.33%
	$J_7(\boldsymbol{\theta}) = 2.9922$	1.43E+003	1.14E-001	5.91E+001	2.35E+001	—	2.11%
Y2	$J_7(\boldsymbol{\theta}) = 3.4956$	2.13E+003	*1.84E-001*	1.11E+002	5.19E+001	—	0.00%
	$J_7(\boldsymbol{\theta}) = 3.995$	**9.00E+002**	1.80E-001	4.75E+001	**6.62E+000**	—	2.57%

Table 3.7: Controller performance in benchmark setup for a simultaneous step change in references Y1 and Y2. In ***bold*** *appears the best value and in italics the worst value in each experiment.*

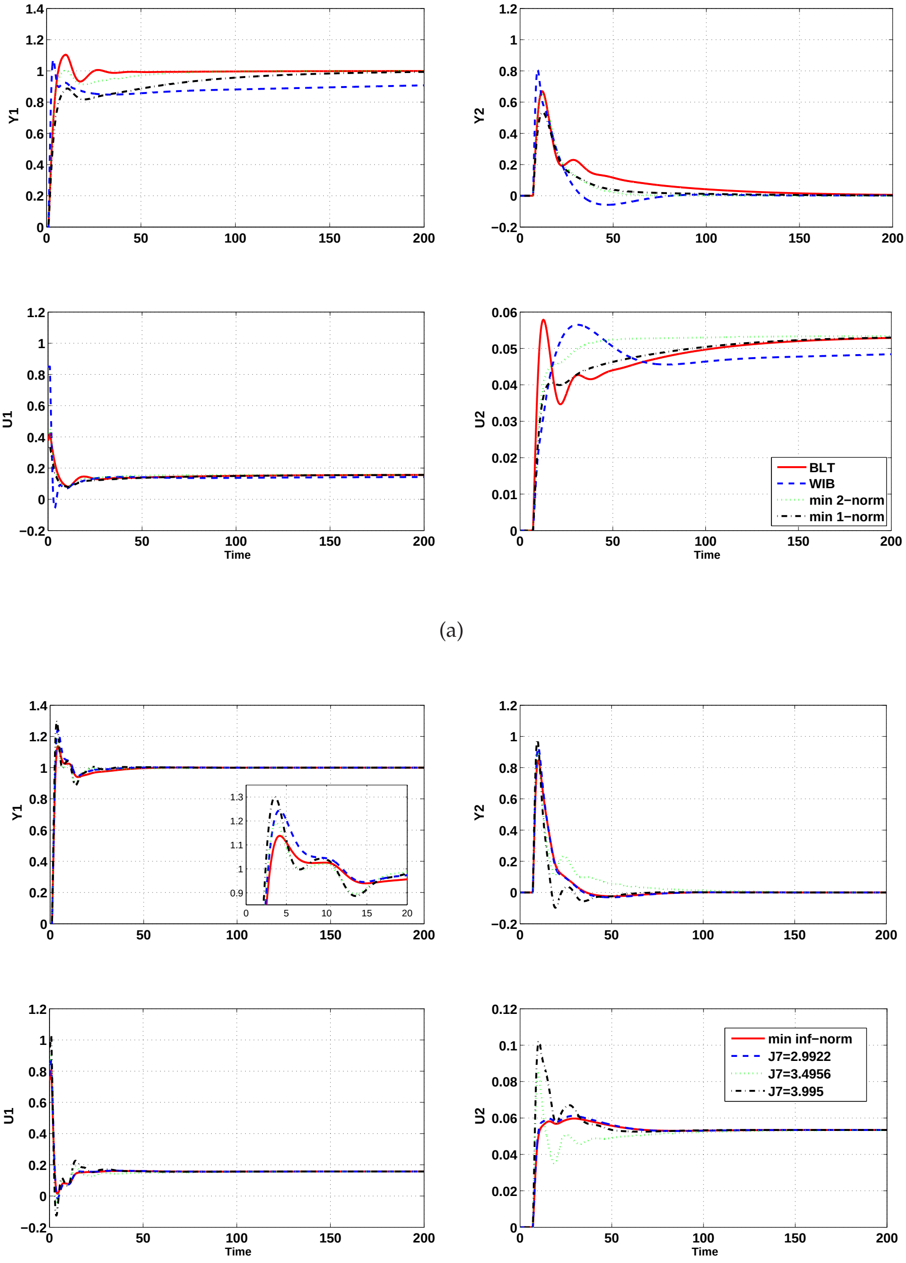
		IAE	IADU	ST	RT	MD	OS
	BLT	5.75E+002	1.01E+000	3.23E+001	2.98E+000	–	26.57%
Unit Step	WIB	**2.35E+002**	1.95E+000	**8.33E+000**	1.39E+000	–	8.04%
Reference	$\min \lVert \hat{\boldsymbol{J}}(\boldsymbol{\theta}) \rVert_2$	5.41E+002	9.57E-001	5.03E+001	3.28E+000	–	13.84%
Y1,Y2	$\min \lVert \hat{\boldsymbol{J}}(\boldsymbol{\theta}) \rVert_1$	*6.35E+002*	**7.30E-001**	*8.34E+001*	*4.68E+000*	–	**7.36%**
	$\min \lVert \hat{\boldsymbol{J}}(\boldsymbol{\theta}) \rVert_\infty$	3.74E+002	1.78E+000	2.55E+001	1.50E+000	–	19.39%
Y1	$J_7(\boldsymbol{\theta}) = 2.9922$	3.67E+002	1.99E+000	1.30E+001	1.35E+000	–	29.46%
	$J_7(\boldsymbol{\theta}) = 3.4956$	3.94E+002	2.33E+000	2.50E+001	1.27E+000	–	26.38%
	$J_7(\boldsymbol{\theta}) = 3.995$	3.86E+002	*2.74E+000*	2.53E+001	**1.11E+000**	–	*35.27%*
	BLT	*1.81E+003*	2.37E-001	2.01E+001	4.77E+000	–	29.61%
Unit Step	WIB	**7.97E+002**	**6.91E-002**	1.73E+001	**2.23E+000**	–	**6.91%**
Reference	$\min \lVert \hat{\boldsymbol{J}}(\boldsymbol{\theta}) \rVert_2$	1.10E+003	1.19E-001	1.83E+001	4.59E+000	–	11.94%
Y1,Y2	$\min \lVert \hat{\boldsymbol{J}}(\boldsymbol{\theta}) \rVert_1$	1.07E+003	1.23E-001	1.80E+001	*5.28E+000*	–	12.28%
	$\min \lVert \hat{\boldsymbol{J}}(\boldsymbol{\theta}) \rVert_\infty$	1.01E+003	1.48E-001	1.72E+001	3.61E+000	–	37.50%
Y2	$J_7(\boldsymbol{\theta}) = 2.9922$	1.04E+003	1.51E-001	1.67E+001	3.49E+000	–	43.74%
	$J_7(\boldsymbol{\theta}) = 3.4956$	1.42E+003	3.31E-001	**1.61E+001**	3.89E+000	–	55.55%
	$J_7(\boldsymbol{\theta}) = 3.995$	1.15E+003	*3.32E-001*	*2.97E+001*	3.68E+000	–	*79.98%*

seen, performances differ. This analysis could assist in scheduling strategies where more than one controller is used. As a final remark, it can be noticed that operational aspects such saturation, initial states, and operational ranges are not considered. MOEAs flexibility allows the use of time function computations to incorporate operational aspects and to redefine the optimization statement with more meaningful objectives.

3.4 Conclusions on this chapter

In this chapter, a MOOD procedure for multivariable PI controller tuning has been presented. The obtained results validate the methodology as a practical approach. Thanks to the visualization capabilities of the LD tool, it is easier perform the controller selection procedure. As the simulations reveal, the MOOD procedure is validated as a useful tool for control purposes.

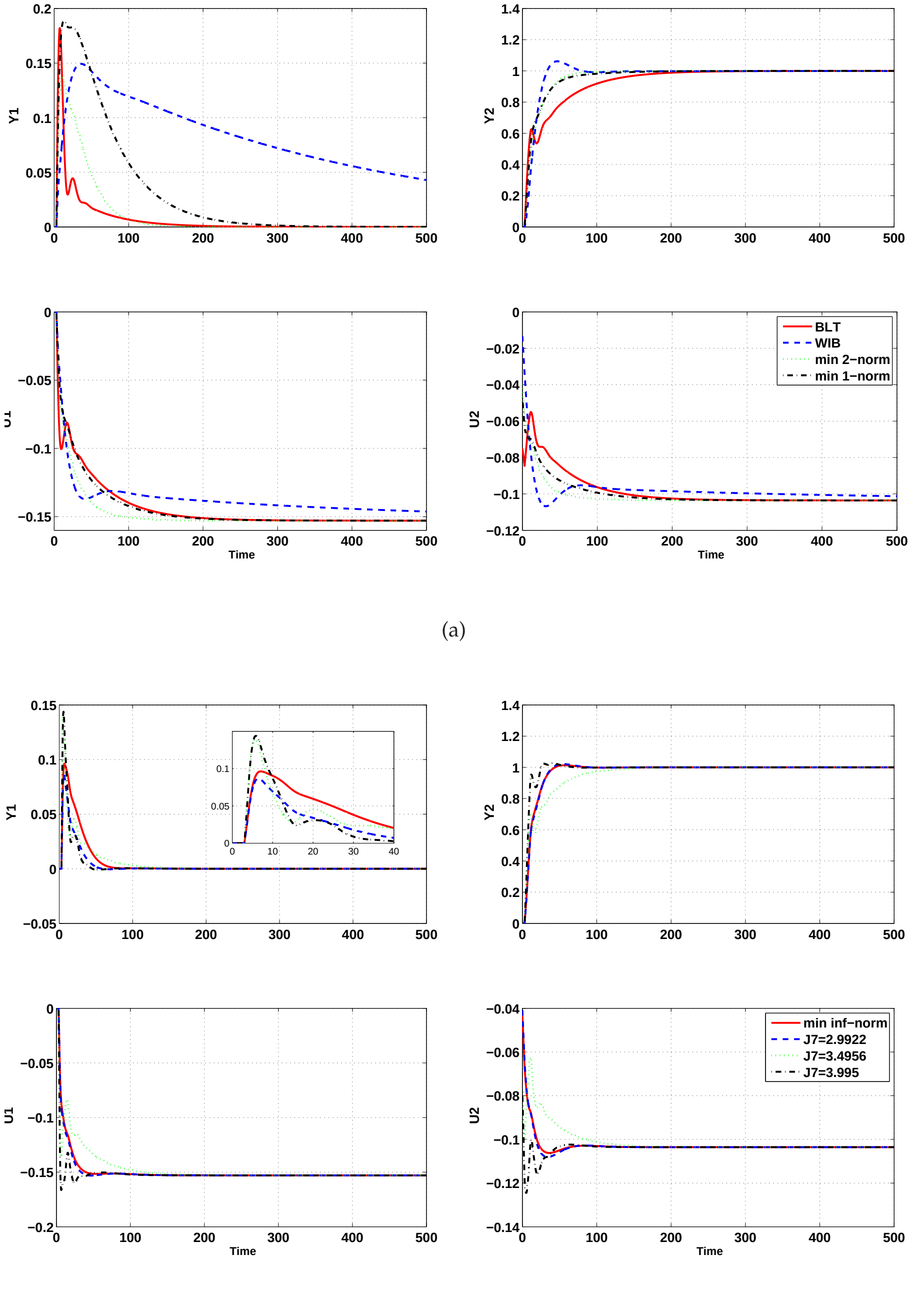
With this approach, it is possible to get the most of the optimization procedure using classical control techniques supported with well-known performance objectives. The common usage of these objectives by the control engineer community allows to have practical bounds and quick interpretations to select suitable controllers. The Pareto front allows to have a better insight about the objective tradeoff and how it changes between solutions.



(a)

(b)

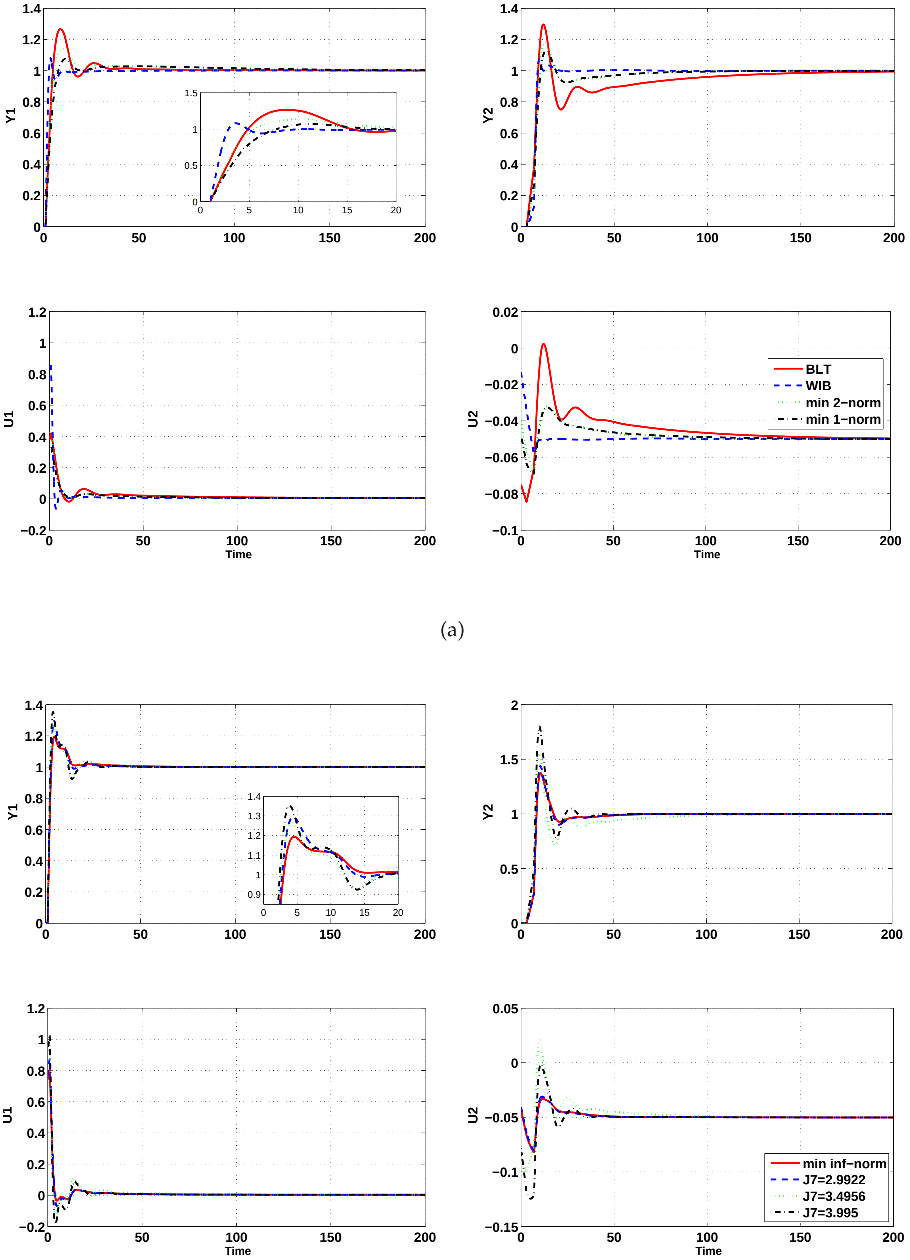
Figure 3.2: Performance in instance 1 for benchmark setup of section 3.3.



(a)

(b)

Figure 3.3: Performance in instance 2 for benchmark setup of section 3.3.



(a)

(b)

Figure 3.4: Performance in instance 3 for benchmark setup of section 3.3.

The MOP definition for the Wood and Berry distillation column will allow further comparisons of MOEAs performance. This MOP provides a useful multiobjective constrained problem for controller tuning in multivariable process, and will help to focus these algorithms into an specific class of engineering design problems.

Nevertheless, two issues are detected:

1. With the aforementioned optimization statement, it is very easy to face problems related with many-objective optimization statements (dominance resistant solutions and exploitation of sub-optimal areas).

2. Sampling for stabilizing PI controllers is solved partially. Constraints imposed are stated to assure stable controllers, nevertheless the two requirements for a controller sampling discussed in section 2.2.1 are not addressed properly.

In the next chapter, an alternative for the first issue is proposed.

CHAPTER 4:

Controller tuning by means of multiobjective optimization algorithms: a global tuning framework

In this chapter a MOOD procedure for multivariable controller tuning is presented. Following the results from the previous chapter, a mechanism to deal with many-objective optimization instances, by means of aggregation functions to fuse similar design objectives, is proposed. Contents of this chapter appear in the following paper:

- **G. Reynoso-Meza, S. García-Nieto, J. Sanchis, X. Blasco**. *Controller tuning by means of multiobjective optimization algorithms: a global tuning framework*. IEEE Transactions on Control Systems. Vol. 21, Issue 2, March 2013. Pp. 445 - 458.

4

Controller tuning by means of multiobjective optimization algorithms: a global tuning framework

> *A process cannot be understood by stopping it. Understanding must move with the flow of the process, must join it and flow with it.*
>
> Paul Atreides

4.1 Aim of this chapter

In this chapter, a MOOD procedure for multivariable controller tuning is presented. It follows the previous chapter dedicated to multivariable PI controller tuning. While the many-objective optimization instance is preserved, an aggregation mechanism is designed. This mechanism is designed to avoid a multiplicative increase in the number of objectives, without compromising the MOOD procedure *philosophy* for controller tuning. The aims of this chapter are:

1. Evaluate current tools for Evolutionary Multiobjective Optimization to find its limitations, concerning desirable features commented in Section 1.4.2.
2. Evaluate current visualization tools for m-dimensional Pareto fronts, design alternatives and design concepts comparison, in order to find its

limitations.

3. Propose a merging technique to limit problematic issues related to many-objective optimization statements.

4. Evaluate the overall approach in a physical process, in order to validate its usability.

This chapter is structured as follows: in Section 5.2 the MOOD procedure for controller tuning is presented, and in Section 5.3 and 5.4 an engineering application example is developed, experimentally evaluated and discussed. Finally, some concluding remarks are given.

4.2 Optimization statement

The use of a process model will lead to a higher degree of reliability for the controller's performance under practical considerations such as saturation, complex tracking references, and/or any kind of constraint. In this chapter, the integral of the absolute magnitude of the error (IAE) and the integral of the absolute value of the derivative control signal (IADU) are used due to their interpretability. Given a model, which will be controlled with a sampling time of T_s with $t \in [t_0, t_f]$ and with controller tuning parameters $\boldsymbol{\theta}$, the IAE and IADU are defined as:

$$IAE(\boldsymbol{\theta}) = \sum_{k=1}^{N} |r_k - y_k| \tag{4.1}$$

$$IADU(\boldsymbol{\theta}) = \sum_{k=1}^{N} |u_k - u_{k-1}| \tag{4.2}$$

Where r_k, y_k and u_k are respectively the setpoint signal, the controlled and manipulated variables at sample k; while N is the number of samples in the time interval $[t_0, t_f]$. The above mentioned objectives are defined for a SISO system. If a MIMO system with ρ inputs and ν outputs is under consideration, it is possible to have as many objectives IAE, $IADU$ as inputs and outputs. Nevertheless, this could lead to an exponential increase in the number of solutions in the Pareto front $\boldsymbol{J}_P^*$, and the analysis of the results could be more difficult. Moreover, a large subset of solutions will probably be undesirable for the DM (for example, controllers with an outstanding performance in one controlled variable at the expense of another). So, it is worthwhile trying to reduce the objective space to facilitate the analysis for the DM without losing any of the advantages of the MOOD procedure [23]. The following objectives are defined:

$$\boldsymbol{J}_E(\boldsymbol{\theta}) = \left[\frac{IAE^{1,1}(\boldsymbol{\theta})}{\Delta R^1}, \frac{IAE^{2,2}(\boldsymbol{\theta})}{\Delta R^2}, \ldots, \frac{IAE^{\nu,\nu}(\boldsymbol{\theta})}{\Delta R^\nu}\right] \tag{4.3}$$

$$\boldsymbol{J}_U(\boldsymbol{\theta}) = \left[\sum_{j=1}^{\nu} \frac{IADU^{1,j}(\boldsymbol{\theta})}{\Delta U^1_{max}}, \sum_{j=1}^{\nu} \frac{IADU^{2,j}(\boldsymbol{\theta})}{\Delta U^2_{max}}, \ldots, \sum_{j=1}^{\nu} \frac{IADU^{\rho,j}(\boldsymbol{\theta})}{\Delta U^\rho_{max}}\right] \tag{4.4}$$

Where $IAE^{i,j}(\boldsymbol{\theta})$ is the $IAE(\boldsymbol{\theta})$ for controlled variable i when there is a setpoint change ΔR^j for controlled variable j; $IADU^{i,j}(\boldsymbol{\theta})$ is the $IADU(\boldsymbol{\theta})$ for control signal i when there is a change in setpoint signal j, and ΔU^i_{max} is the maximum change allowed for control signal i. Vectors (4.3) and (4.4) contain the IAE and IADU values for each variable normalized over an operation range. Because of this, it is possible to perform a comparison between controlled variables and between control signals.

Define a sorting function $\mathcal{Z} : \mathbb{R}^{1\times n} \rightarrow \mathbb{R}^{1\times n}, \mathcal{Z}(\boldsymbol{f}) = \boldsymbol{g}$ so that: $\boldsymbol{g} = [a_1, a_2, a_3, \ldots, a_n]$, where $a_1 \geq a_2 \geq a_3 \geq \ldots a_n$, where each a_i is an element of $\boldsymbol{f}$. The global index for IAE and IADU performance measurements are defined as $J_{\mathcal{E}}(\boldsymbol{\theta})$ and $J_{\mathcal{U}}(\boldsymbol{\theta})$ respectively:

$$J_{\mathcal{E}}(\boldsymbol{\theta}) = \mathcal{Z}(\boldsymbol{J}_E(\boldsymbol{\theta})) \times \boldsymbol{w} \tag{4.5}$$

$$J_{\mathcal{U}}(\boldsymbol{\theta}) = \mathcal{Z}(\boldsymbol{J}_U(\boldsymbol{\theta})) \times \boldsymbol{w} \tag{4.6}$$

Vector $\boldsymbol{w}$ indicates it is most important to optimize the maximum value, thereby assuring a minimum worst performance for all objectives. As inputs and outputs are usually normalized in the range $[0, 1]$ an intuitive value [27] for $\boldsymbol{w}$ is $\boldsymbol{w} = [10^0, 10^{-2}, \ldots, 10^{-n}]^T$.

Please note that this objective reduction is important to facilitate the decision making step. In one hand, the multiobjective approach gives to the DM a better insight concerning the objective tradeoffs; in the other hand, too much information (too many objectives) can hinder the DM task to select a desired solution. This topic, known as many-objective optimization (usually more than 4 objectives) is not trivial, and some algorithms could face several problems due to their diversity improvement mechanisms [142, 35]. The objective reduction is an alternative to face the many-objective optimization issue [90], and with this proposal the relevant information about the conflict between control actions and performance is retained.

Additionally, a measurement for coupling effects is required:

[27] Notice that setting $\boldsymbol{w} = [1, 0, \ldots, 0]$ is equivalent to set $J_{\mathcal{E}}(\boldsymbol{\theta}) = \|\boldsymbol{J}_E(\boldsymbol{\theta})\|_\infty$. Nevertheless, any MOEA would not be able to differentiate, for example, between one solution $\boldsymbol{J}_E(\boldsymbol{\theta}^1) = [0.9, 0.9, 0.9, 0.9, 0.9]$ with $\mathcal{Z}(\boldsymbol{J}_E(\boldsymbol{\theta}^1)) \times \boldsymbol{w} = 0.9$ from another one $\boldsymbol{J}_E(\boldsymbol{\theta}^2) = [0.9, 0.5, 0.01, 0.5, 0.7]$ with $\mathcal{Z}(\boldsymbol{J}_E(\boldsymbol{\theta}^2)) \times \boldsymbol{w} = 0.9$. The latter should be preferred over the former.

$$\boldsymbol{J}_C(\boldsymbol{\theta}) = \left[\max_{i\neq 1} \frac{IAE^{1,i}(\boldsymbol{\theta})}{\Delta R^i_{max}}, \max_{i\neq 2} \frac{IAE^{2,i}(\boldsymbol{\theta})}{\Delta R^i_{max}}, \ldots, \max_{i\neq \nu} \frac{IAE^{\nu,i}(\boldsymbol{\theta})}{\Delta R^i_{max}}\right] \quad i \in [1, 2, \ldots, \nu] \tag{4.7}$$

$$J_{\mathcal{C}}(\boldsymbol{\theta}) = \mathcal{Z}(\boldsymbol{J}_C(\boldsymbol{\theta})) \times \boldsymbol{w} \tag{4.8}$$

Where ΔR^i_{max} is the maximum allowable setpoint step change for controlled variable i.

Finally, it is not possible to rely only on the process model, due to unmodeled dynamics or parametric uncertainty. Therefore, a robustness objective is required to guarantee a robust stability. One possible choice is to use complementary sensitivity function $\mathcal{T}(s)$ with a linearized process model as follows:

$$J_{\mathcal{T}} = \sup_{\omega} \bar{\sigma}\left(\mathcal{T}(\jmath\omega)W(\jmath\omega)\right), \omega \in (\overline{\omega}, \underline{\omega}) \tag{4.9}$$

Usually $\mathcal{T}(s)$ together with weighting function $W(s)$ is stated as a hard constraint ($J_{\mathcal{T}} < 1$). Since $W(s)$ selection is not a trivial task [138], the MOOD procedure can manage this task as an optimization objective (*i.e.*, it will be minimized instead of being used as a hard constraint). The MOOD procedure can deal with constraints in the same way as it deals with each objective and represents a feasible alternative to constraint-handling [29, 124]. This approach, combined with an adequate tool to analyze m-dimensional Pareto fronts, is useful to analyze the impact of relaxing, if possible, one or more constraints.

With the above mentioned objectives, it is possible to build an optimization statement to adjust any kind of parametric controller (see Equation (4.10)). That is, given a control structure with numerical parameters to adjust, the latter MOP can be stated, using information from the simulation process as performance measurement. The objectives cover the most important requirements for a controller: performance, control effort, coupling effects and robustness. Although these performance measurements have been proposed as first approximation, some other measures can be used (or added) by the DM.

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^n} \boldsymbol{J}(\boldsymbol{\theta}) = \left[\; J_{\mathcal{E}}(\boldsymbol{\theta}), J_{\mathcal{U}}(\boldsymbol{\theta}), J_{\mathcal{C}}(\boldsymbol{\theta}), J_{\mathcal{I}}(\boldsymbol{\theta}), J_{\mathcal{T}}(\boldsymbol{\theta}) \;\right] \in \mathbb{R}^5 \tag{4.10}$$

Since the implementation objectives $J_{\mathcal{I}}$ are related with a particular controller, they will be considered according to each specific case. Constraint handling depends on the selected algorithm and its own mechanisms. In general, the guidelines stated in [42] can be used to incorporate them into the cost function evaluation or into the MOP statement as and additional objective [29, 124].

4.3 Experimental setup: The Twin Rotor MIMO System

To show the applicability of the method, two different approaches of controller tuning for a non-linear twin rotor MIMO system (TRMS) are presented.

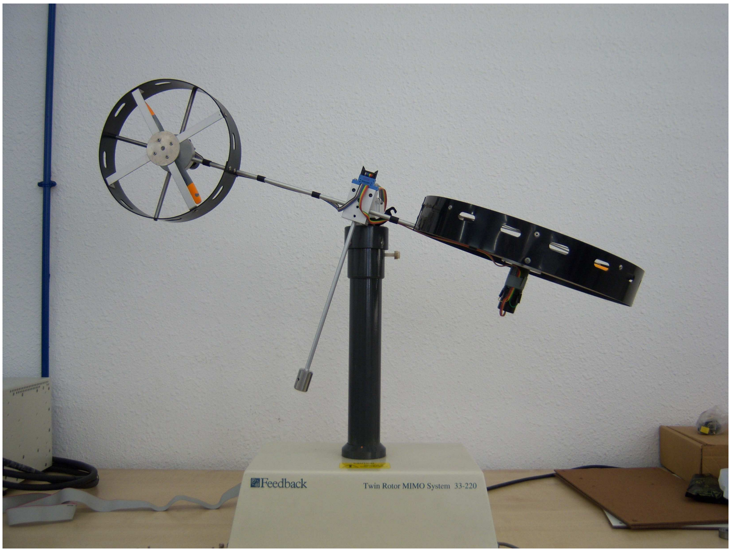


Figure 4.1: Twin Rotor MIMO System (TRMS) setup.

The TRMS is an academic workbench and a useful platform to evaluate control strategies [193, 94, 131] due to its complexity, non-linearities, and inaccessibility of states. It is a TITO (two inputs, two outputs) system, where two DC motors have control over the vertical angle (main angle) and horizontal angle (tail angle) respectively. Both inputs are limited in the normalized range $[-1, 1]$, the main angle being in the range $[-0.5, 0.5]$ rad, and the tail angle in the range $[-3.0, 3.0]$ rad.

The MOOD procedure is validated in two steps:

1. An optimization stage using an identified process model to obtain $\boldsymbol{\Theta}^*_P, \boldsymbol{J}^*_P$.

2. An experimental validation of the approximations $\boldsymbol{\Theta}^*_P, \boldsymbol{J}^*_P$ on the real TRMS.

4.3.1 Optimization stage

A non-linear state space model was identified as a part of the controller tuning-design procedure. Details on the system modeling and the observer design can be consulted in [64].

To evaluate the performance of a given controller, a Simulink© model with the identified non-linear model was used. Two simulations were carried out with different patterns:

- Simulation pattern 1: Setpoint step change for the main angle from 0 rad to 0.4 rad while tail setpoint is maintained at 0.
- Simulation pattern 2: Setpoint step change for the tail angle from 0 rad to 2.4 rad while main setpoint is maintained at 0.

The objectives defined in Equations (4.5), (4.6), (4.8) and (4.9) are used according to a TITO system:

$$J_{\mathcal{E}}(\boldsymbol{\theta}) = T_s \begin{bmatrix} \max\left(\frac{IAE^{1,1}(\boldsymbol{\theta})}{\Delta R^1}, \frac{IAE^{2,2}(\boldsymbol{\theta})}{\Delta R^2}\right) \\ \min\left(\frac{IAE^{1,1}(\boldsymbol{\theta})}{\Delta R^1}, \frac{IAE^{2,2}(\boldsymbol{\theta})}{\Delta R^2}\right) \end{bmatrix}^T \times \boldsymbol{w} \tag{4.11}$$

$$J_{\mathcal{U}}(\boldsymbol{\theta}) = \begin{bmatrix} \max\left(\sum_{j=1}^{2} \frac{IADU^{1,j}(\boldsymbol{\theta})}{\Delta U^1_{max}}, \sum_{j=1}^{2} \frac{IADU^{2,j}(\boldsymbol{\theta})}{\Delta U^2_{max}}\right) \\ \min\left(\sum_{j=1}^{2} \frac{IADU^{1,j}(\boldsymbol{\theta})}{\Delta U^1_{max}}, \sum_{j=1}^{2} \frac{IADU^{2,j}(\boldsymbol{\theta})}{\Delta U^2_{max}}\right) \end{bmatrix}^T \times \boldsymbol{w} \tag{4.12}$$

$$J_{\mathcal{C}}(\boldsymbol{\theta}) = T_s \begin{bmatrix} \max\left(\frac{IAE^{1,2}(\boldsymbol{\theta})}{\Delta R^1_{max}}, \frac{IAE^{2,1}(\boldsymbol{\theta})}{\Delta R^2_{max}}\right) \\ \min\left(\frac{IAE^{1,2}(\boldsymbol{\theta})}{\Delta R^1_{max}}, \frac{IAE^{2,1}(\boldsymbol{\theta})}{\Delta R^2_{max}}\right) \end{bmatrix}^T \times \boldsymbol{w} \tag{4.13}$$

Where $\boldsymbol{w}$ is set to $\boldsymbol{w} = [10^0, 10^{-1}]$. To evaluate $J_{\mathcal{T}}(\boldsymbol{\theta})$ a linearized model is used. As a weighting function for the robustness objective, the transfer function $W(s) = \frac{0.7s+2}{s+1.1}$ will be used.

With the MOOD procedure, any kind of controller can be tuned. In this work, two schemes are used: an ISA-PID controller [7] and a state space controller (see Figures 4.2 and 4.3). For both cases, the controller is required to work with a sampling time of 20/1000 seconds and a saturated control signal in the normalized range ± 1. In both cases, the sp-MODE algorithm (see Algorithm 1.6) has been used, with evolutionary parameters in accordance with Table 1.1 ($F = 0.5$, $Cr = 0.8$, $N_p = 50$, $\boldsymbol{\beta_\epsilon} = [5, 5, 5, 5]$ and 1-norm.)

4.3.2 PID controller tuning

PID controllers currently represent a reliable digital control solution due to their simplicity. They are often used in industrial applications and so there is ongoing research into new techniques for robust PID controller tuning [70, 190, 72, 6, 140]. For this reason, the PID scheme will be the first to be evaluated.

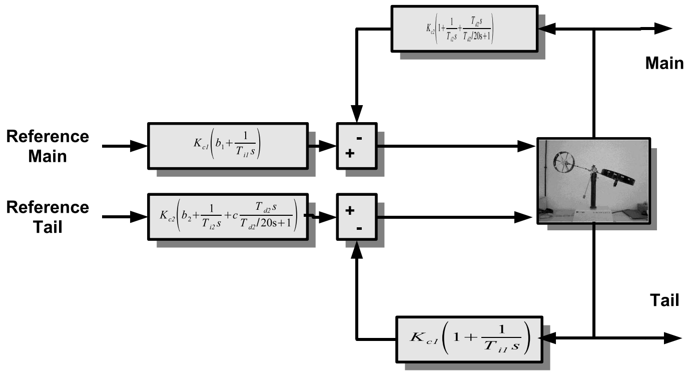

$K_{c2}\left(1+\frac{1}{T_{i2}s}+\frac{T_{d2}s}{T_{d2}/20s+1}\right)$
Main
Reference Main
$K_{c1}\left(b_1+\frac{1}{T_{i1}s}\right)$
-
+
Reference Tail
$K_{c2}\left(b_2+\frac{1}{T_{i2}s}+c\frac{T_{d2}s}{T_{d2}/20s+1}\right)$
+
-
Tail
$K_{c1}\left(1+\frac{1}{T_{i1}s}\right)$

Figure 4.2: PID controller scheme for experimental setup of Section 4.3.

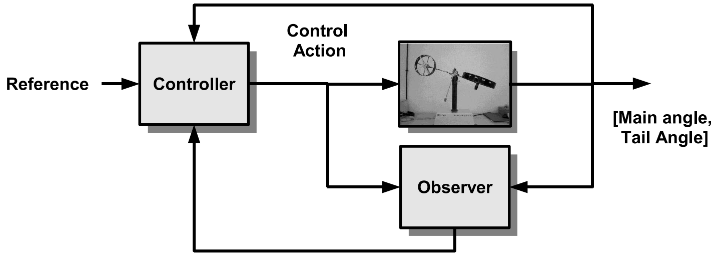


Figure 4.3: State space controller proposal for experimental setup of Section 4.3.

A two degrees of freedom ISA-PID controller with a derivative filter and an antiwindup scheme will be used:

$$\begin{aligned} U(s) &= k_p\left(b+\frac{1}{T_i s}+c\frac{T_d}{T_d/Ns+1}\right)R(s) \\ &- k_p\left(1+\frac{1}{T_i s}+\frac{T_d}{T_d/Ns+1}\right)Y(s) \end{aligned} \tag{4.14}$$

where

k_p is the proportional gain.

T_i represents the integral time (secs).

T_d is the derivative time (secs).

N represents the derivative filter. Common values for this filter lie in the range $N = [3, 20]$.

b is the setpoint weighting for the proportional action.

c is the setpoint weighting for the derivative action.

Table 4.1: MOO statement for the PID controller approach.

	$\min\limits_{\boldsymbol{\theta}\in\mathbb{R}^7} \boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^5$	
$J_{\mathcal{E}}(\boldsymbol{\theta}) =$	$T_s\left[\max\left(\frac{IAE^{Main}_{step}}{0.4}, \frac{IAE^{Tail}_{step}}{2.4}\right) + 10^{-2}\min\left(\frac{IAE^{Main}_{step}}{0.4}, \frac{IAE^{Tail}_{step}}{2.4}\right)\right]$	$K_{p_1,p_2} \in [0, 1]$
$J_{\mathcal{U}}(\boldsymbol{\theta}) =$	$\max\left(\sum \Delta u^{Main}_{step} + \sum \Delta u^{Main}_{pert}, \sum \Delta u^{Tail}_{step} + \sum \Delta u^{Tail}_{pert}\right) +$	$T_{i_1,i_2} \in (0, 100]$
	$10^{-2}\min\left(\sum \Delta u^{Main}_{step} + \sum \Delta u^{Main}_{pert}, \sum \Delta u^{Tail}_{step} + \sum \Delta u^{Tail}_{pert}\right)$	
$J_{\mathcal{C}}(\boldsymbol{\theta}) =$	$T_s\left[\max\left(\frac{IAE^{Main}_{pert}}{(2\cdot 0.5)}, \frac{IAE^{Tail}_{pert}}{(2\cdot 3)}\right) + 10^{-2}\min\left(\frac{IAE^{Main}_{pert}}{(2\cdot 0.5)}, \frac{IAE^{Tail}_{pert}}{(2\cdot 3)}\right)\right]$	$T_{d_2} \in [0, 10]$
$J_{\mathcal{I}}(\boldsymbol{\theta}) =$	$\sup\limits_{\omega} \bar{\sigma}\left(\mathcal{S}(\jmath\omega)\right), \omega \in (10^{-2}, 10^{2})$	$b_{1,2} \in [0, 1]$
$J_{\mathcal{T}}(\boldsymbol{\theta}) =$	$\sup\limits_{\omega} \bar{\sigma}\left(\mathcal{T}(\jmath\omega)W(\jmath\omega)\right), \omega \in (10^{-2}, 10^{2}), s.t. J_5 > 0.8$	

The antiwindup is performed by conditional integration when the output signal is saturated [9]. The strategy to be implemented is a PI controller for the main angle and a PID controller for the tail angle. A setpoint weighting for the derivative action of $c = 0$ and a derivative filter of $N = 20$ will also be used. Therefore, the MOOD procedure will be used to adjust the parameters K_{c1}, T_{i1}, b_1 for the PI controller and K_{c2}, T_{i2}, b_2 and T_d for the PID controller. Both will be tuned under SISO design considerations.

A total of five objectives are defined (see Table 4.1). $J_{\mathcal{E}}(\boldsymbol{\theta})$, $J_{\mathcal{U}}(\boldsymbol{\theta})$, $J_{\mathcal{C}}(\boldsymbol{\theta})$, and $J_{\mathcal{T}}(\boldsymbol{\theta})$ are defined according to Equations (4.11)) (4.12), (4.13) and (4.9) respectively. Objective $J_{\mathcal{I}}(\boldsymbol{\theta})$ is included to prefer controllers with better disturbance rejection.

The $\boldsymbol{\Theta}^*_P$ and $\boldsymbol{J}^*_P$ from the MOOD procedure for PID tuning[28] are shown in Figure 4.4. A total of 471 non-dominated controllers were found (a controllers subset Gk_{1i} is identified for further analysis). The following geometrical remarks (GR) on the level diagrams and their corresponding control remarks (CR) can be seen in Figure 4.4:

GR 1: It can be observed that two different subsets of solutions appear when solutions with $J_{\mathcal{U}}(\boldsymbol{\theta}) \leq 1$ are separated.

CR 1: The IADU performance indicator for control action is a quality indicator to differentiate damping solutions along the Pareto front.

GR 2: For solutions with $J_{\mathcal{U}}(\boldsymbol{\theta}) \leq 1$, the lower $J_{\mathcal{U}}(\boldsymbol{\theta})$, the higher $J_{\mathcal{E}}(\boldsymbol{\theta})$.

[28] A random search with the same number of function evaluations used by the MOEA was performed for comparison purposes. This approach calculates a Pareto front approximation with 161 solutions. The approximation calculated by the MOEA dominates 49 solutions of the random search approach; the random search approximation does not dominate any solution of the MOEA approximation.

Table 4.2: MOO statement for the state space controller approach.

$\min\limits_{\boldsymbol{\theta} \in \mathbb{R}^{16}} \boldsymbol{J}(\boldsymbol{\theta}) \in \mathbb{R}^5$		
$J_{\mathcal{E}}(\boldsymbol{\theta}) =$	$T_s \left[\max \left(\frac{IAE^{Main}_{step}}{0.4}, \frac{IAE^{Tail}_{step}}{2.4} \right) + 10^{-2} \min \left(\frac{IAE^{Main}_{step}}{0.4}, \frac{IAE^{Tail}_{step}}{2.4} \right) \right]$	$\theta_i \in [-10, 10]$
$J_{\mathcal{U}}(\boldsymbol{\theta}) =$	$\max \left(\sum \Delta u^{Main}_{step} + \sum \Delta u^{Main}_{pert}, \sum \Delta u^{Tail}_{step} + \sum \Delta u^{Tail}_{pert} \right) +$	$i \in (1, 2, \ldots, 16)$
	$10^{-2} \min \left(\sum \Delta u^{Main}_{step} + \sum \Delta u^{Main}_{pert}, \sum \Delta u^{Tail}_{step} + \sum \Delta u^{Tail}_{pert} \right)$	
$J_{\mathcal{C}}(\boldsymbol{\theta}) =$	$T_s \left[\max \left(\frac{IAE^{Main}_{pert}}{(2 \cdot 0.5)}, \frac{IAE^{Tail}_{pert}}{(2 \cdot 3)} \right) + 10^{-2} \min \left(\frac{IAE^{Main}_{pert}}{(2 \cdot 0.5)}, \frac{IAE^{Tail}_{pert}}{(2 \cdot 3)} \right) \right]$	
$J_{\mathcal{I}}(\boldsymbol{\theta}) =$	$trace(K * K')$	
$J_{\mathcal{T}}(\boldsymbol{\theta}) =$	$\sup\limits_{\omega} \bar{\sigma} \left(\mathcal{T}(\jmath\omega) W(\jmath\omega) \right), \omega \in (10^{-2}, 10^{2}), s.t. J_5 > 0.8$	

CR 2: For overdamped solutions, the higher the control effort (IADU), the better the performance (IAE).

GR 3: For solutions with $J_{\mathcal{U}}(\boldsymbol{\theta}) \leq 1$, the lower $J_{\mathcal{E}}(\boldsymbol{\theta})$, the higher $J_{\mathcal{I}}(\boldsymbol{\theta})$.

CR 3: For overdamped solutions, the better the performance (IAE), the worse the disturbance rejection ($J_{\mathcal{I}}(\boldsymbol{\theta})$).

GR 4: For solutions with $J_{\mathcal{U}}(\boldsymbol{\theta}) \leq 1$, the lower $J_{\mathcal{E}}(\boldsymbol{\theta})$, the higher $J_{\mathcal{T}}(\boldsymbol{\theta})$.

CR 4: For overdamped solutions, the better performance (IAE), the worse the robustness.

All of these points are well-known considerations in control theory. The Pareto front enables the visualization of this tradeoff between objectives; and the DM can choose a solution that meets his/her own needs and preferences.

4.3.3 State space feedback controller tuning

The above proposal used a PI-PID SISO strategy to address the control of a MIMO system. Such an approach is sometimes not enough to gain satisfactory control in a wide operational working zone, mainly because of the coupling dynamics. For this reason, a matrix gain for a state space (SS) control approach is selected as a second strategy (see Figure 4.3).

The MOOD procedure will be used to adjust a feedback gain matrix $K_{2\times8}$ to control the system. A total of five objectives are defined (see Table 4.2). Objectives $J_{\mathcal{E}}$, $J_{\mathcal{U}}$, $J_{\mathcal{C}}$, and $J_{\mathcal{T}}$ are again defined according to equations (4.11), (4.12), (4.13) and (4.9). Objective $J_{\mathcal{I}}$ is included to have preference over controllers with lower numerical sensitivity, *i.e.* well balanced controllers at the implementation stage.

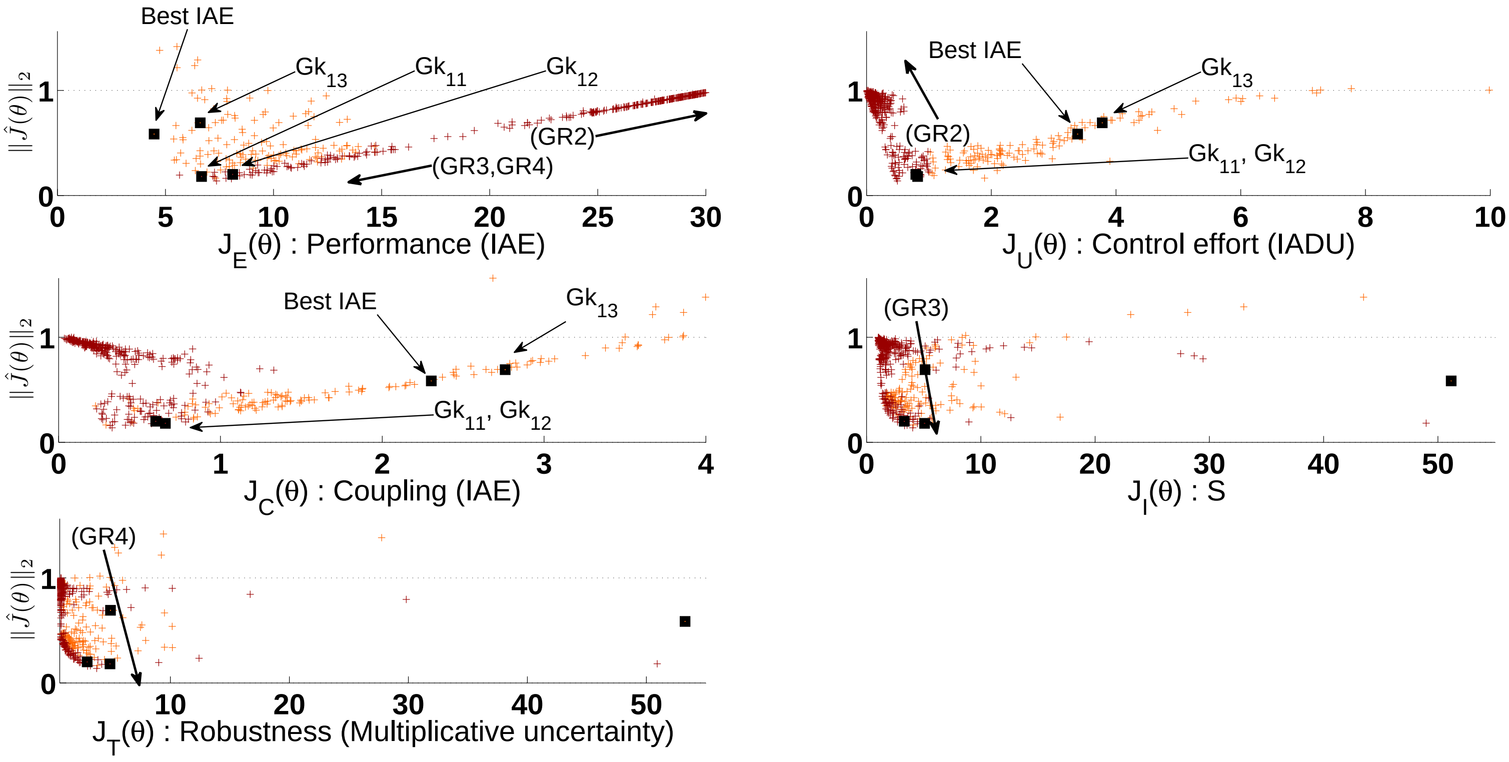


Figure 4.4: $\boldsymbol{J}_P^*$ *for PID controller. Dark solutions match the arbitrary requirement* $J_{\mathcal{U}} \leq 1$.

The Pareto front approximation $\boldsymbol{J}_P^*$[29] is shown in Figure 4.5. As a result, 589 non-dominated solutions were found (a controller subset Gk_{2i} is identified for further analysis). The following geometrical remarks (GR) and their corresponding control remarks (CR) can be seen in Figure 4.5:

GR 1: For solutions with $J_{\mathcal{U}} \leq 1$, the lower $J_{\mathcal{U}}(\boldsymbol{\theta})$, the higher $J_{\mathcal{E}}(\boldsymbol{\theta})$.

CR 1: For overdamped solutions, the higher the control effort (IADU), the better the performance (IAE).

GR 2: For objective $J_{\mathcal{I}}(\boldsymbol{\theta})$, solutions matching the requirement $J_{\mathcal{U}}(\boldsymbol{\theta}) \leq 1$ have the lower trace.

CR 2: Solutions with more balanced coefficients in the matrix gain are solutions that offer less damping responses.

4.4 Experimental validation: The Twin Rotor MIMO System

To validate both approaches, the setpoint pattern in Figure 4.6 is used on the real TRMS [30]. It is important to note that such a pattern is different from the one used at the optimization stage. In this way, it will be possible to evaluate and validate the MOOD procedure. The new pattern evaluates the performance of a given controller in maintaining zero-reference (zone A); a setpoint change in the main angle (zone B); a setpoint change in the tail position (zone C); and simultaneous changes in reference (zone D).

4.4.1 PID controller - experimental results

A subset of three controllers (see Table 4.3) are selected from the Pareto set (Figure 4.4) for further analysis on the TRMS. Controller Gk_{13} is selected due to its performance on $J_{\mathcal{E}}(\boldsymbol{\theta})$; controller Gk_{11} due to its tradeoff for objectives $J_{\mathcal{U}}(\boldsymbol{\theta})$ and $J_{\mathcal{C}}(\boldsymbol{\theta})$ (some performance is sacrificed in order to obtain a better control effort and less coupling between the main and tail closed loops). Finally, controller Gk_{12} is selected due to its robustness (this is a controller capable of working with a larger set of plants because it has a smaller $J_{\mathcal{T}}(\boldsymbol{\theta})$ value). In all cases, it is observed that the robustness requirement $J_{\mathcal{T}}(\boldsymbol{\theta}) < 1$ is not achieved. The reason for this could be: 1) it is not possible to use a

[29] A random search with the same number of function evaluations used by the MOEA was performed for comparison purposes. This approach calculates a Pareto front approximation with 86 solutions. The approximation calculated by the MOEA dominates 85 solutions of the random search approach; the random search approximation does not dominate any solution of the MOEA approximation.

[30] Controllers from Tables 4.3 and 4.6 were implemented in a National Instruments PXI-1002 System.

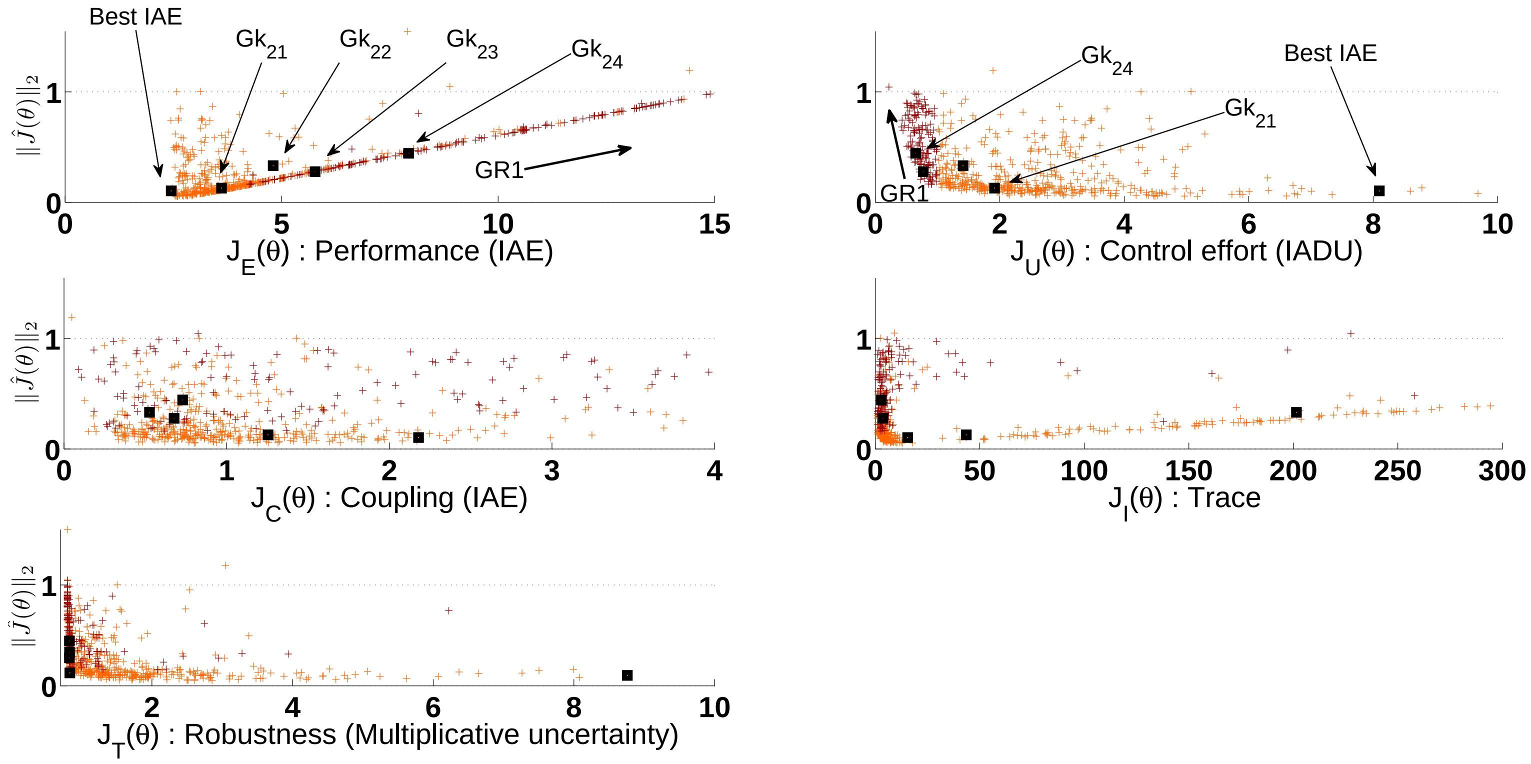


Figure 4.5: $\boldsymbol{J}_P^*$ for the SS controller. Dark solutions match the arbitrary requirement $J_{\mathcal{U}}(\boldsymbol{\theta}) \leq 1$.

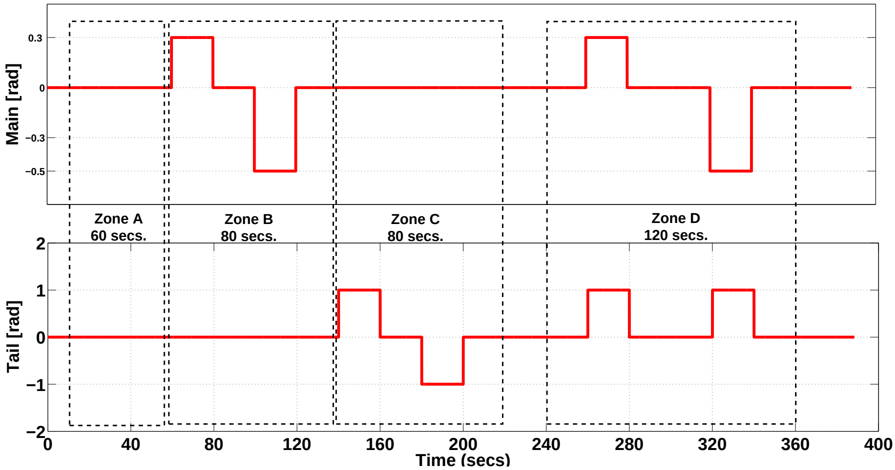


Figure 4.6: Reference trajectory for test on real TRMS. The idle state value for the main angle is around 0.3 rad.

PID scheme to control the system; or 2) the weighting function for robustness has not been chosen correctly (*i.e.* it is an excessive constraint) and the control engineer needs to evaluate if this constraint could be relaxed. After some analysis on the closed loop frequency response, it is determined that it is possible to use these controllers in a small operation range. The performances of these controllers with the reference pattern for the real test (see Figure 4.6) are shown in Tables 4.4, 4.5 and Figure 4.7.

As expected, controller Gk_{12} had the worst performance, but fewer coupling effects and the best control effort on zones C and D. Controller Gk_{13}, as indicated by the Pareto front, has the highest control effort in all cases and the best performance on zones A and D. Finally, controller Gk_{11} presents a good tradeoff between performance and control effort.

Table 4.3: PID controllers selected from $\mathbf{\Theta}_P^*$ (Figure 4.4).

	$J_{\mathcal{E}}(\boldsymbol{\theta})$	$J_{\mathcal{U}}(\boldsymbol{\theta})$	$J_{\mathcal{C}}(\boldsymbol{\theta})$	$J_{\mathcal{I}}(\boldsymbol{\theta})$	$J_{\mathcal{T}}(\boldsymbol{\theta})$	$\boldsymbol{\theta} = (K_{c1}, T_{i1}, b_1, K_{c2}, T_{i2}, T_{d2}, b_2)$
Gk_{11}	6.83	0.82	0.65	4.76	4.58	$\boldsymbol{\theta} = (0.001, 0.006, 0.99, 0.269, 8.258, 1.420, 0.626)$
Gk_{12}	8.60	0.79	0.59	2.94	2.61	$\boldsymbol{\theta} = (0.001, 0.008, 0.68, 0.2533, 8.45, 1.14, 0.84)$
Gk_{13}	6.81	3.76	2.74	4.76	4.58	$\boldsymbol{\theta} = (0.001, 0.006, 0.70, 0.999, 7.396, 1.887, 0.6721)$

Table 4.4: Performance of PI-PID controllers on the real TRMS (Zones A and B)

		Zone A		
		IAE	IADU	Obj
Gk_{11}	Main	4.76E+000	2.85E-002	J_1 =1.31E-001
	Tail	1.07E+001	4.67E+000	$J_2 = 4.67E+000$
		—	—	J_3 =—
Gk_{12}	Main	6.45E+000	3.05E-002	J_1 =2.43E-001
	Tail	3.42E+001	4.81E+000	J_2 =4.81E+000
		—	—	J_3 =—
Gk_{13}	Main	3.58E+000	2.03E-002	$J_1 = 9.89E-002$
	Tail	8.17E+000	1.65E+001	J_2 =1.65E+001
		—	—	J_3 =—
		Zone B		
		IAE	IADU	Obj
Gk_{11}	Main	3.73E+002	2.23E+000	$J_1 = 2.49E+001$
	Tail	1.14E+003	5.74E+001	$J_2 = 5.74E+001$
		—	—	J_3 =3.81E+000
Gk_{12}	Main	4.44E+002	2.11E+000	J_1 =2.96E+001
	Tail	1.27E+003	5.91E+001	J_2 =5.91E+001
		—	—	J_3 =4.24E+000
Gk_{13}	Main	3.86E+002	2.20E+000	J_1 =2.57E+001
	Tail	3.12E+002	1.80E+002	J_2 =1.80E+002
		—	—	$J_3 = 1.04E+000$

Table 4.5: Performance of the PI-PID controllers on the real TRMS (Zones C and D)

		Zone C		
		IAE	IADU	Obj
Gk_{11}	Main	5.68E+001	3.45E-001	J_1 =1.13E+001
	Tail	5.65E+002	4.26E+001	J_2 =4.26E+001
		——	——	$J_3 = 1.14E + 000$
Gk_{12}	Main	5.71E+001	2.74E-001	J_1 =1.28E+001
	Tail	6.42E+002	3.87E+001	$J_2 = 3.87E + 001$
		——	——	$J_3 = 1.14E + 000$
Gk_{13}	Main	6.36E+001	3.69E-001	$J_1 = 8.64E + 000$
	Tail	4.32E+002	1.21E+002	J_2 =1.21E+002
		——	——	J_3 =1.27E+000
		Zone D		
		IAE	IADU	Obj
Gk_{11}	Main	3.97E+002	2.36E+000	J_1 =5.48E+001
	Tail	1.41E+003	7.45E+001	J_2 =7.45E+001
		——	——	J_3 =——
Gk_{12}	Main	6.03E+002	1.97E+000	J_1 =7.76E+001
	Tail	1.87E+003	6.34E+001	$J_2 = 6.34E + 001$
		——	——	J_3 =——
Gk_{13}	Main	3.88E+002	2.19E+000	$J_1 = 3.70E + 001$
	Tail	5.57E+002	2.24E+002	J_2 =2.24E+002
		——	——	J_3 =——

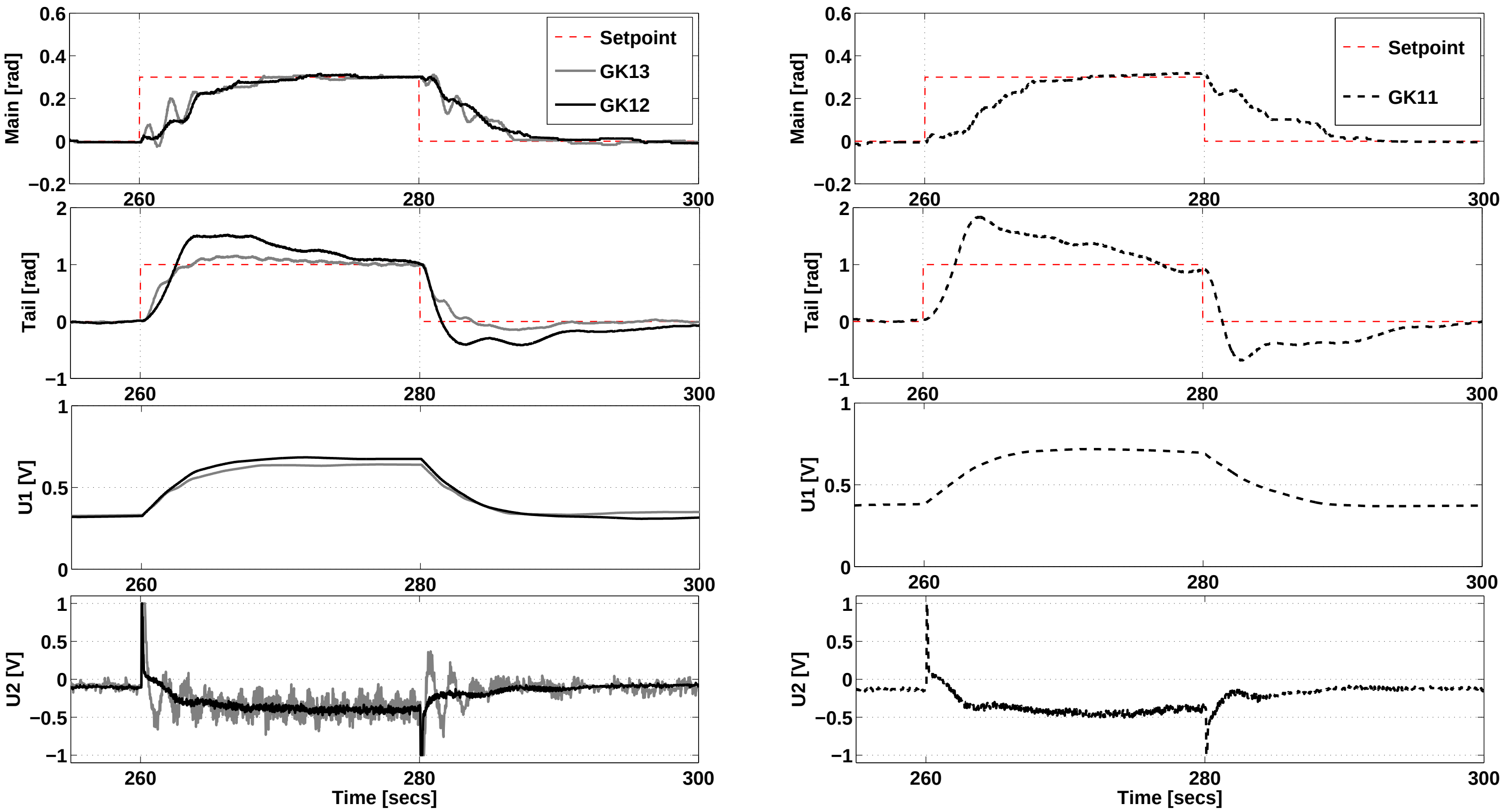


Figure 4.7: *Performance on the real TRMS of the MOOD-PID procedure for the setpoint pattern.*

4.4.2 State space approach - experimental results

A subset of six controllers (Table 4.6) was selected from the Pareto set (Figure 4.5), according to the control requirements and the closed loop frequency response on the linear model. Notice that it is possible to fulfill the requirement $J_{\mathcal{T}}(\boldsymbol{\theta}) < 1$, meaning that a larger set of plants can be controlled by the state space approach. Controller Gk_{21} is selected because it is the controller with the lowest 2-norm on the level diagram, while controller Gk_{22} is selected to analyze the impact of $J_{\mathcal{I}}(\boldsymbol{\theta})$ on performance. Controllers Gk_{23} and Gk_{24} are selected to validate the tradeoff achieved by decreasing the performance in order to gain a better control action and less coupling effects between the main and tail angles. The performance of these controllers with the reference step pattern for the real test (see Figure 4.6) is shown in Tables 4.7, 4.8 and in Figure 4.8.

Table 4.6: State space controller and their performances at the optimization stage.

	$J_{\mathcal{E}}(\boldsymbol{\theta})$	$J_{\mathcal{U}}(\boldsymbol{\theta})$	$J_{\mathcal{C}}(\boldsymbol{\theta})$	$J_{\mathcal{I}}(\boldsymbol{\theta})$	$J_{\mathcal{T}}(\boldsymbol{\theta})$
Gk_{21}	3.61	1.91	1.25	43.58	0.83
Gk_{22}	4.82	1.41	0.53	201.52	0.83
Gk_{23}	5.77	0.77	0.68	3.67	0.83
Gk_{24}	7.93	0.65	0.71	2.96	0.83

Gk_{21} and Gk_{22} are controllers with outstanding performance at the expense of high control efforts ($J_{\mathcal{U}}(\boldsymbol{\theta})$) and larger trace values ($J_{\mathcal{I}}(\boldsymbol{\theta})$). Controller Gk_{21} exhibits more coupling effects as was pointed out by $J_{\mathcal{C}}(\boldsymbol{\theta})$, and noise sensitivity ($J_{\mathcal{I}}(\boldsymbol{\theta})$). Controller Gk_{22} exhibits a better performance than Gk_{21} due to coupling effects ($J_{\mathcal{C}}(\boldsymbol{\theta})$), but also shows a higher noise control effort ($J_{\mathcal{I}}(\boldsymbol{\theta})$).

Controller Gk_{23} and Gk_{24} has almost the same performance for objectives $J_{\mathcal{U}}(\boldsymbol{\theta})$, $J_{\mathcal{C}}(\boldsymbol{\theta})$, $J_{\mathcal{I}}(\boldsymbol{\theta})$, $J_{\mathcal{T}}(\boldsymbol{\theta})$ and it is possible to see the tradeoff predicted by the Pareto front approximation. Controller Gk_{24} shows worse performance than controller Gk_{23}, but with less control effort.

4.4.3 Discussions

With the multiobjective approach and the LD tool it is possible to perform an overall comparison between both control approaches. The comparison will be not limited by using just a pair of solutions (controllers), and the whole set of controllers will be used in accordance with the quality of their performances along the Pareto front approximation.

Table 4.7: Performance of the state space controller on the real TRMS (Zones A and B).

		Zone A		
		IAE	IADU	Obj
Gk_{21}	Main	8.64E+000	3.07E+001	J_1 =2.18E-001
	Tail	1.36E+001	2.17E+001	J_2 =3.07E+001
		—	—	J_3 =——
Gk_{22}	Main	6.47E+000	7.71E+001	$J_1 = 1.88E - 001$
	Tail	1.74E+001	2.90E+001	J_2 =7.71E+001
		—	—	J_3 =—
Gk_{23}	Main	9.96E+000	7.94E+000	J_1 =2.79E-001
	Tail	2.39E+001	8.61E+000	J_2 =8.61E+000
		—	—	J_3 =—
Gk_{24}	Main	9.67E+000	6.71E+000	J_1 =2.66E-001
	Tail	2.19E+001	5.11E+000	$J_2 = 6.71E + 000$
		—	—	J_3 =—
		Zone B		
		IAE	IADU	Obj
Gk_{21}	Main	2.53E+002	1.61E+002	J_1 =1.69E+001
	Tail	1.63E+002	1.24E+002	J_2 =1.61E+002
		—	—	$J_3 = 5.42E - 001$
Gk_{22}	Main	2.11E+002	4.18E+002	$J_1 = 1.40E + 001$
	Tail	3.46E+002	1.59E+002	J_2 =4.18E+002
		—	—	J_3 =1.15E+000
Gk_{23}	Main	3.17E+002	4.85E+001	J_1 =2.11E+001
	Tail	3.28E+002	5.72E+001	J_2 =5.72E+001
		—	—	J_3 =1.09E+000
Gk_{24}	Main	5.79E+002	4.33E+001	J_1 =3.86E+001
	Tail	3.28E+002	3.56E+001	$J_2 = 4.33E + 001$
		—	—	J_3 =1.09E+000

Table 4.8: Performance of the state space controller on the real TRMS (Zones C and D).

		Zone C		
		IAE	IADU	Obj
Gk_{21}	Main	1.34E+002	1.57E+002	$J_1 = 1.01E + 001$
	Tail	5.07E+002	1.10E+002	J_2 =1.57E+002
		—	—	J_3 =2.67E+000
Gk_{22}	Main	4.86E+001	4.02E+002	J_1 =1.25E+001
	Tail	6.26E+002	1.58E+002	J_2 =4.02E+002
		—	—	$J_3 = 9.73E - 001$
Gk_{23}	Main	6.77E+001	3.70E+001	J_1 =1.04E+001
	Tail	5.20E+002	4.23E+001	J_2 =4.23E+001
		—	—	J_3 =1.35E+000
Gk_{24}	Main	1.06E+002	3.09E+001	J_1 =1.46E+001
	Tail	7.28E+002	2.52E+001	$J_2 = 3.09E + 001$
		—	—	J_3 =2.12E+000
		Zone D		
		IAE	IADU	Obj
Gk_{21}	Main	2.90E+002	2.25E+002	J_1 =3.01E+001
	Tail	5.34E+002	1.64E+002	J_2 =2.25E+002
		—	—	J_3 =—
Gk_{22}	Main	2.18E+002	6.37E+002	$J_1 = 2.96E + 001$
	Tail	7.54E+002	2.48E+002	J_2 =6.37E+002
		—	—	J_3 =—
Gk_{23}	Main	3.42E+002	4.99E+001	J_1 =3.61E+001
	Tail	6.64E+002	5.51E+001	J_2 =5.51E+001
		—	—	J_3 =—
Gk_{24}	Main	6.20E+002	5.15E+001	J_1 =6.26E+001
	Tail	1.06E+003	4.23E+001	$J_2 = 5.15E + 001$
		—	—	J_3 =—

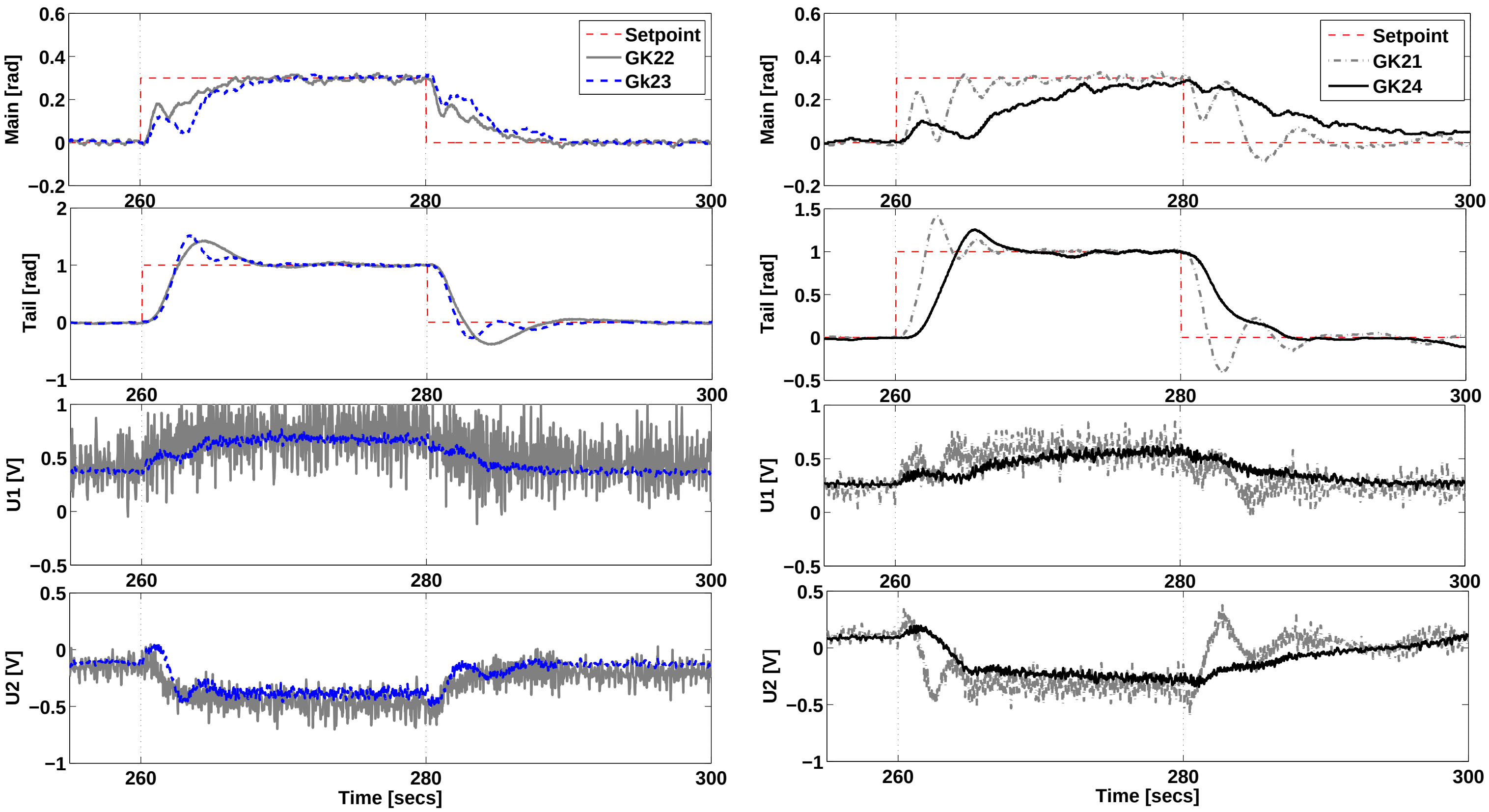


Figure 4.8: Performance on the real TRMS of the MOOD-SS procedure on setpoint pattern.

As objective $J_{\mathcal{I}}(\boldsymbol{\theta})$ corresponds to the particular implementation of each controller, a comparison can be performed in the objective subset $\boldsymbol{J_s}(\boldsymbol{\theta}) = [J_{\mathcal{E}}(\boldsymbol{\theta}), J_{\mathcal{U}}(\boldsymbol{\theta}), J_{\mathcal{C}}(\boldsymbol{\theta}), J_{\mathcal{T}}(\boldsymbol{\theta})]$. A new level diagram, using both set of solutions (with the ideal solution being the minimal offered by two approaches) is built (see Figure 4.9). Again, it is possible to make some geometrical remarks (GR) and their corresponding control remarks (CR):

GR 1: In objective $J_{\mathcal{E}}$ there is a range of solutions where both approaches coincide in the LD (Zone A).

CR 1: There are configurations for each controller capable of reaching the same level of performance in the range $IAE \approx [6, 15]$.

GR 2: For the above mentioned range, solutions of the frontal state space tend to have better values in $J_{\mathcal{C}}(\boldsymbol{\theta})$ and $J_{\mathcal{T}}(\boldsymbol{\theta})$.

CR 2: For the performance range $IAE \approx [6, 15]$ the state space controller gives a better tradeoff for control effort and robustness than a PID controller.

GR 3: Solutions below $\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ (Zone B) correspond to second front solutions. These solutions tend to disperse with larger values in objectives $J_{\mathcal{U}}(\boldsymbol{\theta})$, $J_{\mathcal{C}}(\boldsymbol{\theta})$, and $J_{\mathcal{T}}(\boldsymbol{\theta})$.

CR 3: The state space approach can reach closer values to the ideal solution. Nevertheless, these solutions may include the worst values for control effort, coupling effect, and robustness.

With such graphical analysis, it is possible to see the tradeoff gained by using a modern control strategy such as a state space controller over a PID controller. In some instances, it will be worthwhile seeing if a complex control technique is justified over a classical technique (such as a PID controller) according with the DM preferences.

4.5 Conclusions on this chapter

In this chapter, a new MOOD procedure for controller tuning has been presented. With this procedure, it is possible to achieve a higher degree of flexibility for selecting a solution that matches the desired level of tradeoff between conflicting objectives, such as performance, control effort, and robustness. The approach includes the use of meaningful performance objectives through simulation, and the use of a flexible tool to visualize m-dimensional Pareto fronts.

The MOOD procedure has been used to tune a controller for a non-linear MIMO system. The controller tuning approach has been shown to be flexible when tuning classical PID controllers and state space controllers. It has also

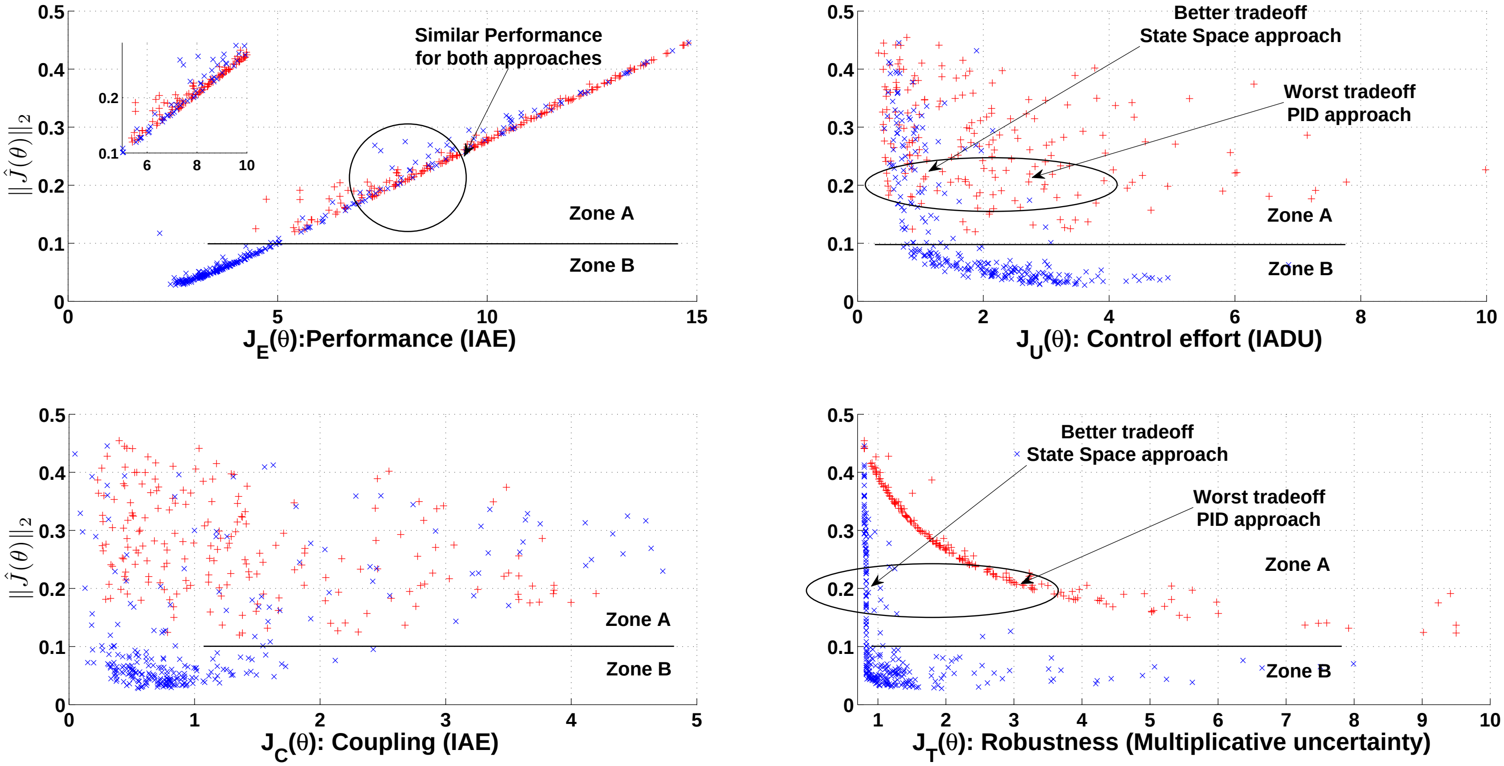


Figure 4.9: Design concept comparison between: PID controllers (+) and state space controllers (x).

been shown to be reliable and robust enough to control the system with different reference patterns. This approach makes it possible to achieve a desired tradeoff between performance and robustness, which leads to better implementation results on a real system than the results achievable by optimizing just a performance measurement. As the tendencies are those predicted by $\boldsymbol{J}_P^*$ from the optimization stage with the process model, the MOOD procedure is validated as a tool for designing different control architectures.

Finally, using the level diagram tool a global comparison has been made between different control approaches, and this is useful to determine if a complex control technique is justified in preference to a classical technique that matches the DM's preferences. Further research will focus on more interpretable objectives for robust control and stability.

Nevertheless, some issues still need to be addressed, due to some limitations on the tools employed:

- Many objectives optimization instances are quite common; even if a grouping technique is provided, the associated problems for this statement remain: persistence of non-Pareto optimal solutions, and a huge and dense Pareto front approximation. This complicates the DM's task of analyzing and selecting a solution to implement.

- Sampling for stabilizing controllers is still partially solved.

- Regarding visualization, design concepts comparison, which is valuable to select one controller over another, is still difficult with the provided tools. To the best of the author's knowledge, there is not a visualization approach specifically oriented to this purpose.

The aforementioned issues will be taken into account in order to improve the MOOD tools used. This will lead to the contributions listed in Part III of this Thesis.

Part III

Contributions on MOOD Tools

PART III:

Contributions on MOOD Tools

This part is devoted to amend the limitations found in the MOOD tools used in Part II (Preliminary Contributions on Controller Tuning). The first contribution is regarding the decision making process, where tools and guidelines for design concepts comparison in m-dimensional Pareto fronts are stated. The second contribution is related to the inclusion of preference handling mechanisms into the EMO process. This improves pertinence, convergence and provides a mechanism to deal with constrained and many-objective optimization statements. Although applications on the scope of this Thesis are related to controller tuning, such improvements can be used in other engineering fields.

CHAPTER 5:

Level Diagrams for design concepts comparison

In this chapter a new approach for design concepts comparison using Level Diagrams visualization is provided. It follows limitations found on Chapters 3 and 4 to compare different MOEAs and controllers (respectively) in m-dimensional Pareto fronts. Contents of this chapter appear in the following paper:

- **G. Reynoso-Meza, X. Blasco, J. Sanchis and J.M. Herrero**. *Comparison of design concepts in multi-criteria decision making using level diagrams*. Information Sciences, Vol. 221, Issue 1, February 2013. Pp. 124-141.

5

Level Diagrams for design concepts comparison

> *There is nothing like looking, if you want to find something. You certainly usually find something, if you look, but it is not always quite the something you were after.*
>
> Bilbo Baggins

5.1 Aim of this chapter

In this chapter, an improvement on Level Diagrams is developed for design concepts comparison. The specifics aims of this chapter are:

- To develop a theorical framework for the visual comparison of design concepts in m-dimensional objective spaces, bringing new facilities on simplicity, persistence and completeness to the designer.
- To provide the designer with a GUI-tool with the aforementioned framework in order to improve interactivity with the data analysis.

The remainder of this chapter is as follows: in Section 5.2 limitations of LD for concepts comparison are discussed. In Section 5.3 the new framework for concept comparison by means of LD is defined. In Section 5.4, the new LD framework is validated as a practical tool for concept comparison with a set of MOP. Finally, some concluding remarks of this chapter are given.

5.2 Design concepts comparison

A further analysis on objective tradeoff by the DM could include the selection and comparison of various design concepts (*i.e.*, different methods) for solving an MOP. However, the higher the dimension of the objective vector, the greater the difficulty in performing that analysis. Moreover, it will be more complex if the aim is to perform a comparison of two (or more) design concepts.

An analysis of the objective exchange when different design concepts are used will provide a better insight into the problem at hand. This new analysis will help the DM to compare different design approaches, and evaluate the circumstances where he/she would prefer one over another; furthermore, the DM can decide whether the use of a complex concept is justified over a simple concept. In this chapter, the LD capabilities are improved by defining a new quality measure to perform an analysis between Pareto fronts (design concepts) for a given MOP statement. This work has been developed based on two assumptions:

- For the DM it is important to compare the degree of improvement of one design concept over other(s). This could be justified by the fact that some of the qualitative preferences of one design concept are important to bear in mind during the final selection. If there are no preferences for the design concepts under consideration, a global Pareto front could be calculated with solutions from all design concepts; in such case, the analysis on a single Pareto front described in [19] with LD visualization would be enough.
- This visualization is complementary, *i.e.* it does not substitute the LD visualization technique shown in [19], but it gives additional information to the DM.

In Chapter 4, an initial approach to compare design concepts with LD was presented. Nevertheless, this analysis is difficult, because it is not possible to have a quantitative measurement to decide which concept performs best. As it is difficult to obtain such an insight with current norms, a new measurement is required for this purpose.

The LD visualization also enables the comparison of Pareto fronts obtained for different design concepts [119] (in this case, controller schemes). In such visualization, it will be possible to analyze the different tradeoffs achieved by different control solutions, and determine under which circumstances it is justified to use one over another. For example, in Figure 5.1, it is possible to see how a PID can achieve a better tradeoff than a PI controller between load rejection and step setpoint change (Zone Y). In the same way, it is possible to determine under which conditions their performance will be the same (Zone W).

To plot the LD, the LD visualization tool (LD-tool) [31] will be used. This is *a posteriori* visualization tool (*i.e.* it is used after the optimization process) that enables the DM to identify preferences zones along the Pareto front, as well as selecting and comparing solutions. With this tool, it is possible to remove objectives or to add new performance measurements, not used in the optimization stage. Furthermore, it is possible to integrate the DM's preferences in a lexicographic environment (as the one proposed by physical programming) to identify preferred solutions.

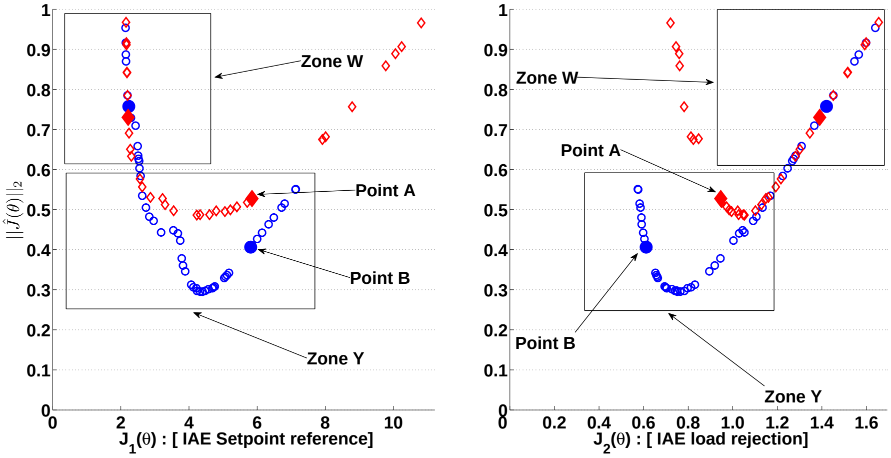


Figure 5.1: Typical LD comparison of a PI (◊) and a PID (◯) controllers in a SISO system.

5.3 Quality measure to compare design concepts

As pointed in [119], when multiple design concepts are evaluated by means of their Pareto fronts, it is necessary to have a measurement to quantify their weaknesses and strengths. Both are essential to bring the usefulness of Pareto fronts for conceptual design evaluation.

Several measurements have been developed to evaluate the Pareto front approximations. Nevertheless, many are incompatible or incomplete [206] with objective vector relations such as strict dominance, dominance or weak dominance (Definitions 2, 3, 4).

To evaluate the relative performance between design concepts, the I_ϵ binary indicator [206, 99] is used. This indicator shows the factor $I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2})$ by which an approximation set $\boldsymbol{J}^*_{p1}$ is worse than another set $\boldsymbol{J}^*_{p2}$ with respect to all the objectives. As detailed in [206], this indicator is complete and compati-

[31] Available at `http://www.mathworks.com/matlabcentral/fileexchange/24042`

ble, and is useful to determine if two Pareto fronts are incomparable, equal, or if one is better than the other (see Table 5.1).

Table 5.1: Interpretations for the I_ϵ indicator.

$I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}) < 1$	→	Every $\boldsymbol{J}^2(\boldsymbol{\theta^2}) \in \boldsymbol{J}^*_{p2}$ is strictly dominated by at least one $\boldsymbol{J}^1(\boldsymbol{\theta^1}) \in \boldsymbol{J}^*_{p1}$.
$I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}) = 1 \wedge I_\epsilon(\boldsymbol{J}^*_{p2}, \boldsymbol{J}^*_{p1}) = 1$	→	$\boldsymbol{J}^*_{p1} = \boldsymbol{J}^*_{p2}$.
$I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}) > 1 \wedge I_\epsilon(\boldsymbol{J}^*_{p2}, \boldsymbol{J}^*_{p1}) > 1$	→	Neither $\boldsymbol{J}^*_{p1}$ weakly dominates $\boldsymbol{J}^*_{p2}$ nor $\boldsymbol{J}^*_{p2}$ weakly dominates $\boldsymbol{J}^*_{p1}$.

Definition 11. *The binary ϵ-indicator $I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2})$ [206] for two Pareto front approximations $\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}$ is defined as:*

$$I_\epsilon(\boldsymbol{J}^*_{p1}, \boldsymbol{J}^*_{p2}) = \max_{\boldsymbol{J}^2(\boldsymbol{\theta}^2) \in \boldsymbol{J}^*_{p2}} \epsilon_{\boldsymbol{J}^2(\boldsymbol{\theta}^2)} \tag{5.1}$$

where

$$\epsilon_{\boldsymbol{J}^2(\boldsymbol{\theta}^2)} = \min_{\boldsymbol{J}^1(\boldsymbol{\theta}^1) \in \boldsymbol{J}^*_{p1}} \epsilon_{\boldsymbol{J}^1(\boldsymbol{\theta}^1), \boldsymbol{J}^2(\boldsymbol{\theta}^2)} \tag{5.2}$$

$$\epsilon_{\boldsymbol{J}^1(\boldsymbol{\theta}^1), \boldsymbol{J}^2(\boldsymbol{\theta}^2)} = \max_{1 \le l \le m} \frac{J^1(\boldsymbol{\theta}^1)_l}{J^2(\boldsymbol{\theta}^2)_l}, \forall \boldsymbol{J}^1(\boldsymbol{\theta}^1) \in \boldsymbol{J}^*_{p1}, \boldsymbol{J}^2(\boldsymbol{\theta}^2) \in \boldsymbol{J}^*_{p2} \tag{5.3}$$

As the binary ϵ-indicator is a scalar measure between Pareto fronts, some modifications are required to build a scalar measure for each design alternative on each concept. The quality indicator $Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j})$ [32] is defined for this purpose.

Definition 12. *The quality indicator $Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j})$ for two design concepts $i, j \in [1, \ldots, K]$ is defined as:*

$$Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j}) = \begin{cases} 1 & \text{if} \min \epsilon_{\boldsymbol{J}^i(\boldsymbol{\theta}^i), \boldsymbol{J}^*_{p_j}} > 1 \\ & \wedge \\ & \min \epsilon_{\boldsymbol{J}^j(\boldsymbol{\theta}^j), \boldsymbol{J}^*_{p_i}} > 1 \\ \min\limits_{\boldsymbol{J}^j(\boldsymbol{\theta}^j) \in \boldsymbol{J}^*_{pj}} \epsilon_{\boldsymbol{J}^i(\boldsymbol{\theta}^i), \boldsymbol{J}^*_{p_j}} & \text{otherwise} \end{cases} \tag{5.4}$$

[32] To avoid problems with this quality indicator when the objective vector has positive and negative values or zero values are involved, a normalization in the range [1,2] for each objective is used as a preliminary step.

where:

$$\epsilon_{\boldsymbol{J}^i(\boldsymbol{\theta}^i),\boldsymbol{J}^*_{p_j}} = \max_{1\leq l\leq m} \frac{J^i(\boldsymbol{\theta}^i)_l}{J^j(\boldsymbol{\theta}^j)_l}, \boldsymbol{J}^i(\boldsymbol{\theta}^i) \in \boldsymbol{J}^*_{p_i}, \forall \boldsymbol{J}^j(\boldsymbol{\theta}^j) \in \boldsymbol{J}^*_{p_j} \tag{5.5}$$

with $\boldsymbol{\theta}^i \in \boldsymbol{\Theta}^*_{p_i}$, $\boldsymbol{J}^i(\boldsymbol{\theta}^i) \in \boldsymbol{J}^*_{p_i}$, $i \neq j$.

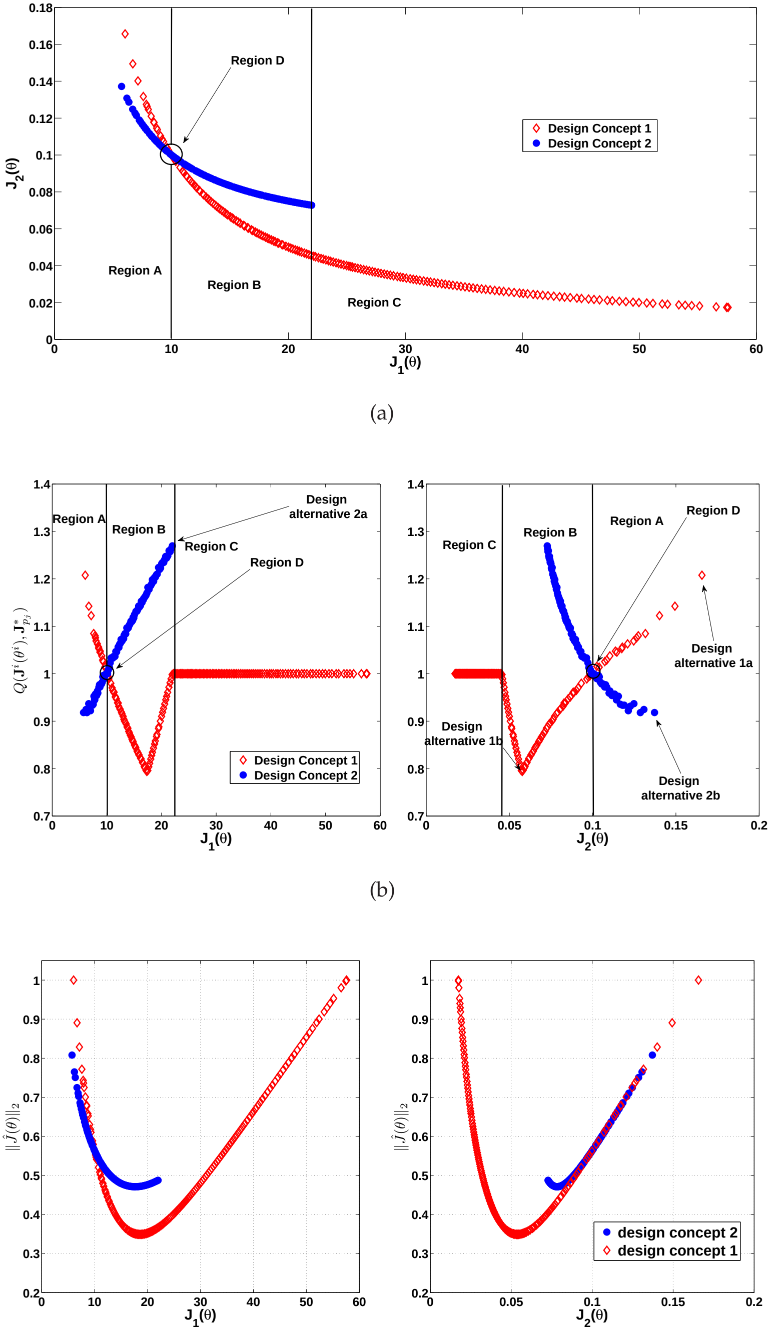
This quality measurement and the LD enable visualizing regions in the Pareto front where a design concept is better or worse than another. Furthermore, this offers a measurement of how much better one design concept performs than the other (see Figure 5.2).

Table 5.2: Comparison methods using the $Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j})$ quality measure and its meaning.

$Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j}) < 1$	→	$\boldsymbol{J}^i(\boldsymbol{\theta^i}) \in \boldsymbol{J}^*_{p_i}$ strictly dominates at least one $\boldsymbol{J}^j(\boldsymbol{\theta^j}) \in \boldsymbol{J}^*_{p_j}$.	$\boldsymbol{J}^i(\boldsymbol{\theta^1}) \in \boldsymbol{J}^*_{p_i}$ has an improvement over a solution $\boldsymbol{J}^j(\boldsymbol{\theta^j}) \in \boldsymbol{J}^*_{p_j}$ by a scale factor of $Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j})$ (at least) for all objectives.
$Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j}) = 1$	→	$\boldsymbol{J}^i(\boldsymbol{\theta^i}) \in \boldsymbol{J}^*_{p_i}$ is not comparable with any solution $\boldsymbol{J}^j(\boldsymbol{\theta^j}) \in \boldsymbol{J}^*_{p_j}$.	$\boldsymbol{J}^i(\boldsymbol{\theta^i}) \in \boldsymbol{J}^*_{p_i}$ is Pareto optimal in $\boldsymbol{J}^*_{p_j}$ or $\boldsymbol{J}^i(\boldsymbol{\theta^i}) \in \boldsymbol{J}^*_{p_i}$ is inside a region in the objective space not covered by $\boldsymbol{J}^*_{p_j}$.
$Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j}) > 1$	→	$\boldsymbol{J}^i(\boldsymbol{\theta^i}) \in \boldsymbol{J}^*_{p_i}$ is strictly dominated by at least one $\boldsymbol{J}^j(\boldsymbol{\theta^j}) \in \boldsymbol{J}^*_{p_j}$.	A solution $\boldsymbol{J}^j(\boldsymbol{\theta^j}) \in \boldsymbol{J}^*_{p_j}$ has an improvement over $\boldsymbol{J}^i(\boldsymbol{\theta^i}) \in \boldsymbol{J}^*_{p_i}$ by a scale of $Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j})$ (at least) for all objectives.

For example, in Figure 5.2 the relationships described in Table 5.2 can be seen. Firstly, due to the quality measurement, it is possible to quickly identify the s-Pareto non-optimal (any solution $Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j}) > 1$) from s-Pareto optimal solutions (any solution $Q(\boldsymbol{J}^i(\boldsymbol{\theta^i}), \boldsymbol{J}^*_{p_j}) \leq 1$). Moreover, the quality measurement enables us to assign a quantitative value about how better or worse a solution is with respect another concept. Further analysis with this quality measurement can be made for particular solutions, or for regions in the LD.

Regarding particular solutions, the following remarks can be noticed for design alternatives 1a, 2a, 1b and 2b (Figure 5.2b):



(a)

(b)

(c)

Figure 5.2: Typical comparison of two design concepts using a 2-D graph (a), level diagrams with quality measure $Q(\cdot,\cdot)$ (b) and level diagrams with 2-norm (c).

- $Q(1a, \boldsymbol{J}^*_{p2}) \approx 1.2$. That is, among the solutions $\boldsymbol{J}^2(\boldsymbol{\theta}^2) \in \boldsymbol{J}^*_{p2}$ which dominate objective vector 1a, the smaller k for a solution $\boldsymbol{J}^2(\boldsymbol{\theta}^2)$ such that $\boldsymbol{J}^{2'}(\boldsymbol{\theta}^2) = k \cdot \boldsymbol{J}^2(\boldsymbol{\theta}^2)$ is weakly dominated by 1a is $k \approx 1.2$.

- $Q(1b, \boldsymbol{J}^*_{p2}) \approx 0.8$. That is, among the solutions $\boldsymbol{J}^2(\boldsymbol{\theta}^2) \in \boldsymbol{J}^*_{p2}$ dominated by objective vector 1b, the bigger k for a solution $\boldsymbol{J}^2(\boldsymbol{\theta}^2)$ such that $\boldsymbol{J}^{2'}(\boldsymbol{\theta}^2) = k \cdot \boldsymbol{J}^2(\boldsymbol{\theta}^2)$ weakly dominates 1b is $k \approx 0.8$.

- $Q(2a, \boldsymbol{J}^*_{p1}) \approx 1.28$. That is, among the solutions $\boldsymbol{J}^1(\boldsymbol{\theta}^1) \in \boldsymbol{J}^*_{p1}$ which dominate objective vector 2a, the smaller k for a solution $\boldsymbol{J}^1(\boldsymbol{\theta}^1)$ such that $\boldsymbol{J}^{1'}(\boldsymbol{\theta}^1) = k \cdot \boldsymbol{J}^1(\boldsymbol{\theta}^1)$ is weakly dominated by 2a is $k \approx 1.28$.

- $Q(2b, \boldsymbol{J}^*_{p1}) \approx 0.9$. That is, among the solutions $\boldsymbol{J}^1(\boldsymbol{\theta}^1) \in \boldsymbol{J}^*_{p1}$ dominated by objective vector 2b, the bigger k for a solution $\boldsymbol{J}^1(\boldsymbol{\theta}^1)$ such that $\boldsymbol{J}^{1'}(\boldsymbol{\theta}^1) = k \cdot \boldsymbol{J}^1(\boldsymbol{\theta}^1)$ weakly dominates 2b is $k \approx 0.9$.

With regard to tendencies, the following remarks can be made:

- The lower $J_1(\boldsymbol{\theta})$, the better the improvement of concept 2 (◯) over concept 1 (◇).

- For $J_2(\boldsymbol{\theta}) \in [0.055, 0.1]$ approximately, the lower $J_2(\boldsymbol{\theta})$, the better the improvement of concept 1 (◇) over concept 2 (◯).

Regarding zones in Figure 5.2b, region A represents the zone where design concept 2 (◯) is better than design concept 1 (◇). Note that the design alternatives from concept 2 have a quality measurement $Q(\boldsymbol{J^2}(\boldsymbol{\theta}^2), \boldsymbol{J}^*_{p1}) < 1$ and design alternatives from concept 1 have a quality measurement $Q(\boldsymbol{J^1}(\boldsymbol{\theta}^1), \boldsymbol{J}^*_{p2}) > 1$. The opposite is true for region B. Region C is a zone reached (covered) only by concept 1 (and thus, it is impossible to compare both concepts). Finally, region D represents a region where both concepts have almost the same exchange between objectives. Please note how this analysis is more difficult with standard norms using LD (see Figure 5.2c).

Finally, note that although it is possible to build a s-Pareto front merging the design alternatives of each concept and analyze its tendencies, it is difficult to measure the improvement of one concept over another. This fact is mainly due to the loss of information after building the s-Pareto front. The LD with the quality measure enables a quantitative a-priori analysis between concepts, and it is possible to decide, for example, if the improvement of one of them is justifiable. For instance, the DM could decide that the improvement by a factor of 0.9 for concept 2 is not justifiable, and decide to only retain concept 1.

While such comparison can be performed by visual inspection in a classical 2D-objective graph (see Figure 5.2a), such a task will be more complex when three or more objectives are considered. Several examples are considered to show the LD performing comparisons on m-dimensional Pareto fronts using this quality measurement.

5.4 Examples

In this section, four examples are analyzed, where different concepts are used to obtain different tradeoffs between objectives. The following examples show analysis on 2, 3, 5 and 6 dimensions for the objective space. The first example is used to show that no information (completeness) or clarity (simplicity) will be lost using the LD instead of a 2-D graphical analysis. The latter cases are presented to illustrate the visualization capacity of LD using the quality measurement for concept comparison in m-dimensional Pareto fronts. It is assumed that any superfluous objective has been previously removed from the analysis; *i.e.* all the information (concerning the number of objectives) required by the DM is depicted.

For all examples, the LD-ToolBox is used to retain the degree of interactivity with the supplied data. Finally, the Pareto fronts are calculated using the sp-MODE algorithm [148, 158], which uses the differential evolution (DE) [183, 182, 41] algorithm, but any other algorithm or procedure could be used since tools for the MCDM step are being developed in this chapter. For the sake of simplicity, the different LDs will be referred as follows: *LD/front/measure*. For example, LD/$\boldsymbol{J}_p^*/\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$, means that a visual representation of Pareto front approximation $\boldsymbol{J}_p^*$ with 2-norm in LD is presented.

5.4.1 Bi-objective truss design problem

The truss design problem is a classical MOO benchmark statement to test algorithms, as well as decision making step procedures. The truss parameters proposed in [165] are used with the design concepts stated in [121, 119]. Two objectives are minimized: squared deflection ($J_1(\boldsymbol{\theta})$ [cm^2]) and total volume ($J_2(\boldsymbol{\theta})$ [cm^3]), with decision variables the cross sections of the bars $\theta_1, \theta_2, \theta_3$ (see Figure 5.3. That is:

$$\min \boldsymbol{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), J_2(\boldsymbol{\theta})] \tag{5.6}$$

$$J_1(\boldsymbol{\theta}) = L \cdot (\frac{\theta_1}{\sin\gamma_1} + \theta_2 + \frac{\theta_3}{\sin\gamma_2}) \tag{5.7}$$

$$J_2(\boldsymbol{\theta}) = 0.25 \cdot \gamma_3 + 0.75 \cdot \gamma_4 \tag{5.8}$$

where

$$\begin{bmatrix} \gamma_3 \\ \gamma_4 \end{bmatrix} = \frac{L}{E} \begin{bmatrix} a_1 & a_2 \\ a_2 & a_3 \end{bmatrix}^{-1} \begin{bmatrix} F \\ F \end{bmatrix} \tag{5.9}$$

$$
\begin{aligned}
a_1 &= \theta_2 + \theta_1 \sin^3 \gamma_1 + \theta_3 \sin^3 \gamma_2 && (5.10)\\
a_2 &= -\theta_1 \sin^2 \theta \cos \gamma_1 + \theta_3 \sin^2 \gamma_2 \cos \gamma_2 && (5.11)\\
a_3 &= \theta_1 \sin \theta \cos^2 \gamma_1 + \theta_3 \sin \gamma_2 \cos^2 \gamma_2 && (5.12)
\end{aligned}
$$

subject to:

$$
\begin{aligned}
\theta_i &< 2.0 && (5.13)\\
\theta_i &> 0.1 && (5.14)\\
\sigma &\geq \frac{E}{L}\left(\gamma_3 \sin \gamma_1 - \gamma_4 \sin \gamma_1\right) \sin \gamma_1 && (5.15)\\
\sigma &\geq \frac{E}{L}\gamma_3 && (5.16)\\
\sigma &\geq \frac{E}{L}\left(\gamma_3 \sin \gamma_2 - \gamma_4 \sin \gamma_2\right) \sin \gamma_2 && (5.17)
\end{aligned}
$$

The other parameters for the truss are: $L = 1\ m$; $\gamma_1 = 45^0$; $\gamma_2 = 30^0$; $F_1 = F_2 = 20\ kN$; $E = 200e9\ Pa$ (Young modulus); and maximum stress accepted in each bar $\sigma = 200E + 006\ Pa$.

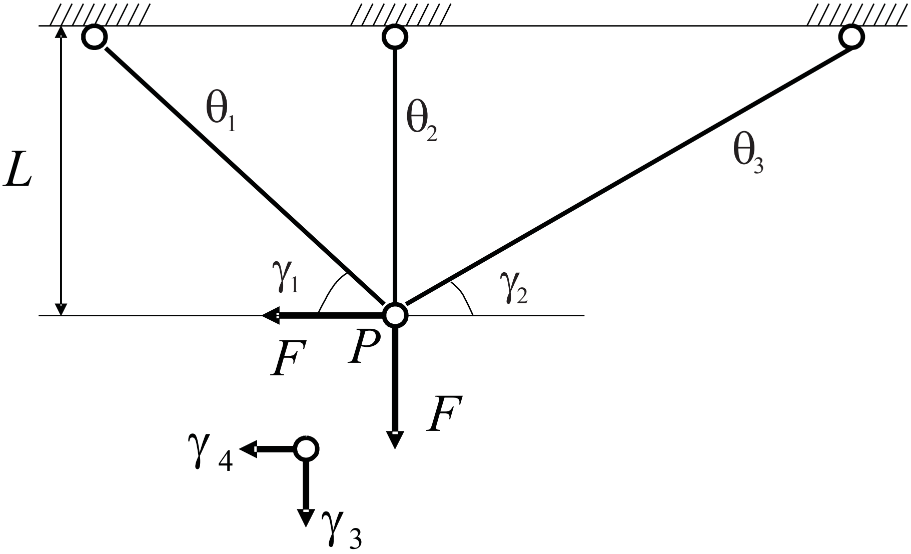


Figure 5.3: Bi-objective truss design problem

Whereas it could be the easiest (from a persistence point of view) to use a straightforward approach with the SCp visualization, this example is included and used to cover the following topics:

- Bring a first approach to the LD visualization and design concept comparison with a (probably well known) multiobjective benchmark problem.
- Introduce some properties of the LD visualization.

In Figure 5.4, the objective exchange for both concepts and its LD/ $\{J^*_{p1}, J^*_{p2}\}$ / $\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ visualization is shown. It is important to notice that the LD visualization by itself is not enough to make an analysis or a comparison between design concepts. This is mainly because there is not a quantitative measurement on the improvement of one over the other.

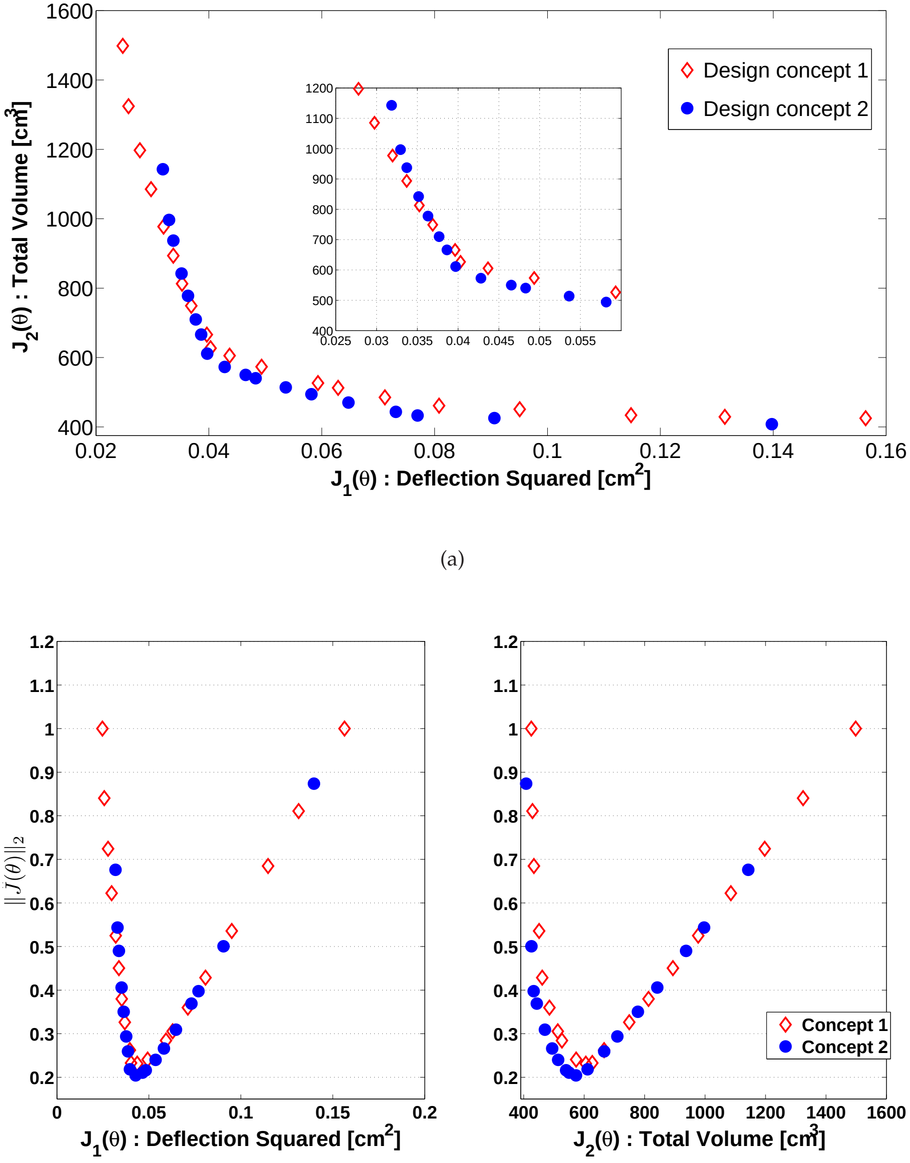
In Figure 5.5, the LD/$\{J^*_{p1}, J^*_{p2}\}$/$Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p_j}\right)$ and the LD/ s-Pareto/ $\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ visualizations are shown. In Figure 5.5a it is possible to identify some interesting remarks:

- In region B, concept 1 (♢) is better than concept 2 (◯). The opposite is true for Zone D.
- The lower $J_2(\boldsymbol{\theta})$, the better the improvement of concept 2 (◯) over concept 1 (♢).
- Region A is covered only by concept 1 (♢).
- In region C, both concepts offer the DM the same type of exchange between objectives.

With this visual inspection, the DM can decide if a 2% of improvement between concepts is justifiable to change the structural design (assuming that one of the concepts is currently being used). Furthermore, the DM knows that if the concept design requires keeping the deflection $J_2(\boldsymbol{\theta}) \in [0.03, 0.04]$, then she/he can select, for example, the easier structure to build.

5.4.2 Disc brake design

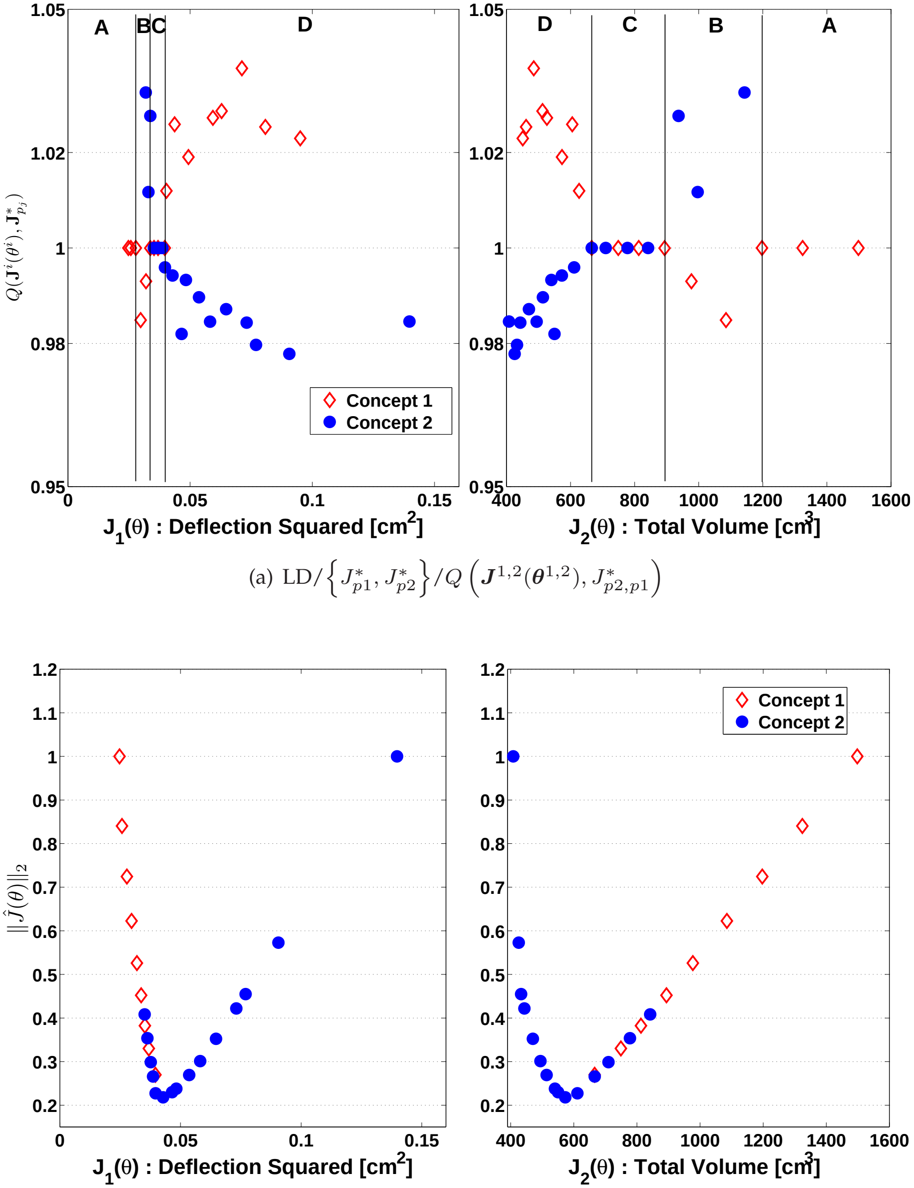
This is the disc brake design proposed in [139]. The design objectives to be minimized are the mass of the brake ($J_1(\boldsymbol{\theta})$ [Kg]), the stopping time ($J_2(\boldsymbol{\theta})$ [secs.]), and the engaging force ($J_3(\boldsymbol{\theta})$ [N]). The decision variables are the inner and outer radius of the disc (θ_1 and θ_2 respectively), the engaging force (θ_3), and the number of friction surfaces (γ). The latter is used to generate two different design concepts: a disc brake design with 4 and 6 friction surfaces respectively. Therefore, the following MOP is stated:



(a)

(b)

Figure 5.4: Objective exchange for two concepts in truss design example (a) and representation using LD/$\{J^*_{p1}, J^*_{p2}\}/\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ (b).



(a) LD/$\left\{J^*_{p1}, J^*_{p2}\right\}$/$Q\left(\boldsymbol{J}^{1,2}(\boldsymbol{\theta}^{1,2}), J^*_{p2,p1}\right)$

(b) LD/s-Pareto/$\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$

Figure 5.5: Concepts comparison for truss design example.

$$\begin{aligned}
\min \boldsymbol{J}(\boldsymbol{\theta}) &= [J_1(\boldsymbol{\theta}), J_2(\boldsymbol{\theta}), J_3(\boldsymbol{\theta})] && (5.18)\\
J_1(\boldsymbol{\theta}) &= 4.9 \cdot 10^{-5}(\theta_2^2 - \theta_1^2)(\gamma - 1) && (5.19)\\
J_2(\boldsymbol{\theta}) &= \frac{9.82 \cdot 10^{6}(\theta_2^2 - \theta_1^2)}{\theta_3 \gamma (\theta_2^3 - \theta_1^3)} && (5.20)\\
J_3(\boldsymbol{\theta}) &= \theta_3 && (5.21)
\end{aligned}$$

subject to:

$$\begin{aligned}
\theta_1 &\geq 55 && (5.22)\\
\theta_1 &\leq 80 && (5.23)\\
\theta_2 &\geq 75 && (5.24)\\
\theta_2 &\leq 110 && (5.25)\\
\theta_3 &\geq 1000 && (5.26)\\
\theta_3 &\leq 3000 && (5.27)\\
0 &\leq (\theta_2 - \theta_1) && (5.28)\\
0 &\leq 30 - 2.5 \cdot (\gamma + 1) && (5.29)\\
0 &\leq 0.4 - \frac{\theta_3}{3.14(\theta_2^2 - \theta_1^2)} && (5.30)\\
0 &\leq 1 - \frac{2.22 \cdot 10^{-3}\theta_3(\theta_2^3 - \theta_1^3)}{3.14(\theta_2^2 - \theta_1^2)^2} && (5.31)\\
0 &\leq \frac{2.66 \cdot 10^{-2}\theta_3 \gamma (\theta_2^3 - \theta_1^3)}{3.14(\theta_2^2 - \theta_1^2)} - 900 && (5.32)
\end{aligned}$$

This example is used to show:

- The complexity increase on visualization when the DM is required to analyze more than two objectives.
- To introduce geometrical remarks and their physical meaning to the problem at hand.

In Figure 5.6, the objective tradeoff in a 3D-objective plot, and the LD/ $\{J^*_{p1}, J^*_{p2}\}/\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ visualization are presented. It is important to notice the complexity increases from 2 to 3 objectives; also, it is more difficult to identify strengths and drawbacks for each of the design concepts. In Figure 5.7, the LD/$\{J^*_{p1}, J^*_{p2}\}$/ $Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p_j}\right)$ and LD/s-Pareto/$\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ visualizations are shown. Some remarks from Figure 5.7 can be made:

- Both design concepts guarantee a useful coverage on $J_3(\boldsymbol{\theta})$.
- No single concept is able to fully cover $J_1(\boldsymbol{\theta})$ or $J_2(\boldsymbol{\theta})$.

- According to the last observation, if $J_1(\boldsymbol{\theta}) \leq 0.5$ is required, concept 1 must be used; however, if $J_2(\boldsymbol{\theta}) \leq 4$ is required, concept 2 must be selected.
- The bigger the mass of the brake, the better the improvement of concept 2 (◯) over concept 1 (◇) in the range $J_1(\boldsymbol{\theta}) \in [0.6, 1.0]$.

Notice how the analysis becomes more difficult as the objective space increases. In this example, using the LD/$\{J^*_{p1}, J^*_{p2}\}$/$Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p_j}\right)$ visualization, it is possible to decide under which circumstances one concept will be preferred over another. For example, when the DM has a strong preference for one concept (due to its flexibility or because it is less complex), he or she can accept sub-optimal solutions. The following example will be used to show the flexibility of the LD/$\{J^*_{p1}, J^*_{p2}\}$/$Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p_j}\right)$ visualization to build a Pareto front for a set of concepts *filtered* with preferences.

5.4.3 Parametric controller design

The next example is a parametric controller design $G(s)$ for the control benchmark proposed at the American Control Conference (ACC) [194]. The MOP statement described in [19] is used. It has six objectives: robust stability ($J_1(\boldsymbol{\theta})$); maximum control effort ($J_2(\boldsymbol{\theta})$), worst case settling time ($J_3(\boldsymbol{\theta})$); noise sensitivity ($J_4(\boldsymbol{\theta})$); nominal control effort ($J_5(\boldsymbol{\theta})$); and nominal settling time ($J_6(\boldsymbol{\theta})$). Two different controller structures $G_1(s)$ (design concept 1) and $G_2(s)$ (design concept 2) will be evaluated:

$$G_1(s) = \frac{\theta_1 s^2 + \theta_2 s + \theta_3}{s^3 + \theta_4 s^2 + \theta_5 s + \theta_6} \tag{5.33}$$

$$G_2(s) = \frac{\theta_1 s + \theta_2}{s^2 + \theta_3 s + \theta_4} \tag{5.34}$$

subject to:

$$-10 \leq \theta_i \leq 25 \tag{5.35}$$

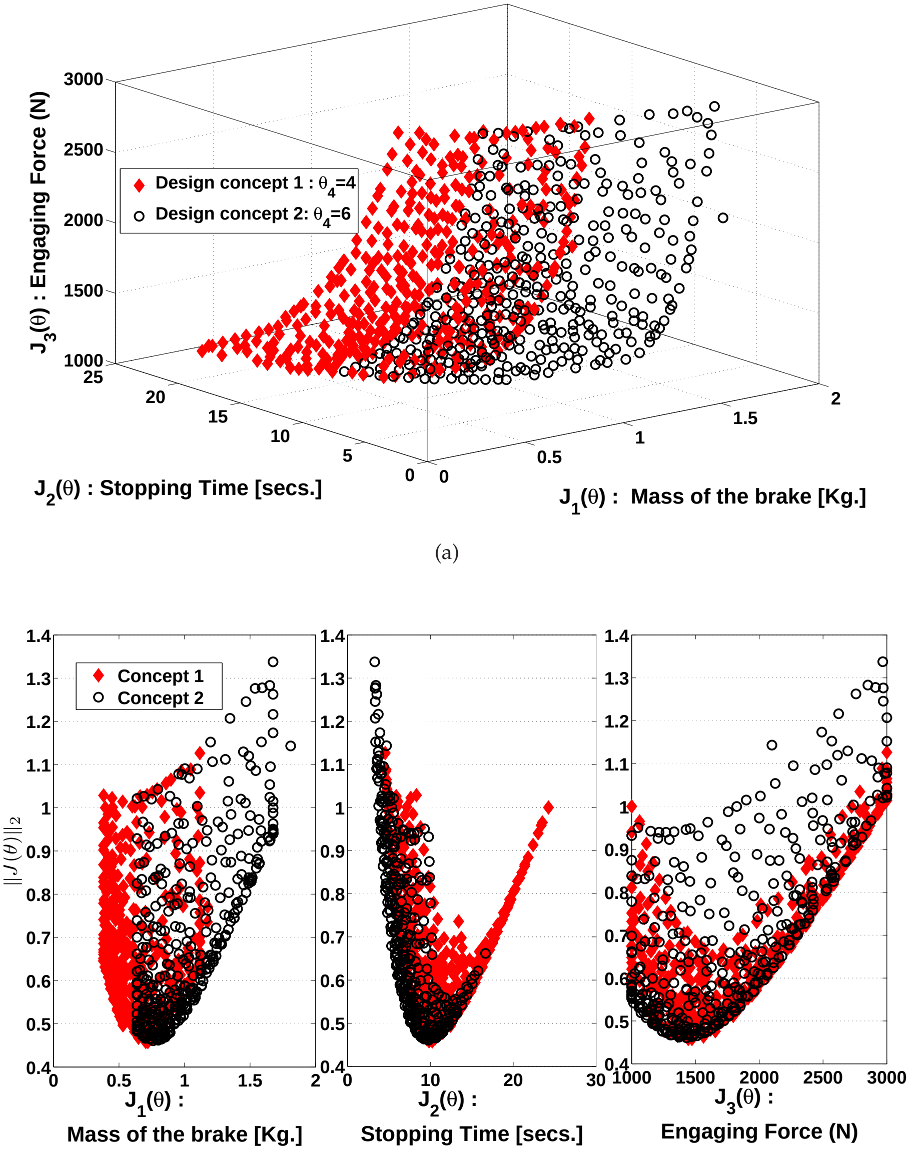
$$J_3(\boldsymbol{\theta}) \leq 100 \tag{5.36}$$

$$J_4(\boldsymbol{\theta}) \leq 1 \tag{5.37}$$

$$J_6(\boldsymbol{\theta}) \leq 50 \tag{5.38}$$

The aims with this example are to:

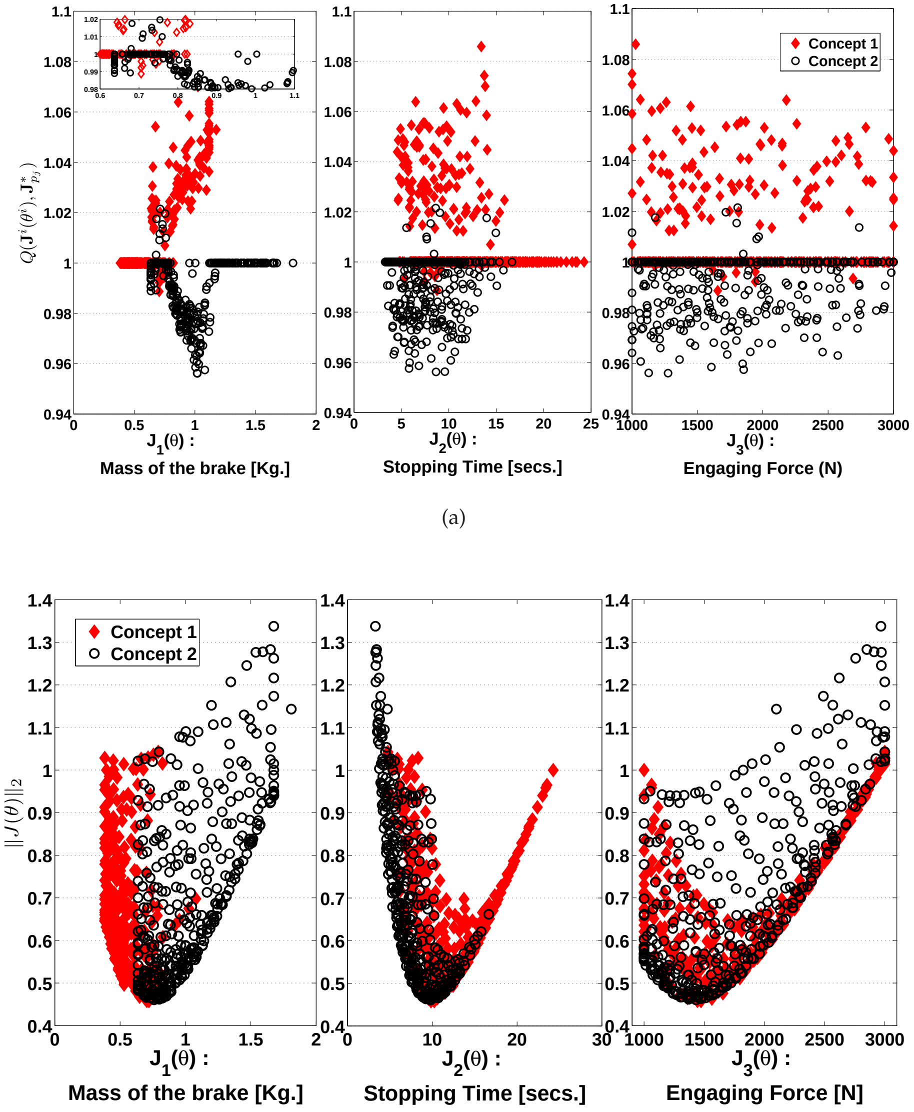
- Demonstrate the viability of the approach with more than three objectives.
- Perform a simple interactive solution selection procedure.



(a)

(b)

Figure 5.6: Objective exchange for two design: concepts in disc brake design example (a) and concepts comparison with LD/$\{J^*_{p1}, J^*_{p2}\}/\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ (b).



(a)

(b)

Figure 5.7: Concepts comparison for disc brake design using LD/$\{J^*_{p1}, J^*_{p2}\}$/$Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p_j}\right)$ (a) and representation of LD/s-Pareto/$\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ (b).

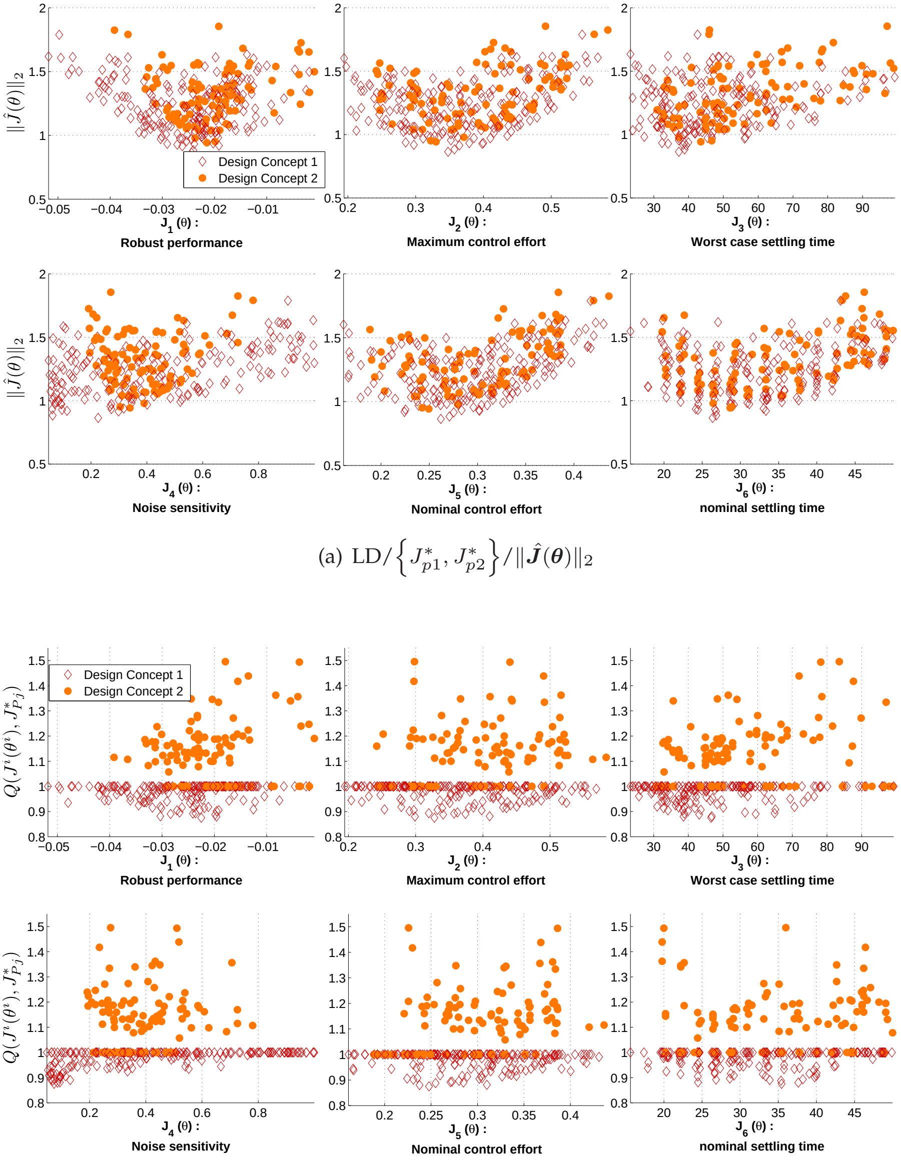
In Figure 5.8, the LD/$\{J^*_{p1}, J^*_{p2}\}/Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p_j}\right)$ visualization is depicted. This figure shows that concept 2 better covers the objective space. Also, there are several solutions from concept 1 dominated by concept 2.

An inspection of LD/$\{J^*_{p1}, J^*_{p2}\}/Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p_j}\right)$ enables to determine under which circumstances one concept will be preferred over another. To show this flexibility, it will be assumed that the DM has a strong preference for concept 2, due to its simplicity and implementation facilities. Although concept 1 has a larger coverage, the DM observes that such covering implies a strong degradation in objective $J_4(\boldsymbol{\theta})$. Also, given the preference for implementing concept 2, the DM decides that small improvements of concept 1 over concept 2 are not justifiable. With this, the DM can build a Pareto front for a set of concepts *filtered* with his/her preferences (fs-Pareto front). For this particular example, the following assumptions will be assumed:

- Concept 2 is preferred over concept 1, when covering the same objective space (*i.e.*, if both concepts have the same type of objective tradeoff, then concept 2 will be always be selected).
- Solutions from concept 2 will be removed from the fs-Pareto front only if $Q(J^1(\boldsymbol{\theta^1}), J^*_{p2}) > 1.10$ (*i.e.*, there is an acceptance region of 10% for a solution from concept 2 even if it is dominated by $\boldsymbol{J}^*_{p1}$). That is, some sub-optimal solutions could be accepted, given the simplicity of the design concept.
- Solutions from concept 2 will be inserted in the fs-Pareto front only if $Q(J^2(\boldsymbol{\theta^2}), J^*_{p1}) < \frac{1}{1.10}$ (*i.e.* a minimum improvement over concept 2 is required for any solution from concept 1 to be considered.)
- The DM would like to incorporate the preferences defined in Table 5.3 to facilitate its selection procedure.

With the aforementioned preferences, an fs-Pareto front is built. This front merges solutions from both concepts with a given set of preferences of the DM. In Figure 5.9 the LD/fs-Pareto/$\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$ visualization is shown. From this point onwards, the usual analysis on LD/front/measure using the LD-Tool [19] can be performed. For example, a solution (designed with a star in the aforementioned figure) could be selected to appreciate the tradeoff and performance when compared with concept 1.

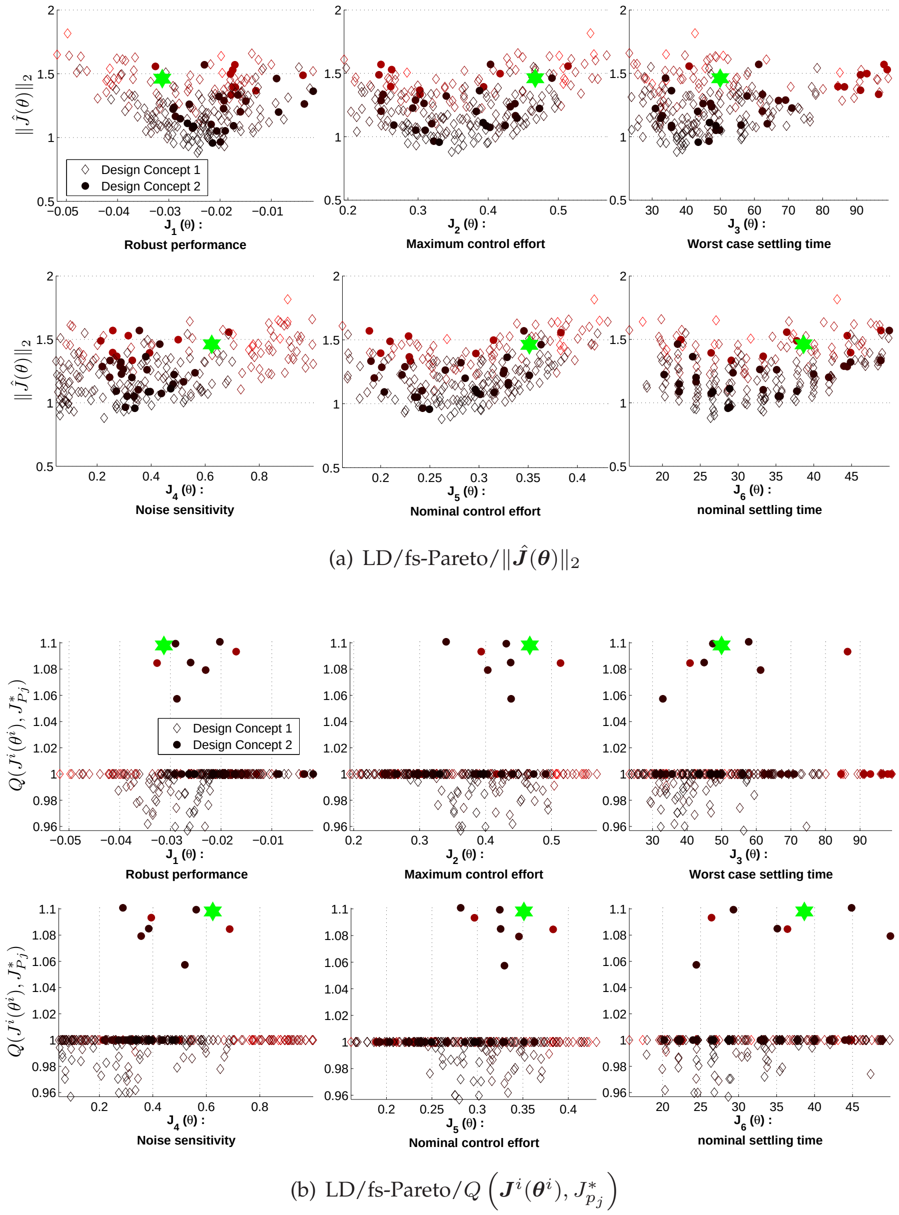
It is important to notice that, even with a large dimensional objective space, the LD visualization remains useful for design alternative selection and concept comparison. Such a filter over the s-Pareto front facilitates the DM analysis task, by allowing her/him to analyze the region and characteristics that are important according to her/his preferences.



(a) $\text{LD}/\left\{J^*_{p1}, J^*_{p2}\right\}/\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$

(b) $\text{LD}/\left\{J^*_{p1}, J^*_{p2}\right\}/Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p_j}\right)$

Figure 5.8: Objective exchange visualization for parametric controller tuning example.



(a) LD/fs-Pareto/$\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$

(b) LD/fs-Pareto/$Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p_j}\right)$

Figure 5.9: Visualization with preferences (see table 5.3) for parametric controller tuning example. The darker the solution, the most preferable according to the DM's preferences.

Table 5.3: Preference matrix for parametric controller design example. Three preference ranges are defined: Highly Desirable(HD), Desirable (D) and Tolerable (T).

		← HD →	← D →	← T →
Objective	J_i^0	J_i^1	J_i^2	J_i^3
Robust performance	-1.00	-0.04	-0.02	-0.001
Maximum control effort	0.00	0.30	0.40	0.500
Worst case settling time	0.00	40.00	60.00	80.000
Noise sensitivity	0.00	0.50	0.60	0.700
Nominal control effort	0.00	0.30	0.40	0.500
Nominal settling time	0.00	30.00	40.00	50.000

5.4.4 Performance evaluation of MOEAs

The final example is a proposal for MOEA visual comparison using the proposed approach. This example is used to:

- Make a proposal for MOEA visual comparison with more than three objectives.
- Show how the comparison of three or more design concepts could be performed.

Three different design concepts (MOEAs) are used to calculate a Pareto front approximation for problem WFG6 (with 32 decision variables and five objectives) from the benchmark test suit for multiobjective optimization described in [84]. Design concepts to be used are:

Concept 1: A DE algorithm with the diversity/pertinency mechanism described in [189].

Concept 2: A standard GA for multiobjective optimization.

Concept 3: A standard DE algorithm for multiobjective optimization.

A target vector solution is defined at $\boldsymbol{J}(\boldsymbol{\theta}) = [1.30, 2.24, 2.79, 5.24, 8.24]$ in all cases. The GA algorithm uses a basic penalty-function mechanism for this purpose. A total of 25 independent runs for each design concept was evaluated and 2000 function evaluations are used.

In order to keep simplicity as a requirement, all the proposals will be compared with a reference Pareto front J_p^{ref} (design concept 0) to avoid the comparison by pairwise design concepts. In this case, it is straightforward to generate such a Pareto front, since it is known. In other instances, a Pareto front for reference could be generated by merging all the solutions obtained and using a dominance filter. This would be helpful to visualize the degree of estrangement of design alternatives in a given design concept from the reference front.

To evaluate the average performance of the design concepts, the m-dimensional attainment surface is calculated as indicated in [98] [33]. The attainment surface at 50% ($\widetilde{J_{pi}^*}^{50\%}$) is calculated for the design concept comparison. In Figure 5.10 the LD/$\left\{\widetilde{J_{p1}^*}^{50\%}, \widetilde{J_{p2}^*}^{50\%}, \widetilde{J_{p3}^*}^{50\%}\right\}$ / $Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J_{p0}^*\right)$ visualization is shown. It is possible to appreciate that design concepts 2 and 3 (circles and points respectively) show almost the same performance in approximating the Pareto front. Concept 1 (squares) shows a better performance, since it is closer to 1 (the Pareto front for reference); furthermore, several solutions seem to be in the Pareto front, or at least, are not dominated by the Pareto front for reference J_{p0}^* [34]. This graphical inspection, as noticed in [98] is helpful because it is important to have a visual validation for MOEA performance evaluation when other metrics are used (such as hypervolume for instance). Notice that while the performance metrics by themselves give an overall-quantitative measure on Pareto front performance, this graphical inspection enables a visualization and comparison of the algorithm performance by sectors.

5.5 Conclusions on this chapter

In this chapter, a new quality measurement $Q(J^i(\boldsymbol{\theta^i}), J_{pj}^*)$ was defined to perform concept comparison using level diagrams (LD). With this measure, and the flexibility of LD to visualize and analyze m-dimensional Pareto fronts, the DM is able to obtain a better insight into objective tradeoffs for different concepts. This was validated in a set of multi-criteria decision making problems of 2, 3, 5 and 6 objectives.

Furthermore, with the $Q(J^i(\boldsymbol{\theta^i}), J_{pj}^*)$ quality measurement it is possible to have a measure of alternative design improvements between design concepts. With this feature, it is possible to build a Pareto front for a set of concepts *filtered* (fs-Pareto) with the DM's preferences. In this fs-Pareto front, the DM can accept sub-optimal solutions for one concept when for example, the improvement of one concept over another is small, and the use of the s-Pareto optimal solution is not justifiable.

[33] C code freely available at `http://dbkgroup.org/knowles/plot_attainments/`

[34] J_{p0}^* is calculated with 1000 solutions uniformly distributed in the known Pareto set.

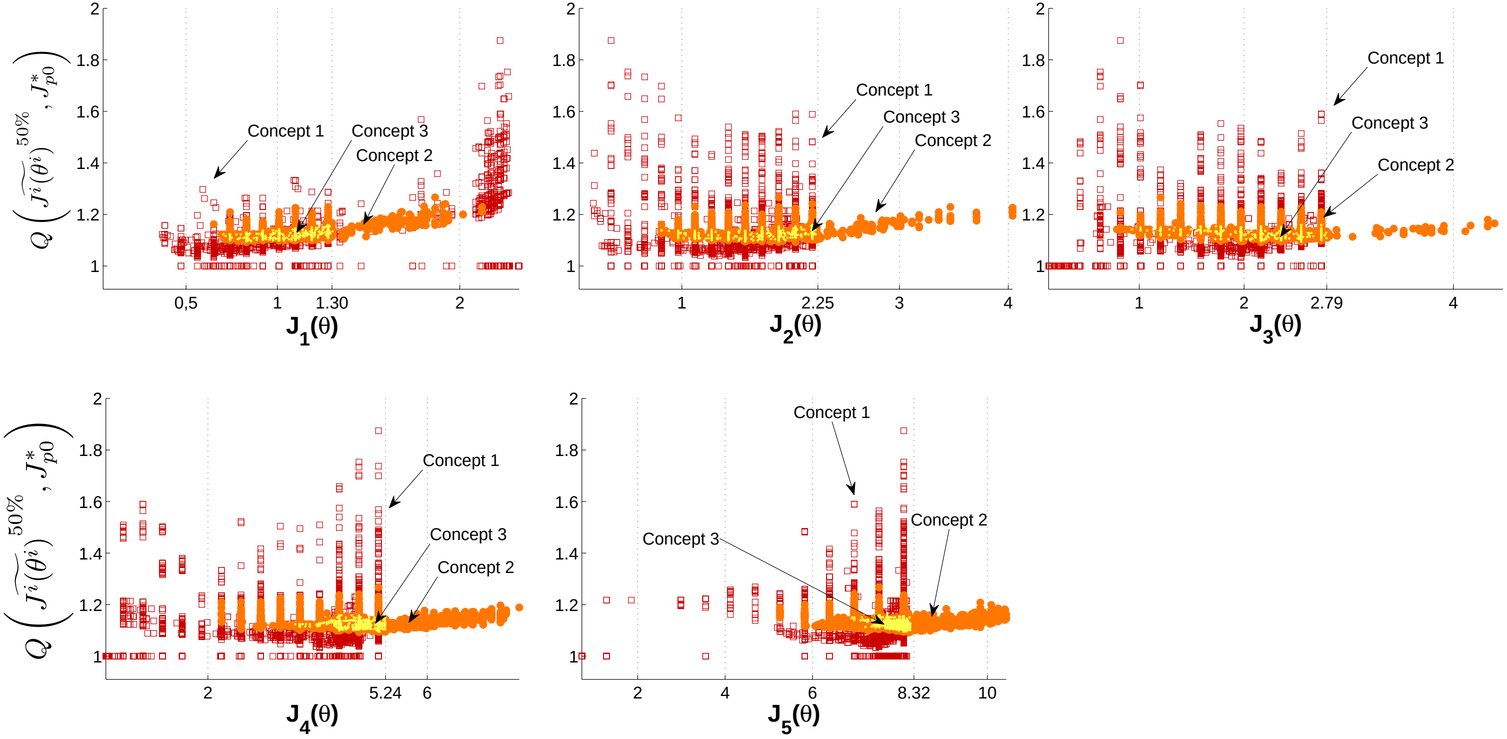


Figure 5.10: Visual comparison among MOEAs using LD / $\left\{\widetilde{J^*_{p1}}^{50\%}, \widetilde{J^*_{p2}}^{50\%}, \widetilde{J^*_{p3}}^{50\%}\right\}$ / $Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p0}\right)$.

Finally, the $Q(J^i(\boldsymbol{\theta^i}), J^*_{pj})$ quality measurement is used to test multiobjective optimization algorithms since it can measure the improvement of one algorithm over another. Although such evaluation can be done by means of the original I_ϵ binary indicator that calculates a single scalar for each Pareto front; with the $Q(J^i(\boldsymbol{\theta^i}), J^*_{pj})$ quality measurement it is possible to analyze the regions in the objective space where an algorithm performs better, thereby obtaining a better understanding of its weaknesses and strengths. This characteristic could be desirable when the algorithms are evaluated with real-world application problems.

CHAPTER 6:

Physical Programming for preference driven Evolutionary Multiobjective Optimization

This chapter is devoted to develop and evaluate a new preference oriented MOEA. Such contribution follows limitations found in Chapters 3 and 4 regarding pertinence of the approximated Pareto front. The mechanism proposed allows to handle the DM's preferences in an intuitive manner, assuring pertinent solutions but avoiding uninteresting ones. A paper with results from this chapter is under review.

6

Physical Programming for preference driven Evolutionary Multiobjective Optimization

I don't know half of you half as well as I should like, and I like less than half of you half as well as you deserve.

Bilbo Baggins

6.1 Aim of this chapter

As commented before, one potentially desirable characteristic of a MOEA is the mechanism for preference handling to calculate pertinent solutions. That is, the capacity to obtain a set of interesting solutions from the DM's point of view.

The specifics aims of this chapter are:

- To develop a theorical framework for preferences inclusion in the EMO process, in order to improve pertinency of solutions in the approximated Pareto front and to deal with many-objective and constrained optimization instances.

- To provide the designer with a Tool-box with the aforementioned framework in order to improve pertinency capabilities in the sp-MODE algorithm.

In this chapter, Physical Programming (PP) is merged with MOEAs as an auxiliary mechanism to improve pertinency of the calculated solutions. Such approach will allow to have solutions more useful for the DM, since it provides a flexible and intuitive coding statement where the MOP is built from his/her point of view. An algorithm to test its viability is developed; nevertheless it could be potentially used by any MOEA. The remainder of this chapter is as follows: in Section 6.2 some preliminaries in multiobjective optimization, physical programming and the MOEA to be used are presented. In Section 6.3 the preference handling mechanism is explained and evaluated in Section 6.4. Finally, some concluding remarks are given.

6.2 Background on Physical Programming

Physical Programming (PP) is a suitable technique for multiobjective engineering design since it formulates design objectives in an understandable and intuitive language for designers. PP is an aggregate objective function (AOF) technique [119] for multiobjective problems that includes the available information in the optimization phase. That enables the designer to express preferences relative to each objective function with more detail. Firstly, PP translates the designer's knowledge into classes[35] with previously defined ranges[36]. This preference set reveals the DM's wishes using physical units for each of the objectives in the MOP. From this point of view, the problem is moved to a different domain where all the variables are independent of the original MOP (see Figure 6.1).

For each objective and set of preferences $\mathfrak{P}$, a class function $\eta_q(\boldsymbol{J}(\boldsymbol{\theta}))|_{\mathfrak{P}}$, $q = [1, \ldots, m]$ is built to translate each objective $J_q(\boldsymbol{\theta})$ to a new image. In the new codomain, all the objectives are equivalent to each other. Afterwards, a PP index $J_{pp}(\boldsymbol{J}(\boldsymbol{\theta})) = \sum\limits_{q=1}^{m} \eta_q(\boldsymbol{J}(\boldsymbol{\theta}))$ is calculated.

In this work, the implementation stated in [165] named as Global Physical Programming (GPP), is a better fit for evolutionary optimization techniques and will be used. This is due to the fact that the original method employs several resources to build the proper class functions $\eta_q(\boldsymbol{J}(\boldsymbol{\theta}))|_{\mathfrak{P}}$, fulfilling a list of convexity and continuity requirements. The interested reader might refer to [165, 118] for a detailed explanation. For the sake of simplicity such details are not reproduced here, since they will be used as basis for the development to be presented in Section 6.3.

The $J_{pp}(\boldsymbol{J}(\boldsymbol{\theta}))$ index is suitable to evaluate the performance of a design alternative, but not that of the design concept. That is, if it is used as it is, it will

[35] The original method states 4 clases: 1S (smaller is better), 2S (larger is better), 3S (a value is better) and 4S (range is better)

[36] According the original method: Highly Desirable (HD), Desirable (D), Tolerable (T), Undesirable (U) and Highly Undesirable (HU)

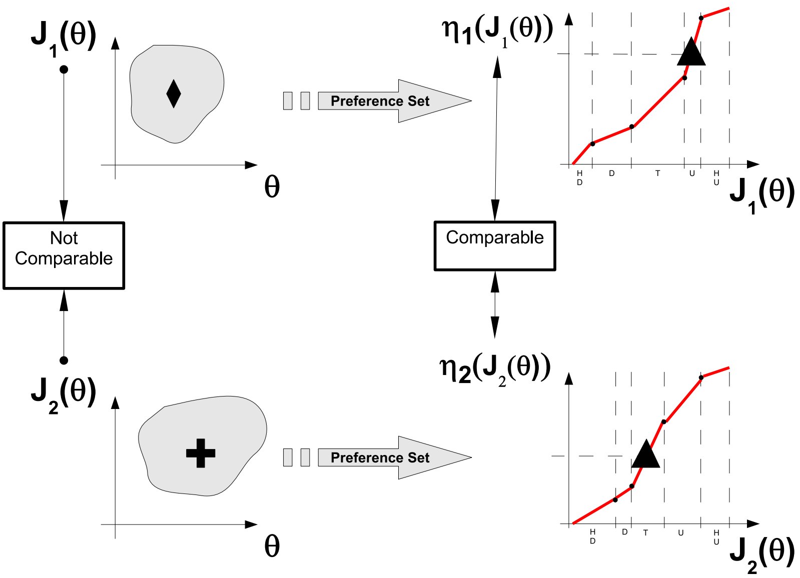


Figure 6.1: Physical Programming (PP) notion. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

evolve the entire population to a single Pareto optimal solution. Therefore, it must be merged with other(s) mechanism(s) to maintain diversity in the Pareto front. Pruning mechanisms seem to be a promising solution for this purpose. Therefore, GPP will be used with sp-MODE algorithm (see Algorithm 1.6 and references [158, 148]), which is a MOEA based on Differential Evolution algorithm [183, 125, 41] and spherical coordinates to prune $\boldsymbol{J}_P^*$. Even if similar algorithms use similar approaches [22, 102, 12], the use of a norm to perform the pruning makes it suitable to incorporate preferences, as detailed below.

6.3 Pertinency improvement mechanism by means of global Physical Programming

Global physical programming is a tool which could be used in different ways by MOEAs. In this case, it will be merged together with a pruning technique, in order to decide which solutions will be archived in an external file. GPP can be used as a selection mechanism in the evolved population and/or in the store and replace mechanism in the external archive A.

6.3.1 Global Physical Programming statements

Given a vector $\boldsymbol{\varphi} \in \mathbb{R}^m$, linear functions will be used for class functions $\eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}}$ as detailed in [165][37], due to its simplicity and interpretability. Firstly an offset between two adjacent ranges is incorporated (see Figure 6.2) to meet the *one versus another* (OVO) rule criterion [18, 20]. Given a set of preferences $\mathfrak{P}$ with M ranges for m objectives:

$$\mathfrak{P} = \begin{pmatrix} J_1^1 & \cdots & J_1^M \\ \vdots & \ddots & \vdots \\ J_m^1 & \cdots & J_m^M \end{pmatrix} \tag{6.1}$$

$\eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}}$, $q = [1, \ldots, m]$ are defined as:

$$\eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}} = \alpha_{k-1} + \delta_{k-1} + \Delta\alpha_k \frac{\varphi_q - J_q^{k-1}}{J_q^k - J_q^{k-1}} \tag{6.2}$$

$$J_q^{k-1} \leq \varphi_q < J_q^k \tag{6.3}$$

where

$$\alpha_0 = 0 \tag{6.4}$$

$$\alpha_1 \in \mathbb{R}^+ \tag{6.5}$$

$$\alpha_k > \alpha_{k-1} \quad (1 < k \leq M) \tag{6.6}$$

$$\Delta\alpha_k = \alpha_k - \alpha_{k-1} \quad (1 \leq k \leq M) \tag{6.7}$$

$$\delta_0 = 0 \tag{6.8}$$

$$\delta_1 \in \mathbb{R}^+ \tag{6.9}$$

$$\delta_k > m \cdot (\alpha_k + \delta_{k-1}) \quad (1 < k \leq M) \tag{6.10}$$

The last inequality guarantees the *one versus others* (OVO) rule, since an objective value in a given range is always greater than the sum of the others in a most preferable range. Therefore, the $J_{gpp}(\boldsymbol{\varphi})$ index is defined as:

$$J_{gpp}(\boldsymbol{\varphi}) = \sum_{q=1}^{m} \eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}} \tag{6.11}$$

The $J_{gpp}(\boldsymbol{\varphi})$ has an intrinsic structure to deal with constraints. If the fulfillment of constraints is required, they will be included in the preference set as objectives. That is, preference ranges will be stated for each constraint and they will be used to compute the $J_{gpp}(\boldsymbol{\varphi})$ index. The $\eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}}$ is shown in Figure 6.2 for the specific case (to be used hereafter) of the following 5 preference ranges:

[37] Hereafter, only 1S classes (the smaller, the better) will be considered.

HD: Highly desirable if $J_q^0 \leq J_q(\varphi) < J_q^1$.

D: Desirable if $J_q^1 \leq J_q(\varphi) < J_q^2$.

T: Tolerable if $J_q^2 \leq J_q(\varphi) < J_q^3$.

U: Undesirable if $J_q^3 \leq J_q(\varphi) < J_q^4$.

HU: Highly undesirable $J_q^4 \leq J_q(\varphi) < J_q^5$.

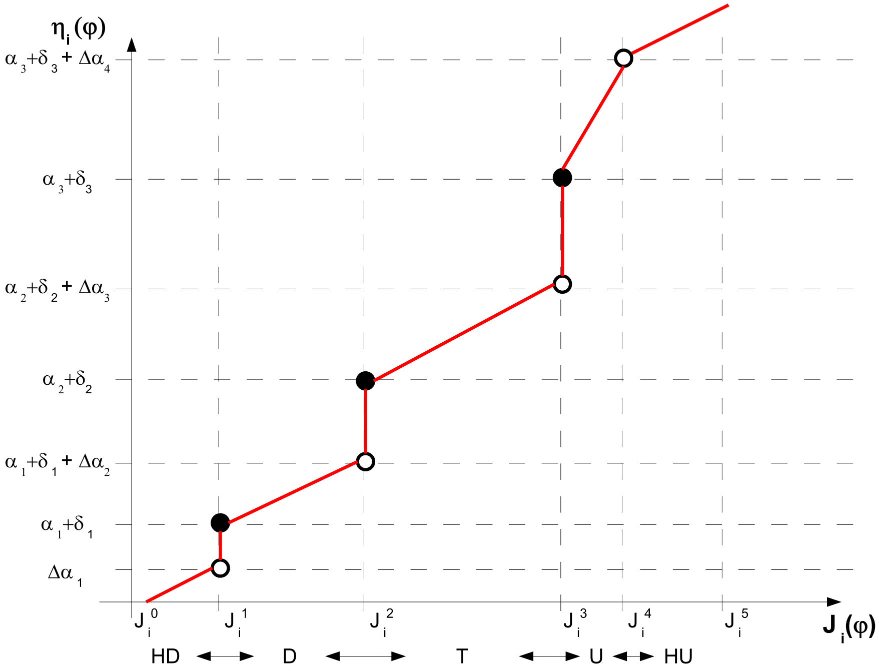


Figure 6.2: New class definition for global physical programming.

Those preferences ranges are defined for the sake of flexibility (as it will be shown) to evolve the population to a pertinent Pareto front. The following definitions will be used (see Figure 6.3):

T_Vector: $\boldsymbol{J}^T = [J_1^3, J_2^3, \cdots, J_m^3]$, *i.e.* the vector with the maximum value for each objective in the tolerable range.

D_Vector: $\boldsymbol{J}^D = [J_1^2, J_2^2, \cdots, J_m^2]$, *i.e.* the vector with the maximum value for each objective in the desirable range.

HD_Vector: $\boldsymbol{J}^{HD} = [J_1^1, J_2^1, \cdots, J_m^1]$, *i.e.* the vector with the maximum value for each objective in the highly desirable range.

T_HypV: The hypervolume of the Pareto front approximation bounded by $\boldsymbol{J}^T$.

D_Hypv: The hypervolume of the Pareto front approximation bounded by $\boldsymbol{J}^D$.

HD_HypV: The hypervolume of the Pareto front approximation bounded by $\boldsymbol{J}^{HD}$.

T_$\boldsymbol{J}_P^*$: the Tolerable Pareto front approximation where all solutions dominate $\boldsymbol{J}^T$.

D_$\boldsymbol{J}_P^*$: the Desirable Pareto front approximation where all solutions dominate $\boldsymbol{J}^D$.

HD_$\boldsymbol{J}_P^*$: the Highly Desirable Pareto front approximation where all solutions dominate $\boldsymbol{J}^{HD}$.

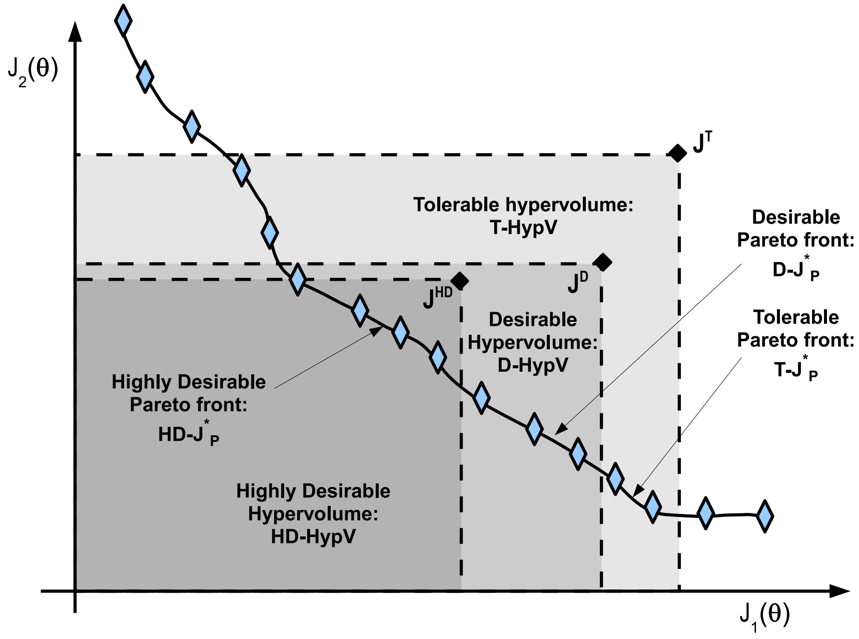


Figure 6.3: Graphical representation of the definitions stated.

6.3.2 Population selection and archive update

The $J_{gpp}(\varphi)$ index will be used as a selection mechanism in the evolutionary technique. Nevertheless, using it through the entire evolution process does not yield a practical approach. This is because the $J_{gpp}(\varphi)$ would lead the entire population to converge to a single solution, since the physical index converges to a single Pareto optimal solution. To avoid this, a mechanism must be designed to evolve the population to a zone of interest and afterwards, promote diversity. In this case, a switch operator is used in DE to change the selection criteria (Algorithm 6.1) at a certain value J_{gpp}^{max}.

This J_{gpp}^{max} value needs to be previously defined. This upper bound on $J_{gpp}(\boldsymbol{\varphi})$ will push the population to evolve to a desired preference region. As five preference ranges are defined, an intuitive selection of such a value is $J_{gpp}^{max} = J_{gpp}(\boldsymbol{J}^T)$. This will guarantee the population evolves into the feasible **T_HypV** and then, perform a selection based on dominance. With this, the evolutionary process has a strong pressure to reach the **T_HypV**; furthermore, it is assured that only tolerable Pareto optimal solutions will be contained in $\boldsymbol{J}_P^*$.

In the case of the archiving strategy to update A, $J_{gpp}(\boldsymbol{\varphi})$ is used as pseudo-norm measurement to select one solution into each spherical sector (see Algorithm 6.2). With this, the *most preferable solution* according to the set of preferences $\mathfrak{P}$ previously defined by the designer will be retained in each spherical sector.

```
Read generation counter G;
Read offspring (child population) $O|_G$ and Subpopulation (parent) $S|_G$;
for i=1:Solutions In Child Population do
    Calculate the physical index $J_{gpp}(\boldsymbol{J}(\boldsymbol{u}^i))$ and $J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}^i))$;
    if $J_{gpp}(\boldsymbol{J}(\boldsymbol{u}^i)) > J_{gpp}^{max}$ AND $J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}^i)) > J_{gpp}^{max}$ then
        if $J_{gpp}(\boldsymbol{J}(\boldsymbol{u}^i)) < J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}^i))$ then
            $\boldsymbol{u}^i$ substitutes $\boldsymbol{\theta}^i$ in the main population $P|_G$
        end
    end
    if $J_{gpp}(\boldsymbol{J}(\boldsymbol{u}^i)) < J_{gpp}^{max}$ AND $J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}^i)) > J_{gpp}^{max}$ then
        $\boldsymbol{u}^i$ substitutes $\boldsymbol{\theta}^i$ in the main population $P|_G$
    end
    if $J_{gpp}(\boldsymbol{J}(\boldsymbol{u}^i)) < J_{gpp}^{max}$ AND $J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}^i)) < J_{gpp}^{max}$ then
        if $\boldsymbol{u}^i \prec \boldsymbol{\theta}^i$ AND $\boldsymbol{\theta}^i \in \boldsymbol{P}|_{\boldsymbol{G}}$ then
            $\boldsymbol{u}^i$ substitutes $\boldsymbol{\theta}^i$ in the main population $P|_G$
        end
    end
end
Selection procedure terminates. Return Parent population $P|_G$;
```

Algorithm 6.1: DE selection procedure with global physical programming.

6.3.3 Tolerable solutions handling

The previously mentioned feature could be enough if the DM will be satisfied in considering solutions which could have all their individual objective values

```
Read generation counter G;
Read $\hat{A}|_G$ to be prune;
Read and update extreme values for $J_{ref}|_G$;
for each member in $\hat{A}|_G$ do
    calculate its normalized spherical coordinates (Definition 3)
end
Build the spherical grid (Definitions 4 and 5);
for each member in $\hat{A}|_G$ do
    calculate its spherical sector (Definition 6)
end
for i=1:SolutionsInParentPopulation do
    if $J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}^i)) > J_{gpp}^{max}$ then
        $\boldsymbol{\theta}^i$ is discarded
    end
    if $J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}^i)) \leq J_{gpp}^{max}$ then
        Compare with the remainder solutions in $A|_G$;
        if no other solution has the same spherical sector then
            it goes to the archive $A|_{G+1}$
        end
        if other solutions are in the same spherical sector then
            it goes to the archive $A|_{G+1}$ if it has the lowest $J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}^i))$
        end
    end
end
Pruning ends. Return $A|_{G+1}$;
```

Algorithm 6.2: Spherical pruning with physical programming index.

in the tolerable region (for example). Nevertheless, most of the times, the DM is willing to accept just *some* of the objectives in a given region (see Figure 6.4). Such a feature can be incorporated in the pruning mechanism by modifying J_{gpp}^{max}. For example, it is assumed that the DM is dealing with a m objectives problem. The following values [38] could be stated for J_{gpp}^{max}:

- If $J_{gpp}^{max} = J_{gpp}(\boldsymbol{J}^T)$, then solutions with their m objectives in the Tolerable region could appear in $\boldsymbol{J}_P^*$.
- If $J_{gpp}^{max} = J_{gpp}([\overbrace{J_1^3, J_2^3, \ldots, J_{m-1}^3,}^{Tolerable} \overbrace{J_m^2}^{Desirable}])$, then solutions with their m objectives in the Tolerable region will not appear in $\boldsymbol{J}_P^*$.

[38] without loss of generality, values are modified starting from objective m towards objective 1.

- If $J_{gpp}^{max} = J_{gpp}([\overbrace{J_1^3, J_2^3, \ldots, J_{m-2}^3}^{Tolerable}, \overbrace{J_{m-1}^2, J_m^2}^{Desirable}])$, then solutions with $m-1$ or more objectives in the Tolerable region will not appear in $\boldsymbol{J}_P^*$.
- If $J_{gpp}^{max} = J_{gpp}([\overbrace{J_1^3, J_2^3, \ldots, J_{m-3}^3}^{Tolerable}, \overbrace{J_{m-2}^2, J_{m-1}^2, J_m^2}^{Desirable}])$, then solutions with $m-2$ or more objectives in the Tolerable region will not appear in $\boldsymbol{J}_P^*$.
- And so on.

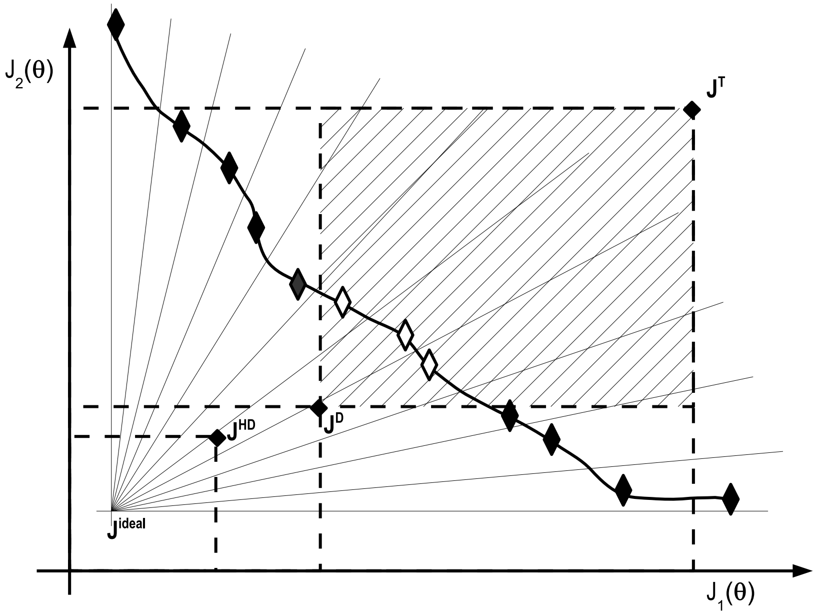


Figure 6.4: Handling of tolerable solutions. The algorithm will avoid (on the designer's request) solutions with several tolerable values (light solutions) according with the J_{gpp}^{max} value defined. In the example, bi-objective vectors with both values in the tolerable zone are omitted.

6.3.4 Multiple preferences coding

The designer may state more than one preference set $\mathfrak{P}$ for a given MOP. This could be the case of many-objective optimization, where the DM is willing to accept some degradation in one objective, if he/she can assure outstanding performance on the remainder (see Figure 6.5). It is assumed that the designer defines K different preference sets; therefore, the Equation (6.11) is redefined as:

$$J_{gpp}(\boldsymbol{\varphi}) = \min_{k=[1,\ldots,K]} \left(\sum_{q=1}^{m} \eta_q(\boldsymbol{\varphi})|_{\mathfrak{P}_k} \right) \tag{6.12}$$

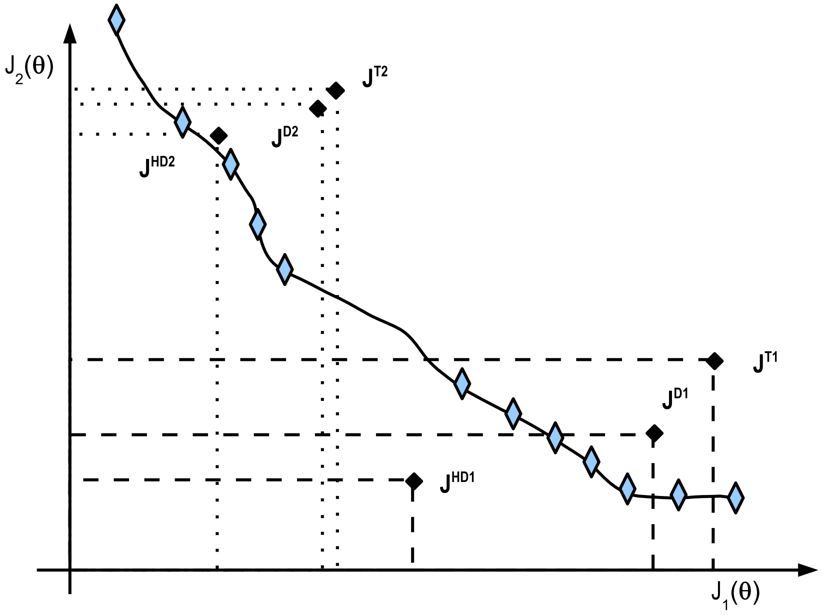


Figure 6.5: Multiple preference set definition.

6.3.5 Pareto front approximation size control

The GPP can be used as a mechanism to dynamically adapt the algorithm according to the desired set of solutions. Instead of reducing the grid size, J_{gpp}^{max} will be modified to retain the most preferable solutions (Figure 6.6). This also will facilitate the grid size selection. A threshold for a desired number of solutions $[\underline{car(J_P^*)}, \overline{car(J_P^*)}]$ will be stated. A simple dynamic adaptation of J_{gpp}^{max} is described in Algorithm 6.3. With this mechanism, the pruning technique will concentrate towards the **HD-HypV**, if the required number of solutions in the **T-HypV** is beyond the bound $\overline{car(J_P^*)}$.

```
if $car(A|_G) > \overline{car(J_P^*)}$ then
    Sort elements on $A|_G$ according to their $J_{gpp}(\boldsymbol{J(\theta)})$ index;
    Substitute $J_{gpp}^{max}$ value with the Physical norm of the $\overline{car(J_P^*)}$-th element;
end
Return $A|_G$;
```

Algorithm 6.3: Dynamic size control.

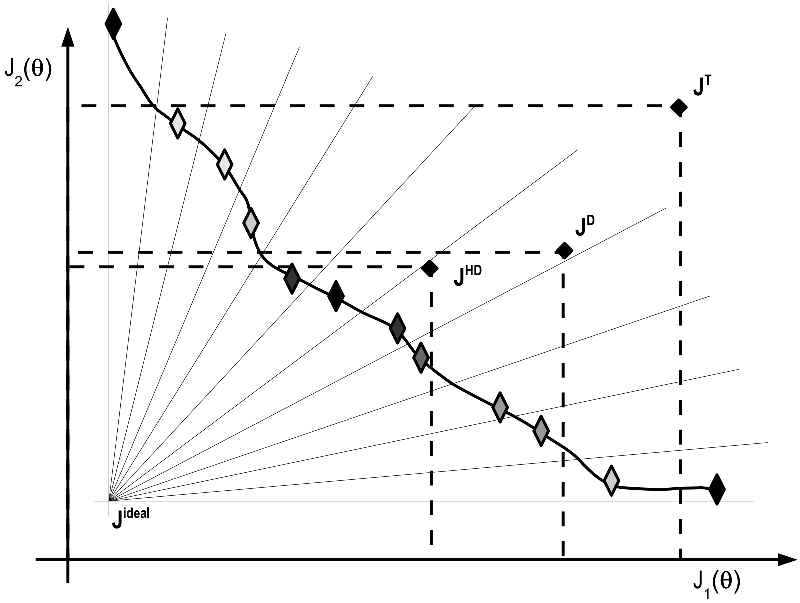


Figure 6.6: Dynamic size control. Less preferable solutions are omitted (on the designer's request) in the external archive A. The darker the solution, the more valuable according to the preference set.

6.3.6 Algorithm proposal: sp-MODE-II. Discussions and insights

With the aforementioned Algorithms 1.4, 6.1, 6.2 and 6.3, it is possible to rewrite Algorithms 1.2 and 1.3 in order to state a proposal using GPP to improve pertinency of the $\boldsymbol{J}_P^*$. Therefore, a spherical pruning with preferences multiobjective differential evolution algorithm (sp-MODE-II) is presented (see Algorithm 6.4). Some insights and discussions are commented below.

Using GPP or related approaches brings the additional task of defining K preference sets $\mathfrak{P}$. Nevertheless, in several instances this supplementary effort could be justifiable, if it brings a Pareto set approximation with more pertinent solutions to the DM, facilitating the decision making step. Therefore the DM must be willing to spend this additional effort at the beginning of the MOP statement definition. If upper and lower bounds on objectives are known and are sufficient to improve pertinency, a simple constraint coding could be used [156].

A statement to discourage the usage of the approach presented here could be the need to define the preference set $\mathfrak{P}$. It is fundamental to have an understanding of the objectives to define the preference ranges. Nevertheless, if the DM has no idea on such values, it could be an indicative of a perfunctory or precipitate selection of the design objectives. Therefore, perhaps the DM should ponder the design objectives stated. One of the advantages of MOEAs concerns using more interpretable and meaningful objectives for the DM and this aspect should be exploited.

Other mechanisms as goal vectors use one or more proposal vectors to redirect the evolution process. Nevertheless, such vectors need to be carefully chosen. A goal vector selected inside or outside of the Pareto front could affect the algorithm performance. The presented proposal brings the flexibility of a better redirection of the evolution process.

Finally, according to the number of solutions, the designer could adopt a *ten times the number of objectives* rule of thumb, based on [119]. In such work, it is noticed that for 2-3 objectives a set of 20-30 design alternatives could be handled by the DM.

```
Generate initial population $P|_0$ with $N_p$ individuals;
Evaluate $P|_0$;
Apply dominance criterion on $P|_0$ to get $A|_0$;
while stopping criterion not satisfied do
    Read generation count $G$;
    Get subpopulation $S|_G$ with solutions in $P|_G$ and $A|_G$;
    Generate offspring $O|_G$ with $S|_G$ using DE operators (Algorithm
    1.4).;
    Evaluate offspring $O|_G$;
    Update population $P|_G$ with offspring $O|_G$ according to $J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}))$
    values (Algorithm 6.1);
    Apply dominance criterion on $O|_G \bigcup A|_G$ to get $\hat{A}|_G$;
    Apply pruning mechanism based on $J_{gpp}(\boldsymbol{J}(\boldsymbol{\theta}))$ (Algorithm 6.2) to
    prune $\hat{A}|_G$ to get $A|_{G+1}$;
    Apply dynamic size control on $A|_{G+1}$ (Algorithm 6.3);
    Update environment variables (if using a self-adaptive mechanism);
    $G = G + 1$;
end
Algorithm terminates. $\boldsymbol{J_P}$ is approximated by $\boldsymbol{J}_P^* = A|_G$;
```

Algorithm 6.4: sp-MODE-II.

6.4 Examples

Different tests are defined to evaluate the presented proposal. In all cases, parameters used for DE are $F = 0.5$ and $Cr = 0.9$ with an initial population of 50 individuals (in accordance to Table 6.1). The subpopulation $S|_G$ uses half of the individuals from $P|_G$ and half of the individuals from $A|_G$. Regarding the GPP index, values $\alpha_k = \frac{k}{10}, k > 0$ and $\delta_k = (m+1) \cdot (\alpha_k + \delta_{k-1}), k > 1$ are

Table 6.1: Guidelines for sp-MODE-II's parameters tuning.

Parameter	Value	Comments
DE algorithm		
F (Scaling factor)	0.5	Recognized as good initial choice according to [183].
Cr (Crossover rate)	$[0.8, 1.0]$	Values recognized for non-separable problems according to [155, 41].
	$[0.1, 0.2]$	Values recognized for separable problems according to [155, 41].
	0.5	Trade-off value for separable and non-separable problems. Default value used (for example) by MOEA/D algorithm [199].
N_p (Population)	50	While a *five to ten times the number of decision variables* rule has been recognized as a thumb rule [183] for single objective optimization, here it is proposed a default size of 50 individuals.
Pruning mechanism		
$\boldsymbol{\beta_\epsilon}$ (Arcs)	$10 \cdot \overbrace{[m, \ldots, m]}^{m-1}$	It is proposed for m-objective problems, to bound the grid size to m^{m-1} hyper spherical sectors.
Pertinency mechanism		
J_{gpp}^{max}	$J_{gpp}(\boldsymbol{J}^T)$	It is proposed as default value. Only solutions with their m objectives in the Tolerable region could appear in $\boldsymbol{J}_P^*$. It will be modified by the algorithm if the size control mechanism (Algorithm 6.3) is activated.
$\overline{car(J_P^*)}$	$10 \cdot m$	It is proposed as default value. In accordance to [119].
α_k	$\frac{k}{10}, k > 0$	It is proposed in accordance to Equation (6.6).
δ_k	$(m+1) \cdot (\alpha_k + \delta_{k-1}), k > 1$	It is proposed in accordance to Equation (6.10).

Table 6.2: Preferences for the 3-bar truss design problem. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

					Preference Set A						
	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{\theta})$	0.00		0.06		0.08		0.10		0.20		0.50
$J_2(\boldsymbol{\theta})$	0.00		550		650		750		1000		1500
$G_1(\boldsymbol{\theta})$	<2E+008		<2E+008		<2E+008		<2E+008		2E+008		2E+008
$G_2(\boldsymbol{\theta})$	<2E+008		<2E+008		<2E+008		<2E+008		2E+008		2E+008
$G_3(\boldsymbol{\theta})$	<2E+008		<2E+008		<2E+008		<2E+008		2E+008		2E+008

used.

6.4.1 The 3-bar truss design problem

The first problem is the well-known bi-objective 3-bar truss design. The truss design problem is a classical MOP benchmark statement to test algorithms, as well as decision making step procedures. The truss parameters proposed in [165] are used. Two objectives are minimized: deflection squared ($J_1(\boldsymbol{\theta})$, [cm^2]) and total volume ($J_2(\boldsymbol{\theta})$, [cm^3]), where each bar is constrained to a maximum stress $\sigma < 2E + 008$ [Pa]. Characteristics to be evaluated with this benchmark problem are:

- The implicit constraint handling mechanism with $J_{gpp}(\boldsymbol{\varphi})$ and the pruning technique.
- The capacity to build a $\mathbf{T\_}\boldsymbol{J}_P^*$, $\mathbf{D\_}\boldsymbol{J}_P^*$ and $\mathbf{HD\_}\boldsymbol{J}_P^*$ in one single run.
- Pareto front approximation control size.

As a rule of thumb, it will be stated that the DM looks for a Pareto set approximation with $10 \cdot m = 20$ solutions (approximately). Preferences stated for this problem are shown in Table 6.2. Notice that constraints are incorporated as additional objectives; since the designer is just interested in calculating solutions with $G_i(\boldsymbol{\theta}) < 2E{+}008$, no further specifications are required below the tolerable value. These additional objectives are used to calculate the $J_{gpp}(\boldsymbol{\varphi})$ index, but they will not be used for partitioning the space, unless required.

The results of one single typical run are shown in Figure 6.7. The following observations can be made:

1. In Figure 6.7a, a typical run with a grid size of 500 spherical sectors is shown. In this case, it is important to notice that the algorithm is capable of reaching the **HD_$\boldsymbol{J}_P^*$**, but using computational resources without regarding DM's interesting areas.

2. In Figure 6.7b, the algorithm is executed again, with the same parameters, but using the $J_{gpp}(\boldsymbol{\varphi})$ index. In this case, the algorithm spends resources taking care of **T_HypV**.

3. Finally, in figure 6.7c, the algorithm is executed again, with the size control mechanism. Note how the algorithm concentrates in the **HD_HypV**, sacrificing all solutions in the **T_HypV**, and retaining few solutions from the desirable Pareto front.

Certainly, for bi-objectives problems (most of the cases) it is not difficult to attain a Pareto front approximation such as the one in Figure 6.7a and apply successive filtering to reach the highly desirable region. Nevertheless, with cost function evaluations with high computational load (complex simulations for example) it could be a desirable characteristic to evolve quickly to the desirable zone.

Usually, it is not difficult to perform an analysis on bi-objective problems, for tradeoff and preference articulation. Next, a problem with three objectives is analyzed, to evaluate the flexibility of the $J_{gpp}(\boldsymbol{\varphi})$ index.

6.4.2 The DTLZ2 benchmark problem

The second benchmark example is the DTLZ2 problem [45], with three objectives and ten decision variables. The Pareto front is the surface contained in the first quadrant of a hypersphere with unitary radius. It is used to show the following characteristics:

- Capacity to build a T-$\boldsymbol{J}_P^*$

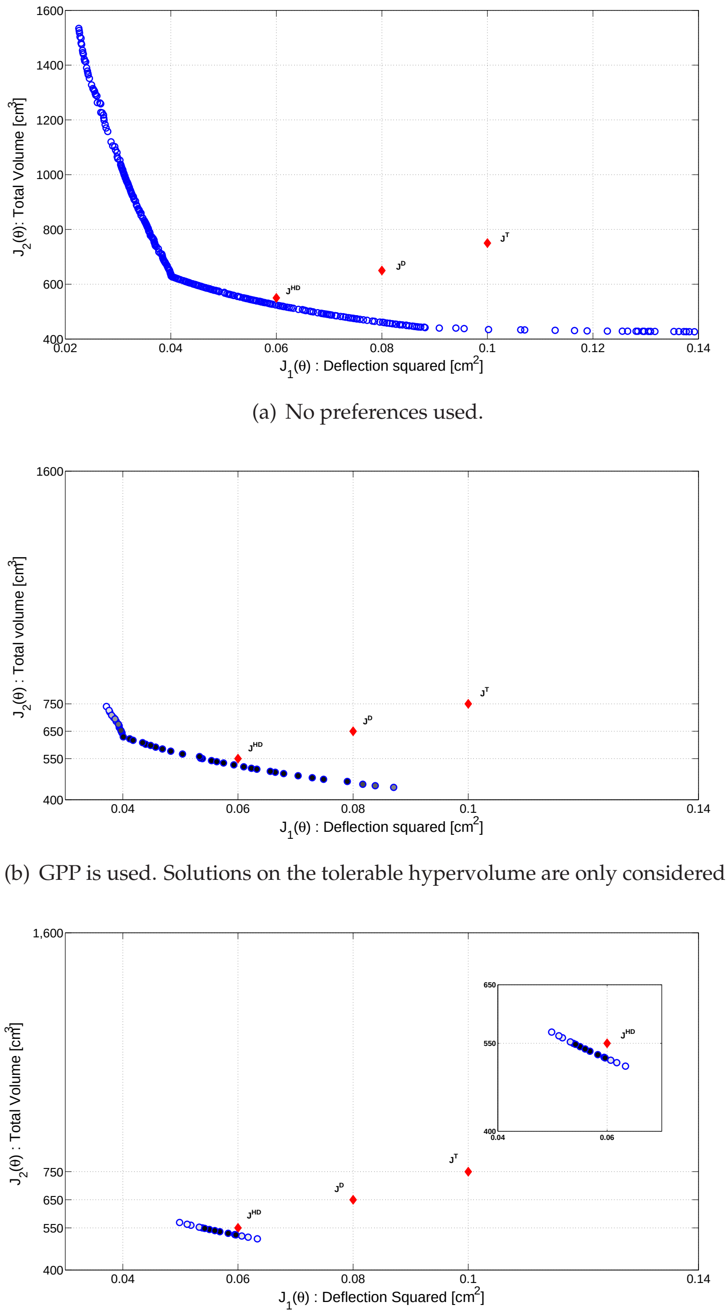
- Capacity to build a T-$\boldsymbol{J}_P^*$, with solutions with at least one objective in the desirable hypervolume (or equivalently, solutions with two objectives in the tolerable value).

Preferences stated for this problem are shown in Table 6.3. Furthermore, it will be stated that the DM has some interest to keep the objectives in the desirable zone and is willing to accept two of them (and not more) in the tolerable region; for this reason, J_{gpp}^{max} is adapted to $J_{gpp}^{max} = J_{gpp}([J_1^3, J_2^3, J_3^2])$.

In Figure 6.8, results for the following algorithm executions are shown:

- Execution with no preferences ($*$), used for comparison purposes.

- Execution to build a T-$\boldsymbol{J}_P^*$ (blue circles ○).



(a) No preferences used.

(b) GPP is used. Solutions on the tolerable hypervolume are only considered.

(c) Size control. Original grid size is always preserved

Figure 6.7: Algorithm performance in the bi-objective 3 bar truss design benchmark problem.

Table 6.3: Preferences for the DTLZ2 benchmark problem. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Set A											
	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{\theta})$	0.00		0.05		0.10		0.40		1.00		10
$J_2(\boldsymbol{\theta})$	0.00		0.30		0.40		0.60		1.00		10
$J_3(\boldsymbol{\theta})$	0.00		0.50		0.80		0.90		1.00		10

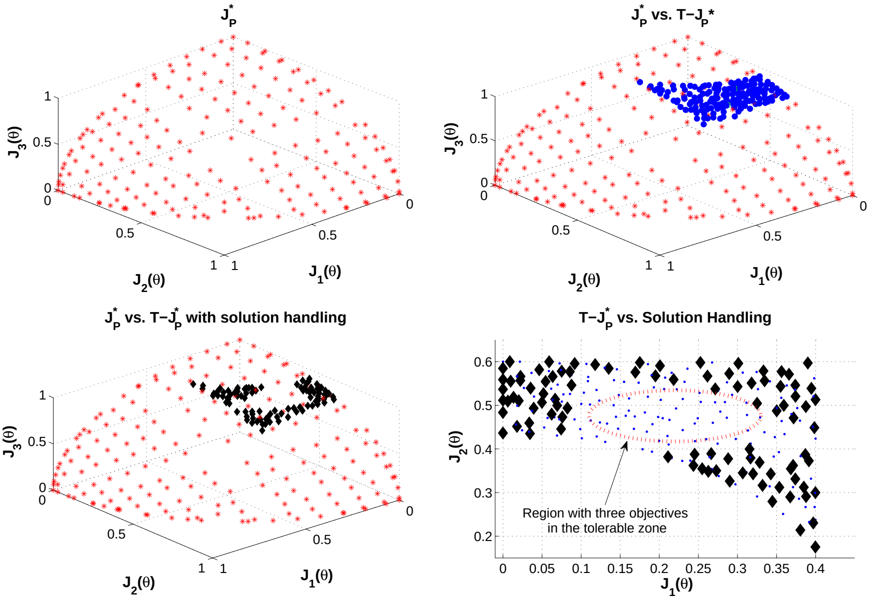


Figure 6.8: Performance of the tolerable solution handling. Pareto front approximation (red-∗), tolerable Pareto front (blue-o) and Pareto front approximation with at most two values in the tolerable zone (black-◊) are depicted.

- Execution to build a T-$\boldsymbol{J}_P^*$ with at most 2 objectives in the tolerable zone (black diamonds $\Diamond$).

Regarding the T-$\boldsymbol{J}_P^*$, the algorithm reaches the zone according to the preferences defined earlier. The (apparently) irregular distribution is owe to the selection of the most preferable solution inside each spherical sector. With regard the tolerable Pareto front with at most two objectives in the tolerable region, the algorithm is capable of avoiding solutions in the middle of the tolerable region ($J_2(\boldsymbol{\theta}) \approx [0.4, 0.5]$), which corresponds to solutions with $J_{gpp}(\varphi) > J_{gpp}([0.4, 0.6, 0.8])$ and consequently, not of interest to the DM (any solution with $J_{gpp}([0.4, 0.6, 0.8]) < J_{gpp}(\varphi) \leq J_{gpp}([0.4, 0.6, 0.9])$ has its three objectives in the tolerable region). This fact shows the flexibility of the approach to evolve its population towards different regions of interest and avoiding uninteresting regions to the DM.

The flexibility to reach regions of interest and the capacity to adapt the archive size according to the desired number of solutions will allow the algorithm to deal with many-objective optimization. A many-objective problem is then used to show the ability of the approach to deal with higher dimensions.

6.4.3 Parametric controller tuning

The following example is a parametric controller design for the control benchmark proposed at the American Control Conference (ACC) [194]. The MOP statement described in [19] is used. It has 6 objectives: robust stability ($J_1(\boldsymbol{\theta})$), maximum control effort ($J_2(\boldsymbol{\theta})$), worst case settling time ($J_3(\boldsymbol{\theta})$), noise sensitivity ($J_4(\boldsymbol{\theta})$), nominal control effort ($J_5(\boldsymbol{\theta})$) and nominal settling time ($J_6(\boldsymbol{\theta})$). Additionally, a constraint on the overshoot in the nominal model is imposed: $G_1(\boldsymbol{\theta}) < 2$. One controller structure will be evaluated:

$$C(s) = \frac{\theta_1 s^2 + \theta_2 s + \theta_3}{s^3 + \theta_4 s^2 + \theta_5 s + \theta_6} \tag{6.13}$$

Characteristics to be evaluated are:

- The implicit constraint handling mechanism with GPP and the pruning technique.
- Capacity to build **T_$\boldsymbol{J}_P^*$**, **D_$\boldsymbol{J}_P^*$** and **HD_$\boldsymbol{J}_P^*$** in a many-objective optimization framework.
- Capacity for Pareto front approximation control size.

The preference set in Table 6.4 is used. The calculated Pareto front for reference is depicted in Figure 6.9 using the LD-Tool[39] which is an application

[39] GUI for Matlab is available at: `http://www.mathworks.com/matlabcentral/fileexchange/24042`

Table 6.4: Preferences for parametric controller tuning example. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

	Preference Set A										
	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{\theta})$	-0.01		-0.005		-0.001		-0.0005		-0.0001		0
$J_2(\boldsymbol{\theta})$	0.85		0.900		1.000		1.5000		2.0000		10
$J_3(\boldsymbol{\theta})$	14.00		16.000		18.000		21.0000		25.0000		50
$J_4(\boldsymbol{\theta})$	0.50		0.900		1.200		1.4000		1.5000		10
$J_5(\boldsymbol{\theta})$	0.50		0.700		1.000		1.5000		2.0000		10
$J_6(\boldsymbol{\theta})$	10.00		11.000		12.000		14.0000		15.0000		50
$G_1(\boldsymbol{\theta})$	2.00		2.000		2.000		2.0000		5.0000		10

developed in Matlab© for Pareto front visualization using Level Diagrams [19]. With the LD-Tool, a color scheme can be used to depict the interesting solutions to the DM according to her/his preference set definition. A geometrical remark is relevant in the figure: in $J_1(\boldsymbol{\theta})$ two different and isolated regions in the Pareto front fulfil the designer's preferences.

Comparing the results from Figure 6.9 with the solution presented by the ev-MOGA algorithm in [19] and provided within the LD-Toolbox where the preference set is used *a posteriori*. While it is possible to identify the pertinent regions in the Pareto front, a lot of computational resources were used in the remainder areas of the objective space. With the usage of GPP for preferences articulation it is possible to focus the searching process in the area of interest. As a consequence in this case, the sp-MODE-II is capable of finding solutions in the **T-HypV** but the ev-MOGA is not. This is due to the use of additional information embedded in the optimization process provided by the DM.

6.4.4 Performance evaluation with other approaches

The last example is the pollution monitoring MOP stated and solved by the IBEA algorithm in [189]. It states 5 objectives, each one representing the expected information loss in 5 different monitoring stations.

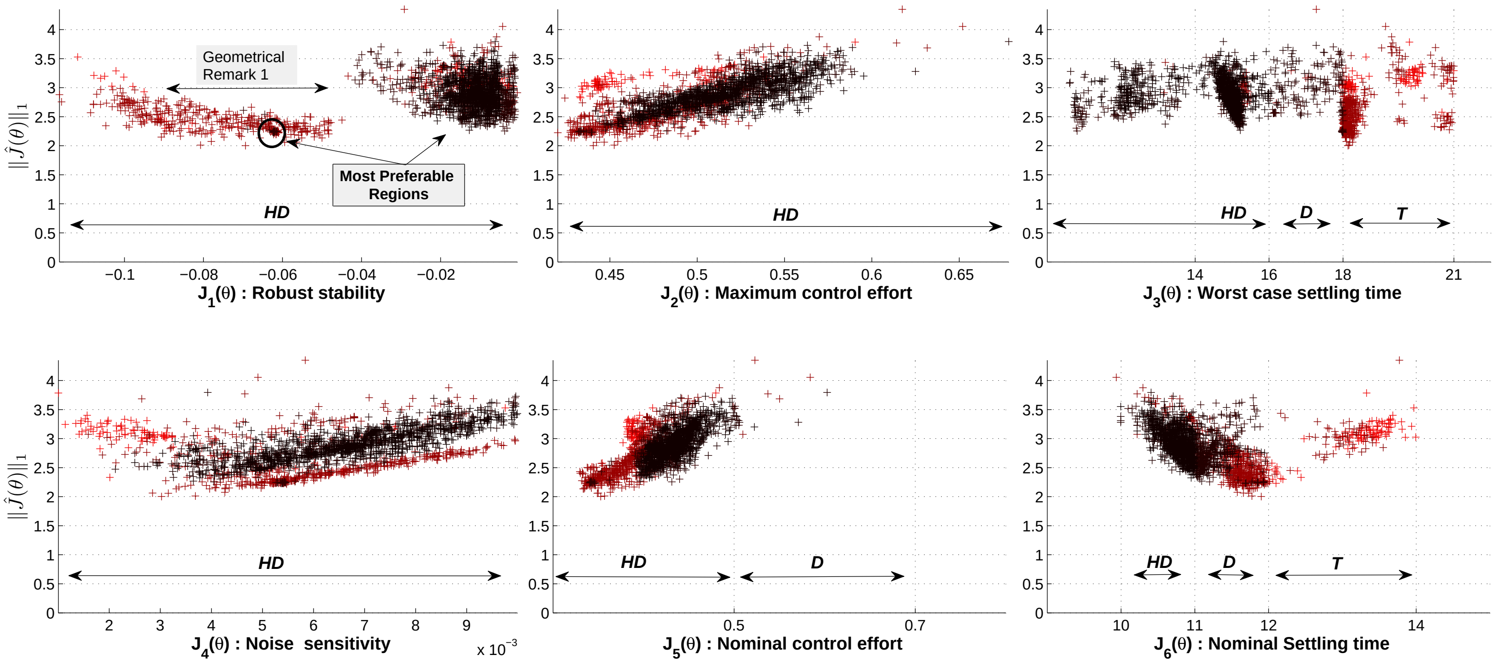


Figure 6.9: Visualization of the Preferable Pareto front of the parametric controller tuning example using Level Diagrams.

$$\begin{aligned}
\min \boldsymbol{J}(\boldsymbol{\theta}) &= [J_1(\boldsymbol{\theta}), J_2(\boldsymbol{\theta}), J_3(\boldsymbol{\theta}), J_4(\boldsymbol{\theta}), J_5(\boldsymbol{\theta})] && (6.14)\\
J_1(\boldsymbol{\theta}) &= -u_1(\theta_1,\theta_2) - u_2(\theta_1,\theta_2) - u_3(\theta_1,\theta_2) + 10 && (6.15)\\
J_2(\boldsymbol{\theta}) &= -u_1(\theta_1 - 1.2, \theta_2 - 1.5) - u_2(\theta_1 - 1.2, \theta_2 - 1.5)\\
&\quad -u_3(\theta_1 - 1.2, \theta_2 - 1.5) + 10 && (6.16)\\
J_3(\boldsymbol{\theta}) &= -u_1(\theta_1 + 0.3, \theta_2 - 3.0) - u_2(\theta_1 + 0.3, \theta_2 - 3.0)\\
&\quad -u_3(\theta_1 + 0.3, \theta_2 - 3.0) + 10 && (6.17)\\
J_4(\boldsymbol{\theta}) &= -u_1(\theta_1 - 1.0, \theta_2 + 0.5) - u_2(\theta_1 - 1.0, \theta_2 + 0.5)\\
&\quad -u_3(\theta_1 - 1.0, \theta_2 + 0.5) + 10 && (6.18)\\
J_5(\boldsymbol{\theta}) &= -u_1(\theta_1 - 0.5, \theta_2 - 1.7) - u_2(\theta_1 - 0.5, \theta_2 - 1.7)\\
&\quad -u_3(\theta_1 - 0.5, \theta_2 - 1.7) + 10 && (6.19)
\end{aligned}$$

where:

$$\begin{aligned}
u_1(\theta_1,\theta_2) &= 3(1-\theta_1)^2 \exp^{-\theta_1^2-(\theta_2+1)^2} && (6.20)\\
u_2(\theta_1,\theta_2) &= -10(\theta_1/4 - \theta_1^3 - \theta_2^5)\exp -\theta_1^2 - \theta_2^2 && (6.21)\\
u_3(\theta_1,\theta_2) &= \frac{1}{3}\exp -(\theta_1+1)^2 - \theta_2^2 && (6.22)
\end{aligned}$$

subject to:

$$\begin{aligned}
\theta_1 &\in [-4.9, 3.2] && (6.23)\\
\theta_2 &\in [-3.5, 6.0] && (6.24)
\end{aligned}$$

The IBEA algorithm is an indicator based MOEA [205], which uses the ϵ indicator [206] to evolve the entire Pareto front approximation. When using a **T_Vector**, **D_Vector** or **HD_Vector** it evolves the Pareto front approximation towards the **T_HypV**, **D_HypV** and **HD_HypV**.

Characteristics to be evaluated are:

- The performance of the strategy with respect to a state of the art technique.
- The capacity to build multiple **T_**$\boldsymbol{J}_P^*$, **D_**$\boldsymbol{J}_P^*$ and **HD_**$\boldsymbol{J}_P^*$ in many-objective optimization instances.
- The capacity for Pareto front approximation size control.

Two instances will be evaluated: a single physical matrix A and two simultaneous physical matrices B, C (Table 6.6). In each case, 2000 function evaluations are used and 50 solutions are required by the DM in $\boldsymbol{J}_P^*$. An IBEA

Table 6.5: Hypervolume achieved in 201 runs

		Single Physical Matrix		Simultaneous Physical Matrixes			
		Preference Set A		Preference B		Preference C	
		sp-MODE-II	IBEA	sp-MODE-II	IBEA	sp-MODE-II	IBEA
HD	Best	0	0	**0.1782**	0.0148	**0.3987**	0.0
	Median	0	0	**0.1241**	0.0	**0.3147**	0.0
	Worst	0	0	**0.0293**	0.0	**0.0952**	0.0
	Mean	0	0	**0.1187**	0.00007	**0.3091**	0.0
	std	0	0	**0.0308**	0.0010	**0.0549**	0.0
D	Best	**1.2533**	0.6902	**56.8633**	52.2928	**123.9829**	18.5836
	Median	**1.1586**	0.6342	**49.7769**	9.1762	**47.7898**	6.0634
	Worst	**0.5880**	0.5611	**21.0424**	0.0000	**20.8970**	0.0000
	Mean	**1.1320**	0.6324	**48.1316**	8.7044	**47.7064**	5.8899
	std	0.0971	**0.0219**	**6.2359**	6.6696	9.9826	**4.6920**
T	Best	**260.0377**	259.1223	515.4334	**585.4359**	**1.9292E+003**	**1.7819E+003**
	Median	146.6786	**242.2504**	429.8147	**461.2593**	1.3083E+003	**1.5312E+003**
	Worst	123.8105	**216.6591**	218.4111	**257.7210**	**1.0425E+003**	0.8677E+003
	Mean	184.3619	**242.0966**	423.7853	**456.3398**	1.3138E+003	**1.5044E+003**
	std	**4.7779**	7.2756	51.6358	**40.5184**	0.1173E+003	**0.1073E+003**

Table 6.6: Preferences Set A for example 4. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Set A

	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{\theta})$	6		7		9		10		11		12
$J_2(\boldsymbol{\theta})$	6		7		8		10		11		12
$J_3(\boldsymbol{\theta})$	6		7		9		10		11		12
$J_4(\boldsymbol{\theta})$	6		7		8		10		11		12
$J_5(\boldsymbol{\theta})$	6		7		9		10		11		12

Preference Set B

	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{\theta})$	5		10		11		12		13		15
$J_2(\boldsymbol{\theta})$	5		9		10		11		12		15
$J_3(\boldsymbol{\theta})$	5		8		9		10		11		15
$J_4(\boldsymbol{\theta})$	5		7		8		9		10		15
$J_5(\boldsymbol{\theta})$	5		6		7		8		9		15

Preference Set C

	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{\theta})$	5		8		9		10		14		15
$J_2(\boldsymbol{\theta})$	5		8		9		10		14		15
$J_3(\boldsymbol{\theta})$	5		11		12		13		14		15
$J_4(\boldsymbol{\theta})$	5		11		12		13		14		15
$J_5(\boldsymbol{\theta})$	5		11		12		13		14		15

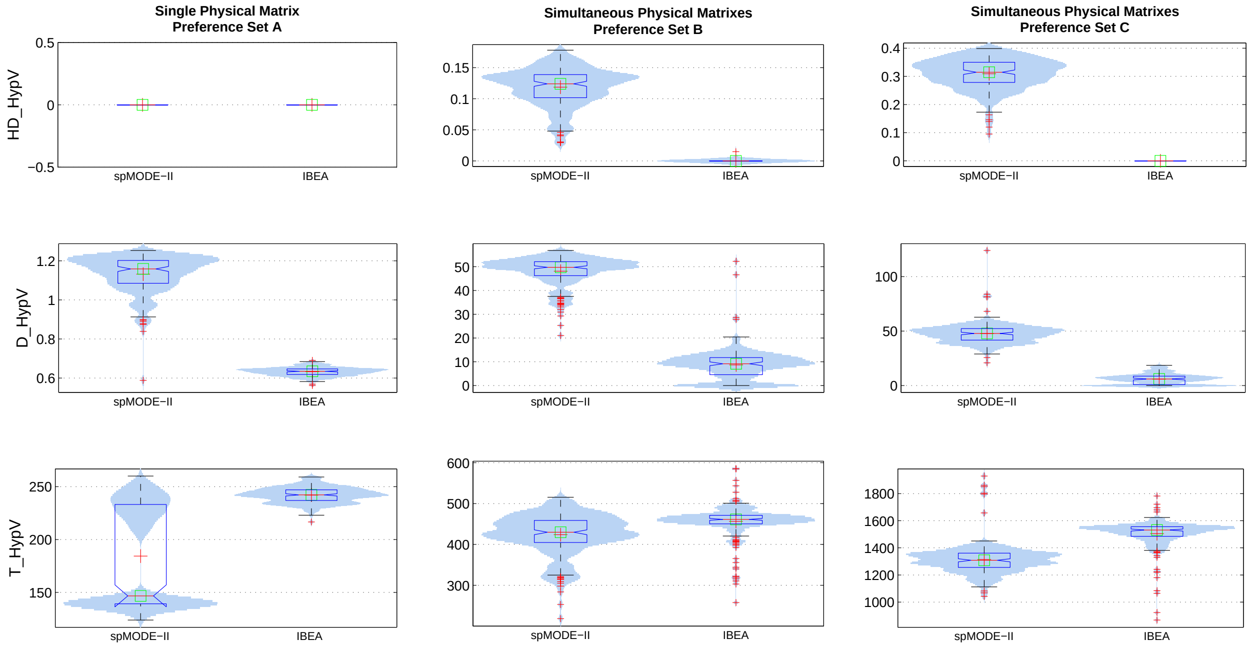


Figure 6.10: Hypervolume achieved in 201 runs.

using DE algorithm ($\rho = 0.05$) has been coded for this purpose. For both algorithms values of $F = 0.5$ and $CR = 0.9$ are defined and 201 independent runs are carried out.

In Table 6.5 the results of 201 independent runs for the best, median, worst and mean values of the achieved **HD_HypV**, **D_HypV** and **T_HypV** are depicted. Statistical differences have been validated with Wilcoxon test at 95% significance level [47]. Regarding instance 1, none of the algorithms is capable of reaching the **HD_HypV**. The sp-MODE-II algorithm is able to achieve a better performance than IBEA for **D_HypV**, at expenses of the **T_HypV**. Regarding instance 2, only the sp-MODE-II algorithm is capable of finding solutions in the **HD_HypV** for both preference sets. Again, sp-MODE-II presents the same behavior when compared with IBEA for **D_HypV** and **T_HypV**. This is due to the implicit pressure on $J_{gpp}(\boldsymbol{\theta})$ index to continue improving the solutions obtained from the **T_HypV** to the **HD_HypV**.

Attainment surfaces for comparison ($\Diamond$ for sp-MODE-II and x for IBEA) are depicted in Figure 6.11 as described in [151] for MOEAs comparison in many-objective optimization instances by means of Level Diagrams[40]. Taxonomy to identify the visualizations is adopted from the same reference[41]. It is possible to notice the following remarks on Figure 6.11:

Remark 1: In instance 1 (single preference condition) the 50% attainment surface of the sp-MODE-II algorithm fails to converge at the **D_HypV**. This impossibility is due to Objective 1: the algorithm fails to handle this objective to reach the **T_**$\boldsymbol{J}_P^*$. As a consequence, the IBEA algorithm dominates this attainment surface portion.

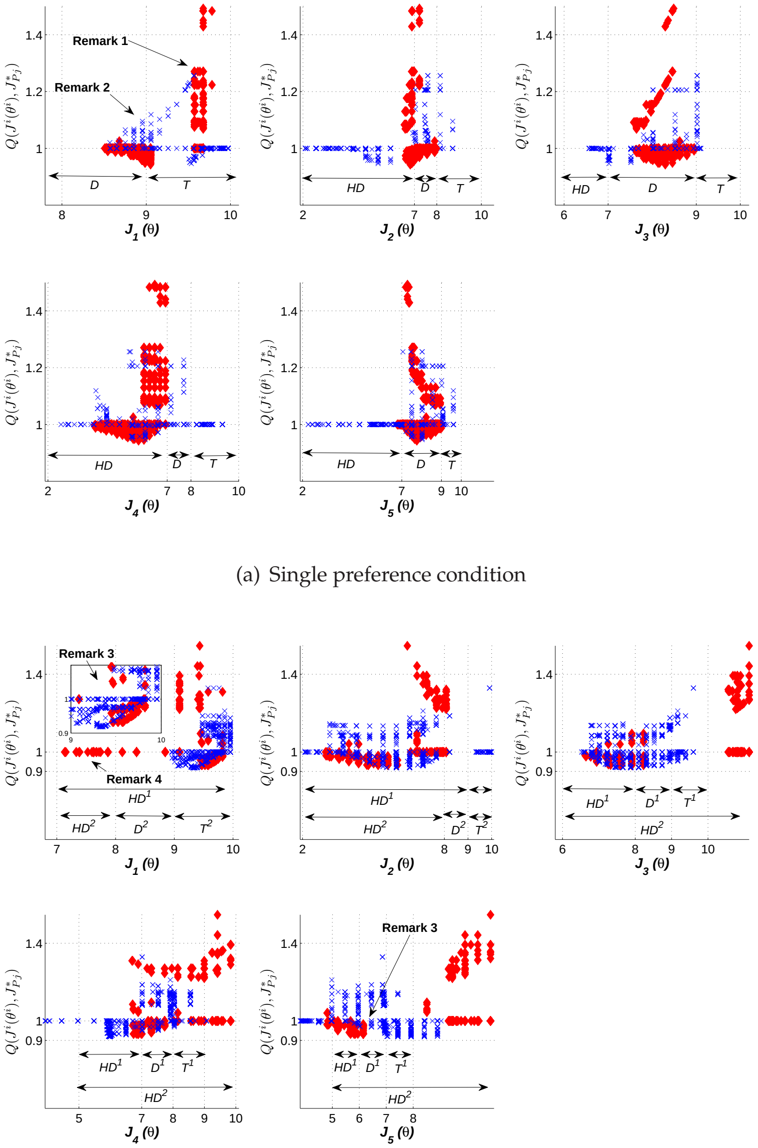
Remark 2: In the same case, when the 50% attainment surface of the sp-MODE-II algorithm converges at the **D_HypV**, it tends to dominate the IBEA algorithm.

Remark 3: According to instance 2 (simultaneous preference conditions), there is a change in the covering for $J_1(\boldsymbol{\theta})$ and $J_5(\boldsymbol{\theta})$ (Figure 6.11b). That could indicate structural differences between algorithms, since one of them dominates the other in one of the objectives, at the price of being dominated in the other.

Remark 4: According to Table 6.5, the IBEA is capable of reaching the **HD_HypV** of the preference set B, but incapable of reaching the **HD_HypV** of the preference set C. In Figure 6.11b, it is possible to appreciate this fact and it is possible to specify where it happens: in Objective $J_1(\boldsymbol{\theta})$ the IBEA fails to reach the **HD_HypV** of preference set C.

[40] Available for Matlab at: `http://www.mathworks.com/matlabcentral/fileexchange/39458`

[41] *LD/front/measure*. For example, LD/$\boldsymbol{J}_p^*/\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$, means that a visual representation of Pareto front approximation $\boldsymbol{J}_p^*$ with 2-norm in LD is presented.



(a) Single preference condition

(b) Simultaneous preference conditions

Figure 6.11: LD/$\left\{\widetilde{J^*_{p1}}^{50\%}, \widetilde{J^*_{p2}}^{50\%}\right\}$/Q $\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{pj}\right)$ visualization for attainment surface performance. Red $\Diamond$ for sp-MODE-II and blue x for IBEA are used. Detailed explanation on the remarks is given within the thesis.

6.5 Conclusions on this chapter

In this chapter, a mechanism for handling preferences in EMO using GPP has been presented. Techniques for preference handling could be a desirable characteristic for a MOEA, since they improve pertinency of solutions. Therefore, it could make easier the DM's task of analyzing and selecting a preferable solution from the approximated Pareto front. The embedment of the articulation of preferences in a MOEA could reinforce the holistic MOOD procedure.

The approach developed fulfills the requirements for preferences coding in an intuitive and meaningful scheme in order to generate pertinent solutions. As a consequence, pertinent approximations of the Pareto front are built. Furthermore, the approach has shown to be effective for constrained and many-objective optimization statements.

According to the examples provided, a competitive algorithm has been developed. Although it has been presented as a particular algorithm, the technique could be potentially applied to any EMO approach. The developed algorithm (sp-MODE-II) is flexible to articulate preferences and is DM-oriented since it provides features for multiple preference sets, size control of the approximated front (based on preferences) and capacity to handle the level of tolerance of a design alternative.

The summary of using this framework is:

1. To code the DM's preferences in an intuitive and meaningful language.
2. To include a constraint handling mechanism into the optimization statement.
3. The possibility to refine the pruning mechanism, specifying which solutions are not allowed to remain in the spherical grid, according the DM's preferences.
4. The possibility to create a control mechanism for the quantity of solutions required in the Pareto front approximation, with regard for designer's interesting solutions.

Part IV

Final Contributions on Controller Tuning

PART IV:

Final Contributions on Controller Tuning

This part is devoted to present final contributions using the MOOD procedure for controller tuning. Such contributions focus again on multivariable processes (identified as a potential research area in Chapter 2) and on many-objective optimization instances. Basic ideas from Part II (Preliminary Contributions on Controller Tuning) are used, and the limitations found on MOOD tools implemented are amended with the developments from Part III (Contributions on MOOD Tools). Firstly, a sampling mechanism explicitly designed for PID controllers will be stated; secondly, it is used together with the ideas and tools from Parts II and III for multivariable controller tuning.

CHAPTER 7:

Stochastic sampling of stable PID controllers

Contributions on MOOD tools presented in Part III can be used straightforwardly in controller tuning applications; nevertheless it is worthwhile to develop explicit mechanisms to close the gap between them and EMO, before facing specific engineering control problems. In this chapter a stochastic sampling procedure for PID controllers is proposed, to guarantee that (1) any sampled controller will stabilize the closed loop and (2) any stabilizing controller could be sampled. Contents of this chapter appear in the following conference paper:

- **G. Reynoso-Meza, J. Sanchis, X. Blasco and J.M. Herrero**. *A stabilizing PID controller sampling procedure for stochastic optimizers*. The 19th World Congress of the International Federation of Automatic Control. Accepted on February 13th., 2014.

7

Stochastic sampling of stable PID controllers

Control, control, you must learn control!

Grand master Yoda

7.1 Aim of this chapter

In general, stochastic optimizers are characterized by the randomness used in the search process, which is helpful to avoid local minima. Evolutionary or nature inspired algorithms are very popular stochastic optimizers and they have been used extensively for PID-like controller tuning [152]. Such algorithms use matching and mutation operators to induce such randomness into the search process. Nevertheless, in the case of PID controller tuning, this process has to be carefully performed; besides the problematic (and often uncommented) preliminary step of defining the boundaries in the search process, not all parameter combinations in a PID guarantees closed loop stability given a nominal process.

The specific aim of this chapter is:

- To develop and implement a simple sampling procedure for stochastic optimizers with a focus on first order plus dead time (FOPDT) processes, in order to generate randomly PID parameters which guarantee closed loop stability.

This procedure not only assures sampling stabilizing PID parameters but also guarantees that all stabilizing controllers are taken into account. Such sampling could be potentially used for any stochastic optimizer saving computational resources to achieve an optimal solution (given a cost function) and improving its convergence. The remainder of this work is as follows: in section 2 it is presented the sampling procedure, while in section 3 it is validated its usefulness with two different test cases. Finally, some concluding remarks are given.

7.2 Sampling Procedure

Firstly, it will be briefly explained how is characterized the subset of stabilizing PID controllers $\mathbb{C}$ and afterwards, how to sample controller from such subset.

7.2.1 Computation of stable set $\mathbb{C}$

In this chapter FOPDT transfer functions $P(s)$ and PID controllers $C(s)$ are considered. Both are described by the equations:

$$P(s) = \frac{k}{Ts+1}e^{-Ls} \tag{7.1}$$

$$C(s) = k_p + \frac{k_i}{s} + k_d s \tag{7.2}$$

where k is the process proportional gain, T the time constant and L the lag of the system; k_p, k_i, k_d the proportional, integral and derivative gains of the controller. According to [169], the set $\mathbb{C} \subset \mathbb{R}^3$ of stabilizing controllers given the $P(s)$ process is given by subsets $\mathbf{T}$, $\mathbf{\Delta}$ and $\mathbf{Q}$ (Figure 7.1).

The range of k_p values for which $P(s)$ is stable is given by:

$$-\frac{1}{k} < k_p < k_u \tag{7.3}$$

where k_u is usually known as the ultimate gain

$$k_u = \frac{1}{k}\left[\frac{T}{L}\alpha_1 \sin(\alpha_1) - \cos(\alpha_1)\right] \tag{7.4}$$

and α_1 is the solution of the equation

$$\tan(\alpha) = -\frac{T}{T+L}\alpha, \alpha \in [0, \pi] \tag{7.5}$$

The complete stabilization region is given by:

1. For each $k_p \in [-\frac{1}{k}, \frac{1}{k})$, the corresponding region in (k_i, k_d) space is the quadrilateral $\boldsymbol{Q}$ of Figure 7.1.

2. For $k_p = \frac{1}{k}$, the corresponding region in (k_i, k_d) space is the triangle $\mathbf{\Delta}$ of Figure 7.1.

3. For each $k_p \in [\frac{1}{k}, k_u)$, the corresponding region in (k_i, k_d) space is the trapezoid $\boldsymbol{T}$ of Figure 7.1, where $k_u = \frac{1}{k}\left[\frac{T}{L}\alpha_1 \sin(\alpha_1) - \cos(\alpha_1)\right]$.

Where relevant variables m_j, b_j, w_j, $j = [1, 2]$ are calculated as follows:

$$m_j = \frac{L^2}{z_j^2} \tag{7.6}$$

$$b_j = -\frac{L}{k \cdot z_j}\left[\sin(z_j) + \frac{T}{L} z_j \cos(z_j)\right] \tag{7.7}$$

$$w_j = \frac{z_j}{kL}\frac{(1 + kk_p)[1 + \cos(z_j)]}{\sin(z_j)} \tag{7.8}$$

with z_1, z_2 being the roots of

$$k \cdot k_p + \cos(z) - \frac{T}{L} z \sin(z) = 0 \tag{7.9}$$

Therefore, set $\mathbb{C}$ is composed by subsets $\mathbf{T}$ (for $k_p \in [-\frac{1}{k}, \frac{1}{k})$), $\boldsymbol{\Delta}$ (for $k_p = \frac{1}{k}$) and $\bigcup \mathbf{Q}$ (for $k_p \in [\frac{1}{k}, k_u)$).

7.2.2 Sampling from stable set $\mathbb{C}$

It is stated that, given a subset $\mathbb{S} \in \mathbb{R}^3$ and a process $P(s)$, two important features for PID $C(s)$ parameter sampling from $\mathbb{S}$ should be fulfilled:

1. Any sampled controller $C(s)$ from $\mathbb{S}$ must stabilize the closed loop.
2. Any stabilizing controller $C(s)$ of the process $P(s)$ must be contained in $\mathbb{S}$.

A common approach for feature 1 is to define bounds on the parameter which avoid all non-stable but also some stable PID parameters; therefore, feature 2 is not fulfilled. A second alternative, is to bound the search space with all stable PID parameters, but including non-stable parameters, which are verified while the algorithm is running. In such case, feature 1 is unfulfilled, and computational (CPU time) resources could be misspent.

To sample stabilizing controllers fulfilling both features, it is required to sample on $\mathbb{C}$. It is assumed that controllers matching $k_p > 0$, $ki > 0$, $k_d > 0$ are the only accepted by the designer to stabilize $P(s)$. According to this an injective function $\mathbb{R}^3 \to \mathbb{R}^3$ to map from the unitary cube to $\mathbb{C}$ is defined:

$$\begin{aligned} F(a, b, c) &= [k_p, k_i, k_d] \\ \{a, b, c\} &\in [0, 1] \end{aligned} \tag{7.10}$$

Where:

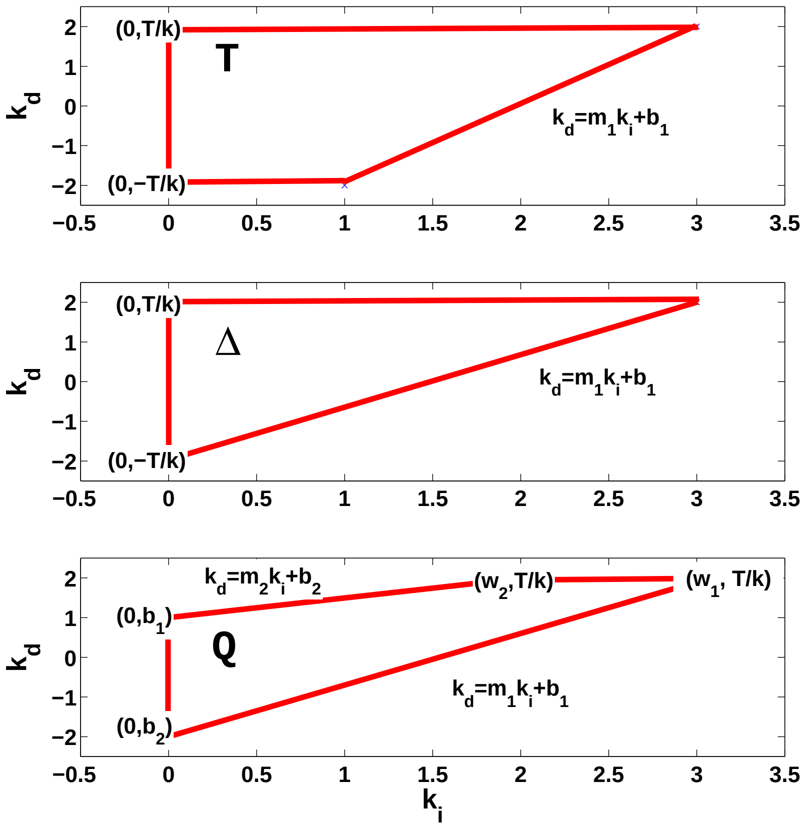


Figure 7.1: PID stable regions.

$$k_p = a \cdot k_u \tag{7.11}$$

$$k_d = c \cdot \frac{T}{K} \tag{7.12}$$

$$k_i = \begin{cases} b \cdot \frac{k_d - b_1}{m_1} & if \quad k_p \leq 1/k \\ \frac{k_d - b_2}{m_2} + b \cdot \left[\frac{k_d - b_1}{m_1} - \frac{k_d - b_2}{m_2}\right] & if \quad 1/k < k_p \end{cases} \tag{7.13}$$

The algorithm to generate stabilizing controllers is depicted in Algorithm 10.

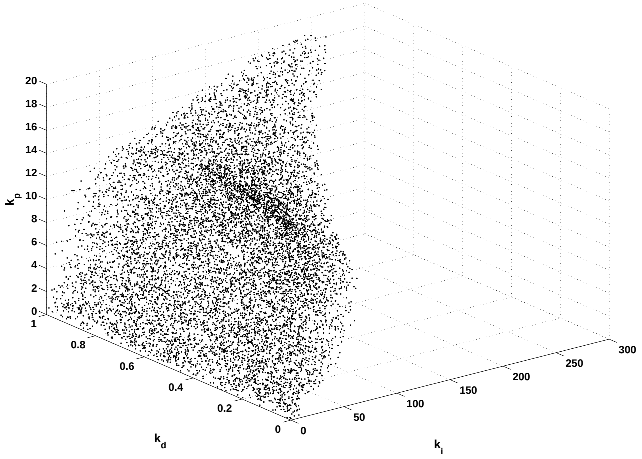
Simple analytic solutions using Newton-Raphson optimization method could be employed. Line 1 and 2 could be calculated *off-line* if a fixed FOPDT is under consideration. If not required, line 10 could be omitted. In figure 7.2 a

```
Read values $k, T, L$;
Calculate $k_u$ (Equation (7.4));
Read values $a, b, c$;
Calculate $k_p$ (Equation (7.11));
Calculate $k_d$ (Equation (7.12));
Determine roots $z_j, j = [1, 2]$ (Equation (7.9));
Determine values $m_j, b_j, j = [1, 2]$ (Equations (7.6),(7.7));
Determine $k_i$ (Equation (7.13));
Return $k_p, k_i, k_d$;
Return $T_i = k_p/k_i$ $T_d = k_d/k_p$;
```

Algorithm 7.1: Stable PID stochastic sampling.

simple random sampling using Algorithm is shown.

Next, it will be used this approach for several stochastic optimizers to evaluate its performance.



(a) Stabilizing controllers in (k_i, k_d, k_p) space.

Figure 7.2: Stabilizing controllers for system $\frac{1}{s+1}e^{-0.1s}$ (1e6 samples are plotted) from stochastic sampling in the unitary cube.

7.3 Performance Evaluation

Three different stochastic optimizers to evaluate the performance of the sampling will be used. In this chapter, it will be focused on single objective optimization, due to limitation of space. The optimizers under consideration are:

DE: Differential Evolution algorithm [41, 182, 183]. It belongs to the family of evolutionary strategies, and it is a very popular and compact algorithm. A version available from Matlab central is used.[42]

SA: Simulated annealing algorithm ([184]). A heuristic optimizer emulating annealing process in materials for search in the decision space. The version available in the Matlab optimization toolbox is used.

CMAES: Covariance Matrix Adaptation Evolution Strategy ([75]). An algorithm from the family of estimation of distribution algorithms. The version available from the authors for Matlab is used.[43]

In all cases, algorithms have been downloaded from their respective sources and used with their standard parameters. That is, no further effort on tuning algorithm parameters has been spent.

The process under consideration is the benchmark for PID control 2012 described by [133]. It proposes a boiler control problem [132, 59] based on the work of [141]. This work improves the model provided in [14] by adding a non-linear combustion equation with a first order lag to model the excess oxygen in the stack and the stoichiometric *air-to-fuel* ratio for complete combustion. The non-linear explicit model is described by the following equations:

$$\dot{\boldsymbol{x_1}}(t) = c_{11}\boldsymbol{x_4}(t)\boldsymbol{x_1}^{\frac{9}{8}} + c_{12}\boldsymbol{u_1}(t-\tau_1) - c_{13}\boldsymbol{u_3}(t-\tau_3) \tag{7.14}$$

$$\begin{aligned}\dot{\boldsymbol{x_2}}(t) = {} & c_{21}\boldsymbol{x_2}(t) \\ & + \frac{c_{22}\boldsymbol{u_2}(t-\tau_2) - c_{23}\boldsymbol{u_1}(t-\tau_1) - c_{24}\boldsymbol{u_1}(t-\tau_1)\boldsymbol{x_2}(t)}{c_{25}\boldsymbol{u_2}(t-\tau_2) - c_{26}\boldsymbol{u_1}(t-\tau_1)}\end{aligned} \tag{7.15}$$

$$\dot{\boldsymbol{x_3}}(t) = -c_{31}\boldsymbol{x_1}(t) - c_{32}\boldsymbol{x_4}(t)\boldsymbol{x_1}(t) + c_{33}\boldsymbol{u_3}(t-\tau_3) \tag{7.16}$$

$$\dot{\boldsymbol{x_4}}(t) = -c_{41}\boldsymbol{x_4}(t) + c_{42}\boldsymbol{u_1}(t-\tau_1) + c_{43} + \boldsymbol{n_5}(t) \tag{7.17}$$

$$\boldsymbol{y_1}(t) = c_{51}\boldsymbol{x_1}(t-\tau_4) + \boldsymbol{n_1}(t) \tag{7.18}$$

$$\boldsymbol{y_2}(t) = c_{61}\boldsymbol{x_1}(t-\tau_5) + \boldsymbol{n_2}(t) \tag{7.19}$$

$$\begin{aligned}\boldsymbol{y_3}(t) = {} & c_{70}\boldsymbol{x_1}(t-\tau_6) + c_{71}\boldsymbol{x_3}(t-\tau_6) \\ & + c_{72}\boldsymbol{x_4}(t-\tau_6)\boldsymbol{x_1}(t-\tau_6) + c_{73}\boldsymbol{u_3}(t-\tau_3-\tau_6) \\ & + c_{74}\boldsymbol{u_1}(t-\tau_1-\tau_6) \\ & + \frac{[c_{75}\boldsymbol{x_1}(t-\tau_6) + c_{76}]\,[1 - c_{77}\boldsymbol{x_3}(t-\tau_6)]}{\boldsymbol{x_3}(t-\tau_6)\,[\boldsymbol{x_1}(t-\tau_6) + c_{78}]} \\ & + c_{79} + \boldsymbol{n_3}(t)\end{aligned} \tag{7.20}$$

$$\boldsymbol{y_4}(t) = [c_{81}\boldsymbol{x_4}(t-\tau_7) + c_{82}]\,\boldsymbol{x_1}(t-\tau_7) + \boldsymbol{n_4}(t) \tag{7.21}$$

[42] `http://www.mathworks.com/matlabcentral/fileexchange/38962`

[43] `https://www.lri.fr/~hansen/cmaesintro.html`

Table 7.1: Default parameters of the SA algorithm. These values are obtained by executing in Matlab© R2009b the command line `options=saoptimset(’simulannealbnd’)`.

AnnealingFcn:	@annealingfast
TemperatureFcn:	@temperatureexp
AcceptanceFcn:	@acceptancesa
TolFun:	1.0000e-006
StallIterLimit:	’500*numberofvariables’
MaxFunEvals:	’3000*numberofvariables’
TimeLimit:	Inf
MaxIter:	Inf
ObjectiveLimit:	-Inf
Display:	’final’
DisplayInterval:	10
HybridFcn:	[]
HybridInterval:	’end’
PlotFcns:	[]
PlotInterval:	1
OutputFcns:	[]
InitialTemperature:	100
ReannealInterval:	100
DataType:	’double’

Table 7.2: Default parameters of the CMAES algorithm (Version 3.54.beta). These values are obtained by executing in Matlab© R2009b the command line `options=cmaes`.

StopFitness:	'-Inf stop if f(xmin) < stopfitness, minimization'
MaxFunEvals:	'Inf maximal number of fevals'
MaxIter:	'1e3*(N5)2/sqrt(popsize) maximal number of iterations'+
StopFunEvals:	'Inf stop after resp. evaluation, possibly resume later'
StopIter:	'Inf stop after resp. iteration, possibly resume later'
TolX:	'1e-11*max(insigma) stop if x-change smaller TolX'
TolUpX:	'1e3*max(insigma) stop if x-changes larger TolUpX'
TolFun:	'1e-12 stop if fun-changes smaller TolFun'
TolHistFun:	'1e-13 stop if back fun-changes smaller TolHistFun'
StopOnStagnation:	'on stop when fitness stagnates for a long time'
StopOnWarnings:	'yes 'no'=='off'==0, 'on'=='yes'==1 '
StopOnEqualFunctionValues:	'2 N/3 number of iterations'+
DiffMaxChange:	'Inf maximal variable change(s), can be Nx1-vector'
DiffMinChange:	'0 minimal variable change(s), can be Nx1-vector'
WarnOnEqualFunctionValues:	'yes 'no'=='off'==0, 'on'=='yes'==1 '
LBounds:	'-Inf lower bounds, scalar or Nx1-vector'
UBounds:	'Inf upper bounds, scalar or Nx1-vector'
EvalParallel:	'no objective function FUN accepts NxM matrix, with M>1?'
EvalInitialX:	'yes evaluation of initial solution'
Restarts:	'0 number of restarts '
IncPopSize:	'2 multiplier for population size before each restart'
PopSize:	'(4 floor(3*log(N))) population size, lambda'+
ParentNumber:	'floor(popsize/2) AKA mu, popsize equals lambda'
RecombinationWeights:	'superlinear decrease or linear, or equal'
DiagonalOnly:	'0*(1100*N/sqrt(popsize)) C is diagonal for given iterations, 1==always'+
Noise:	[1x1 struct]
CMA:	[1x1 struct]
Resume:	'no resume former run from SaveFile'
Science:	'on off==do some additional (minor) problem capturing, NOT IN USE'
ReadSignals:	'on from file signals.par for termination, yet a stumb'
Seed:	'sum(100*clock) evaluated if it is a string'
DispFinal:	'on display messages like initial and final message'
DispModulo:	'100 [0:Inf], disp messages after every i-th iteration'
SaveVariables:	'on [on|final|off][-v6] save variables to .mat file'
SaveFilename:	'variablescmaes.mat save all variables, see SaveVariables'
LogModulo:	'1 [0:Inf] if >1 record data less frequently after gen=100'
LogTime:	'25 [0:100] max. percentage of time for recording data'
LogFilenamePrefix:	'outcmaes files for output data'
LogPlot:	'off plot while running using output data files'
UserData:	'for saving data/comments associated with the run'
UserDat2:	''

Where $\boldsymbol{\dot{x}_1}(t)$, $\boldsymbol{\dot{x}_2}(t)$, $\boldsymbol{\dot{x}_3}(t)$, $\boldsymbol{\dot{x}_4}(t)$ are the states of the system; $\boldsymbol{y_1}(t)$, $\boldsymbol{y_2}(t)$, $\boldsymbol{y_3}(t)$, $\boldsymbol{y_4}(t)$ the observed states; c_{ij}, τ_i and $\boldsymbol{n_i}$ are non-linear coefficients, time constants and noise models, respectively, determined to improve the accuracy oh the model.

The benchmark problem stated a 3x3 MIMO system with a measured load disturbance:

$$\begin{aligned}\begin{bmatrix} Y_1(s) \\ Y_2(s) \\ Y_3(s) \end{bmatrix} &= \begin{bmatrix} P_{11}(s) & 0 & P_{13}(s) \\ P_{21}(s) & P_{22}(s) & 0 \\ P_{31}(s) & 0 & P_{33}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \\ U_3(s) \end{bmatrix} \\ &+ \begin{bmatrix} P_{1d}(s) \\ 0 \\ P_{3d}(s) \end{bmatrix} D(s) \end{aligned} \tag{7.22}$$

Where the inputs are fuel flow [%], air flow [%] and water flow [%] while the outputs are steam pressure [%], oxygen level [%] and water level [%]. This is a verified model, useful to propose, evaluate and compare different kinds of tuning/control techniques [69, 163, 137, 170, 160].

For such benchmark a reduced MIMO and SISO version were available to evaluate different controller tuning procedures. For this chapter only the SISO PID controller tuning (Figure 7.3), where the fuel flow is manipulated to control the steam pressure, is considered.

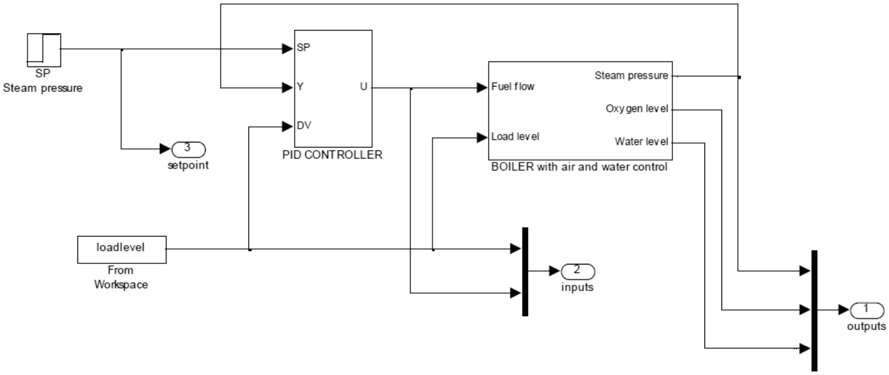


Figure 7.3: SISO loop for Boiler benchmark (taken from [133]).

The benchmark also defined an index ($I_{benchmark}$) to evaluate the performance for a given controller. It is an aggregate objective function, which combines ratios of IAE, ITAE and IAVU indexes using a base case PI controller

($[k_p, T_i] = [2.5, 50]$). More details can be consulted in [133]. The benchmark index is used to tune a PID controller using an identified model $G(s)$[44] to be evaluated later in the original model.

$$G(s) = \frac{0.3934}{45.6794s + 1} e^{-3.42s} \tag{7.23}$$

7.3.1 Performance test 1: Convergence improvement

The aim of the first test design is to validate the hypothesis that the stochastic sampling will improve each algorithm performance to calculate optimal PID parameters[45]. For this purpose, 2000 function evaluations have been used for each algorithm and 51 independent runs are carried out. Each one will be executed twice, in order to test the sampling proposal:

1. An execution using standard bounds on k_p, T_i, T_d.
2. An execution using the stochastic sampling previously explained.

Therefore, the following optimization problem is defined for the former instance:

$$\begin{aligned}
\min J(k_p, T_i, T_d) &= I_{benchmark}|_{G(p(s)} && (7.24)\\
&s.t. && \\
k_p &\in \left(0,\, k_u|_{G(s)}\right] && (7.25)\\
T_i &\in [0, 100] && (7.26)\\
T_d &\in [0, 100] && (7.27)\\
Re[\lambda] &< 0 && (7.28)
\end{aligned}$$

while for the latter:

$$\begin{aligned}
\min J(a, b, c) &= I_{benchmark}|_{G(p(s)} && (7.29)\\
&s.t. && \\
a &\in (0, 1] && (7.30)\\
b &\in [0, 1] && (7.31)\\
c &\in [0, 1] && (7.32)
\end{aligned}$$

In the former case, a basic penalty function is used to identify feasible from non-feasible solutions. The results are shown in Tables 7.3, 7.4 and 7.5. In the

[44] This model was obtained with a step response experiment using the identification toolbox of Matlab©.

[45] Optimal according to the selected cost function $I_{benchmark}$.

case of the DE algorithm (Table 7.3), there are no statistical differences according to the Wilcoxon test at 95% [47], although it can be notice that DE algorithm without the coding is not capable of finding a suitable controller in all executions. Regarding SA and CMA-ES algorithms (Tables 7.4 and 7.5) the Wilcoxon test reveals that the instance using the sampling proposed leads to better results on both cases. In each instance, algorithms seem to be converging to a PID controller with parameters $[k_p, T_i, T_d] = [3.99, 29.41, 0]$. That is, a controller without derivative gain. The performance of such controller and its comparison with the reference case (using files and guidelines from the benchmark) are depicted in Figure 7.4.

Table 7.3: Performance of DE algorithm in test 1.

	Stochastic Sampling	Standard bounds
minimum	**0.6247**	**0.6247**
median	**0.6448**	0.6480
maximum	**22.0645**	56.4709
mean	**3.4526**	6.7177
variance	**37.4191**	167.2447
success	**51**	47

Table 7.4: Performance of SA algorithm in test 1.

	Stochastic Sampling	Standard bounds
minimum	**0.6247**	0.6255
median	**0.6248**	0.6943
maximum	**0.6847**	0.9726
mean	**0.6265**	0.7415
variance	**0.0002**	0.0350
success	**51**	**51**

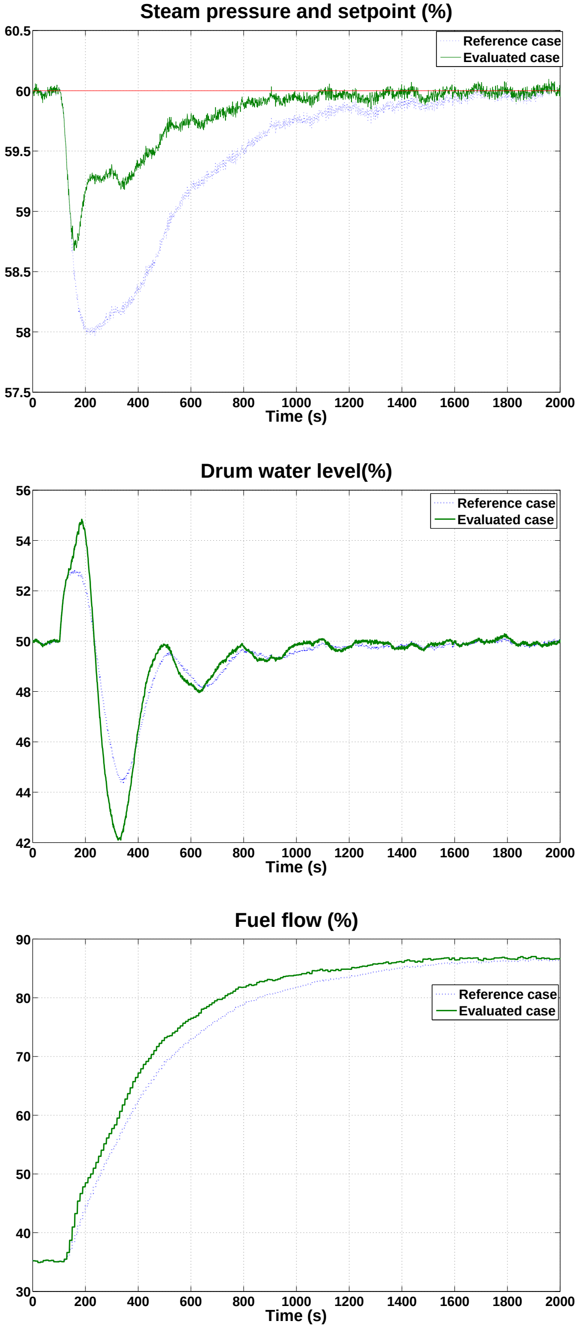


Figure 7.4: Performance of the PI controller $[k_p, T_i] = [3.99, 29.41]$ *and its comparison with the reference case* $[k_p, T_i] = [2.5, 50]$.

Table 7.5: Performance of CMA-ES algorithm in test 1.

	Stochastic Sampling	Standard bounds
minimum	**0.6247**	**0.6247**
median	**0.6247**	0.6255
maximum	**14.9956**	18.3715
mean	**0.9166**	1.3141
variance	**4.0489**	11.8571
success	**51**	**51**

7.3.2 Performance test 2: multidisciplinary optimization

The aim of the second test design is to validate

- The efficacy of the proposal, due to a more efficient number of function evaluations required.
- The usefulness on this sampling for multidisciplinary optimization design.

A PI for the aforementioned process will be tuned. Nevertheless, it will be assumed a design phase for the boiler, where it is decided to carry a multidisciplinary design approach of plant design and control. As detailed in [161] and [24], multidisciplinary optimization approaches could bring new and interesting solutions for designer, since it shows a better performance than sequential design procedures. This approach has been used before with success involving PI and PID controllers ([10, 13, 92, 103]).

It is assumed a hypothetic case where from the nominal model identified, the plant could be (re)designed as:

$$G'(s) = f(l) = \frac{0.3934 \cdot (4 - 3l)}{45.6794 \cdot (4 - 3l)s + 1} e^{-3.42s} \tag{7.33}$$

with free parameter $l \in [0.7, 1.3]$. In a physical sense, with a fixed step reference change it means that:

- it is possible to (re)design a plant with a quicker response (*i.e.* , with a lower time constant) but at the expense of more energy required to get to the reference (*i.e.* , a lower gain).
- it is possible to (re)design a plant with a lower response (*i.e.* , with a high time constant) that requires less energy to get to the reference (i.e. a bigger gain).

Only the CMA-ES algorithm will be used. The termination condition will be defined by the algorithm itself. Therefore, the following optimization problems are defined:

$$\begin{aligned} \min J(k_p, T_i, l) &= I_{benchmark}|_{G'(p(s)=f(l)} && (7.34)\\ & s.t. \\ k_p &\in \left(0, \max{(k_u)}|_{G'(s)=f(l)}\right] && (7.35)\\ T_i &\in [0, 100] && (7.36)\\ l &\in [0.7, 1.3] && (7.37)\\ Re[\lambda] &< 0 && (7.38) \end{aligned}$$

and

$$\begin{aligned} \min J(a, b, l) &= I_{benchmark}|_{G'(p(s)=f(l)} && (7.39)\\ & s.t. \\ a &\in (0, 1] && (7.40)\\ b &\in [0, 1] && (7.41)\\ l &\in [0.7, 1.3] && (7.42) \end{aligned}$$

Again, in the former case, a basic penalty function is used to identify stabilizing from non-stabilizing solutions. In Tables 7.6 and 7.7 the results are shown. While there are not statistical differences (according Wilcoxon test) in the value achieved in the benchmark index $I_{benchmark}$, there are differences in the function evaluations used (see Figure 7.5). Therefore, the sampling proposal has been useful to reduce the quantity of function evaluations required to get an optimal value.

7.4 Conclusions on this chapter

In this chapter, it has been presented a simple coding statement to sample internal stable PID controllers given a FOPDT process. This coding could be potentially used by stochastic optimizers, to improve their efficiency in the global search procedure. In summary, the main advantages of this proposal are:

- Less function evaluations used, since the algorithm is always sampling candidate solutions in a space where all controllers stabilize the closed loop.
- All the stabilizing controllers are contained in the space; therefore *a priori* the algorithm is minding all the possible tuning configurations.

Table 7.6: Performance of the CMA-ES algorithm in test 2.

	Stochastic Sampling	Standard bounds
minimum	**0.5855**	**0.5855**
median	**0.5855**	**0.5855**
maximum	**0.5855**	**0.5855**
mean	**0.5855**	**0.5855**
variance	**1.8e-27**	2.8e-14
success	**51**	**51**

Table 7.7: Function evaluation required by CMA-ES algorithm in test 2.

	Stochastic Sampling	Standard bounds
minimum	**1815**	2137
median	**2417**	3229
maximum	**3537**	4447
mean	**2415**	3146
variance	**1.0e5**	2.3e5

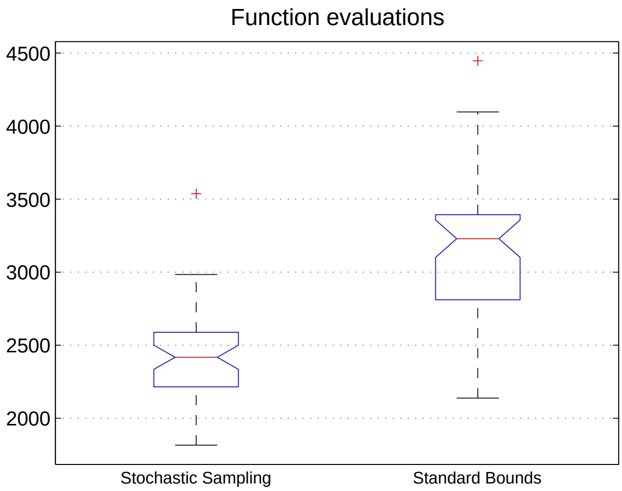


Figure 7.5: Function evaluation required by CMA-ES algorithm in test 2.

- No guesses or hints are made about the decision search space for the evolutionary algorithm; the search space is straightforward bounded.
- It is an approach suitable for multidisciplinary optimization, where the system itself could be subject to (re)design.

While simple examples have been shown in this chapter, the approach is suitable to face more demanding optimizations statements by including constraints. Limitations and future work are:

- The code is presented just for FOPDT processes; future work should focus on extend the sampling to process of higher complexity.
- Only PI and PID controllers can be tuned with this procedure; it should be extended to PID controllers with derivative filter and setpoint weighting.
- It is limited to SISO processes; it will be interesting to use the same approach in MIMO statements.

CHAPTER 8:

Minding the gap: a preference driven multiobjective optimization design procedure for controller tuning

This chapter is devoted to present final contributions on multivariable controller tuning using the MOOD procedure. For this purpose basic ideas from Part II (Preliminary Contributions on Controller Tuning) are used, while the limitations found on MOOD tools implemented are amended with the developments from Part III (Contributions on MOOD Tools). Likewise, the sampling mechanism stated in the previous chapter is used, in order to improve convergence and optimize computational resources. A paper with contents from this chapter is currently under development.

8

Minding the gap: a preference driven multiobjective optimization design procedure for controller tuning

It's the end. But the moment has been prepared for.

The 4th. Doctor

8.1 Aim of this chapter

In this chapter the developed tools for the MOOD procedure are implemented in two control engineering benchmarks. The specific aims of this chapter are to:

- Employ tools developed in Chapters 5, 6 and 7 for the MOOD procedure in order to validate their usability for controller tuning purposes.
- State potential control engineering benchmark problems in order to serve as a test suite for performance comparison among MOEAs and MCDM methodologies.

The remainder of this chapter is as follows: in Section 8.2 a summary on the MOOD procedure to be used is presented; Section 8.3 is dedicated to solve anew (for comparison purposes) the benchmark setup of section 3.3, the Wood & Berry distillation column; Section 8.4 is devoted to propose a controller tuning for the TITO version of the Boiler Control problem of [133]. Finally, some concluding remarks are given.

8.2 MOOD procedure with preferences for multivariable PI controller tuning

This chapter is dedicated to PI controller tuning in MIMO processes by means of the MOOD procedure. As commented in chapter 1, the MOOD procedure consists of (at least) three main steps: the MOP definition (measurement); the multiobjective optimization process (search); and the MCDM stage (decision making). Tools and guidelines for each of these steps are commented below:

MOP definition: the preference inclusion methodology using the GPP index will be used (Chapter 6) in order to be able to deal with many-objective and constrained optimization instances, assuring a pertinent approximation of the Pareto front. The *ten times the number of objectives* thumb of rule for the quantity of solutions required in the approximated Pareto front commented in Chapter 6, based on [119], is adopted. Also a clear distinction among design objectives and design objectives for decision making is stated. That is, in which subspace the DM would like to perform a decision making analysis, by identifying objectives that should be minimized and minded in the search process, but they are not meant to be used for decision making.

EMO process: the sp-MODE-II algorithm (from Chapter 6) will be used because of its features regarding the expected Pareto front and the optimization instance (Chapter 1). In the former case, these features are related to the capabilities to produce a well spread and pertinent Pareto front (Features 2 and 3, respectively). In the latter case, because of its capabilities to deal with constrained and many-objective optimization instances (Features 4 and 5, respectively). Also, the stochastic sampling procedure presented in Chapter 7 will be used; as this sampling is limited to single loops, the overall stability is checked through the evolutionary process and incorporated in the preference set as a constraint.

MCDM step: Level Diagrams will be used due to their capabilities to depict m-dimensional Pareto fronts and for design concepts comparison, developed in Chapter 5. The taxonomy to identify the visualizations is adopted from the same chapter [46].

[46] *LD/front/measure*. For example, LD/$\boldsymbol{J}_p^*$/$\|\hat{\boldsymbol{J}}(\boldsymbol{\theta})\|_2$, means that a visual representation of Pareto front approximation $\boldsymbol{J}_p^*$ with 2-norm in LD is presented.

According to Chapter 2, while several works focus on PI-PID controller tuning using EMO, few of them deal with MIMO instances. Furthermore, few of them use some mechanism for pertinency improvement in many-objective optimization statements for these problems. Therefore, it is justified to test the MOOD procedure with the proposals contained in this thesis in the aforementioned control engineering benchmarks. In all instances, it is assumed that commonly used tuning techniques don't fulfill all the designer's requirements and therefore, the MOOD procedure is employed. A standard CPU[47] is used to calculate the Pareto front approximations for such benchmarks.

With the aforementioned tools and procedure, limitations detected on Part II (preliminary contributions on controller tuning) regarding many-objective optimization, merging design objectives and the quantity (and quality) of the approximated Pareto front for the MCDM step are amended; as a consequence, the results are further improved.

8.3 Benchmark setup: the Wood & Berry distillation column process

The first example is the well-known distillation column model defined by Wood & Berry [16, 195], used previously in Chapter 3 and solved anew for comparison purposes. The aims of this example are:

- To provide a many-objective optimization statement for MIMO processes, in order to validate the sp-MODE-II algorithm.
- To provide a comparison between algorithm's performance to build a $\mathbf{T\_}\boldsymbol{J}_P^*$, $\mathbf{D\_}\boldsymbol{J}_P^*$ and $\mathbf{HD\_}\boldsymbol{J}_P^*$ in many-objective optimization instances.

Three different algorithms (concepts) to approximate the $\mathbf{T\_}\boldsymbol{J}_P^*$, $\mathbf{D\_}\boldsymbol{J}_P^*$ and $\mathbf{HD\_}\boldsymbol{J}_P^*$ are used:

Concept 1: sp-MODE-II algorithm of Chapter 6.

Concept 2: A DE algorithm with the diversity/pertinency mechanism described in [189], used for indicator based evolutionary algorithms (IBEA). This technique, used previously in Chapter 6, is used because it is a state of the art technique capable of handling design preferences.

Concept 3: A stochastic sampling approach. This process is used for comparison purposes, since it has been noticed in [35] that for many-objective optimization instances, stochastic sampling procedures could be competitive with MOEAs.

[47] DELL T1500 computer, Windows 7 system, processor Intel Core i7, 2.93 GHz with 8.00 GB RAM.

For concepts 1 and 2, $Cr = 0.5$ and $F = 0.5$ are used in the DE mechanism (general values in accordance with Tables 1.1 and 6.1). A budget of 6000 FEs is fixed in three instances and the same stochastic sampling approach of Chapter 7 is used. A total of 201 independent runs are carried out in each case.

8.3.1 Design problem statement

The process $P(s)$ (Equation (8.1)) will be controlled with a decentralized PI controller structure $C(s)$ (Equation (8.8)). The designer is interested in improving the performance of the initial solution $\boldsymbol{\theta}_0$ (stated by the BLT [109] methodology).

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} = \begin{bmatrix} \frac{12.8e^{-s}}{16.7s+1} & \frac{-18.9e^{-3s}}{21s+1} \\ \frac{6.6e^{-7s}}{10.9s+1} & \frac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \tag{8.1}$$

$$C(s) = \begin{bmatrix} k_{p1}\left(1 + \frac{1}{T_{i1}s}\right) & 0 \\ 0 & k_{p2}\left(1 + \frac{1}{T_{i2}s}\right) \end{bmatrix} \tag{8.2}$$

The same objective statement from Chapter 3 is used:

$$\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta}) = [-k_{i1}, -k_{i2}, L_{cm}, M_{s1}, M_{s2}, M_{p1}, M_{p2}] \in \mathbb{R}^7 \tag{8.3}$$

subject to:

$$0 \leq \theta_i \leq 1, i = [1, 2, 3, 4] \tag{8.4}$$

$$G_1(\boldsymbol{\theta}) = Re[\lambda]_{max} < 0 \tag{8.5}$$

where $k_{i1} = \frac{k_{p1}}{T_{i1}}$, $k_{i2} = \frac{k_{p2}}{T_{i2}}$ are the integral gains of controller $C_{11}(s)$ and $C_{22}(s)$ respectively; L_{cm} (Equation (3.5)) is the closed loop log modulus; M_{s1} and M_{s2} are the maximum values of the sensitivity function (Equation (2.1)) for individual control loops 1 and 2 respectively; M_{p1} and M_{p2} are the maximum values of the complementary sensitivity function (Equation (2.3)) for individual control loops 1 and 2, respectively.

Given the stochastic statement of the previous chapter, $\theta_i \in [0, 1]$ for $i = \{1, \cdots, 4\}$ the constraint to improve stability of the original statement is no longer needed. It is expected to achieve a solution with better $J_1(\boldsymbol{\theta}_0)$, $J_2(\boldsymbol{\theta}_0)$ than the BLT ($\boldsymbol{\theta}_0$) solution. Also, limits of $(M_{s1}, M_{s2}) < 2.0$, $(M_{p1}, M_{p2}) < 1.5$ and $L_{cm} < 4.0$ are imposed.

For the sp-MODE-II (Concept 1) the preference matrix is depicted in Table 8.1. For the IBEA technique (Concept 2), two different approaches are evaluated: using as preference vector $J_{pref} = \boldsymbol{J}^T$ with $\rho = 0.01$ (named IBEA$_1$) and using $J_{pref} = \boldsymbol{J}^D$ with $\rho = 0.001$ (named IBEA$_2$). Since seven objectives are defined, 70 solutions are required by the DM in the approximated Pareto front. Therefore sp-MODE-II algorithm uses a spherical grid of 70 arcs and a population of 70 individuals; IBEA employs a population of 70 individuals.

8.3.2 Results and Discussions

In Table 8.2 the achieved **T_HypV**, **D_HypV** and **HD_HypV** are depicted. Statistical significance was validated with the Wilcoxon test at 95% with Bonferroni correction [47]. In Figure 8.1 a visual representation of such statistics is shown. MOEAs were capable of reaching better **T_HypV** values than the stochastic approach. This validates the usage of MOEAs to approximate the $\mathbf{T\_}\boldsymbol{J}_P^*$ in the many-objective optimization instance of Equation (8.3). Regarding the computational cost, a time base of 1180 seconds is used (median value of the time required to evaluate 6000 FEs, with the available CPU resources, in 201 runs). Afterwards, the time required to execute each algorithm with the same number of FEs for each run is calculated and divided by the time base. This lead to the median values of time ratio depicted at the end of the Table 8.2. The dominance filter is the algorithm with the lower time performance (less than 1% of additional computational cost), while for the IBEA algorithm has the highest (more than 100% of additional computational cost).

Regarding the MOEAs, the sp-MODE-II algorithm was capable to approximate solutions in the $\mathbf{T\_}\boldsymbol{J}_P^*$ and $\mathbf{D\_}\boldsymbol{J}_P^*$ Pareto fronts consistently. The IBEA$_1$ approach is not capable of reaching the **T_HypV** for some executions, and is incapable of reaching the **D_HypV**. The IBEA$_2$ approach is able to find solutions in the **T_HypV** for all executions, but not for the **D_HypV**.

In Figure 8.2 a LD / $\left\{ \widetilde{J_{p1}^*}^{50\%}, \widetilde{J_{p2}^*}^{50\%}, \widetilde{J_{p3}^*}^{50\%} \right\}$ / $Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J_{p0}^*\right)$ visualization of attainment surfaces of IBEA$_1$ (J_{p1}, red ◇), IBEA$_2$ (J_{p2}, orange ◯) and sp-MODE-II (J_{p3}, yellow □) is depicted. A Pareto front approximation with all the solutions from all executions of each design concept evaluated has been used as reference front. The set of solutions close to 1 indicate[48] that the algorithm approximates better the $\mathbf{T\_}\boldsymbol{J}_P^*$ and $\mathbf{D\_}\boldsymbol{J}_P^*$ Pareto fronts. Since the sp-MODE-II attainment surface lies practically on 1, it indicates its superiority to reach the pertinent Pareto front approximation. The difference among algorithms with objectives $J_1(\boldsymbol{\theta})$ and $J_2(\boldsymbol{\theta})$ can be noticed. While the attainment surface of sp-MODE-II is inside the **T_HypV**, the attainment surfaces of the IBEA$_1$ and IBEA$_2$ cover some part of the undesirable hypervolume. It seems that the pressure mechanism of the sp-MODE-II algorithm, which focuses first on individual solutions getting to the **T_HypV** and afterwards on the Pareto front approximation, is useful and makes the difference between the approaches.

[48] According to que comparison methods developed for this visualization in Chapter 5 and depicted in Table 5.2.

Table 8.1: Preferences Set for experimental setup of Section 8.3. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

	Preference Set										
	←	HD	→ ←	D	→ ←	T	→ ←	U	→ ←	HU	→
Objective	J_i^0		J_i^1		J_i^2		J_i^3		J_i^4		J_i^5
$J_1(\boldsymbol{\theta})$	$1.6 \cdot J_1(\boldsymbol{\theta}_0)$		$1.4 \cdot J_1(\boldsymbol{\theta}_0)$		$1.2 \cdot J_1(\boldsymbol{\theta}_0)$		$J_1(\boldsymbol{\theta}_0)$		$0.8 \cdot J_1(\boldsymbol{\theta}_0)$		$0.6 \cdot J_1(\boldsymbol{\theta}_0)$
$J_2(\boldsymbol{\theta})$	$1.6 \cdot J_2(\boldsymbol{\theta}_0)$		$1.4 \cdot J_2(\boldsymbol{\theta}_0)$		$1.2 \cdot J_2(\boldsymbol{\theta}_0)$		$J_2(\boldsymbol{\theta}_0)$		$0.8 \cdot J_2(\boldsymbol{\theta}_0)$		$0.6 \cdot J_2(\boldsymbol{\theta}_0)$
$J_3(\boldsymbol{\theta})$	0.0		1.0		2.0		4.0		8.0		16.0
$J_4(\boldsymbol{\theta})$	1.2		1.4		1.8		2.0		3.0		5.0
$J_5(\boldsymbol{\theta})$	1.2		1.4		1.8		2.0		3.0		5.0
$J_6(\boldsymbol{\theta})$	1.0		1.2		1.4		1.5		3.0		5.0
$J_7(\boldsymbol{\theta})$	1.0		1.2		1.4		1.5		3.0		5.0
$G_1(\boldsymbol{\theta})$	<0.0		<0.0		<0.0		0.0		1.0		2.0

Finally, in Figure 8.3, the Pareto front approximation for the median value **T_HypV** of the **T_**$\boldsymbol{J}_P^*$ for sp-MODE-II algorithm is presented. From here, the DM will evaluate and analyze the approximated Pareto front to select a solution according to his/her needs. Given that the main aim of this benchmark setup is to compare MOEAs, in the MCDM step it will be enough to select the solution with the better GPP index. In Figure 8.4 the design alternative with the minimum GPP index is compared with the BLT tuning under the same experiments from Chapter 3.

*Table 8.2: Pertinency level reached for benchmark setup 8.3. **T_HypV**, **D_HypV** and **HD_-HypV** achieved in 201 independent runs for each design concept.*

		sp-MODE-II algorithm (0%, 99%, 100%)	IBEA$_1$ algorithm **T_Vector**, $\rho = 0.01$ (0%, 0.5%, 100%)	IBEA$_2$ algorithm **D_Vector**, $\rho = 0.001$ (0%, 74%, 100%)	Random Sampling (0%, 0.5%, 68%)
HD	Best	0.0E+000	0.0E+000	0.0E+000	0.0E+000
	Median	0.0E+000	0.0E+000	0.0E+000	0.0E+000
	Worst	0.0E+000	0.0E+000	0.0E+000	0.0E+000
	Mean	0.0E+000	0.0E+000	0.0E+000	0.0E+000
	std	0.0E+000	0.0E+000	0.0E+000	0.0E+000
D	Best	**1.7E-009**	8.3E-011	1.2E-009	4.2E-010
	Median	**5.3E-010**	0.0E+000	8.2E-011	0.0E+000
	Worst	0.0E+000	0.0E+000	0.0E+000	0.0E+000
	Mean	**5.8E-010**	4.2E-013	1.8E-010	2.1E-012
	std	3.8E-010	5.9E-012	2.4E-010	2.9E-011
T	Best	**7.7E-006**	4.2E-006	1.6E-006	2.3E-006
	Median	**5.8E-006**	2.6E-006	9.5E-007	3.7E-007
	Worst	**4.0E-006**	9.0E-009	1.6E-007	0.0E+000
	Mean	**5.9E-006**	2.6E-006	9.4E-007	5.7E-007
	std	7.3E-007	7.8E-007	2.6E-007	6.1E-007
Time Performance		1.7271	2.2051	2.2195	1.0059

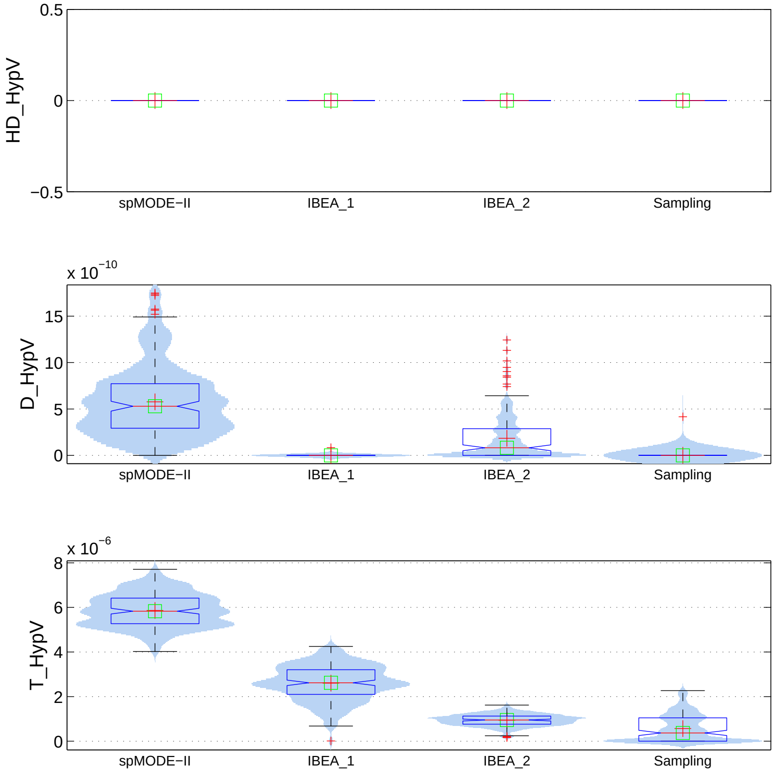


Figure 8.1: Pertinency level reached for benchmark setup 8.3. ***T_HypV*** *and* ***D_HypV*** *achieved in 201 independent runs for each design concept.*

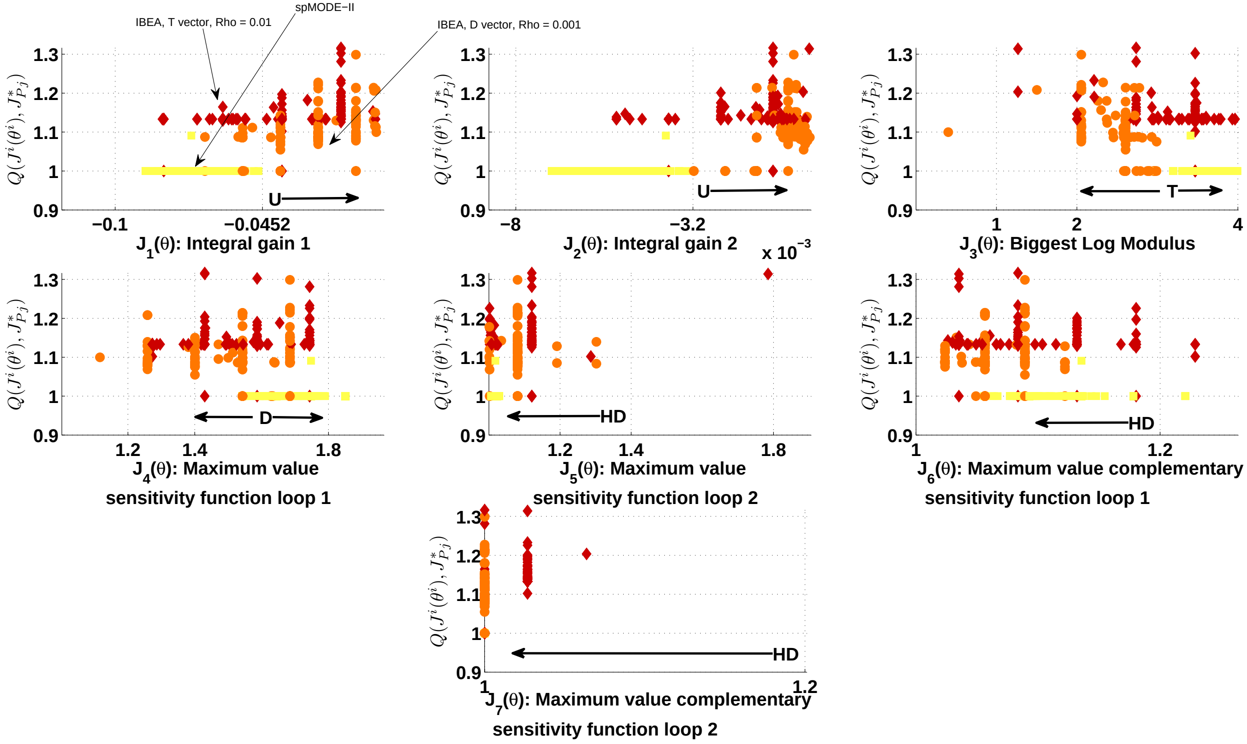


Figure 8.2: LD / $\left\{\widetilde{J^*_{p1}}^{50\%}, \widetilde{J^*_{p2}}^{50\%}, \widetilde{J^*_{p3}}^{50\%}\right\}$ / $Q\left(\boldsymbol{J}^i(\boldsymbol{\theta}^i), J^*_{p0}\right)$ visualization of attainment surfaces of IBEA$_1$ (J_{p1}, red ⋄), IBEA$_2$ (J_{p2}, orange ◯) and sp-MODE-II (J_{p3}, yellow □) for the benchmark setup of Section 8.3.

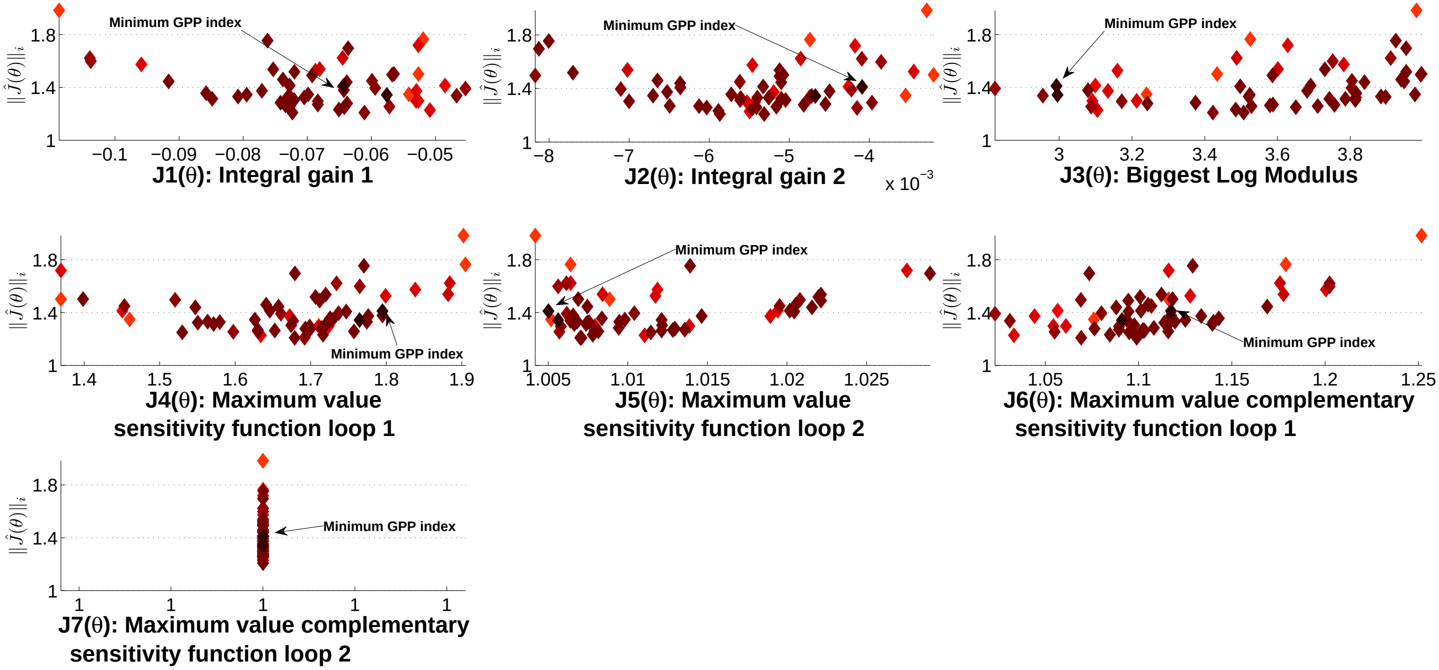


Figure 8.3: Pareto front approximation with sp-MODE-II algorithm of the benchmark setup of Section 8.3. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

8.4 Benchmark setup: the Boiler Control problem

The process under consideration is the benchmark for PID control 2012 described in [133]. It is a benchmark which proposes a boiler control problem ([132, 59]) based on the work of [141]. The aims of this example are:

- To provide a many-objective optimization statement for MIMO processes under *quasi-real* conditions, in order to validate the MOOD procedure for controller tuning.

Quasi-real conditions makes reference to the following steps:

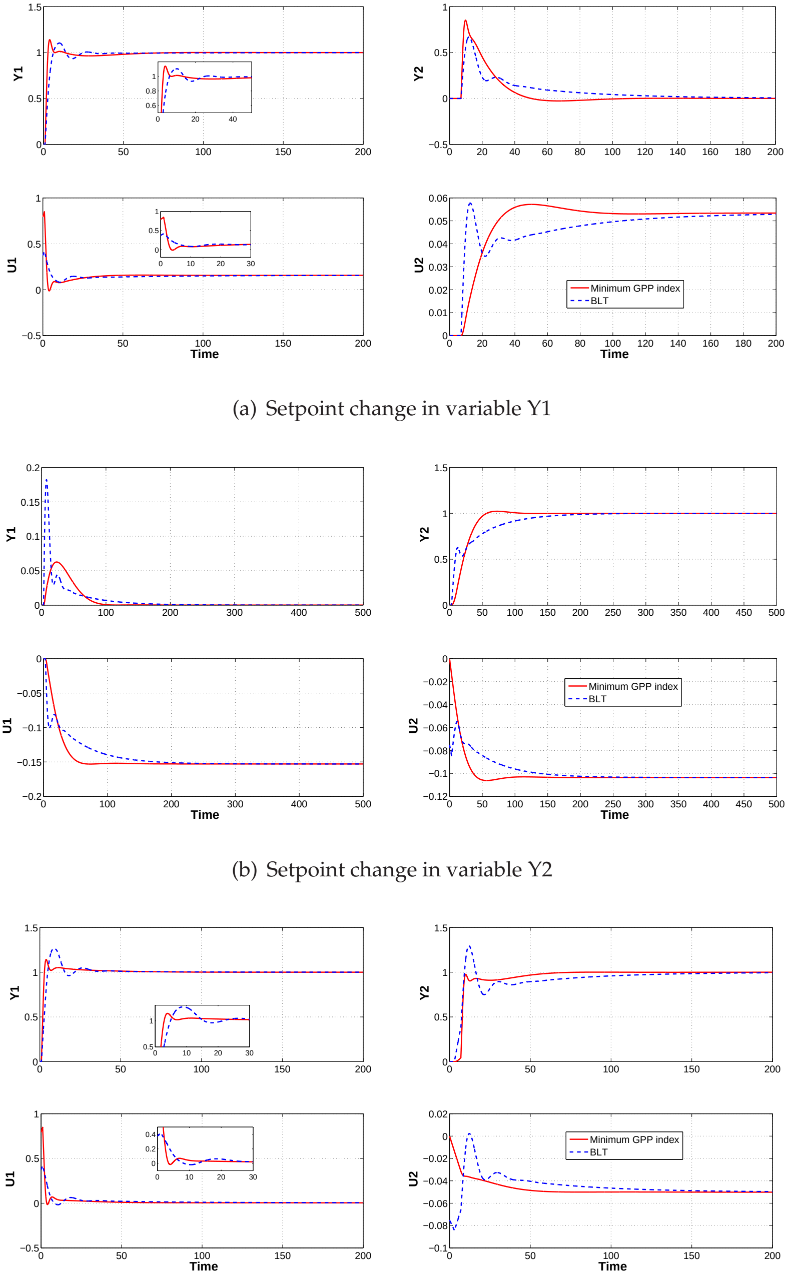
1. Consider the (original) non-linear model simulation as the *real process*.
2. Step changes are used to obtain simplified linear models from the *real process*.
3. Controllers are tuned using the aforementioned approximated models.
4. The selection procedure will be made according to experiments on the approximated models.
5. The selected controller will be implemented in the *real process*.

The original problem stated a 3x3 MIMO system with a measured load disturbance:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \\ Y_3(s) \end{bmatrix} = \begin{bmatrix} P_{11}(s) & 0 & P_{13}(s) \\ P_{21}(s) & P_{22}(s) & 0 \\ P_{31}(s) & 0 & P_{33}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \\ U_3(s) \end{bmatrix} + \begin{bmatrix} P_{1d}(s) \\ 0 \\ P_{3d}(s) \end{bmatrix} D(s) \tag{8.6}$$

Where the inputs are fuel flow $U_1(s)$ [%], air flow $U_2(s)$ [%] and water flow $U_3(s)$ [%], while the outputs are steam pressure $Y_1(s)$ [%], oxygen level $Y_2(s)$ [%] and water level $Y_3(s)$ [%]. $D(s)$ is a measured load disturbance. This is a verified model, useful to propose, evaluate and compare different kinds of tuning/control techniques ([69, 163, 137, 170, 160]).

Here the reduced TITO version of the benchmark is used:



(a) Setpoint change in variable Y1

(b) Setpoint change in variable Y2

(c) Setpoint changes in variables Y1 and Y2

Figure 8.4: Performance of the BLT tuning and the controller with the lowest GPP index from Figure 8.3 for a given setpoint change for benchmark setup of Section 8.3.

$$\begin{aligned}\begin{bmatrix} Y_1(s) \\ Y_3(s) \end{bmatrix} &= \begin{bmatrix} P_{11}(s) & P_{13}(s) \\ P_{31}(s) & P_{33}(s) \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_3(s) \end{bmatrix} \\ &+ \begin{bmatrix} G_{1d}(s) \\ G_{3d}(s) \end{bmatrix} D(s) \end{aligned} \tag{8.7}$$

with the proposed controller:

$$C(s) = \begin{bmatrix} k_{p1}\left(1+\frac{1}{T_{i1}s}\right) & 0 \\ 0 & k_{p2}\left(1+\frac{1}{T_{i2}s}\right) \end{bmatrix} \tag{8.8}$$

8.4.1 Design problem statement

The identified reduced model[49] at the operation point is shown in Equations (8.9), (8.10) and depicted in Figure (8.5).

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{13}(s) \\ P_{31}(s) & P_{33}(s) \end{bmatrix} = \begin{bmatrix} \frac{0.3727e^{-3.1308s}}{55.68s+1} & \frac{-0.1642}{179.66s+1} \\ \frac{0.0055\cdot(166.95s-1)}{31.029s^2+s} & \frac{0.0106e^{-9.28s}}{s} \end{bmatrix} \tag{8.9}$$

$$G(s) = \begin{bmatrix} G_{1d}(s) \\ G_{3d}(s) \end{bmatrix} = \begin{bmatrix} \frac{-0.78266e^{-17.841s}}{234.69s+1} \\ \frac{-0.0014079e^{-7.1872s}}{7.9091s^2+s} \end{bmatrix} \tag{8.10}$$

The optimization statement to minimize is:

$$\min_{\boldsymbol{\theta}} \boldsymbol{J}(\boldsymbol{\theta}) = [J_1(\boldsymbol{\theta}), J_2(\boldsymbol{\theta}), J_3(\boldsymbol{\theta}), J_4(\boldsymbol{\theta}), J_5(\boldsymbol{\theta})] \tag{8.11}$$

subject to:

$$0 \leq \theta_i \leq 1, i = [1,2,3,4] \tag{8.12}$$

$$G_1(\boldsymbol{\theta}) = Re[\lambda]_{max} < 0 \tag{8.13}$$

Objectives stated are:

$J_1(\boldsymbol{\theta})$**:** Stabilizing time for $Y_1(s)$ at presence of a step Load disturbance $D(s)$.

[49] Nominal linear models have been identified using simple step tests with the Matlab© identification toolbox.

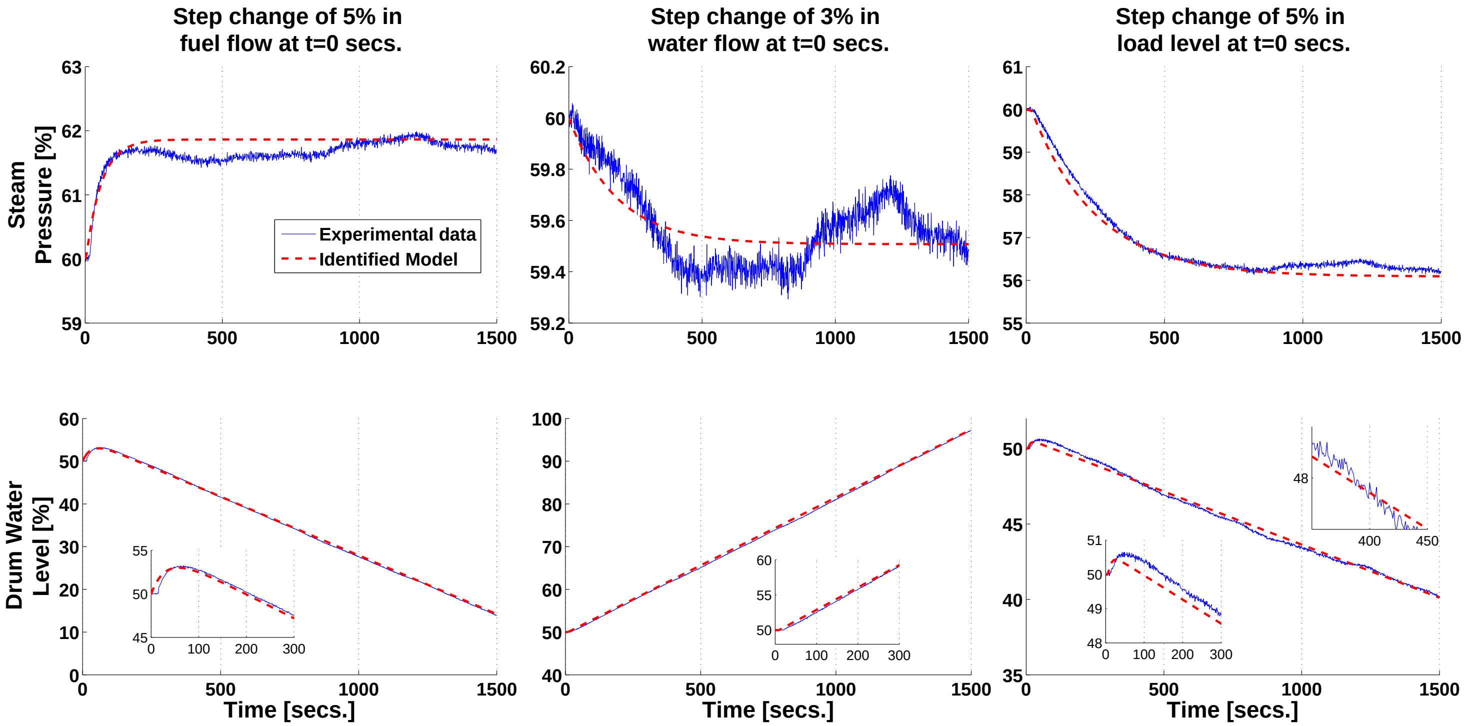


Figure 8.5: Identified reduced model of the Boiler process (benchmark setup from Section 8.4).

$J_2(\boldsymbol{\theta})$**:** Stabilizing time for $Y_2(s)$ at presence of a step Load disturbance $D(s)$.

$J_3(\boldsymbol{\theta})$: Biggest log modulus for overall robustness (Equation (3.5)).

$J_4(\boldsymbol{\theta})$**:** Maximum value of sensitivity function M_s for loop 1 (Equation (2.1)).

$J_5(\boldsymbol{\theta})$**:** Maximum value of sensitivity function M_s for loop 2 (Equation (2.1)).

Design objectives to perform a MCDM stage are $J_1(\boldsymbol{\theta}), \cdots, J_3(\boldsymbol{\theta})$. That means that 30 solutions are required and the objective space is just partitioned in three dimensions; notice that all five objectives are still minding due to the preference matrix. The preference matrix for the design objectives is depicted in Table 8.3. Given the stochastic statement of the previous chapter, $\theta_i \in [0, 1]$ for $i = 1, \cdots, 4$. A total of 25 runs are carried, and the Pareto front with the median value of the **T_HypV** is used as representative solution for further analysis in the MCDM step.

Table 8.3: Preferences Set A for experimental setup of Section 8.4. Five preference ranges have been defined: highly desirable (HD), desirable (D), tolerable (T) undesirable (U) and highly undesirable (HU).

Preference Set A						
	← HD	→← D	→← T	→← U	→← HU	→
Objective	J_i^0	J_i^1	J_i^2	J_i^3	J_i^4	J_i^5
$J_1(\boldsymbol{\theta})$ [s]	600	700	800	900	1500	2000
$J_2(\boldsymbol{\theta})$ [s]	600	900	1000	1600	1800	2000
$J_3(\boldsymbol{\theta})$ [-]	1.0	2	3	6	12	24
$J_4(\boldsymbol{\theta})$ [-]	1.0	1.2	1.4	1.8	1.9	2.0
$J_5(\boldsymbol{\theta})$ [-]	1.0	1.2	1.4	1.8	1.9	2.0
$G_1(\boldsymbol{\theta})$ [-]	<0.0	<0.0	<0.0	0.0	1.0	2.0

8.4.2 Results and Discussions

In Figure 8.6, the median value for the **T_HypV** of the **T_**$\boldsymbol{J}_P^*$ for the sp-MODE-II algorithm is presented. It is important noting the compactness of the approximated Pareto front towards the **D_HypV**. This facilitates the DM procedure to select a solution to implement. Also the simulation response of the approximated design alternatives is depicted in Figure 8.7.

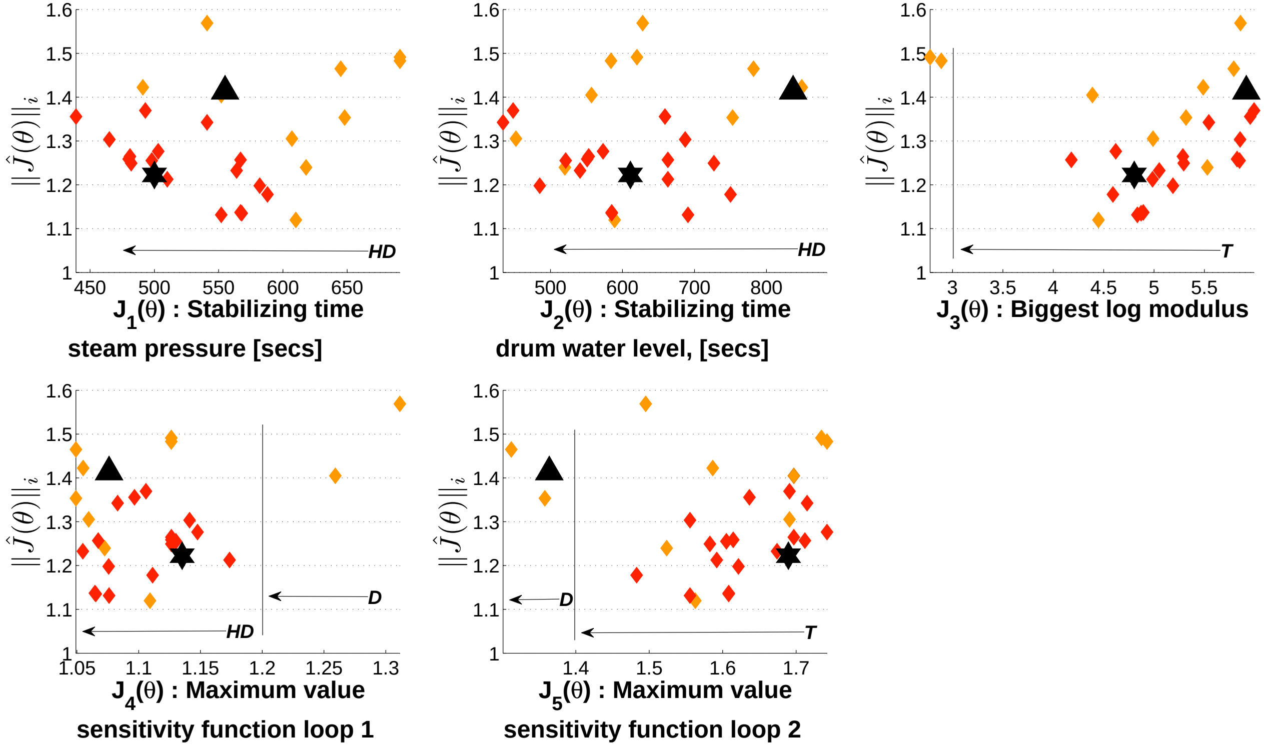


Figure 8.6: Pareto Front of the benchmark setup of Section 8.4. Design alternative with the lowest GPP index (dark △) and the selected design alternative (dark ★) are depicted.

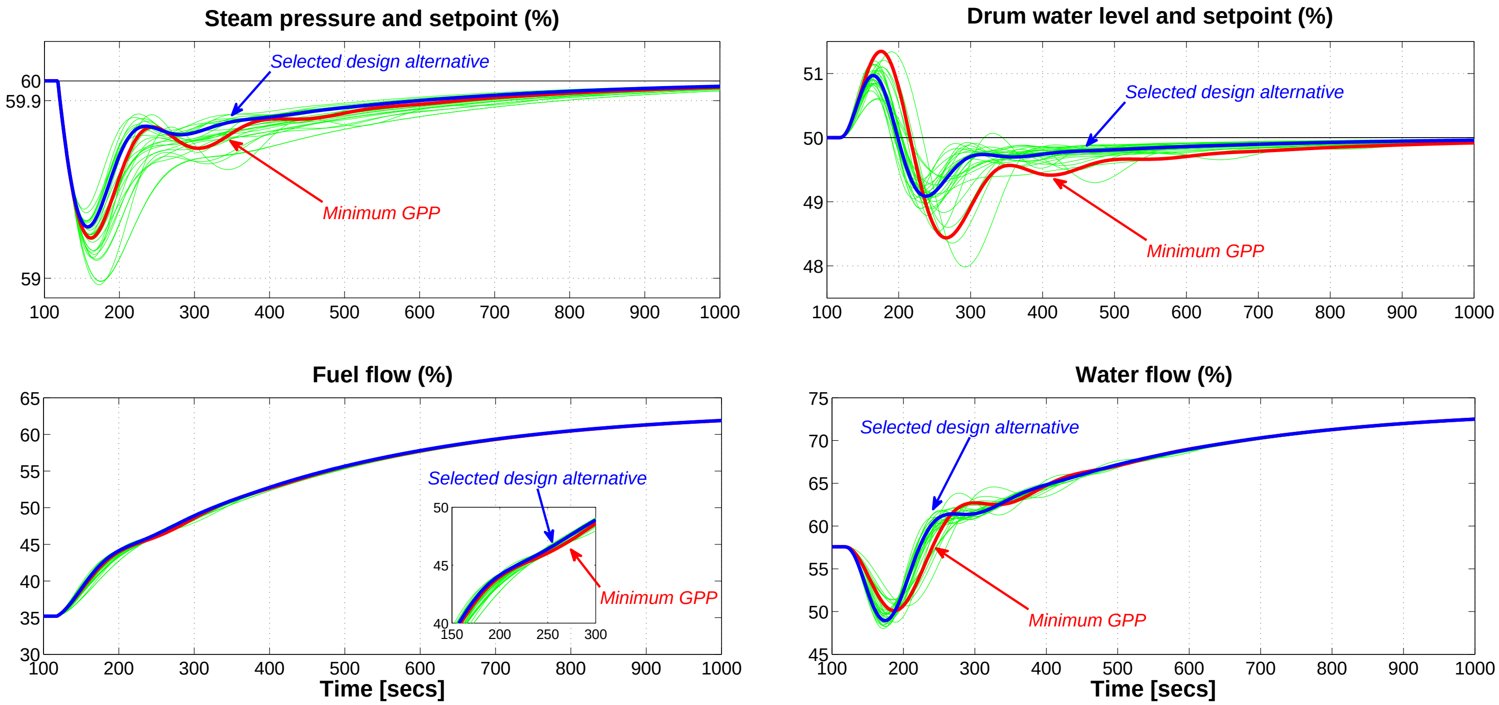


Figure 8.7: Simulation performance of the approximated Pareto front from Figure 8.6.

The preference set stated (Table 8.3) to calculate the GPP index has been useful to incorporate the designer's preferences among conflicting objectives. Nevertheless, although this index has been useful to evolve the evolutionary process towards the pertinent Pareto front, it is not capable to describe by itself the tradeoff among objectives. That is, since the Pareto front is unknown *a priori*, it is impossible to predict its shape by just analyzing the lowest GPP index obtained. Therefore, the MCDM stage is still a necessary step for the designer. By using the sp-MODE-II, it has been assured to have a pertinent approximation of the Pareto front, with an interesting set of solutions from the point of view of the designer. By consequence, it facilitates the decision making process.

Two solutions are marked in the Figure 8.6: the design alternative with the lowest GPP index (black $\triangle$ in the approximated Pareto front) and a solution which sacrifices a desirable value in $J_5(\boldsymbol{\theta})$ (individual loop performance) in order to get a better overall performance $J_3(\boldsymbol{\theta})$ (black ★ in the approximated Pareto front). The differences between both design alternatives can be appreciated in Figure 8.7; the latter has a smooth response in the drum water level when compared with the former. Given the above, this design alternative is preferred over the one with the minimum GPP index.

This design alternative has been implemented in the *real process*, and the performance index defined by the benchmark $I_{benchmark}$ is shown in Table 8.5. Such index is an aggregate objective function, which combines ratios of the IAE, ITAE and IADU indexes using as a base case two PI controllers $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2}] = [2.5, 50, 1.25, 50]$. More details can be consulted in [133].

Firstly, a PI digital implementation without filtering the measured signal is employed. Notice that performance indexes related with ratios of IAE and ITAE are better than the reference controller. Nevertheless, the performance indicator $I_{benchmark}$ is worse in the case of Test 1 (Figure 8.8) and it has almost the same performance than Test 2 (Figure 8.10). This is due to the weighting factor used for the control action; the design alternative selected is more sensitive to noise and therefore, the IADU ratio is bigger.

Using a first order filter with $\tau_f = 10$, the performance related to the control action is improved[50] and as a consequence, the overall index $I_{benchmark}$ (Figures 8.9 and 8.11). Therefore, the proposed PI controllers have a performance (regarding this metric) which is better than the reference controllers.

There are differences between performance on the nominal model and real model with regard to stabilizing times. A more suitable approach would be to define a reliable based optimization, using a stochastic sampling with the expected uncertainties in the models. In summary, the methodology is effective, bringing a controller fulfilling all the requirements with a better performance than the reference controller.

[50] Such filter still guarantees overall stability in the control loop for the nominal process of Equation (8.9)

A comparison in equal conditions with other approaches dealing with the benchmark (refereed at the beginning of this benchmark setup) is not possible. In [69] a feedforward mechanism is used that is not included in this proposal; in [163] a data driven approach is used (*i.e.* a *on the fly* tuning technique); in [137] a 2x2 PI controller matrix is proposed; finally in [170] and [160] results reported are not evaluated under the benchmark guidelines.

Table 8.4: Simulation validation and performance achieved of the selected design alternative (without filtering) for the benchmark setup from Section 8.4. Ratios of IAE (RIAE), ITAE (RITAE) e IADU (RIADU) with respect to the PI reference case are depicted.

	RIAE1	RIAE2	RIAE3	RITAE1	RITAE3	RIADU1	RIADU2	$I_{benchmark}(0.25)$
Test1	0.8032	0.9991	0.5581	0.0000	0.0000	1.8736	9.8487	1.5117
Test2	0.7882	0.9647	0.2644	0.7137	0.0000	2.4576	4.8029	1.0169

Table 8.5: Simulation validation and performance achieved of the selected design alternative ($\tau_f = 10$) for experimental setup of Section 8.4. Ratios of IAE (RIAE), ITAE (RITAE) e IADU (RIADU) with respect to the PI reference case are depicted.

	RIAE1	RIAE2	RIAE3	RITAE1	RITAE3	RIAVU1	RIAVU2	$I_{benchmark}(0.25)$
Test1	0.8042	0.9991	0.5614	0.0000	0.0000	0.9637	1.2320	0.8325
Test2	0.7093	0.9673	0.3307	0.7085	0.0000	0.7574	1.4825	0.7279

8.5 Conclusions on this chapter

With the aforementioned examples, ideas developed in Part II (Chapters 3 and 4) for multivariable controller tuning in many-objective optimization instances were used. Gaps and limitations noticed in such preliminary contributions were amended with the proposed tools from Part III.

The developed tools and methodologies from Part III have been validated as useful in the MOOD procedure for controller tuning. The sp-MODE-II algorithm (Chapter 6) has shown to be practical and useful for preference inclusion into the evolutionary process. This allows to obtain pertinent Pareto front approximations for many-objective optimization instances. The stochastic sampling approach (Chapter 7) has supported the evolutionary process, assuring that any sampled controller is stable and furthermore, any stable controller could be potentially sampled.

Finally, the methodologies for MOEAs comparison using attainment surfaces (Chapter 5) has been useful to identify limitations on MOEAs. Likewise, level diagrams have shown to be a useful tool for the MCDM step to select a desired design alternative.

Limitations and future work are:

- The stochastic sampling procedure is used just for single PI control loops; future work should focus on extending the sampling to MIMO processes.
- Only PI and PID controllers can be tuned with this procedure; it should be extended to (a) PID controllers with derivative filter and setpoint weighting, and (b) more complex controllers.
- A strategy to design the feedforward compensator by means of the MOOD procedure should be developed.
- Experimental validation on physical setups (as in Chapter 4) need to be carried out.
- It should be worthwhile to evaluate different optimization instances for multivariable controller tuning as multidisciplinary or reliable based statements.
- While the LD visualization is a powerful tool to analyze an m-dimensional Pareto front, it was also required to incorporate information from the time response of the approximated (and pertinent) Pareto front. Therefore, it seems to be a promising area for development to build visualizations approaches for the specific application of controller tuning.

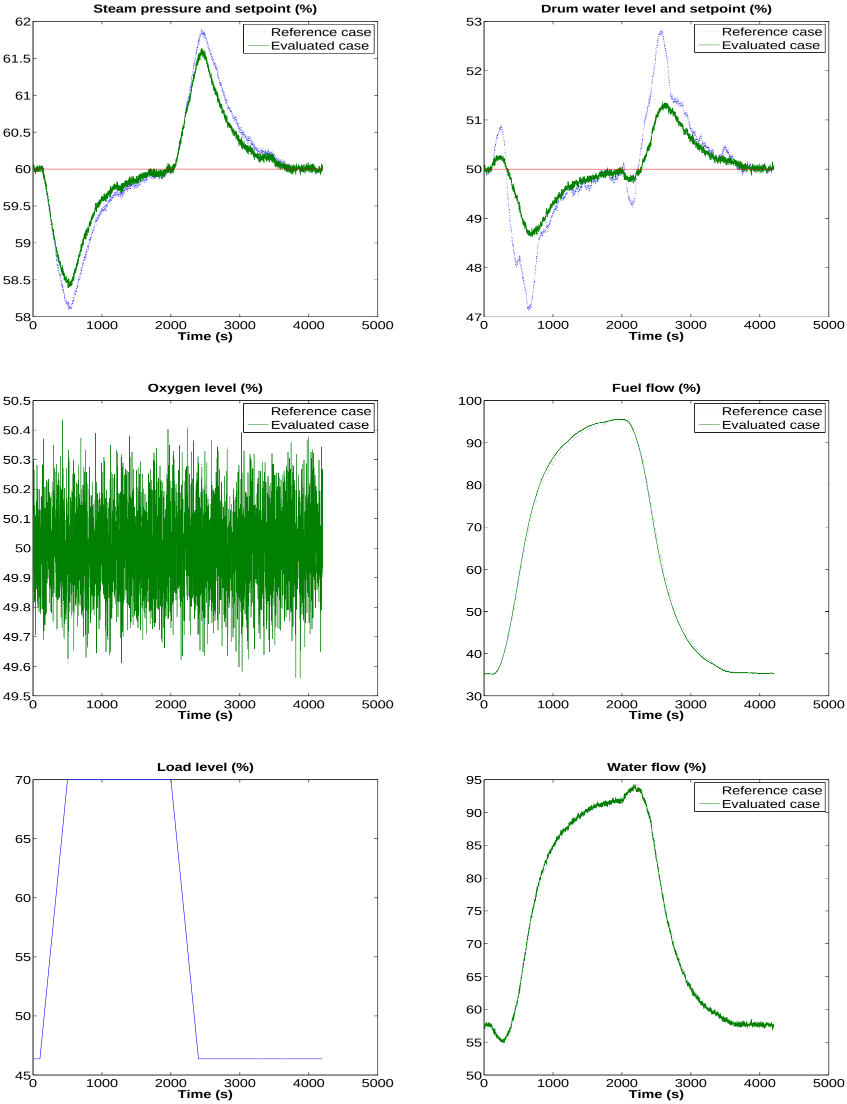


Figure 8.8: Performance of the PI controller $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [1.533, 29.549, 5.315, 125.778]$ *(without filter) and its comparison with the reference case* $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2}] = [2.5, 50, 1.25, 50]$ *for test 1 ($JM = 1.5117$) in benchmark setup 8.4.*

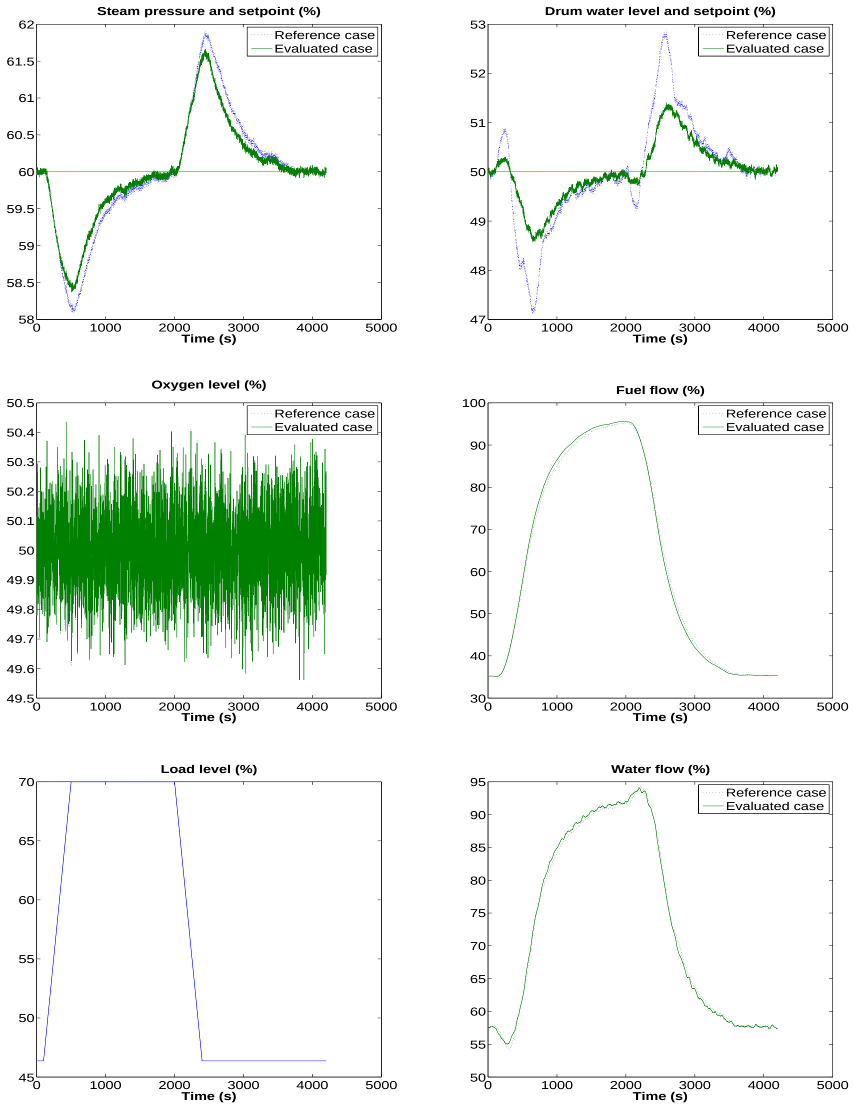


Figure 8.9: Performance of the PI controller $[k_p, T_i, k_p, T_i] = [1.533, 29.549, 5.315, 125.778]$ *(*$\tau_f = 10$*) and its comparison with the reference case* $[k_{p1}, T_{i_1}, k_{p2}, T_{i_2}] = [2.5, 50, 1.25, 50]$ *for test 1 (*$JM = 0.8325$*) in benchmark setup 8.4.*

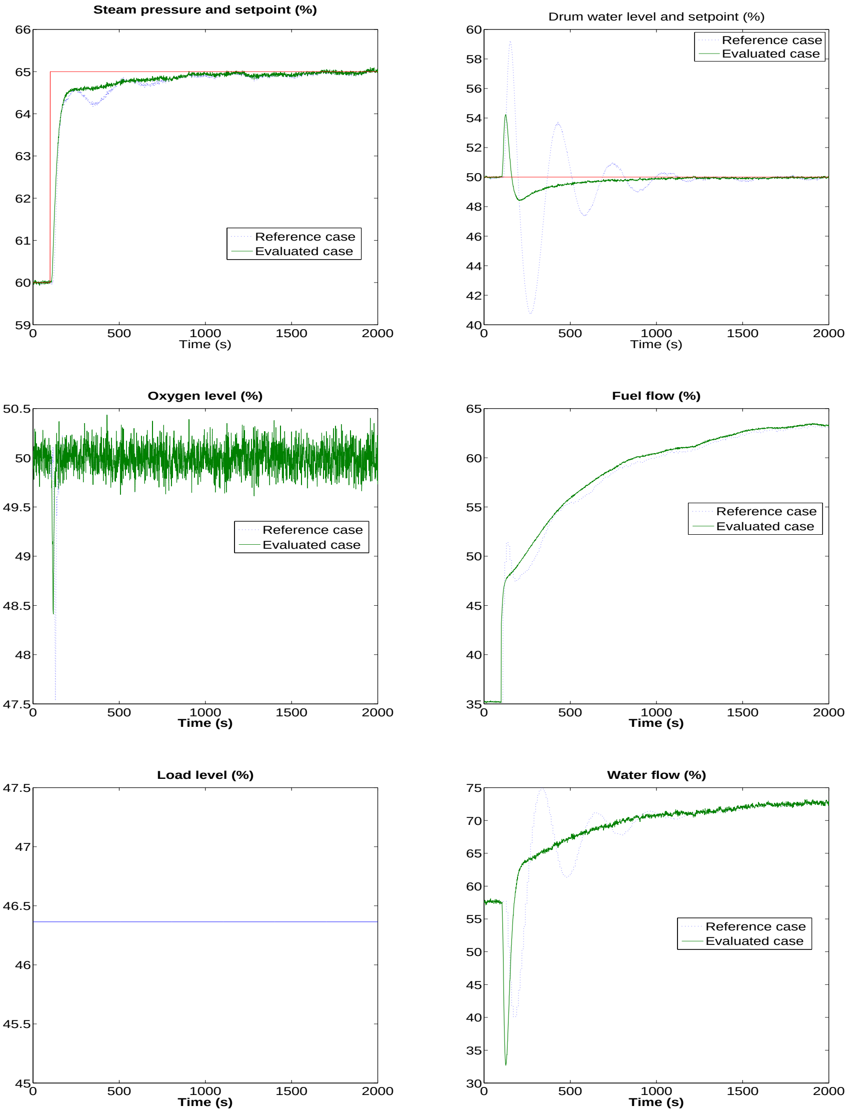


Figure 8.10: Performance of the PI controller $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [1.533, 29.549, 5.315, 125.778]$ *(without filter) and its comparison with the reference case* $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [2.5, 50, 1.25, 50]$ *for test 2 (*$JM = 1.0169$*) in benchmark setup 8.4.*

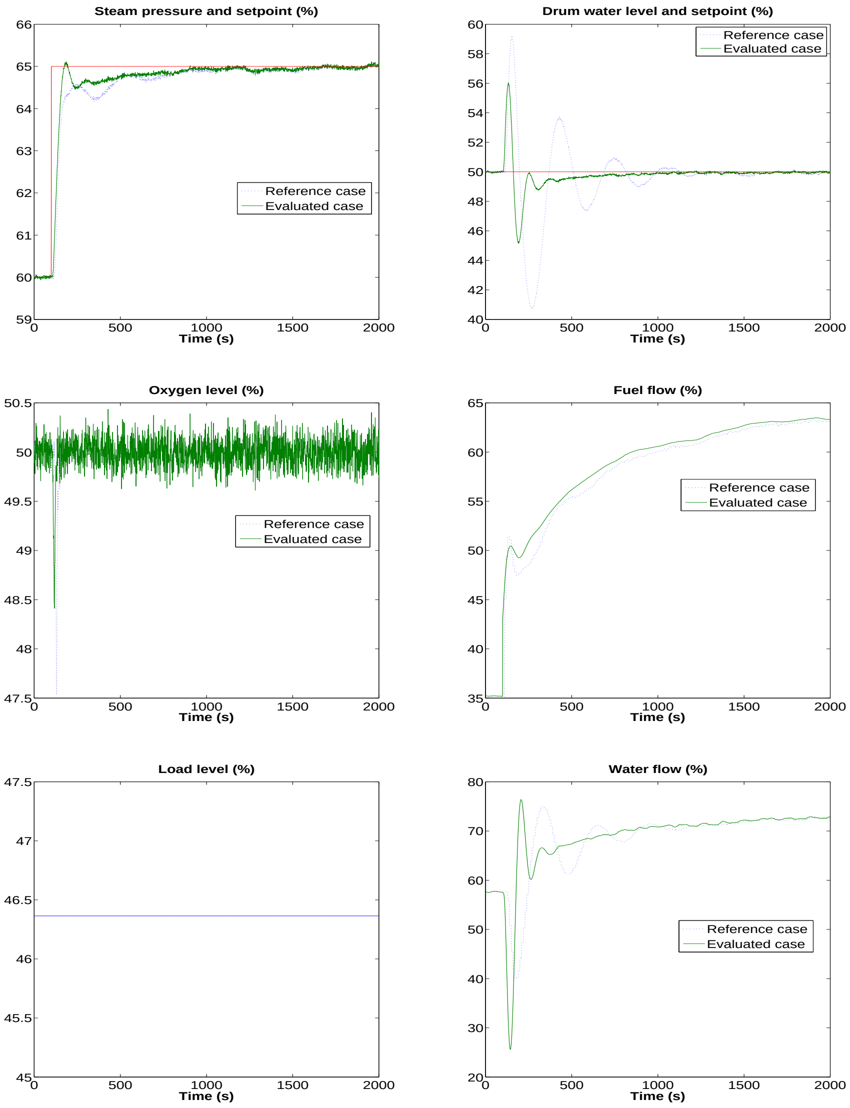


Figure 8.11: Performance of the PI controller $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [1.533, 29.549, 5.315, 125.778]$ *(*$\tau_f = 10$*) and its comparison with the reference case* $[k_{p_1}, T_{i_1}, k_{p_2}, T_{i_2},] = [2.5, 50, 1.25, 50]$ *for test 2 (*$JM = 0.7279$*) in benchmark setup 8.4.*

Conclusions of this Thesis

Conclusions of this thesis

This song is ending. But the story never ends.

Ood Σ

I have devoted this thesis to addressing the controller tuning problem by means of a Multiobjective Optimization Design (MOOD) procedure, employing Evolutionary Multiobjective Optimization (EMO). The main purpose was to develop a holistic procedure that takes all the steps in the MOOD procedure into account: the multiobjective problem (MOP) statement; the EMO process; and the Multi-Criteria Decision Making (MCDM) step. Accordingly, I have proposed tools and guidelines for each of these steps and validated them through various experimental scenarios and examples. The contributions of this research were commented in the introduction, and partial conclusions have been made in each chapter. Below some general conclusions are discussed, as well as ideas for further research.

The multiobjective problem statement

Regarding the MOP, incorporating preferences has been shown to be a perfect link for the three steps of the MOOD procedure, as it brings an alternative to evolve the population towards the pertinent Pareto front, and consequently, facilitates the MCDM step. It was important to develop a meaningful framework for the designer in order to facilitate such preference inclusion. The GPP approach was shown to be useful for this purpose and it was successfully merged with MOEA.

It is true that using GPP or related approaches brings the additional task of defining the preference sets. Nevertheless, in several instances, this supplementary work can be justifiable if it brings a Pareto set approximation with more pertinent solutions to the DM. Therefore, the DM must be willing to employ this additional effort at the beginning of the MOP statement definition. If upper and lower bounds on objectives are known and are sufficient to improve pertinency, a simple constraint coding can be used [156]. As commented before, the preference set definition statement could discourage the use of this tool. It is fundamental to have an understanding of the objectives to define the preference ranges. If the DM has no ideas regarding such values, this could be indicative of a perfunctory or precipitate selection of design objectives. Therefore, perhaps the DM shall ponder the design objectives stated.

The question from Chapter 2 regarding *What kind of problems are worthwhile addressing with MOOD?* has been partially answered for PI-like controllers by addressing MIMO processes. Nevertheless, developing more complex controllers that cover more desirable features for EMO remains a potential application (Table 2.6).

The evolutionary multiobjective optimization process

Regarding EMO, in this thesis the feature related to pertinency of solutions has been exploited in order to design a mechanism (and algorithm) that closes the gap between the search process and decision making.

This new algorithm with a preference inclusion mechanism enables an improvement in the pertinency of the approximated Pareto front, and also helps to deal efficiently with many-objective and constrained optimization instances. Therefore, this is one of the main contributions of this thesis.

For the specific case of PID controller tuning, it has been stated that, given a subset $\mathbb{S} \in \mathbb{R}^3$ and a process $P(s)$, two important features for PID $C(s)$ parameter sampling from $\mathbb{S}$ should be fulfilled:

1. Any sampled controller $C(s)$ from $\mathbb{S}$ must stabilize the closed loop.
2. Any stabilizing controller $C(s)$ of the process $P(s)$ must be contained in $\mathbb{S}$.

The sampling procedure presented in Chapter 7 deals successfully with this issue. Nevertheless, other approaches should be evaluated for more complex controllers, and in the case of the PID controller, to extend it to MIMO processes, since the stability of the single loops does not guarantee the stability of the overall process.

Future work should be oriented to multi-disciplinary optimization, where a true holistic design procedure can be carried out. For this purpose, hybridization techniques should be explored to guarantee the requirements for stochastic sampling of the selected controllers and deal more efficiently with, for example, expensive and reliability based optimization statements.

The multi-criteria decision making step

Design concept comparison for different types of controllers, or different algorithms, to approximate a Pareto front, is a valuable tool for a visual inspection of an m-dimensional Pareto front.

Using the attainment surface to compare MOEAs seems to be a good alternative. Nevertheless, due to the spreading nature of many-objective optimization instances, it is necessary to ensure that algorithms are covering the same Pareto front surface.

A new version of the LD-Tool has been developed and made available. Since the question: *is it possible to define visualization approaches for analyzing multidimensional data and a multidimensional Pareto front that fits better for controller tuning?* remain unanswered, future work should be oriented towards the development of visualizations specifically oriented to controller tuning. As it has been noted in Chapter 8, time response information was valuable for making a final decision on the design alternative to implement in the Boiler benchmark setup from Section 8.4.

The multiobjective optimization design for controller tuning procedure

The MOOD procedure has proven its usefulness for controller tuning; nevertheless, validation for other processes and physical devices needs to be carried out. One of the objectives of this thesis is to highlight some useful guidelines for this procedure. These are summarized below:

Mind the gap. There are several MOEAs available for practitioners. Any effort on new algorithms should be focussed on mechanisms to close the gap between optimization and decision making instead of *new* evolutionary techniques. A good starting point is analyzing the desirable features of EMO that could be used for this purpose in controller tuning (Table 2.6).

Ten times rule. Approximating a dense Pareto front is useful, however approximating a compact and pertinent Pareto front is even more useful. Ten times the number of objectives seem to be a reasonable size for a Pareto front approximation.

Tolerability is essential. It is fundamental to have an understanding of the objectives to define a preference range.[51] If the DM has no idea on such values, that could be an indicative of a perfunctory or precipitate selection of the design objectives. Therefore, perhaps the DM should ponder the design objectives stated.

Where to make a decision. In addition to design objectives and constraints, a third category is included in this thesis: objectives that should be minimized and taken into account during the search process, but which are not meant to be used for decision making.

[51] Or at least, the tolerable values for design objectives.

Controller sampling. Any sampled controller must be stable; all stabilizing controllers must be potentially sampled.

Keep it simple. The MOOD procedure is a powerful tool to analyze objective exchange and select a preferable solution. Nevertheless, it is not a procedure that can be used for every design instance (and in this case, for any controller tuning problem). The following two questions could be helpful for identifying a potential design problem:

- *Is it difficult to find a controller with a reasonable balance among design objectives?*
- *Is it worthwhile analyzing the tradeoff among controllers (design alternatives)?*

If the answer is yes to both questions, then the MOOD procedure could be an appropriate tool for the problem. Otherwise, tuning techniques (rules) or AOF based approaches could fit better for the problem at hand.

Future directions

This investigation will now focus on developing new tools to address two differing MOPs: multidisciplinary and reliability based optimization statements. The former statement will enable the engineering design process to be handled in a fully holistic way. This is because various engineering fields will be taken into account from the beginning of the design process. Mechatronic devices will be a good starting point because mechanical design and control can be easily merged. In the latter statement, reliability based approaches could produce interesting results on robustness performance and it should be possible to obtain more information regarding the degradation of a performance objective (in addition to the robustness measure). Frequency based measures, despite their usefulness in obtaining a theoretical understanding on robustness performance, could be meaningless from a practical point of view to the DM.

Regarding the specific topic of controller tuning; firstly, stochastic sampling for PID controllers will be generalized for PID/N controllers for MIMO processes. Secondly, research will move to state space feedback controllers. While PID controllers are the most used control structure, state space feedback controllers may be the second most commonly employed. This means that developing a proper sampling procedure (similar to the procedure developed for PID controllers) is also a primary goal.

Contributions

Although the contributions of this thesis were focused on controller tuning applications, the EMO process and the MOOD procedure are transversal research fields. This means that they could have an impact across different design domains. For instance, contributions on parametric model identification [149], aerospace trajectory optimization [155], synthetic biology[154], multidisciplinary optimization [157], control education [18, 20], were addressed. In some cases, these applications are under development and shaping future post-doctoral research lines. These applications are listed in Chapter I.

Contributions explicitly related to and contained in this Thesis are commented and listed below:

- A new visualization approach for design concept comparison in m-dimensional Pareto fronts.
- A new preference inclusion mechanism for evolutionary multiobjective optimization to approximate a pertinent Pareto front; using this mechanism, many-objective optimizations and constrained optimization instances can be solved efficiently.
- A new stochastic sampling procedure for a PID controllers in the evolutionary optimization framework has been given.
- General guidelines for the MOOD procedure for controller tuning have been made through examples.

Such results have been made available through the following publications:

- G. Reynoso-Meza, J. Sanchis, X. Blasco, M. Martínez. Evolutionary Algorithms for PID controller tuning: Current Trends and Perspectives. Revista Iberoamericana de Automática e Informática Industrial. 2013; 10: 251-268.
- G. Reynoso-Meza, X. Blasco, J. Sanchis, M. Martínez. Controller tuning by means of evolutionary multiobjective optimization: current trends and applications. Control Engineering Practice. July 2014, Vol. 28, Pp. 58-73.
- G. Reynoso-Meza, J. Sanchis, X. Blasco, J.M. Herrero. Multiobjective evolutionary algorithms for multivariable PI controller design. Expert Systems with Applications. Vol. 39, Issue 9, July 2012. Pp. 7895-7907.
- G. Reynoso-Meza, S. García-Nieto, J. Sanchis, X. Blasco. Controller tuning by means of multiobjective optimization algorithms: a global tuning framework. IEEE Transactions on Control Systems. Vol. 21, Issue 2, March 2013. Pp. 445 - 458.
- G. Reynoso-Meza, X. Blasco, J. Sanchis and J.M. Herrero. Comparison of design concepts in multi-criteria decision making using level diagrams. Information Sciences, Vol. 221, Issue 1, February 2013. Pp. 124-141.

- G. Reynoso-Meza, J. Sanchis, X. Blasco and S. García-Nieto. Physical Programming for preference driven Evolutionary Multi-objective Optimization. Applied Soft Computing. Submitted for review.
- G. Reynoso-Meza, J. Sanchis, X. Blasco and J.M. Herrero. A stabilizing PID controller sampling procedure for stochastic optimizers. The 19th World Congress of the International Federation of Automatic Control. Submitted for review.

and in the following software available at:

`www.mathworks.com/matlabcentral`

MODE: Multi-objective Differential Evolution Algorithm with Spherical Pruning©. Software for MOEAs development at:
`.../fileexchange/38962`.

sp-MODE: Multi-objective Differential Evolution Algorithm with Spherical Pruning©. Software developed in [148] and used in Chapters 3 and 4:
`.../fileexchange/39215`.

LD-Tool: Level Diagrams for multiobjective decision making and Design Concepts Comparison©. Software developed in [151] and presented in Chapter 5:
`.../fileexchange/39458`.

sp-MODE-II: Preference based Multi-objective Differential Evolution Algorithm with Spherical Pruning©. Software developed in Chapter 6:
`.../fileexchange/authors/289050`.

References

He lived in a world where paper and gold were valued in rough equivalency. He had never seen so much paper in his life.

Stephen King, *The Dark Tower II.*

[1] Advances in evolutionary multi-objective optimization. In *Search Based Software Engineering*, G. Fraser and J. Teixeira de Souza, Eds., vol. 7515 of *Lecture Notes in Computer Science*. 2012.

[2] ALAM, M., AND TOKHI, M. Designing feedforward command shapers with multi-objective genetic optimisation for vibration control of a single-link flexible manipulator. *Engineering Applications of Artificial Intelligence 21*, 2 (2008), 229 – 246.

[3] ALGOUL, S., ALAM, M., HOSSAIN, M., AND MAJUMDER, M. Multi-objective optimal chemotherapy control model for cancer treatment. *Medical and Biological Engineering and Computing 49* (2011), 51–65. 10.1007/s11517-010-0678-y.

[4] ASLAM, T., AND NG, A. Multi-objective optimization for supply chain management: A literature review and new development. In *Supply Chain Management and Information Systems (SCMIS), 2010 8th International Conference on* (2010).

[5] ÅSTRÖM, K., AND HÄGGLUND, T. The future of PID control. *Control Engineering Practice 9*, 11 (2001), 1163 – 1175.

[6] ÅSTRÖM, K., PANAGOPOULOS, H., AND HÄGGLUND, T. Design of PI controllers based on non-convex optimization. *Automatica 34*, 5 (1998), 585 – 601.

[7] ÅSTRÖM, K. J. *Control system design*. 2002.

[8] ÅSTRÖM, K. J., AND HÄGGLUND, T. *Advanced PID Control*. ISA - The Instrumentation, Systems, and Automation Society, Research Triangle Park, NC 27709, 2005.

[9] Åström, K. J., and Rundqwist, L. Integrator windup and how to avoid it. In *American Control Conference* (1989), pp. 1693 – 1698.

[10] Avigad, G., Moshaiov, A., and Brauner, N. Towards a general tool for mechatronic design. In *Control Applications, 2003. CCA 2003. Proceedings of 2003 IEEE Conference on* (2003).

[11] Bader, J., and Zitzler, E. Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 1 (2011), 45 – 76.

[12] Batista, L., Campelo, F., Guimarães, F., and Ramírez, J. Pareto cone ϵ-dominance: Improving convergence and diversity in multiobjective evolutionary algorithms. In *Evolutionary Multi-Criterion Optimization*, R. Takahashi, K. Deb, E. Wanner, and S. Greco, Eds., vol. 6576 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2011, pp. 76–90. 10.1007/978-3-642-19893-9-6.

[13] Behbahani, S., and de Silva, C. System-based and concurrent design of a smart mechatronic system using the concept of mechatronic design quotient (MDQ). *IEEE/ASME Transactions on Mechatronics* (2008).

[14] Bell, R., and Åström, K. J. Dynamic models for boiler-turbine alternator units: Data logs and parameter estimation for a 160 MW unit". Technical Report ISRN LUTFD2/TFRT--3192--SE, Department of Automatic Control, Lund University, Sweden, jun 1987.

[15] Bemporad, A., and de la Peña, D. M. Multiobjective model predictive control. *Automatica 45*, 12 (2009), 2823 – 2830.

[16] Berry, M. W. Terminal composition control of a binary distillation column. Master's thesis, Deptartment of Chemical and Petroleum Engineering, University of Alberta., Edmonton, Alberta, 1973.

[17] Beyer, H.-G., and Sendhoff, B. Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering 196*, 33-34 (2007), 3190 – 3218.

[18] Blasco, X., García-Nieto, S., and Reynoso-Meza, G. Autonomous trajectory control of a quadricopter vehicle. simulation and evaluation. *Revista Iberoamericana de Automática e Informática Industrial 9*, 2 (2012), 194 – 199.

[19] Blasco, X., Herrero, J., Sanchis, J., and Martínez, M. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences 178*, 20 (2008), 3908 – 3924.

[20] Blasco, X., Reynoso-Meza, G., and García-Nieto, S. Resultados del concurso de ingeniería de control 2012 y convocatoria 2013. *Revista Iberoamericana de Automática e Informática Industrial 10*, 2 (2013), 240 – 244.

[21] Bonissone, P., Subbu, R., and Lizzi, J. Multicriteria decision making (MCDM): a framework for research and applications. *IEEE Computational Intelligence Magazine* (2009).

[22] Branke, J., Schmeck, H., Deb, K., and Reddy S, M. Parallelizing multi-objective evolutionary algorithms: cone separation. In *Evolutionary Computation, 2004. CEC2004. Congress on* (2004).

[23] Brockhoff, D., and Zitzler, E. Objective reduction in evolutionary multiobjective optimization: Theory and applications. *Evolutionary Computation 17*, 2 (2009), 135 – 166.

[24] Chai, T., Jin, Y., and Sendhoff, B. Evolutionary complex engineering optimization: opportunities and challenges. *Computational Intelligence Magazine 8*, 3 (2013), 12 – 15.

[25] Chipperfield, A., Bica, B., and Fleming, P. Fuzzy scheduling control of a gas turbine aero-engine: a multiobjective approach. *IEEE Transactions on Industrial Electronics* (2002).

[26] Chuk, O. D., and Kuchen, B. R. Supervisory control of flotation columns using multi-objective optimization. *Minerals Engineering 24*, 14 (2011), 1545 – 1555.

[27] Coello, C. Handling preferences in evolutionary multiobjective optimization: a survey. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* (2000), vol. 1, pp. 30 –37 vol.1.

[28] Coello, C. An introduction to multi-objective particle swarm optimizers. In *Soft Computing in Industrial Applications*, A. Gaspar-Cunha, R. Takahashi, G. Schaefer, and L. Costa, Eds., vol. 96 of *Advances in Intelligent and Soft Computing*. Springer Berlin / Heidelberg, 2011, pp. 3–12. $10.1007/978 - 3 - 642 - 20505 - 7_1$.

[29] Coello, C. A. C. Theorical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer methods in applied mechanics and engineering*, 191 (2002), 1245 – 1287.

[30] Coello, C. A. C., and Lamont, G. B. *Applications of Multi-Objective evolutionary algorithms*, advances in natural computation vol. 1 ed. World scientific publishing, 2004.

[31] Coello, C. A. C., Lamont, G. B., and Veldhuizen, D. A. V. Multi-criteria decision making. In *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic and Evolutionary Computation Series. Springer US, 2007, pp. 515–545.

[32] Coello, C. A. C., Veldhuizen, D. V., and Lamont, G. *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic press, 2002.

[33] Coello Coello, C. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Computational Intelligence Magazine* (2006).

[34] Coello Coello, C. Evolutionary multi-objective optimization: Basic concepts and some applications in pattern recognition. In *Pattern Recognition*, J. Martínez-Trinidad, J. Carrasco-Ochoa, C. Ben-Youssef Brants, and E. Hancock, Eds., vol. 6718 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2011, pp. 22–33. 10.1007/978-3-642-21587-2_3.

[35] Corne, D. W., and Knowles, J. D. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation* (New York, NY, USA, 2007), GECCO '07, ACM, pp. 773 – 780.

[36] Crews, J. H., Mattson, M. G., and Buckner, G. D. Multi-objective control optimization for semi-active vehicle suspensions. *Journal of Sound and Vibration 330*, 23 (2011), 5502 – 5516.

[37] Cruz, C., González, J., and Pelta, D. A. Optimization in dynamic environments: a survey on problems, methods and measures. *Soft Computing 15* (2011), 1427–1448.

[38] Cvetkovic, D., and Parmee, I. Preferences and their application in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* (2002).

[39] Das, I., and Dennis, J. Normal-boundary intersection: a new method for generating the Pareto surface in non-linear multicriteria optimization problems. *SIAM Journal on Optimization 8* (1998), 631 – 657.

[40] Das, S., Maity, S., Qu, B.-Y., and Suganthan, P. Real-parameter evolutionary multimodal optimization - a survey of the state-of-the-art. *Swarm and Evolutionary Computation 1*, 2 (2011), 71 – 88.

[41] Das, S., and Suganthan, P. N. Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation PP*, 99 (2010), 1 –28.

[42] DEB, K. An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering 186*, 2-4 (2000), 311 – 338.

[43] DEB, K., PRATAP, A., AGARWAL, S., AND MEYARIVAN, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation 6*, 2 (2002), 124 – 141.

[44] DEB, K., AND SAHA, A. Multimodal optimization using a bi-objective evolutionary algorithm. *Evolutionary Computation* (2012), 27–62.

[45] DEB, K., TIELE, L., LAUMMANS, M., AND ZITZLER, E. Scalable test problems for evolutionary multi-objective optimization. Tech. Rep. TIK-Technical Report No. 112, Institut fur Technische Informatik und Kommunikationsnetze, ETH Zurich, Feb. 2001.

[46] DELGADO, M., CUELLAR, M., AND PEGALAJAR, M. Multiobjective hybrid optimization and training of recurrent neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* (2008).

[47] DERRAC, J., GARCÍA, S., MOLINA, D., AND HERRERA, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation 1*, 1 (2011), 3 – 18.

[48] DIXON, R., AND PIKE, A. ALSTOM benchmark challenge II on gasifier control. *IEE Proceedings Control Theory and Applications* (2006).

[49] DORIGO, M., AND STÜTZLE, T. Ant colony optimization: Overview and recent advances. In *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds., vol. 146 of *International Series in Operations Research & Management Science*. Springer US, 2010, pp. 227–263.

[50] EFSTRATIADIS, A., AND KOUTSOYIANNIS, D. One decade of multi-objective calibration approaches in hydrological modelling: a review. *Hydrological Sciences Journal 55*, 1 (2010), 58–78.

[51] EIBEN, A. E., HINTERDING, R., AND MICHALEWICZ, Z. Parameter control in evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on 3*, 2 (1999), 124 – 141.

[52] FADAEE, M., AND RADZI, M. Multi-objective optimization of a stand-alone hybrid renewable energy system by using evolutionary algorithms: A review. *Renewable and Sustainable Energy Reviews 16*, 5 (2012), 3364 – 3369.

[53] FARINA, M., DEB, K., AND AMATO, P. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Transactions on Evolutionary Computation 8*, 5 (2004), 425–442.

[54] FAZENDEIRO, P., DE OLIVEIRA, J., AND PEDRYCZ, W. A multiobjective design of a patient and anaesthetist-friendly neuromuscular blockade controller. *IEEE Transactions on Biomedical Engineering* (2007).

[55] FAZLOLLAHI, S., MANDEL, P., BECKER, G., AND MARÉCHAL, F. Methods for multi-objective investment and operating optimization of complex energy systems. *Energy 45*, 1 (2012), 12 – 22.

[56] FAZZOLAR, M., ALCALÁ, R., NOJIMA, Y., ISHIBUCHI, H., AND HERRERA, F. A review of the application of multi-objective evolutionary fuzzy systems: Current status and further directions. *IEEE Transactions on Fuzzy Systems*, 45–65.

[57] FENG, G. A survey on analysis and design of model-based fuzzy control systems. *IEEE Transactions on Fuzzy Systems* (2006).

[58] FERNANDEZ, E., LOPEZ, E., LOPEZ, F., AND COELLO, C. A. C. Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method. *Information Sciences 181*, 1 (2011), 44 – 56.

[59] FERNÁNDEZ, I., RODRÍGUEZ, C., GUZMAN, J., AND BERENGUEL, M. Control predictivo por desacoplo con compensación de perturbaciones para el benchmark de control 2009-2010. *Revista Iberoamericana de Automática e Informática Industrial Apr. 8*, 2 (2011), 112 – 121.

[60] FIGUEIRA, J., GRECO, S., AND EHRGOTT, M. *Multiple criteria decision analysis: State of the art surveys*. Springer international series, 2005.

[61] FLEMING, P., AND PURSHOUSE, R. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10 (2002), 1223 – 1241.

[62] FONSECA, C., AND FLEMING, P. Multiobjective optimization and multiple constraint handling with evolutionary algorithms-I: A unified formulation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* (1998).

[63] FONSECA, C., AND FLEMING, P. Multiobjective optimization and multiple constraint handling with evolutionary algorithms-II: Application example. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* (1998).

[64] GABRIEL, C. Modelling, simulation and control of a twin rotor mimo-system. Master's thesis, Universitat Politècnica de València., Avalaible at `http://personales.upv.es/gilreyme/mood4ct/files/TRMS.zip`, 2009.

[65] Gacto, M., Alcalá, R., and Herrera, F. A multi-objective evolutionary algorithm for an effective tuning of fuzzy logic controllers in heating, ventilating and air conditioning systems. *Applied Intelligence 36* (2012), 330–347. 10.1007/s10489-010-0264-x.

[66] García, J. J. V., Garay, V. G., Gordo, E. I., Fano, F. A., and Sukia, M. L. Intelligent multi-objective nonlinear model predictive control (iMO-NMPC): Towards the on-line optimization of highly complex control problems. *Expert Systems with Applications 39*, 7 (2012), 6527 – 6540.

[67] García-Nieto, S., Martínez, M., Blasco, X., and Sanchis, J. Nonlinear predictive control based on local model networks for air management in diesel engines. *Control Engineering Practice 16*, 12 (2008), 1399 – 1413.

[68] Garcia-Alvarado, M., and Ruiz-Lopez, I. A design method for robust and quadratic optimal mimo linear controllers. *Chemical Engineering Science 65*, 11 (2010), 3431 – 3438.

[69] Garrido, J., Márquez, F., and Morilla, F. Multivariable PID control by inverted decoupling: application to the benchmark PID 2012. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)* (March 2012).

[70] Ge, M., Chiu, M.-S., and Wang, Q.-G. Robust PID controller design via LMI approach. *Journal of process control*, 12 (2002), 3 – 13.

[71] Goh, C.-K., Teoh, E.-J., and Tan, K. C. Hybrid multiobjective evolutionary design for artificial neural networks. *IEEE Transactions on Neural Networks* (2008).

[72] Goncalves, E. N., Palhares, R. M., and Takahashi, R. H. A novel approach for H_2/H_∞ robust PID synthesis for uncertain systems. *Journal of process control*, 18 (2008), 19 – 26.

[73] Gong, W., Cai, Z., and Zhu, L. An efficient multiobjective Differential Evolution algorithm for engineering design. *Structural and Multidisciplinary Optimization 38* (2009), 137 – 157. 10.1007/s00158-008-0269-9.

[74] Hajiloo, A., Nariman-zadeh, N., and Moeini, A. Pareto optimal robust design of fractional-order PID controllers for systems with probabilistic uncertainties. *Mechatronics 22*, 6 (2012), 788 – 801.

[75] Hansen, N. The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation. Advances on estimation of distribution algorithms*, J. Lozano, P. Larranaga, I. Inza, and E. Bengoetxea, Eds. Springer, 2006, pp. 75 – 102.

[76] Harik, G., Lobo, F., and Goldberg, D. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation 3*, 4 (1999), 287 –297.

[77] Hernández-Díaz, A. G., Santana-Quintero, L. V., Coello, C. A. C., and Molina, J. Pareto-adaptive ϵ-dominance. *Evolutionary Computation*, 4 (2007), 493 – 517.

[78] Herrero, J., Martínez, M., Sanchis, J., and Blasco, X. Well-distributed Pareto front by using the ϵ-MOGA evolutionary algorithm. In *Computational and Ambient Intelligence*, F. S. et al., Ed., vol. LNCS 4507. Springer-Verlag, 2007, pp. 292 – 299.

[79] Herreros, A., Baeyens, E., and Perán, J. R. Design of PID-type controllers using multiobjective genetic algorithms. *ISA Transactions 41*, 4 (2002), 457 – 472.

[80] Herreros, A., Baeyens, E., and Perán, J. R. MRCD: a genetic algorithm for multiobjective robust control design. *Engineering Applications of Artificial Intelligence*, 15 (2002), 285 – 301.

[81] Houska, B., Ferreau, H. J., and Diehl, M. ACADO toolkit an open source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods 32*, 3 (2011), 298–312.

[82] Huang, L., Wang, N., and Zhao, J.-H. Multiobjective optimization for controller design. *Acta Automatica Sinica 34*, 4 (2008), 472 – 477.

[83] Huang, V., Qin, A., Deb, K., Zitzler, E., Suganthan, P., Liang, J., Preuss, M., and Huband, S. Problem definitions for performance assessment on multi-objective optimization algorithms. Tech. rep., Nanyang Technological University, Singapore, 2007.

[84] Huband, S., Hinstong, P., Barone, L., and While, L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation 10*, 5 (2006), 477 – 506.

[85] Hung, M.-H., Shu, L.-S., Ho, S.-J., Hwang, S.-F., and Ho, S.-Y. A novel intelligent multiobjective simulated annealing algorithm for designing robust PID controllers. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans* (2008).

[86] Inselberg, A. The plane with parallel coordinates. *The Visual Computer 1* (1985), 69–91.

[87] Iruthayarajan, M. W., and Baskar, S. Evolutionary algorithms based design of multivariable PID controller. *Expert Systems with Applications 36*, 5 (2009), 9159 – 9167.

[88] IRUTHAYARAJAN, M. W., AND BASKAR, S. Covariance matrix adaptation evolution strategy based design of centralized PID controller. *Expert Systems with Applications 37*, 8 (2010), 5775 – 5781.

[89] ISAKSSON, A., AND GRAEBE, S. Derivative filter is an integral part of PID design. *IEE Proceedings on Control Theory and Applications* (2002).

[90] ISHIBUCHI, H., TSUKAMOTO, N., AND NOJIMA, Y. Evolutionary many-objective optimization: A short review. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on* (2008).

[91] JAMALI, A., HAJILOO, A., AND NARIMAN-ZADEH, N. Reliability-based robust Pareto design of linear state feedback controllers using a multi-objective uniform-diversity genetic algorithm (MUGA). *Expert Systems with Applications 37*, 1 (2010), 401 – 413.

[92] JIACHUAN, W., ZHUN, F., TERPENNY, J., AND GOODMAN, E. Knowledge interaction with genetic programming in mechatronic systems design using bond graphs. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* (2005).

[93] JIN, Y., AND SENDHOFF, B. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* (2008).

[94] JUANG, J.-G., LIN, R.-W., AND LIU, W.-K. Comparison of classical control and intelligent control for a mimo system. *Applied Mathematics and Computation 205*, 2 (2008), 778 – 791. Special Issue on Advanced Intelligent Computing Theory and Methodology in Applied Mathematics and Computation.

[95] KARABOGA, D., GORKEMLI, B., OZTURK, C., AND KARABOGA, N. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review* (2012), 1–37.

[96] KENNEDY, J., AND EBERHART, R. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on* (1995).

[97] KIM, H.-S., AND ROSCHKE, P. N. Fuzzy control of base-isolation system using multi-objective genetic algorithm. *Computer-Aided Civil and Infrastructure Engineering 21*, 6 (2006), 436 – 449.

[98] KNOWLES, J. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In *Intelligent Systems Design and Applications, 2005. ISDA '05. Proceedings. 5th International Conference on* (2005).

[99] Knowles, J., Thiele, L., and Zitzler., E. A tutorial on the performance assessment of stochastic multiobjective optimizers. Tech. Rep. TIK report No. 214, Computer Engineering and networks laboratory. ETH Zurich, 2006.

[100] Kollat, J. B., and Reed, P. A framework for visually interactive decision-making and design using evolutionary multi-objective optimization (VIDEO). *Environmental Modelling & Software 22*, 12 (2007), 1691 – 1704.

[101] Konak, A., Coit, D. W., and Smith, A. E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety 91*, 9 (2006), 992 – 1007. Special Issue - Genetic Algorithms and Reliability.

[102] Kramer, O., and Koch, P. Rake selection: A novel evolutionary multi-objective optimization algorithm. In *KI 2009: Advances in Artificial Intelligence*, B. Mertsching, M. Hund, and Z. Aziz, Eds., vol. 5803 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2009, pp. 177–184. 10.1007,978-3-642-04617-9,23.

[103] Lamanna, R., Vega, P., Revollar, S., and Alvarez, H. Diseño simultáneo de proceso y control de una torre sulfitadora de jugo de caña de azúcar. *Revista Iberoamericana de Automática e Informática Industrial 6*, 3 (2009), 32 – 43.

[104] Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 3 (2002), 263 – 282.

[105] Laurí, D., Salcedo, J., García-Nieto, S., and Martínez, M. Model predictive control relevant identification: multiple input multiple output against multiple input single output. *IET Control Theory & Applications 4*, 9 (2010), 1756–1766.

[106] Lotov, A., and Miettinen, K. Visualizing the Pareto frontier. In *Multiobjective Optimization*, J. Branke, K. Deb, K. Miettinen, and R. Slowinski, Eds., vol. 5252 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2008, pp. 213–243.

[107] Low, K.-S., and Wong, T.-S. A multiobjective genetic algorithm for optimizing the performance of hard disk drive motion control system. *IEEE Transactions on Industrial Electronics* (2007).

[108] Lozano, M., Molina, D., and Herrera, F. *Soft Computing: Special Issue on scalability of evolutionary algorithms and other metaheuristics for large-scale continuous optimization problems.*, vol. 15. Springer-Verlag, 2011.

[109] LUYBEN, W. L. Simple method for tuning SISO controllers in multivariable systems. *Industrial and Engineering Chemistry Process Design*, 25 (1986), 654 – 660.

[110] LYGOE, R., CARY, M., AND FLEMING, P. A many-objective optimisation decision-making process applied to automotive diesel engine calibration. In *Simulated Evolution and Learning*, K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakroborty, S. Das, J. Dutta, S. Gupta, A. Jain, V. Aggarwal, J. Branke, S. Louis, and K. Tan, Eds., vol. 6457 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2010, pp. 638–646. 10.1007/978-3-642-17298-4 72.

[111] MALLIPEDDI, R., AND SUGANTHAN, P. Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization. Tech. rep., Nanyang Technological University, Singapore, 2009.

[112] MALLIPEDDI, R., AND SUGANTHAN, P. Ensemble of constraint handling techniques. *IEEE Transactions on Evolutionary Computation* (2010).

[113] MANSOURI, S. A., GALLEAR, D., AND ASKARIAZAD, M. H. Decision support for build-to-order supply chain management through multiobjective optimization. *International Journal of Production Economics 135*, 1 (2012), 24 – 36.

[114] MARINAKI, M., MARINAKIS, Y., AND STAVROULAKIS, G. Fuzzy control optimized by a multi-objective particle swarm optimization algorithm for vibration suppression of smart structures. *Structural and Multidisciplinary Optimization 43* (2011), 29–42. 10.1007/s00158-010-0552-4.

[115] MARLER, R., AND ARORA, J. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, 26 (2004), 369 – 395.

[116] MARTINS, J. R. R. A., AND LAMBE, A. B. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal 51*, 9 (2013), 2049–2075.

[117] MARTÍNEZ, M., HERRERO, J., SANCHIS, J., BLASCO, X., AND GARCÍA-NIETO, S. Applied Pareto multi-objective optimization by stochastic solvers. *Engineering applications of artificial intelligence 22* (2009), 455 – 465.

[118] MARTÍNEZ, M., SANCHIS, J., AND BLASCO, X. Multiobjective controller design handling human preferences. *Engineering Applications of Artificial Intelligence 19* (2006), 927 – 938.

[119] MATTSON, C. A., AND MESSAC, A. Pareto frontier based concept selection under uncertainty, with visualization. *Optimization and Engineering 6* (2005), 85–115.

[120] MEEUSE, F., AND TOUSAIN, R. L. Closed-loop controllability analysis of process designs: Application to distillation column design. *Computers and Chemical Engineering 26*, 4-5 (2002), 641 – 647.

[121] MESSAC, A., ISMAIL-YAHAYA, A., AND MATTSON, C. The normalized normal constraint method for generating the Pareto frontier. *Structural and Multidisciplinary Optimization*, 25 (2003), 86 – 98.

[122] MESSAC, A., AND MATTSON, C. Generating well-distributed sets of Pareto points for engineering design using Physical Programming. *Optimization and Engineering 3* (2002), 431–450. 10.1023/A:1021179727569.

[123] METAXIOTIS, K., AND LIAGKOURAS, K. Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. *Expert Systems with Applications 39*, 14 (2012), 11685 – 11698.

[124] MEZURA-MONTES, E., AND COELLO, C. A. C. Constraint-handling in nature-inspired numerical optimization: Past, present and future. *Swarm and Evolutionary Computation 1*, 4 (2011), 173 – 194.

[125] MEZURA-MONTES, E., REYES-SIERRA, M., AND COELLO, C. Multi-objective optimization using differential evolution: A survey of the state-of-the-art. *Advances in Differential Evolution*, SCI 143 (2008), 173 – 196.

[126] MICHALEWICZ, Z. Quo vadis, evolutionary computation? In *Advances in Computational Intelligence*. Springer, 2012, pp. 98–121.

[127] MIETTINEN, K. M. *Nonlinear multiobjective optimization*. Kluwer Academic Publishers, 1998.

[128] MININNO, E., NERI, F., CUPERTINO, F., AND NASO, D. Compact differential evolution. *IEEE Transactions on Evolutionary Computation 15*, 1 (2011), 32 –54.

[129] MOLINA-CRISTÓBAL, A., GRIFFIN, I., FLEMING, P., AND OWENS, D. Linear matrix inequialities and evolutionary optimization in multiobjective control. *International Journal of Systems Science 37*, 8 (2006), 513 – 522.

[130] MONJE, C. A., VINAGRE, B. M., FELIU, V., AND CHEN, Y. Tuning and auto-tuning of fractional order controllers for industry applications. *Control Engineering Practice 16*, 7 (2008), 798 – 812.

[131] MONTES DE OCA, S., PUIG, V., WITCZAK, M., AND QUEVEDO, J. Fault-tolerant control of a two-degree of freedom helicopter using LPV techniques. pp. 1204 –1209.

[132] MORILLA, F. Benchmark 2009-10 grupo temático de ingeniería de control de cea-ifac: Control de una caldera. Avalaible at `www.cea-ifac.es/w3grupos/ingcontrol`, Febrero 2010.

[133] MORILLA, F. Benchmar for PID control based on the boiler control problem. Available at `http://www.dia.uned.es/~fmorilla/benchmark09_10/`, 2012. Internal report, UNED Spain.

[134] MOSCATO, P., AND COTTA, C. A modern introduction to memetic algorithms. In *Handbook of Metaheuristics*, M. Gendreau and J.-Y. Potvin, Eds., vol. 146 of *International Series in Operations Research & Management Science*. Springer US, 2010, pp. 141–183.

[135] MUNRO, M., AND AOUNI, B. Group decision makers' preferences modelling within the goal programming model: An overview and a typology. *Journal of Multi-Criteria Decision Analysis 19*, 3-4 (2012), 169–184.

[136] NERI, F., AND COTTA, C. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation 2*, 0 (2012), 1 – 14.

[137] OCHI, Y. PID controller design for MIMO systems by applying balanced truncation to integral-type optimal servomechanism. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)* (March 2012).

[138] ORTEGA, M., AND RUBIO, F. Systematic design of weighting matrices for the H_∞ mixed sensitivity problem. *Journal of Process Control*, 14 (2004), 89 – 98.

[139] OSYCZKA, A., AND KUNDU, S. A modified distance method for multicriteria optimization, using genetic algorithms. *Computers & Industrial Engineering 30*, 4 (1996), 871 – 882.

[140] PANAGOPOULOS, H., ÅSTRÖM, K., AND HÄGGLUND, T. Design of PID controllers based on constrained optimisation. *Control Theory and Applications, IEE Proceedings -* (2002).

[141] PELLEGRINETTI, G., AND BENTSMAN, J. Nonlinear control oriented boiler modeling-a benchmark problem for controller design. *IEEE Transactions on Control Systems Technology* (1996).

[142] PURSHOUSE, R., AND FLEMING, P. On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation 11*, 6 (2007), 770 – 784.

[143] PURSHOUSE, R., JALBA, C., AND FLEMING, P. Preference-driven co-evolutionary algorithms show promise for many-objective optimisation. In *Evolutionary Multi-Criterion Optimization*, R. Takahashi, K. Deb, E. Wanner, and S. Greco, Eds., vol. 6576 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2011, pp. 136 – 150. 10.1007/978-3-642-19893-9 10.

[144] QU, B., AND SUGANTHAN, P. Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection. *Information Sciences 180*, 17 (2010), 3170 – 3181. Including Special Section on Virtual Agent and Organization Modeling: Theory and Applications.

[145] RAMÍREZ-ARIAS, A., RODRÍGUEZ, F., GUZMÁN, J., AND BERENGUEL, M. Multiobjective hierarchical control architecture for greenhouse crop growth. *Automatica 48*, 3 (2012), 490 – 498.

[146] RAO, J. S., AND TIWARI, R. Design optimization of double-acting hybrid magnetic thrust bearings with control integration using multi-objective evolutionary algorithms. *Mechatronics 19*, 6 (2009), 945 – 964.

[147] REED, P., HADKA, D., HERMAN, J., KASPRZYK, J., AND KOLLAT, J. Evolutionary multiobjective optimization in water resources: The past, present, and future. *Advances in Water Resources 51*, 1 (2013), 438 – 456.

[148] REYNOSO-MEZA, G. Design, coding and implementation of a multi-objective optimization algorithm based on differential evolution with spherical pruning: applications for system identification and controller tuning. Master's thesis, Universitat Politècnica de València., 2009.

[149] REYNOSO-MEZA, G., BLASCO, X., AND SANCHIS, J. Multi-objective design of PID controllers for the control benchmark 2008-2009 (in spanish). *Revista Iberoamericana de Automática e Informática Industrial 6*, 4 (2009), 93 – 103.

[150] REYNOSO-MEZA, G., BLASCO, X., AND SANCHIS, J. Optimización evolutiva multi-objetivo y selección multi-criterio para la ingeniería de control. In *X Simposio CEA de Ingeniería de Control* (March 2012).

[151] REYNOSO-MEZA, G., BLASCO, X., SANCHIS, J., AND HERRERO, J. M. Comparison of design concepts in multi-criteria decision-making using level diagrams. *Information Sciences 221* (2013), 124 – 141.

[152] REYNOSO-MEZA, G., BLASCO, X., SANCHIS, J., AND MARTÍNEZ, M. Evolutionary algorithms for PID controller tuning: Current trends and perspectives (in spanish). *Revista Iberoamericana de Automática e Informática Industrial 10*, 3 (2013), 251 – 268.

[153] Reynoso-Meza, G., García-Nieto, S., Sanchis, J., and Blasco, X. Controller tuning using multiobjective optimization algorithms: a global tuning framework. *IEEE Transactions on Control Systems Technology 21*, 2 (2013), 445–458.

[154] Reynoso-Meza, G., Montagud, A., Sanchis, J., and Urchueguía, J. Simulation of the synechocystis sp. PCC6803 metabolic behavior using stoichiometric representations and multiobjective evolutionary algorithms. In *Proceedings of the 12th international conference on systems biology (ICSB 2011)* (August 2011).

[155] Reynoso-Meza, G., Sanchis, J., Blasco, X., and Herrero, J. Hybrid DE algorithm with adaptive crossover operator for solving real-world numerical optimization problems. In *Evolutionary Computation (CEC), 2011 IEEE Congress on* (june 2011), pp. 1551 –1556.

[156] Reynoso-Meza, G., Sanchis, J., Blasco, X., and Herrero, J. M. Multiobjective evolutionary algortihms for multivariable PI controller tuning. *Expert Systems with Applications 39* (2012), 7895 – 7907.

[157] Reynoso-Meza, G., Sanchis, J., Blasco, X., and Herrero, J. M. A stabilizing PID controller sampling procedure for stochastic optimizers. In *Memories of the 19th World Congress IFAC 2014 (Accepted)*. 2014.

[158] Reynoso-Meza, G., Sanchis, J., Blasco, X., and Martínez, M. Design of continuous controllers using a multiobjective differential evolution algorithm with spherical pruning. In *Applications of Evolutionary Computation*. Springer, 2010, pp. 532–541.

[159] Reynoso-Meza, G., Sanchis, J., Blasco, X., and Martínez, M. Multiobjective design of continuous controllers using differential evolution and spherical pruning. In *Applications of Evolutionary Computation, Part I* (2010), C. D. Chio, S. Cagnoni, C. Cotta, M. Eber, A. Ekárt, A. I.Esparcia-Alcaráz, C.-K. Goh, J. J.Merelo, F. Neri, M. Preuss, J. Togelius, and G. N.Yannakakis, Eds., vol. LNCS 6024, Springer-Verlag, pp. 532–541.

[160] Rojas, J. D., Morilla, F., and Vilanova, R. Multivariable PI control for a boiler plant benchmark using the virtual reference feedback tuning. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)* (March 2012).

[161] Roy, R., Hinduja, S., and Teti, R. Recent advances in engineering design optimisation: Challenges and future trends. *CIRP Annals - Manufacturing Technology 57*, 2 (2008), 697 – 715.

[162] Ruzika, S., and Wiecek, M. Successive approach to compute the bounded Pareto front of practical multiobjective optimization problems. *SIAM Journal on Optimization 20* (2009), 915 – 934.

[163] Saeki, M., Ogawa, K., and Wada, N. Application of data-driven loop-shaping method to multi-loop control design of benchmark PID 2012. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)* (March 2012).

[164] Sanchis, J., Martínez, M., Blasco, X., and Salcedo, J. V. A new perspective on multiobjective optimization by enhanced normalized normal constraint method. *Structural and multidisciplinary optimization*, 36 (2008), 537 – 546.

[165] Sanchis, J., Martínez, M. A., Blasco, X., and Reynoso-Meza, G. Modelling preferences in multiobjective engineering design. *Engineering Applications of Artificial Intelligence 23* (2010), 1255 – 1264.

[166] Santana-Quintero, L., Montaño, A., and Coello, C. A review of techniques for handling expensive functions in evolutionary multiobjective optimization. In *Computational Intelligence in Expensive Optimization Problems*, Y. Tenne and C.-K. Goh, Eds., vol. 2 of *Adaptation Learning and Optimization*. Springer Berlin Heidelberg, 2010, pp. 29–59.

[167] Saridakis, K., and Dentsoras, A. Soft computing in engineering design - a review. *Advanced Engineering Informatics 22*, 2 (2008), 202 – 221. Network methods in engineering.

[168] Sidhartha, and Panda. Multi-objective PID controller tuning for a facts-based damping stabilizer using non-dominated sorting genetic algorithm-II. *International Journal of Electrical Power and Energy Systems 33*, 7 (2011), 1296 – 1308.

[169] Silva, G., Datta, A., and Bhattacharyya, S. New results on the synthesis of PID controllers. *Automatic Control, IEEE Transactions on 47*, 2 (2002), 241–252.

[170] Silveira, A., Coelho, A., and Gomes, F. Model-free adaptive PID controllers applied to the benchmark PID12. In *Proceedings of the IFAC Conference on Advances in PID Control (PID'12)* (March 2012).

[171] Singh, H., Isaacs, A., and Ray, T. A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Transactions on Evolutionary Computation 15*, 4 (2011), 539 –556.

[172] SONG, Z., AND KUSIAK, A. Optimization of temporal processes: A model predictive control approach. *IEEE Transactions on Evolutionary Computation 13*, 1 (2009), 169 –179.

[173] SÖRENSEN, K. Metaheuristics - the metaphor exposed. *International Transactions in Operational Research* (2013), DOI 10.1111/itor.12001.

[174] SRINIVAS, M., AND PATNAIK, L. Genetic algorithms: a survey. *Computer 27*, 6 (1994), 17 – 26.

[175] SRINIVASAN, S., AND RAMAKRISHNAN, S. Evolutionary multi objective optimization for rule mining: a review. *Artificial Intelligence Review 36* (2011), 205–248. 10.1007/s10462-011-9212-3.

[176] STENGEL, R. F., AND MARRISON, C. I. Robustness of solutions to a benchmark control problem. *Journal of Guidance, Control and Dynamics.*, 15 (1992), 1060 – 1067.

[177] STEWART, G., AND SAMAD, T. Cross-application perspectives: Application and market requirements. In *The Impact of Control Technology*, T. Samad and A. Annaswamy, Eds. IEEE Control Systems Society, 2011, pp. 95 – 100.

[178] STEWART, P., GLADWIN, D., AND FLEMING, P. Multiobjective analysis for the design and control of an electromagnetic valve actuator. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 221* (2007), 567 – 577.

[179] STEWART, P., JEWELL, G., CLARK, R., AND FLEMING, P. Controllability analysis of multi objective control systems. In *Computer Aided Control System Design, 2002. Proceedings. 2002 IEEE International Symposium on* (2002), pp. 74 – 79.

[180] STEWART, P., STONE, D., AND FLEMING, P. Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop. *Engineering Applications of Artificial Intelligence 17*, 3 (2004), 275 – 284.

[181] STEWART, P., ZAVALA, J. C., AND FLEMING, P. Automotive drive by wire controller design by multi-objective techniques. *Control Engineering Practice 13*, 2 (2005), 257 – 264.

[182] STORN, R. SCI: Differential evolution research: Trends and open questions. vol. LNCS 143. Springer, Heidelberg, 2008, pp. 1 – 31.

[183] STORN, R., AND PRICE, K. Differential evolution: A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization 11* (1997), 341 – 359.

[184] SUMAN, B., AND KUMAR, P. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society 57*, 10 (2005), 1143–1160.

[185] SUN, Y., ZHANG, C., GAO, L., AND WANG, X. Multi-objective optimization algorithms for flow shop scheduling problem: a review and prospects. *The International Journal of Advanced Manufacturing Technology 55* (2011), 723–739. 10.1007/s00170-010-3094-4.

[186] SUTTORP, T., AND IGEL, C. Multi-objective optimization of support vector machines. In *Multi-Objective Machine Learning*, Y. Jin, Ed., vol. 16 of *Studies in Computational Intelligence*. Springer Berlin / Heidelberg, 2006, pp. 199–220. 10.1007/3-540-33019-4_9.

[187] TAN, W., LIU, J., FANG, F., AND CHEN, Y. Tuning of PID controllers for boiler-turbine units. *ISA Transactions 43*, 4 (2004), 571 – 583.

[188] TAVAKOLI, S., GRIFFIN, I., AND FLEMING, P. Multi-objective optimization approach to the PI tuning problem. In *Proceedings of the IEEE congress on evolutionary computation (CEC2007)* (2007), pp. 3165 – 3171.

[189] THIELE, L., MIETTINEN, K., KORHONEN, P. J., AND MOLINA, J. A preference-based evolutionary algorithms for multi-objective optimization. *Evolutionary Computation*, 3 (2009), 411 – 436.

[190] TOSCANO, R. A simple robust PI/PID controller design via numerical optimization approach. *Journal of process control*, 15 (2005), 81 – 88.

[191] VILANOVA, R., AND ALFARO, V. M. Robust PID control: an overview (in spanish). *Revista Iberoamericana de Automática e Informática Industrial 8*, 3 (2011), 141 – 158.

[192] WANG, Y., AND YANG, Y. Particle swarm optimization with preference order ranking for multi-objective optimization. *Information Sciences 179*, 12 (2009), 1944 – 1959. Special Section: Web Search.

[193] WEN, P., AND LU, T.-W. Decoupling control of a twin rotor mimo system using robust deadbeat control technique. *IET Control Theory Applications 2*, 11 (2008), 999 – 1007.

[194] WIE, B., AND BERNSTEIN, D. S. Benchmark problems for robust control design. *Journal of Guidance, Control and Dynamics.*, 15 (1992), 1057 – 1059.

[195] WOOD, R. K., AND BERRY, M. W. Terminal composition control of a binary distillation column. *Chemical Engineering Science 28*, 9 (1973), 1707 – 1717.

[196] XUE, Y., LI, D., AND GAO, F. Multi-objective optimization and selection for the PI control of ALSTOM gasifier problem. *Control Engineering Practice 18*, 1 (2010), 67 – 76.

[197] YUSUP, N., ZAIN, A. M., AND HASHIM, S. Z. M. Evolutionary techniques in optimizing machining parameters: Review and recent applications (2007-2011). *Expert Systems with Applications 39*, 10 (2012), 9909 – 9927.

[198] ZAVALA, V. M., AND FLORES-TLACUAHUAC, A. Stability of multiobjective predictive control: An utopia-tracking approach. *Automatica 48*, 10 (2012), 2627 – 2632.

[199] ZHANG, Q., AND LI, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* (2007).

[200] ZHANG, Q., ZHOU, A., ZHAO, S., SUGANTHAN, P., LIU, W., AND TIWARI, S. Multiobjective optimization test instances for the cec 2009 special session and competition. Tech. Rep. CES-887, University of Essex and Nanyang Technological University, 2008.

[201] ZHAO, S.-Z., IRUTHAYARAJAN, M. W., BASKAR, S., AND SUGANTHAN, P. Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization. *Information Sciences 181*, 16 (2011), 3323 – 3335.

[202] ZHOU, A., QU, B.-Y., LI, H., ZHAO, S.-Z., SUGANTHAN, P. N., AND ZHANG, Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation 1*, 1 (2011), 32 – 49.

[203] ZIO, E., AND BAZZO, R. Level diagrams analysis of pareto front for multiobjective system redundancy allocation. *Reliability Engineering & System Safety 96*, 5 (2011), 569 – 580.

[204] ZIO, E., AND RAZZO, R. Multiobjective optimization of the inspection intervals of a nuclear safety system: A clustering-based framework for reducing the pareto front. *Annals of Nuclear Energy*, 37 (2010), 798–812.

[205] ZITZLER, E., AND KÜNZLI, S. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature - PPSN VIII*, X. Yao, E. Burke, J. Lozano, J. Smith, J. Merelo-Guervós, J. Bullinaria, J. Rowe, P. Tino, A. Kabán, and H.-P. Schwefel, Eds., vol. 3242 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2004, pp. 832–842. 10.1007/978-3-540-30217-9 84.

[206] ZITZLER, E., THIELE, L., LAUMANNS, M., FONSECA, C., AND DA FONSECA, V. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation 7*, 2 (2003), 117 – 132.

Author

Gilberto Reynoso Meza (20-01-1979, Tampico, Tamaulipas, México) received his M.Sc. degree (2005) in Automation and Control from the Tecnológico de Monterrey, Campus Monterrey (México) and his B.Sc. (2001) in Mechanical Engineering from the Tecnológico de Monterrey, Campus Querétaro (México). Until the submission of this thesis (June 2014), he has been with the Universitat Politècnica de València (Spain) as a member of the Predictive Control and Heuristic Optimization (CPOH) research staff.
His main research interests are computational intelligence methods for control engineering, multiobjective optimization, many-objective optimization, multi-criteria decision making and evolutionary algorithms.